Project: 图编辑距离GED

李康为 段庆宏 孙祯鹏

lkw123@pku.edu.cn 1900013087@pku.edu.cn 1900013007@pku.edu.cn

1. 简介

图编辑距离是图模式匹配技术中常用的方法之一。基于图编辑距离的匹配方法能够处理多种类型的图数据，因而受到了学术界的广泛关注。我们小组通过研读论文，对一些现有的图编辑距离算法进行复现，实现了A*-GED 和Munkres 等方法，并选择Alkane和Acyclic 数据集作为操作对象，得到对应的输出并对实验结果加以分析。

2. 相关工作

经过调研，我们对GED领域的典型工作进行横向对比与分析，进而确定本组重点实现的算法形式。

2.1. 精确图编辑距离

1. 基于启发式搜索技术的精确图编辑距离

   - 利用一般A*算法来解决GED问题时，其相当于宽度优先搜索算法，至多需遍历完整个搜索树空间才能得到最优解，因此，称此搜索方法为Plain-A*搜索，时间复杂度为指数级。
   - 相应地，可以对启发式函数进行优化，实现耗时的进一步缩减。譬如利用Hausdorff编辑距离函数来替换A*算法中的启发式函数[7]，以及采用BP代价矩阵框架的启发式函数进行优化等[6]。

2. 基于深度优先搜索的精确图编辑距离

   - 通过基于上下界剪枝策略的深度优先算法来对GED问题进行求解[1]，与一般A*算法相比，DF-GED大幅度减小了所需拓展的集合空间，减少了算法的计算时间。
   - 而在优化方面，可以使用分布式方法解决图编辑距离问题[2]，称为D-DF 算法。将大量图数据分割成子图，再将子图发送到各工作站利用DF算法求解。

2.2. 近似图编辑距离

对于精确图编辑距离而言，在处理大规模的图时复杂度较高，适用性不佳。因此，对于复杂度更低的近似图编辑距离的研究也至关重要。

1. 二分图编辑距离

   如今众多图编辑距离问题的求解都是基于BP代价矩阵框架进行的。在将源图g_1 和目标图g_2 边编辑代价整合进点编辑代价的情况下，不考虑边结构，将g_1 和g_2 中的点视为同一个新图中的点，图g_1 和g_2 之间点的所有可能指派视为此新构成的图的边，此时指派代价即为编辑操作代价，模型化出一个完全二分图，并构建一个二分代价阵框架，也就是所谓的BP代价阵框架。

2. 上下界方法

   图编辑距离上界是指算法得出的距离值比精确图编辑距离稍大；同理，下界指算法得出的距离值稍小于精确距离，上下界方法利用生成距离的上下界来逼近精确编辑距离[8]。

3. 其它近似算法

   除此之外，还有一些多项式时间复杂度的近似算法在图编辑距离问题上取得了很好的成效。文献[3]基于遗传算法的优化方法最终得出次优解，文

献[9]对A*算法的启发函数进行改进，首先对估值函数$h(\lambda)$ 限定步数以减少遍历时间，同时将启发式函数重新定义为:

$$f(\lambda) = \frac{g(\lambda) + h(\lambda)}{t^{|\lambda|}}$$

其中人为设定参数$t > 1$，λ则表示当前编辑操作的数量。如此的改进可以有效的加快计算速度并得到次优解。

3. 生成基准数据

在对实验结果进行准确性与效率的分析时，得到对应的基准数据（也就是GED的真实值）是必不可少的。我们根据一些已有工具分别对Alkane和Acyclic数据集进行处理，生成了正确的GED值的json文件，在后续的实验结果分析中作为参考标准来进行评估。

3.1. Alkane数据集

PyTorch Geometric（PyG）是一个基于PyTorch的用于处理不规则数据（比如图）的库，其运行速度出色，同时还集成了SimGNN论文[4]中提出的方法以及LINUX等数据集。我们将Alkane数据集导入，对train训练集和test测试集加以整理，将数据按照选题所要求的形式进行进一步的规范，具体体现在:

- 对Alkane数据集，依据图的属性将每个图对应转换为torch_geometric.data.Data的数据结构，构建出符合实际应用特点的Dataset。
- 根据特定的循环方式计算出Alkane数据集中两两之间的图编辑距离，并格式化输出到output.json中作为基准数据。

在具体的实现上，出于安装包与框架等环境的便捷性而言，我们选择借助Google Colab平台来完成。

首先安装Torch 框架并安装PyTorch Geometric 拓展库，要注意版本的匹配问题。

接下来对数据集进行处理，与要求的Alkane 数据集进行比对，将GXL 格式的图数据转换为PyG 所提供的数据集中的gexf 格式，并对图的属性进行检测从而

```
!pip install torch===1.4.0

!pip install torch-geometric \
  torch-sparse==latest+cu101 \
  torch-scatter==latest+cu101 \
  torch-cluster==latest+cu101 \
  -f https://pytorch-geometric.com/whl/torch-1.4.0.html
```

得到自定义的数据集，定义为myDataset。在此基础上得到myDataset的每张图之间的图编辑距离并格式化输出到json 文件中作为基准数据。

3.2. Acyclic数据集

与Alkane数据集相类似的，采取C++库GEDLIB [5]来对Acyclic 数据集中的数据加以处理生成基准数据。

4. EGED方法

4.1. 算法思路

通过构造代价函数从而计算出优先队列中点的优先级，然后通过传统的A*算法计算。通过更新传统代价函数得到效果更好的代价函数解决。

$$\begin{aligned} GED(G_1, G_2) \geq\ & \Gamma(L_{V_1/V'}, L_{V_2/f(V')}) \\ & + \Gamma(L_{E_1/E'}, L_{E_2/f(E')}) + \Delta, \Gamma(X, Y) \\ & = max(|X|, |Y|) - |X \cap Y| \end{aligned}$$

上述估计方法比较粗略，对于边而言并没有完全利用映射条件。只考虑到了两个点都在映射中的情形，对于一个点在映射中，另一个不在映射中的情形没有分开讨论。

因此可以进行如下优化：对于图$G = (V, E, l)$，$N_I(V) = \{(u, v) \in E : u, v \in V\}$ 为V 的内部节点，$N_O(V) = \{(u, v) \in E : u \in V \wedge v \notin V\}$ 为V 的外部节点。则

$$c_d = \sum_{u \in V_G} |d_O(u) - d_O(f(u))| + \frac{1}{2}|d_I(u) - d_I(f(u))|$$

4.2. 实现

与传统A*算法实现类似，采用c语言的方式用优先

队列实现。

4.2.1 优化方法

- 对于映射多的节点优先展开，这样可以尽快得到最优解。
- 对于初始情形，找到比较特殊的节点作为展开节点。例如节点标签数量较小的点和度数较大的点。同时寻找的映射也最好具有相同特征。由于数据集比较特殊，从1开始寻找映射效果也较好。
- 对于映射后下一个节点的选取，应选择连通分支内的节点，由于数据集是连通图，无需考虑，只需按节点顺序即可。
- 基于Alkane，Acyclic数据集的特殊性考虑，可以先计算出$\{(1,1),(2,2),\cdots\}$的代价，如果大于等于这个初始代价直接返回即可。
- 使得映射比较规律，可以将映射(i,i)对的数量也作为节点的比较指标。

4.2.2 CSI-GED

可以考虑基于CSI-GED精确图编辑距离。CSI-GED与一般A*算法相比，通过边的搜索代替点的搜索使得搜索空间大范围缩小，因此性能更好。主要体现在:

- CSI-GED是基于寻找边映射的搜索方法相比于点映射，CSI-GED在稀疏图中搜索空间较小。
- CSI-GED对于公共子结构较为敏感，会优先筛选出有相同结构的映射。
- 边映射相比于点映射有额外的限制条件，即新映射的边符合维护的点映射集。
- 对于给定的边映射，可以通过边映射得出对应的部分点映射。由于点数较小的图到点数大的图不存在删除操作，因此为映射的点为游离点，该点的度数为0。对于顶点标号一致的图，可以随意制定该点映射。
- 这里的边映射存在删边操作，即对应空集，但是未映射的点没有删除。
- 与点映射的代价函数基本类似。

4.3. 实验结果

相比于传统的A*算法有改善，但效果不明显。

5. AGED方法

根据论文[10]的思路，可以将图编辑距离问题转化为匹配问题。

5.1. 算法思路

先考虑二分图最大权匹配：对于二分图(X,Y)，有边权矩阵C_{ij}。假设存在点标X_i,Y_i，要求$X_i+Y_i >= C_{ij}$ 。若二分图中所有满足$X_i+Y_i \geq C_{ij}$ 的边$< i,j >$构成的子图（相等子图）有完备匹配，则这个完备匹配就是二分图的最大权匹配。

Proof：对于二分图任意完备匹配，如果它包含于相等子图，那么它的边权和等于所有顶点的顶标和；如果它有的边不包含于相等子图，那么它的边权和小于所有顶点的顶标和。所以相等子图的完备匹配一定是二分图的最大权匹配。

以上即为KM算法。然后我们需要构造图编辑到二分图最大权匹配的转换，这里选择以顶点为核心构造代价矩阵，通过KM算法求出最佳顶点匹配后再计算边的修改就是确定性的。

5.2. 实现

5.2.1 KM算法

设$X_i = \max\limits_{1\geq j\geq |Y|}(C_ij)$，$Y_i = 0$。

按照匈牙利算法的思路找到交错路径（$X_i+Y_i = C_{ij}$），必然会在某一Y上点不满足$Xi- = d, Yi+ = d$，形成新的可匹配边，d是从路径中的X点出发，最小的$X_i+Y_i = C_{ij}$。

5.2.2 构造GED到二分图

我们首先只考虑点匹配，对于$|V1|! = |V2|$ 的情况，会有点被删除或插入，故需要新增空节点ϵ，对于

正常的C_{ij}，代价为替换X_i到ϵ删除，ϵ到X_i插入，考虑边修改的影响：

- X_i到Y_i，相邻边E(Xi)到E(Yi)的修改代价
- X_i到ϵ，相邻边E(Xi)的删除代价
- ϵ到Y_i，相邻边E(Yi)的插入代价

原始形式使用的Munkres 算法对于解决分配问题是最佳的。每个节点编辑操作都是单独考虑的（仅考虑局部结构），因此无法动态推断边缘上的隐含操作。通过对顶点的相邻边集进行比较，可以得出考虑边信息的代价矩阵。

5.3. 实验结果

通过实验结果发现，采用的代价函数在点权相同，边权相同的Alkane和Acyclic数据集上表现不佳同时点的二次相邻信息没有考虑到，可以以此对代价函数进行优化。

6. 任务分工

- 李康为：根据论文对DF-GED 进行实现，并通过C++库GEDLIB 以及图神经网络库PyTorch Geometric 生成基准数据对实验结果进行检验分析，撰写报告。
- 段庆宏：选取了新的代价函数，改进了传统的A*算法。并结合有关CSI-GED的方法对传统A*算法中所暴露的问题加以改进。
- 孙祯鹏：使用KM算法对顶点进行重新匹配，BP矩阵为点权差和相邻边权重差。

References

[1] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau. An exact graph edit distance algorithm for solving pattern recognition problems. In *4th International Conference on Pattern Recognition Applications and Methods 2015*, 2015.

[2] Z. Abu-Aisheh, R. Raveaux, J.-Y. Ramel, and P. Martineau. A distributed algorithm for graph edit distance. *DBKDA 2016*, page 76, 2016.

[3] S. Auwatanamongkol. Inexact graph matching using a genetic algorithm for image recognition. *Pattern Recognition Letters*, 28(12):1428–1437, 2007.

[4] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, and W. Wang. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 384–392, 2019.

[5] D. B. Blumenthal, S. Bougleux, J. Gamper, and L. Brun. Gedlib: a c++ library for graph edit distance computation. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 14–24. Springer, 2019.

[6] S. Fankhauser, K. Riesen, and H. Bunke. Speeding up graph edit distance computation through fast bipartite matching. In *International Workshop on Graph-Based Representations in Pattern Recognition*, pages 102–111. Springer, 2011.

[7] A. Fischer, R. Plamondon, Y. Savaria, K. Riesen, and H. Bunke. A hausdorff heuristic for efficient computation of graph edit distance. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 83–92. Springer, 2014.

[8] D. Justice and A. Hero. A binary linear programming formulation of the graph edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1200–1214, 2006.

[9] M. Neuhaus, K. Riesen, and H. Bunke. Fast suboptimal algorithms for the computation of graph edit distance. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 163–172. Springer, 2006.

[10] F. Serratosa. Fast computation of bipartite graph matching. *Pattern Recognition Letters*, 45:244–250, 2014.