优化GTS的启发式搜索算法

成员：向东伟、林天梁

GTS的A*算法简述

GTS在本质上可以抽象成为一个搜索问题：起始状态（搜索树）为输入的实例Graph，对节点适用的rules拓展搜索树，到达新的可达状态。对于搜索树的遍历方式，通常具有广度优先搜索和深度优先搜索。A*算法是对于深度优先搜索算法的一种有效优化算法。A*算法通过利用问题的规则和特点来制定一些启发规则，由此来改变节点的扩展顺序，将最有希望扩展出最优解的节点优先扩展，使得尽快找到最优解。

针对GTS系统应用A*算法的Python代码如下：

```
def astar(question, rules, goal):
    q = queue.PriorityQueue()
    q.put(QueueElem(question, [], 0))

    while not q.empty():
        elem = q.get()
        graph = elem.graph
        path = elem.path
        heuristicValue = elem.value

        if len(Maper.getMaps(graph, goal)) != 0:
            return (graph, path)

        for rule in rules:
            maps = Maper.getMaps(graph, rule)
            for map in maps:
                newGraph = Transformer.transform(rule.transRules, graph, map)
                newPath = list(path)
                newPath.append(rules.index(rule))
                newHValue = len(newPath) + getHeuristicValue(newGraph)
                q.put(QueueElem(newGraph, newPath, newHValue))

    return None
```

问题的再次建模

不难看出，设计启发式搜索算法的关键是估算起始节点经过该节点到达目标节点的最佳路径的代价 cost(G)。这个代价可以分为两部分：

1. f(G)：起始节点到当前节点的路径，这一部分是已知的
2. h(G)：该节点到达目标节点的最佳路径的代价，这部分未知。需要设计一个高效的算法对h(G)进行一个有效的估计（h(G)准确值是多项式时间难解的，否则也不需要设计搜索算法了）。

在*Heuristic Search for the Analysis of Graph Transition Systems*[1]中，作者提出一些在估价函数中可能需要考量的因素，例如：

1. 需要插入和删除的图节点
2. 每一个图节点出度与入度的变化
3. 当前图和目标图的二进制编码的 Hamming 距离

在*Heuristic Search-Based Planning for Graph Transformation Systems*[2]中，作者针对研究的问题实例提出了一些特定的考察因素。进一步，作者通过机器学习的方法（支持向量机）对每一因素设定相应的权重，采用这些因素的线性组合作为h(G)。

实质上，GTS问题类似于GED问题（图编辑距离）：通常的GED问题以增删一条边、一个顶点作为一个基本操作，而GTS以应用一条Rule作为一个基本操作。另一个区别在于GED要求增操作之后的结果与目标图完全的同构，而GTS要求操作所得graph的某一子图同构于目标图。从这个角度出发，h(G)可以考察G与目标图之间的相似度，更确切地说，**考察G与目标图之间未确定顶点映射关系，以rule为基本单位来衡量的相似度。**而且h(G)的时间复杂度必须多项式时间可解，最好是$O(max(\mid E(G) \mid, \mid V(G) \mid))$的。

优化h(G)：基于graph2vec

该问题的求解过程是一个不断应用实例对图进行转化进行深度检索的过程，在每一步的子图匹配中会消耗大量的算力，因此我们试图将GTS中应用规则，进行转换的检索过程泛化，在不丢失太多精确度的情况下用向量化之后的线性操作替代子图匹配和转换的过程。

$Conjecture$ 1.我们假定图转换问题中规则的应用有两条非常良好的性质：

1. 任意图中都有与每一种规则的LHS相匹配的同构子图
2. 规则的适配顺序不影响图转换的结果，即应用规则的操作具有可交换性

在上述两条假设的前提下，我们进行如下的操作：

利用性能足够良好的`graph2vec`，将原有的图转换为一个维度给定的向量。将匹配的目标$goal$也可转化为一个向量。如果能实现将规则$rule$向量化，即可对当前图到目标的步数作出预计。

$Conjecture$ 2.假定

$$
\begin{gathered}
for\ graph2vec(G) : G \to R^m \\
\forall V_1 \cap V_2 = \phi \qquad graph2vec(G_1 \cup G_2) = graph2vec(G_1) + graph2vec(G_2) \\
\forall V_2 \subseteq V_1\ and\ E_2 \subseteq E_1 \qquad graph2vec(G_1 - G_2) = graph2vec(G_1) - graph2vec(G_2)
\end{gathered}
$$

即`graph2vec`为一特定条件下的同态映射。

在满足$Conjecture$ 1&2的情况下，我们假设当前图G应用$rule_i$之后成为G'，则有

$$
\begin{gathered}
\forall G - LHS_i \cup RHS_i = G' \\
graph2vec(G) + graph2vec(RHS_i) - graph2vec(LHS_i) = graph2vec(G')
\end{gathered}
$$

则可定义映射

$$
\begin{gathered}
\Gamma : < LHS, RHS, NAC > \to R^m \quad s.t. \\
\Gamma(rule_i) = graph2vec(RHS_i) - graph2vec(LHS_i)
\end{gathered}
$$

由此，在给定当前图G，目标$goal$的情况下，一个合理的转化路径

$$
G \xrightarrow{rule_{j_1}} G_1 \xrightarrow{rule_{j_2}} \ldots \xrightarrow{rule_{j_t}} G_t
$$

$$
\begin{gathered}
goal =< V_g, E_g >, G_t < V_t, E_t > \\
V_g \subseteq V_t, E_g \subseteq E_t
\end{gathered}
$$

可以向量化表示为

$$graph2vec(G)+\sum_{i=1}^{t}\Gamma(rule_{j_i})=graph2vec(G_t)$$

因此原问题：

给定图G，目标$goal$，一系列规则$rule=<LHS,RHS,NAC>$，预估由当前图G到一个目标图使得$goal$为其子图的预计距离

可以转化为两种方案：

一、

给定向量$a=graph2vec(G)$，目标向量$g=graph2vec(goal)$，一系列规则向量$r_i=\Gamma(rule_i),i=1,2,\dots,m$，寻找适配规则序列后得到的向量与目标向量的几何距离最短的情况下，最小的适配次数。

即

$$\min\sum_{i=1}^{m}\alpha_i$$
$$s.t.\min|a+\sum_{i=1}^{m}\alpha_i r_i-g|$$

考虑向量列$\{r_i\}$是否线性无关：

i. 如果所有向量均线性无关，则最小化问题转化为以下问题：

令$v=a-g$，则问题转化为：

$$\min L=[(\alpha_1 r_{11}+\alpha_2 r_{21}+\cdots+\alpha_m r_{m1}+v_1)^2+\cdots+(\alpha_1 r_{1n}+\alpha_2 r_{2n}+\cdots+\alpha_m r_{mn}+v_1)^2]$$

则分别对α_i求导得：

$$\frac{\partial L}{\partial\alpha_1}=2r_{11}(\alpha_1 r_{11}+\alpha_2 r_{21}+\cdots+\alpha_m r_{m1}+v_1)+\cdots+2r_{1n}(\alpha_1 r_{1n}+\alpha_2 r_{2n}+\cdots+\alpha_m r_{mn}+v_1)=0$$

可知每一个都得到一个线性方程，即为一个m元1次方程组，共m个方程。

则系数矩阵为

$$A=\begin{bmatrix}r_{11}^2+r_{12}^2+r_{13}^2+\cdots+r_{1n}^2 & r_{11}r_{21}+r_{12}r_{22}+\cdots+r_{1n}r_{2n} & \dots & r_{11}r_{m1}+r_{12}r_{m2}+\dots r_{1n}r_{mn}\\ r_{21}r_{11}+r_{22}r_{12}+\cdots+r_{2n}r_{1n} & r_{21}^2+r_{22}^2+\cdots+r_{2n}^2 & \dots & r_{21}r_{m1}+r_{22}r_{m2}+\dots r_{2n}r_{mn}\\ \vdots & \ddots & & \vdots\\ \vdots & & \ddots & \vdots\\ r_{m1}r_{11}+r_{m2}r_{12}+\cdots+r_{mn}r_{1n} & \dots & \dots & r_{m1}^2+r_{m2}^2+r_{m3}^2+\cdots+r_{mn}^2\end{bmatrix}$$

则可定义矩阵

$$R=\begin{bmatrix} r_{11} & r_{21} & \cdots & r_{m1} \\ r_{12} & r_{22} & \cdots & r_{m2} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ r_{1n} & r_{2n} & \cdots & r_{mn} \end{bmatrix}$$

可观察得$A=R^TR$，由于R列满秩，故$rank(R)=min\{n,m\}$

由于$rank(R^TR)=rank(R)$（用基础解系可证）

可得$rank(A)=rank(R)=min\{n,m\}$.

则：当$n\geq m$时，矩阵A满秩，故一定有解。

当$n<m$时，矩阵A不满秩序，因此有很多解或无解。但是由于向量的维度是$graph2vec$的一个先觉变量，且n的增大能够增加映射的信息量，因此我们可以人为将向量的维度定在至少大于m的水平上。

综上，上述方程一定有解。如果存在负解，为了方便处理，我们直接将负解定为0.（直观上貌似这样仍然是最小，但是暂时还没有数学证明）

ii. 如果向量之间线性相关，则进行如下讨论：

- 寻找向量$r_p=t_1r_1+\cdots+t_mr_m,t_i\geq 0$
  - 将$\sum_{i=1}^m t_i\leq 1$的向量直接丢弃.
  - 将$\sum_{i=1}^m t_i>1$的向量记录系数之后暂时丢弃.
- 对于丢弃结束之后的向量，可得到k个互相之间不存在正系数线性表出的向量。可以利用上述方法进行同样的处理，可以得到一组正表出系数α_i.
- 对每一个之前暂时丢弃的向量做处理，令$\theta=\min\frac{\alpha_i}{t_i}$，调整$\alpha_i:=\alpha_i-\theta t_i$.将当前的向量的系数调整为1.则可得整体$\alpha$之和缩小.
- 全部处理完毕之后得到最终的规模估计

考虑到最终我们需要的终点位置与目标$goal$向量需要满足的是子图同构关系而非同构，因此预估空间距离可能不够精确，我们引入第二种等价方法：

二、

给定向量$a=graph2vec(G)$，目标向量$g=graph2vec(goal)$，一系列规则向量$r_i=\Gamma(rule_i),i=1,2,\ldots,m$，寻找适配规则序列后得到的向量与目标向量的空间夹角最小的情况下，最小的适配次数。

即问题转化为

$$\max_{\alpha_i\geq 0}\cos\theta=\frac{(a+\sum_{i=1}^m\alpha_ir_i)dot(g)}{|a+\sum_{i=1}^m\alpha_ir_i\ ||\ g|}$$

当两者夹角最小时我们可以认为当前的终点与目标非常接近。

上述中a,g,r_i均为给定向量，因此即为一个m元函数.同时满足每个自变量大于0。

则为一KKT条件下的拉格朗日优化问题，利用拉格朗日方法可以解得上述问题的解。

误差与复杂度分析

我们在最开始讨论时设定的两个假设是非常强的条件，往往不足以满足，但是为了避免陷入运算上的低效，我们在不造成过多误差的条件下可以适当地将条件放宽。在复杂度方面，算法两个处理思路中有两个待解决的问题：

1. 如何快速的选择可丢弃的向量，即找到可被其他向量正系数表出的向量。
2. 如何快速的求导和求解方程，这两点需要进一步的讨论和优化。

综上，我们可以利用良好的$graph2vec$来进行两个方向的考虑，预估当前图到目标图的最短步数。

graph2vec

我们尝试提出一些`graph2vec`的方法，并且尝试简要分析其优劣。

Naïve method

将节点G映射为$(\mid V(G) \mid + \mid E(G) \mid)$维的特征向量。这样看似能够最大限度的保留原节点G的信息，但是对应用rule毫无意义：这样实际上rule的维度一定低于$graph2vec(G)$的维度，并不能合理的进行线性运算。因此这种方法暂时是不可行的。

Customed method

即人为选取节点G的一些特征构成特征向量。我们在这里列出一些与问题无关的，但较为重要的特征以供参考：

1. 顶点、边的个数
2. 顶点出度、入度的均值
3. 顶点中i-出度、i-入度的顶点个数
4. 顶点的i-距离邻居的类型、个数
5. 顶点的聚类系数[3]
6. 顶点邻居的平均度数
7. 顶点邻居的平均聚类系数[3]

如果针对特定问题，选取与问题高度相关的特征构成特征向量，则启发函数退化为*Heuristic Search-Based Planning for Graph Transformation Systems*[2]中描述的算法形式。但是，在$graph2vec$中，由于考虑到了"不同rule对这些特征改变的大小并不相同"这一问题，直观上看效果应该好于*Heuristic Search-Based Planning for Graph Transformation Systems*中描述的算法。事实上，这种方法已经很好的应用在GED问题上[3]。

根据特定问题人为选取特征，在**学习样本不足**或者**问题规模较小，rule、goal相对简单**的情况下不失为一种高效的方法。

ML methods

一种**问题无关**的搜索优化策略是：如果能够采用机器学习的方法，挖掘在特定GTS问题下Transfrom过程中最重要的特征改变，那么可以就可以让这些特征作为$graph2vec$的维度。

虽然目前Graph Embedding问题是一个机器学习/数据挖掘学界的热点，但是Graph Embedding主要是针对将图中单个节点向量化，即用一个低维，稠密的向量去表示图中的点，该向量表示能反映图中的结构。这些方法对于我们将graph转化为向量的目标没有帮助。

graph2vec的展望

目前，我们希望引入机器学习方法来解决h(G)的优化。目前存在的问题表现在以下三方面：

1. 如何产生机器学习的样本。我们的设想是采用蒙特卡洛方法生成一些可以应用较少rule生成能够匹配goal的问题实例。
2. 应用何种学习方法。这种学习方法需要能够挖掘在特定GTS问题下Transfrom过程中最重要的特征改变。
3. 不同特征的数值表示需要归一化。

另一h(G)优化思路：基于谱分解

基于矩阵计算的谱分解方法可应用于GED[4]，即把G_A和G_B的近似距离定义为

$$d(G_A\ ,\ G_B) = \sqrt{\sum(\lambda_{A_i} - \lambda_{B_i})^2}$$

其中λ_i为G_A邻接矩阵A的标准化拉普拉斯矩阵 ***Normalized Laplacian Matrix*** 的特征值：

$$Normalized\ Laplacian\ L_{norm} = D^{-1/2}AD^{-1/2}$$

如果需要应用在h(G)上，则还需确定邻接矩阵的非0项的数值，即确定：

$$w : EdgeType \rightarrow R$$
$$s.t.\ L[i][j] = w(Edge(i,j))$$

这样问题可以建模为：

$$h(G) = min(d(G_{sub}, goal)) = min(\sqrt{\sum(\lambda_{sub_i} - \lambda_{goal_i})^2})$$
$$s.t.\ G_{sub} \subset G,\ \ |\ V(G_{sub})\ | = |\ V(goal)|$$

这种方法的问题有：

1. 某些结构不同的graph具有相同的特征值序列，某些改变会导致特征值序列发生很大变化
2. SVD求解特征值的时间复杂度为$O(n^3)$，相对较高

参考文献

[1]. Heuristic Search for the Analysis of Graph Transition Systems

[2]. Heuristic Search-Based Planning for Graph Transformation Systems

[3]. NetSimile: A Scalable Approach to Size-Independent Network Similarity arXiv:1209.2684 [cs.SI]

[4]. A study of graph spectra for comparing graphs and trees Richard C. Wilson, Ping Zhu *Pattern Recognition 41 (2008) 2833--2841*

[5]. Pattern Vectors from Algebraic Graph Theory Richard C. Wilson, Edwin R. Hancock, and Bin Luo *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 27, NO. 7, JULY 2005*