图模型匹配: 一种新的凹松弛函数及算法

刘智勇 [1]

摘 要 将问题中的置换矩阵放松为双随机矩阵是近年来近似图匹配算法的一个重要发展方向. 它的本质在于将离散的图匹配问题转换成一个连续优化问题, 而一般来讲, 相对于离散优化, 连续优化问题的近似求解将更为容易. 但随之带来的一个问题是如何有效地将连续优化得到的双随机矩阵重新映射回一个置换矩阵. 最近文献中提出了一种针对于无向无自环图的凹松弛 (Concave relaxation) 函数, 使得算法中的双随机矩阵可以平滑地收敛到一个置换矩阵, 并得到优异的匹配精度. 但除了无向且无自环图, 文献中还没有针对其他类型图模型的凹松弛函数. 本文提出一种针对于有向无自环图匹配问题的凹松弛函数, 并在此基础上给出一种图匹配算法. 大量对比实验验证了本文提出模型及算法的有效性.

关键词 图模型匹配, Frank-Wolfe 算法, 凹松弛函数, 凸松弛函数

引用格式 刘智勇. 图模型匹配: 一种新的凹松弛函数及算法. 自动化学报, 2012, **38**(5): 725−731

DOI 10.3724/SP.J.1004.2012.00725

Graph Matching: a New Concave Relaxation Function and Algorithm

LIU Zhi-Yong[1]

Abstract Recently, approximate graph matching based on relaxing the permutation matrix to a doubly stochastic matrix has become an important and popular topic. The key point lies in which approximation over a continuous set is usually easier to implement than that over a discrete one. However, a consequent trouble related to such a relaxation is how to properly map the doubly stochastic matrix back to a permutation one. In the literature, a concave relaxation function for matching problem between the undirected graphs without self-loops was recently proposed, such that the doubly stochastic matrix can converge to a permutation one in a smooth way, and got a state-of-art performance on matching accuracy. Unfortunately, except for the undirected graphs without self-loops, there are no concave relaxation proposed for any other types of graph models. In this paper, we propose a concave relaxation for the directed graphs without self-loops, based on which a graph matching algorithm is then presented. Extensive experimental comparisons witness the validity of the proposed methods.

Key words Graph matching, Frank-Wolfe algorithm, concave relaxation, convex relaxation

Citation Liu Zhi-Yong. Graph matching: a new concave relaxation function and algorithm. *Acta Automatica Sinica*, 2012, **38**(5): 725−731

图模型是一种典型的结构描述模型, 除了节点本身, 它还可以描述节点与节点之间的关系. 这种丰富的描述能力使得图模型在很多场合都得到应用. 如在计算机视觉和模式识别中, 图模型经常用于描述目标特征[1], 相对于基于向量的特征表示, 图模型可以清晰地描述特征与特征之间的某种相互关系, 这使得它特别适合描述一些复杂目标或场景, 如遥感图像等. 在很多基于图模型的应用中, 图匹配是其不可或缺的重要一环. 例如当利用图模型来描述图像内容, 那么图像匹配的本质其实就是图匹配[2]. 图匹配算法还在其他很多领域得到应用, 例如用于蛋白质相互作用 (Protein-protein interaction, PPI) 网络的对齐和匹配[3] 以及搜索引擎的排名计算[4] 等.

图匹配问题本质上是一个 NP 问题, 精确求解两个大小为 N 的图匹配问题的复杂度是 $O(N!)$. 所以除非图的规模很小, 否则在实际应用中一般都利用近似算法, 在问题的求解精度和所需时间之间寻求平衡. 近三十年来, 人们从不同的角度提出了很多图匹配的近似算法. 它们大概可以分成三大类: 基于树匹配近似的方法[5−6]、基于谱分解的方法[7] 和基于问题松弛的方法[8−11]. 基于树匹配的方法是将图模型近似为树结构以降低算法的复杂度, 此类方法的性能与待匹配的图模型结构有很大关系, 当图的结构和树结构差别较大时此类方法将产生较大误

收稿日期 2011-09-14 录用日期 2011-11-20
Manuscript received September 14, 2011; accepted November 20, 2011
国家自然科学基金 (60975002) 资助
Supported by National Natural Science Foundation of China (60975002)
本文责任编委 刘一军
Recommended by Associate Editor LIU Yi-Jun
1. 中国科学院自动化研究所复杂系统管理与控制国家重点实验室, 北京 100190
1. State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190

差. 谱分解方法建立在图像的邻接矩阵 (Adjacency matrix) 特征分解基础上. 当两个图是同构的, 那么它们的邻接矩阵具有相同的特征值和特征向量, 因此谱分解的方法尤其适合用于解决同构图的匹配问题. 但这个结论反过来却不成立, 即邻接矩阵具有相同的特征值和特征向量的两个图却不一定是同构的, 所以谱分解并不是基于一种合理的距离度量基础上的. 这就意味当两个图不是同构时, 谱分解方法得到的匹配结果将有可能非常差.

基于问题松弛的方法是近年来图匹配近似方法研究中的一个重要方向. 它本质上将图匹配的离散组合优化问题转化成为一个连续优化问题, 然后得到此连续优化问题的一个近似解, 最后将这个连续域上得到的解通过某种方式映射回一个离散解. 典型的基于问题松弛的图匹配方法包括标号松弛 (Relaxation labeling)[8−9]、逐步分配 (Graduated assignment)[10]、路径跟随 (Path following)[11] 等. 标号松弛算法对其中一个图的每个节点赋予一个概率标号, 然后通过计算节点的连接度或者编辑距离 (Edit distance) 来更新标号. 标号松弛算法的一个最大问题在于它不能保证每个标号只是对应到一个节点, 因为它采用的是一种从节点到标号的单向对应策略. 逐步分配和路径跟随这两种算法都直接基于图的邻接矩阵定义目标函数, 且都是将问题中的置换矩阵放松为双随机矩阵 (Doubly stochastic matrix). 给定两个具有相同尺寸的图模型 $G_D = (V_D, E_D)$ 和 $G_M = (V_M, E_M)$, 这里 V 和 E 分别表示图的节点和边集合, 图匹配问题可用下式表示:

$$P = \arg\min_{P\in\mathcal{P}} f(A_D, A_{P(M)}),\ \ A_{P(M)} = PA_MP^{\mathrm{T}} \tag{1}$$

式中 $\mathcal{P}$ 表示置换矩阵集, $A_{P(M)}$ 表示图 M 的邻接矩阵经过置换矩阵 P 变换得到的邻接矩阵. 逐步分配和路径跟随算法中的松弛方法都将上述问题中的定义域 $\mathcal{P}$ 放松为双随机矩阵集, 即 $P \in \mathcal{D}$. 不同的是, 逐步分配算法采用了一种更为一般的方式来定义代价函数 $f(\cdot)$, 而路径跟随算法的代价函数采用了邻接矩阵差的 Frobenius 范数平方. 更重要的不同在于, 逐步分配算法通过引入一个参数来控制目标函数是否或在多大程度上远离凸函数. 此参数一开始设置很小, 对应的目标函数接近凸函数, 因此可以求得其全局最优解 (一般为一个双随机矩阵). 然后此参数逐步增大, 当大到一定程度后双随机矩阵则慢慢逼近置换矩阵, 从而得到问题的求解. 但这个过程和原匹配问题并不等价, 即使此过程可以收敛到全局最优解也不能保证为原匹配问题的最优解. 而且逐步分配算法采用了一种双归一化的粗糙操作来约束 P 为一双随机矩阵, 因此当用于解决非同构图的匹配问题时, 逐步分配算法得到的解往往不尽如人意. 另一方面, 针对于无向无自环图 (Undirected graphs without self-loops) 的匹配问题路径跟随算法引入了一种凹松弛函数, 它和图匹配原问题具有相同的全局最优点. 通过将此凹松弛函数和凸松弛函数 (见式 (3)) 以不同的权重进行线性组合路径跟随算法提出了一系列的目标函数, 同时在算法中逐步提高凹松弛函数在目标函数中的比重, 目标函数将逐步变成一个凹函数, 则算法将会收敛到一个置换矩阵 (凹函数的极小值在其凸包的顶点即置换矩阵处得到). 从另外一个角度来看, 这个过程也可视为优化此凹松弛函数的一类路径跟随算法. 凹松弛函数和图匹配原问题具有相同的最优解, 但欲找到此最优解同样也是一个 NP 问题. 而凸凹松弛两个函数为同一问题的松弛函数, 所以凸函数得到的解为此算法提供了一个较为合适的初始值, 通过逐步提高凹函数的比重, 此过程本质上提供了一种用于优化此凹松弛函数的 Gradual nonconvexity 算法[12]. 当以一个合适的步长来逐步提高凹函数的比例, 此算法取得了非常优异的匹配效果[11]. 但是找到图匹配问题的凹松弛函数是非常困难的, 因为图匹配问题的目标函数 (见式 (2) 和式 (6)) 本质上是一个凸函数. 文献中只有路径跟随算法针对于无向无自环图匹配问题提出了一种凹松弛函数, 而对于其他类型的图模型匹配问题则还没有提出相应的凹松弛函数.

在本文中, 我们将提出一种针对于有向无自环图 (Directed graphs without self-loops) 匹配问题的凹松弛函数, 并在此基础上提出一种类似于路径跟随的图匹配算法. 在几种常用图匹配算法的对比测试中, 本文方法取得了优异的匹配精度. 本文结构如下: 第 1 节简要回顾路径跟随算法. 第 2 节中提出一种针对于有向无自环图匹配问题的凹松弛函数, 并基于此提出相应的图匹配算法. 第 3 节对比测试几种常用的算法, 最后第 4 节为本文结论.

1 路径跟随算法简要回顾

当式 (1) 中的代价函数选取为邻接矩阵差的 Frobenius 范数平方, 图匹配问题可以写成如下形式:

$$\begin{aligned}\min F(P) = &\parallel A_D - PA_MP^{\mathrm{T}} \parallel_{\mathrm{F}}^2 = \\ &\parallel A_DP - PA_M \parallel_{\mathrm{F}}^2 = \\ &\mathrm{vec}(P)^{\mathrm{T}}Q\mathrm{vec}(P)\\ \text{s.t.}\ \ &P \in \mathcal{P}\end{aligned} \tag{2}$$

上式中的

$$Q = (I \otimes A_D - A_M^{\mathrm{T}} \otimes I)^{\mathrm{T}}(I \otimes A_D - A_M^{\mathrm{T}} \otimes I)$$

是一个对称的正定实矩阵, 但是在同构图的匹配问题中退化成了半正定实矩阵, 式中的 $\otimes$ 表示矩阵张量乘. 当把上述函数中的 P 从置换矩阵放松为双随机阵, 那么就得到了图匹配问题的如下凸松弛问题, 即

$$\begin{aligned} &\min F_0(P) = \mathrm{vec}(P)^{\mathrm{T}} Q \mathrm{vec}(P) \\ &\text{s.t. } P \in \mathcal{D} \end{aligned} \tag{3}$$

其中, $\mathcal{D}$ 表示双随机矩阵的集合. 因为双随机矩阵集是置换矩阵集的凸包 (Convex hull), 那么上述问题就变成了一个约束凸优化问题, 它的全局最优解可以通过某种凸规划方法得到, 例如 Frank-Wolfe 算法. 但凸规划得到的解一般来讲并不在凸包的顶点 (在此问题中 $\mathcal{D}$ 的顶点即为置换矩阵), 因此通过凸松弛得到的解还需要以某种方式映射成为一个置换矩阵. 通常来讲这种映射可以通过最大分配 (Maximal assignment) 原则得到, 即

$$X = \arg\max_{X \in \mathcal{P}} \langle X, P \rangle \tag{4}$$

上述 $\langle X, P \rangle$ 表示矩阵 X 和 P 的 Frobenius 内积. 此最大分配问题可利用匈牙利算法[13] 进行求解, 从而得到一种图匹配的解. 这种图匹配算法称为 QCV (Quadratic convex) 算法. 但这种强制把凸松弛问题得到的双随机矩阵映射为置换矩阵的方法会带来较大的误差, 这一点也可以通过后面的实验比较反映出来. 为了使得 P 矩阵可以平滑地收敛到一个置换矩阵, 且保证算法和原问题具有相同的最优解, 路径跟随算法提出了下面的针对于无向无自环图的凹松弛函数:

$$\begin{aligned} F_1(P) = &- \mathrm{tr}(\triangle P) - \\ &2\mathrm{vec}(P)^{\mathrm{T}}(L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}})\mathrm{vec}(P), \ \ P \in \mathcal{D} \end{aligned} \tag{5}$$

其中, $\triangle_{ij} = (D_M(i,i) - D_D(j,j))^2$ [1], D 和 L 分别表示图的 Degree 矩阵和 Laplacian 矩阵. 上述函数之所以为图匹配问题的凹松弛函数源自于以下两点. 首先, 无向图意味着 L_M 和 L_D 都是对称的半正定矩阵, 所以 $L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}}$ 是一个对称的半正定矩阵, 也就是说上述函数是一个凹函数. 其次, 当 $P \in \mathcal{P}$ 最小化函数 $F_1(P)$ 等价于图匹配的原问题 (见式 (2)), 即

$$\arg\min_{P \in \mathcal{P}} F(P) = \arg\min_{P \in \mathcal{P}} F_1(P) \tag{6}$$

[1] 原文献 [11] 中的 i, j 应该交换位置.

但全局最小化上述凹函数也是一个 NP 问题, 所以为了近似优化上述的凹松弛函数, 路径跟随算法将上述的凸凹松弛函数线性组合得到如下的一系列目标函数:

$$F_\gamma(P) = \gamma F_0(P) + (1-\gamma)F_1(P), \ \ P \in \mathcal{D} \tag{7}$$

式中 $\gamma \in [0,1]$ 控制目标函数多大程度上非凸 (或非凹). 较大的 γ 意味着 $F_\gamma(P)$ 倾向于凸函数, 而较小的 γ 意味着 $F_\gamma(P)$ 倾向于凹函数. 当 γ 逐渐从 1 减小到 0, 最小化上述目标函数也就最终收敛到了一个置换矩阵, 也即得到了凹松弛函数的一个局部最优解.

此算法本质上类似用于凹函数优化的 Gradual nonconvexity 算法, 即一开始通过凸函数优化得到较为接近最优解区域的一个初始值, 然后从此初始值出发, 通过逐步提高目标函数的非凸性从而逼近原优化问题. 在无向无自环图的匹配问题上, 路径跟随算法表现出了优异的匹配性能. 需要指出的是, 虽然凹松弛函数具有和原问题同样的最优解, 但上述的算法并不能保证其收敛到凹函数的全局最优解, 因为目标函数具有大量的局部极小值 (最多至 $N!$ 个), 并且路径跟随算法本质上是一种局部搜索的算法. 另一方面, 上述的凹松弛函数只适用于无向无自环图的匹配问题, 因为当用于有向图时, 不对称的 $L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}}$ 并不能保证为一个半正定矩阵[14], 即函数 F_1 不能保证为凹函数. 下面我们将针对于有向无环图的匹配问题提出一个凹松弛函数, 进而在此基础上提出一种图匹配算法.

2 一种新的凹松弛函数及图匹配算法

2.1 凹松弛函数

下面我们首先构建一个凹函数, 然后在此基础上提出一种针对于有向无自环图匹配问题的凹松弛函数. 凹函数构建如下:

$$F_c(P) = -\mathrm{vec}(P)^{\mathrm{T}}(S_M \otimes S_D)\mathrm{vec}(P), \ \ P \in \mathcal{D} \tag{8}$$

式中 S_M 和 S_D 分别定义如下:

$$\begin{aligned} S_M &= L_M + L_M^{\mathrm{T}} - \sigma_M I \\ S_D &= L_D + L_D^{\mathrm{T}} - \sigma_D I \end{aligned} \tag{9}$$

σ_M 和 σ_D 分别表示对称矩阵 $L_M + L_M^{\mathrm{T}}$ 和 $L_D + L_D^{\mathrm{T}}$ 最小的特征值. 对于有向无自环图的匹配问题, 这两个参数一般来讲为负数. 显然 S_M 和 S_D 皆为对称实矩阵, 同时, 这两个矩阵的特征值皆不小于 0, 所以这两个矩阵皆为半正定对称实矩阵, 也就意味 $S_M \otimes S_D$ 也为一个半正定矩阵. 因此, $F_c(P)$ 是一个凹函数.

进一步分析函数 $F_c(P)$:

$$\begin{aligned}F_c(P) = &-\mathrm{vec}(P)^{\mathrm{T}}(S_M \otimes S_D)\mathrm{vec}(P) = \\ &-\mathrm{vec}(P)^{\mathrm{T}}[L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D + \\ &L_M^{\mathrm{T}} \otimes L_D + L_M \otimes L_D^{\mathrm{T}} - \sigma_M I \otimes (L_D^{\mathrm{T}} + L_D) - \\ &\sigma_D(L_M^{\mathrm{T}} + L_M) \otimes I + \sigma_M\sigma_D I]\mathrm{vec}(P) = \\ &-\mathrm{vec}(P)^{\mathrm{T}}(L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D)\mathrm{vec}(P) - \\ &\mathrm{vec}(P)^{\mathrm{T}}[L_M^{\mathrm{T}} \otimes L_D + L_M \otimes L_D^{\mathrm{T}} - \sigma_M I \otimes (L_D^{\mathrm{T}} + \\ &L_D) - \sigma_D(L_M^{\mathrm{T}} + L_M) \otimes I + \sigma_M\sigma_D I]\mathrm{vec}(P) = \\ &-\mathrm{vec}(P)^{\mathrm{T}}(L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D)\mathrm{vec}(P) - \\ &\mathrm{vec}(P)^{\mathrm{T}}B\mathrm{vec}(P) \end{aligned} \tag{10}$$

定义

$$\begin{aligned}B = &L_M^{\mathrm{T}} \otimes L_D + L_M \otimes L_D^{\mathrm{T}} - \\ &\sigma_M I \otimes (L_D^{\mathrm{T}} + L_D) - \\ &\sigma_D(L_M^{\mathrm{T}} + L_M) \otimes I + \sigma_M\sigma_D I\end{aligned}$$

不难看出 B 是一个对称的实矩阵. 利用矩阵 B 的特征值分解, 下面我们找出 $-\mathrm{vec}(P)^{\mathrm{T}}B\mathrm{vec}(P)$ 项的一个下界,

$$\begin{aligned}&-\mathrm{vec}(P)^{\mathrm{T}}B\mathrm{vec}(P) = -\mathrm{vec}(P)^{\mathrm{T}}U\Lambda U^{\mathrm{T}}\mathrm{vec}(P) \geq \\ &\quad -\lambda_{\max}\mathrm{vec}(P)^{\mathrm{T}}UU^{\mathrm{T}}\mathrm{vec}(P) = \\ &\quad -\lambda_{\max}\mathrm{vec}(P)^{\mathrm{T}}\mathrm{vec}(P)\end{aligned} \tag{11}$$

上式中 $\lambda_{\max}$ 表示矩阵 B 的最大特征值. 因此,

$$\begin{aligned}0 \geq F_c(P) \geq &-\mathrm{vec}(P)^{\mathrm{T}}[L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + \\ &L_M \otimes L_D + \lambda_{\max}I]\mathrm{vec}(P), \ \ P \in \mathcal{D}\end{aligned} \tag{12}$$

因为 $L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D + \lambda_{\max}I$ 是对称实矩阵, 且上述二项式始终不大于 0, 所以 $-\mathrm{vec}(P)^{\mathrm{T}}(L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D + \lambda_{\max}I)\mathrm{vec}(P)$ 是一个凹函数. 在此基础上, 我们最终得到如下针对于有向无自环图匹配问题的凹松弛函数:

$$\begin{aligned}F_2(P) = &-\mathrm{tr}(\triangle P) - \mathrm{vec}(P)^{\mathrm{T}}\left[L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + \right. \\ &\left. L_M \otimes L_D + \lambda_{\max}I\right]\mathrm{vec}(P)\end{aligned} \tag{13}$$

上述函数显然是一个凹函数, 因为它只是在一个凹函数的基础上增加了一个线性项. 欲说明上述函数是针对于有向无自环图匹配问题的凹松弛函数, 我们需要进一步证实当 $P \in \mathcal{P}$, 最小化函数 $F_2(P)$ 等价于最小化图匹配问题的原始目标函数. 这一点说明如下:

$$\begin{aligned}\arg\min_{P\in\mathcal{P}} F_2(P) = &\arg\min_{P\in\mathcal{P}}\{-\mathrm{tr}(\triangle P) - \mathrm{vec}(P)^{\mathrm{T}} \times \\ &[L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D + \lambda_{\max}I]\mathrm{vec}(P)\} = \\ &\arg\min_{P\in\mathcal{P}}\{-\mathrm{tr}(\triangle P) - \mathrm{vec}(P)^{\mathrm{T}} \times \\ &[L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}} + L_M \otimes L_D]\mathrm{vec}(P) - \lambda_{\max}N\} = \\ &\arg\min_{P\in\mathcal{P}}\{-\mathrm{tr}(\triangle P) - 2\mathrm{vec}(P)^{\mathrm{T}} \times \\ &[L_M^{\mathrm{T}} \otimes L_D^{\mathrm{T}}]\mathrm{vec}(P)\} = \\ &\arg\min_{P\in\mathcal{P}} F_1(P) = \arg\min_{P\in\mathcal{P}} F(P)\end{aligned} \tag{14}$$

因此, 函数 $F_2(P)$ 为有向无自环图匹配问题的一个凹松弛函数.

我们需要进一步求得 $F_2(P)$ 中的参数 $\lambda_{\max}$, 即 B 的最大特征值. B 是一个 $N^2 \times N^2$ 的矩阵, 当 N 较大时, 直接求解 $\lambda_{\max}$ 的计算量将非常大, 所以转而求 $\lambda_{\max}$ 的一个上界. 给定一个 $n \times n$ 的对称实矩阵, 假设它最大的元素为 b, 最小的元素为 a, 那么它的特征值的上界由以下公式给出[15]:

若 $a < -|b|$:

$$\lambda_{\max} \leq \begin{cases}\dfrac{n(b-a)}{2}, & n \text{ 为奇数} \\ \dfrac{\left(nb + \sqrt{b^2 + (n^2-1)a^2}\right)}{2}, & n \text{ 为偶数}\end{cases}$$

若 $a \geq -|b|$:

$$\lambda_{\max} \leq na$$

精确计算矩阵 B 的最大和最小值也有较大的复杂度 ($\mathrm{O}(N^4)$), 估算其下界和上界如下:

$$\begin{aligned}a = &\ \min\{L_M^{\mathrm{T}} \otimes L_D\} + \min\{L_M \otimes L_D^{\mathrm{T}}\} - \\ &\max\{\sigma_M I \otimes (L_D^{\mathrm{T}} + L_D)\} - \\ &\max\{\sigma_D(L_M^{\mathrm{T}} + L_M) \otimes I\} + \min\{\sigma_M\sigma_D I\}\end{aligned}$$

$$\begin{aligned}b = &\ \max\{L_M^{\mathrm{T}} \otimes L_D\} + \max\{L_M \otimes L_D^{\mathrm{T}}\} - \\ &\min\{\sigma_M I \otimes (L_D^{\mathrm{T}} + L_D)\} - \\ &\min\{\sigma_D(L_M^{\mathrm{T}} + L_M) \otimes I\} + \max\{\sigma_M\sigma_D I\}\end{aligned}$$

上述两个公式中计算后面三项的复杂度为 $\mathrm{O}(N^2)$. 对于前面两项, 不难发现 $\min\{L_M^{\mathrm{T}} \otimes L_D\} = \min\{L_M \otimes L_D^{\mathrm{T}}\}$ 且 $\max\{L_M^{\mathrm{T}} \otimes L_D\} = \max\{L_M \otimes L_D^{\mathrm{T}}\}$. 假设 L_M 和 L_D 的最大和最小值分别求得为 ζ_M, η_M 和 ζ_D, η_D, 那么有:

$$\begin{aligned}\min\{L_M^{\mathrm{T}} \otimes L_D\} &= \min\{\zeta_M\eta_D, \eta_M\zeta_D\} \\ \max\{L_M^{\mathrm{T}} \otimes L_D\} &= \max\{\zeta_M\zeta_D, \eta_M\eta_D\}\end{aligned}$$

所以计算最大最小值上下界的复杂度皆为 $\mathrm{O}(N^2)$.

最后, 在得到有向无自环图匹配问题的凹松弛函数 $F_2(P)$ 的基础上, 类似于路径跟随算法我们提出以下的一系列目标函数:

$$F_\gamma(P) = \gamma F_0(P) + (1-\gamma)F_2(P),\ \ P \in \mathcal{D} \quad (15)$$

2.2 基于 Frank-Wolfe 的图匹配算法

在以下提出的图匹配算法中, 目标函数中的 γ 一开始设置为 1, 然后逐步减小至 0. 这样, 目标函数就从凸函数逐渐变成一个凹函数, 从而使得 P 逐渐收敛至一个置换矩阵. 算法的整体框架如下:

算法 1.

GraphMatching (A_D, A_M)

$P^n \leftarrow 1_{N\times N}/N$, $n \leftarrow 0$, $\gamma \leftarrow 1$

while $\gamma \geq 0$ & $P^n \notin \mathcal{P}$

　　do $P^{n+1} \leftarrow$ Frank-Wolfe(P^n, γ)

　　$\gamma \leftarrow \gamma - \delta\gamma$, $n \leftarrow n+1$

return (P^n)

算法中 $\delta\gamma$ 是 γ 的变化步长, 它的选取原则是使得在连续两次迭代之间的目标函数之差恰好比一个预设常数小, 即如果小于此常数, 则 $\delta\gamma \leftarrow 2\delta\gamma$, 否则 $\delta\gamma \leftarrow \delta\gamma/2$. 对于每一个固定的 γ, 则使用 Frank-Wolfe 算法对目标函数进行优化. Frank-Wolfe 算法是一种典型的简约梯度算法, 它的基本原理是将当前得到的目标函数梯度方向投影到函数的约束域, 从而使得算法始终在约束域内执行. Frank-Wolfe 算法包括以下几个步骤[16]:

算法 2.

步骤 1. 初始化 $P^0 \leftarrow P^*$, $t \leftarrow 0$, 这里 P^* 表示上一次主迭代得到的 P, 即算法 1 中的 P^n.

步骤 2. 求解下述的线性规划问题

$$\begin{aligned} &\min \langle \nabla F_\gamma(P^t), X^t \rangle \\ &\text{s.t. } X^t \in \mathcal{D} \end{aligned} \quad (16)$$

得到一个置换矩阵 X^t(因为线性规划的解在凸包顶点处得到). 上述线性规划中的梯度 $\nabla F_\gamma(P)$ 由下式给出:

$$\begin{aligned} \nabla F_\gamma(P) &= \gamma \nabla F_0(P) + (1-\gamma)\nabla F_2(P) \\ \nabla F_0(P) &= 2(A_D^{\mathrm{T}} A_D P - A_D^{\mathrm{T}} P A_M - \\ &\qquad A_D P A_M^{\mathrm{T}} + P A_M A_M^{\mathrm{T}}) \\ \nabla F_2(P) &= -\triangle^{\mathrm{T}} - L_D^{\mathrm{T}} P L_M - L_G P L_M^{\mathrm{T}} - \lambda_{\max} P \end{aligned} \quad (17)$$

步骤 3. 线性搜索找出 $\alpha \in [0,1]$ 使得其最小化 $F_\gamma(P^t + \alpha(X^t - P^t))$, 同时更新 $P^{t+1} \leftarrow P^t + \alpha(X^t - P^t)$.

步骤 4. 如果 $\langle \nabla F_\gamma(P^{t+1}),\ P^{t+1} - X^t \rangle < \varepsilon |F_\gamma(P^{t+1}) + \langle \nabla F_\gamma(P^{t+1}), X^t - P^{t+1} \rangle|$, 输出 P^{t+1}, 退出; 否则, 使得 $t \leftarrow t+1$, 返回步骤 2.

上述 Frank-Wolfe 算法中, 步骤 2 中的线性规划可由匈牙利算法解决, 步骤 3 中的线性搜索可由 Backtracking 算法解决[17]. 整个算法的存储复杂度为 $\mathrm{O}(N^2)$, 而计算复杂度则大约为 $\mathrm{O}(N^3)$, 因为匈牙利算法和矩阵乘的计算复杂度皆为 $\mathrm{O}(N^3)$.

3 实验对比及分析

下面我们利用三个实验来比较四种图匹配算法: Umeyama 算法[7], 前面提到的 QCV 算法, 逐步分配算法 (Graduated assignment, GA)[10] 和本文所提出的算法 (Ours). GA 算法中的两个图模型的边之间的相似度量定义为

$$C_{aibj} = \begin{cases} 0, & G_{Dab} \text{或} G_{Mij} \text{为空} \\ 1 - (G_{Dab} - G_{Mij})^2, & \text{其他} \end{cases}$$

算法参数采用文献 [10] 中的参数. 本文算法中的参数 ε 设为 0.001. 所有算法基于 Matlab 完成, 但其中的匈牙利算法利用 C 程序生成的 mex 文件执行, 它的执行效率要远高于相对应的 m 文件.

实验中用到的数据分别描述如下: 第 1 个实验是在随机产生的两个图之间进行匹配, 第 2 个实验中的每个图模型对中的第二个图是在第一个图的基础上加上一定程度的噪声得到, 第 3 个实验中待匹配图的尺寸规模将逐步增大.

在第 1 个实验中随机产生具有 8 个节点的 100 对有向无自环图, 每个图模型的产生方式如下: 1) 设置一个常数 $\eta \in [0,1]$ (对应于图的稀疏度); 2) 对应邻接矩阵中的每个非对角元素, 产生一个随机数 $\sigma \in [0,1]$; 3) 如果 $\sigma > \eta$, 设置此非对角元素为一个随机数 $\zeta \in [0,1]$; 否则此元素为空, 即 0. 因为图的尺寸相对较小, 所以可以利用穷搜索 (Optimal result, OPT) 的方法得到它们之间的最优匹配作为对比. 五种方法得到的实验结果列于表 1 中.

表 1 中的均值和标准差表示 100 次匹配结果的均值和方差. 从表 1 的结果可以看出, 本文提出的方法要远远好于其余的三种方法, 同时也说明对于解决将凸松弛得到的双随机矩阵映射回置换矩阵的问题, 基于凸凹松弛结合的逐步收敛的方法 (Ours) 要远好于基于最大分配的直接投影方法 (QCV).

在第 2 个实验中, 我们将定义一个噪声水平参数 $\alpha \in [0,1]$. 在随机产生得到第一个图模型 A_D 的基础上, 第二个图模型 A_M 将依据以下步骤产生:

步骤 1. 设置 $A_M \leftarrow A_D$, 并且对每个 $(A_M)_{ij}$, $i \neq j$, 产生两个随机数 r_1 和 $r_2 \in [0,1]$.

表 1 五种方法在 100 组随机有向无自环图匹配中的实验精度对比结果

Table 1 Experimental results of the matching accuracies of the five algorithms on 100 random generated graph pairs

匹配差 (Matching error)	OPT	Umeyama	QCV	GA	Ours
均值 (Mean)	5.2754	11.8887	8.3875	8.7470	6.1231
标准差 (Standard deviation)	0.8102	2.2723	1.9357	1.6678	1.2435
最佳匹配次数 (No. of optimal matching)	100	0	1	0	21

步骤 2. 如果 $(A_M)_{ij} > 0$: 如果 $r_1 < \alpha$, $(A_M)_{ij} \leftarrow 0$; 否则, $(A_M)_{ij} \leftarrow (A_M)_{ij} + \alpha r_2$.

步骤 3. 如果 $(A_M)_{ij} = 0$: 如果 $r_1 < \alpha$, $(A_M)_{ij} \leftarrow r_2$.

步骤 4. 随机产生一个置换矩阵 P, 设置 $A_M \leftarrow PA_MP^{\mathrm{T}}$.

我们设置 11 个噪声水平, 即 $\alpha = 0, 0.1, 0.2, \cdots, 1$, 然后在每个噪声水平上, 随机产生 50 对尺寸为 8 的图模型, 然后统计在每个噪声水平上的匹配误差. 实验结果显示在图 1 中. 从实验结果中可以观察到以下几点:

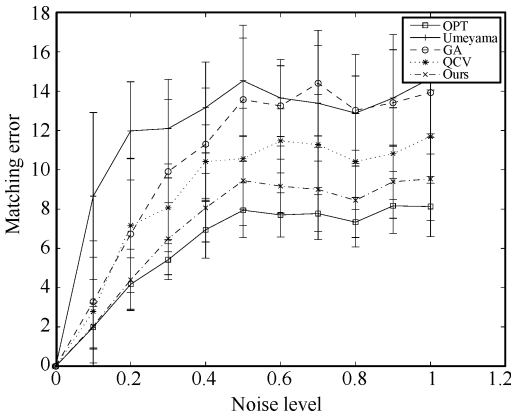

图 1 五种图匹配方法在不同噪声水平下的匹配结果

Fig. 1 Matching accuracies of the five algorithms for matching graphs with different noise levels

1) 不同的方法在噪声为 0 时, 即同构图的匹配中都可以得到最优匹配.

2) Umeyama 的方法最容易受噪声影响, 这一点可以从 $\alpha = 0.1$ 时的表现观察得到. 当噪声较小时, Umeyama 方法却产生了很大的误差. 这一点和我们前面的论断是一致的, 即当待匹配的两个图稍远离同构时, 谱分析可能产生很大的误差.

3) 当待匹配的图模型不同构时, 本文提出的方法远好于其他三种算法.

在第 3 个实验中, 除了比较不同算法的匹配精度, 我们还将比较它们的计算复杂度. 实验共分 12 组, 每组中待匹配图的尺寸规模逐渐增大, 从 5 以步长 5 增加到 60. 每组包含有 50 对图模型, 每一对都以固定噪声水平 $\alpha = 0.2$, 依据实验 2 中数据产生的方式产生. 匹配精度结果显示在图 2 中, 从图 2 可以看出, 本文提出的方法在不同尺寸规模的图模型匹配上都要远好于其他三种算法.

四种算法的平均计算时间显示在图 3 中. 图中横坐标表示的是图模型尺寸的对数值, 纵坐标是平均计算时间的对数值, 通过计算每条曲线的斜率可以得到每种算法的复杂度. 其中 QCV 和本文提出的

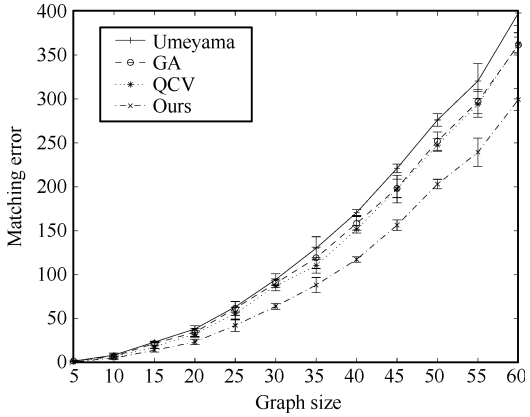

图 2 四种图匹配方法在不同图模型规模下的匹配结果

Fig. 2 Matching accuracies of the four algorithms for matching graphs with different sizes

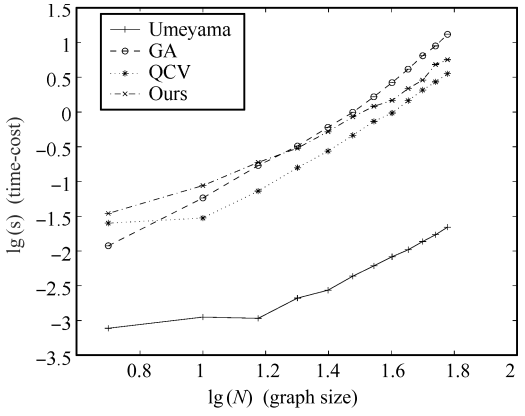

图 3 四种图匹配方法在不同图模型规模下的平均计算复杂度

Fig. 3 Average computational complexities of the four algorithms for matching graphs with different sizes

算法复杂度都大概在 $O(N^3)$ (分别为 $O(N^{2.76})$ 和 $O(N^{2.83})$, 而 GA 算法的复杂度要大于这两种算法, 达到了 $O(N^{3.62})$, Umeyama 算法复杂度最小, 只有大约 $O(N^{2.28})$.

4 结论

将图匹配问题中的置换矩阵放松为双随机矩阵从而得到图匹配问题的近似解是近年来图匹配方法研究中的一个重要方向, 但此类方法中存在一个问题是如何将得到的双随机矩阵重新映射回置换矩阵. 通过引入原问题的凹松弛函数, 最近提出的路径跟随算法在无向无自环图匹配问题上取得了非常好的匹配效果. 但除了无向无自环图, 文献中还没有针对其他类型图模型的凹松弛函数. 本文提出了一种针对于有向无自环图模型的凹松弛函数, 并给出了相应的图匹配算法. 几种常用图匹配方法的对比测试说明了本文所提出方法的有效性. 下一步的工作将针对其他类型的图模型匹配问题提出相应的凹松弛函数.

References

1 Eshera M A, Fu K S. An image understanding system using attributed symbolic representation and inexact graph-matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986, **8**(5): 604−618

2 Duchenne O, Bach F, Kweon I S, Ponce J. A tensor-based algorithm for high-order graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, **33**(12): 2383−2395

3 Singh R, Xu J B, Berger B. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences of the United States of America*, 2008, **105**(35): 12763−12768

4 Sambhoos K, Nagi R, Sudit M, Stotz A. Enhancements to high level data fusion using graph matching and state space search. *Information Fusion*, 2010, **11**(4): 351−364

5 Wu H, Ling T W, Dobbie G, Bao Z, Xu L. Reducing graph matching to tree matching for XML queries with ID references. *Lecture Notes in Computer Science*. Berlin: Springer, 2010. 391−406

6 Fernandez M L, Valiente G. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 2001, **22**(6−7): 753−758

7 Umeyama S. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988, **10**(5): 695−703

8 Luo B, Hancock R. Structural graph matching using the EM algorithm and singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001, **23**(10): 1120−1136

9 Zheng Kai-Jie, Gao Yu-Tao, Peng Ji-Gen. A new relaxation model for weighted graph matching. *Acta Automatica Sinica*, 2010, **36**(8): 1200−1203
(郑开杰, 高玉涛, 彭济根. 赋权图匹配问题的一种新的松弛模型. 自动化学报, 2010, **36**(8): 1200−1203)

10 Gold S, Rangarajan A. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1996, **18**(4): 377−388

11 Zaslavskiy M, Bach F, Vert J P. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009, **31**(12): 2227−2242

12 Blake A, Zisserman A. *Visual Reconstruction*. Cambridge: The MIT Press, 1987

13 Kuhn H W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 1955, **2**(1−2): 83−97

14 Li J S. Kronecker products of positive semidefinite matrices. *Journal of Mathematical Research and Exposition*, 1997, **17**(3): 327−334

15 Zhan X Z. Extremal eigenvalues of real symmetric matrices with entries in an interval. *SIAM Journal on Matrix Analysis and Applications*, 2006, **27**(3): 851−860

16 Frank M, Wolfe P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956, **3**(1−2): 95−110

17 Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge: Cambridge University Press, 2004

刘智勇 中国科学院自动化研究所复杂系统管理与控制国家重点实验室副研究员. 主要研究方向为模式识别, 图像处理, 机器学习.
E-mail: zhiyong.liu@ia.ac.cn
(**LIU Zhi-Yong** Associate professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interest covers pattern recognition, image processing, and machine learning.)