

1 实验研究

我们对我们的算法进行了广泛的实验研究来评估其有效性, 高效性及扩展性。我们在化学分子结构上测试我们的算法。对于化学结构, 节点特征包括数值特征和原子布尔特征。数值特征包括元素种类, 原子部分电荷, 原子电子亲和势, 原子自由电子数目和原子价态等等。布尔特征包括原子是否在供体中, 是否在末端碳中, 是否在环中, 是否为负, 是否是轴向的等等。在实验中, 我们仅用一个原子特征: 元素种类。

我们将我们的方法和小波分配核, C-tree, GraphGrep 还有 gIndex 进行对比。我们的算法, WA 算法, GraphGrep 和 gIndex 是基于 C++ 实现的, 用 g++ 进行编译。C-tree 是用 Java 实现的, 用 Sun JDK 1.5.0 编译。所有的实验都是在 Intel Xeon EM64T 3.2GHz, 4G 内存, Linux 系统这一平台上测试的。

WA, G-Hash, C-tree, GraphGrep 和 gIndex 的参数是这样设置的。对于 WA 和 G-hash, h 取 2, 用 *haar* 小波函数, 对于 C-tree, 用默认值即将最小子节点数 m 设为 20, 最大 M 设为 $2m - 1$, 用 NBM 方法进行图映射。对于 GraphGrep 和 gIndex, 全部采用默认参数。

1.1 数据集

我们选用许多数据集来进行试验。前五个数据集是从 Jorissen/Gilson 数据集获得的已有数据。接下来六个是从 BindingDB 数据集中抽取的, 最后一个 NCI/NIH 艾滋病筛选集里的, 表 1 显示了这些数据集和其基本情况。

1.1.1 Jorissen 数据集

Jorissen 数据集主要为一些包含活动蛋白质的化学结构信息。我们以药物对特定蛋白质的吸引力作为目标值。我们选取了 5 个包含 100 个分子结构的蛋白质作为测试目标, 其中 50 个化学结构连接着蛋白质, 另外 50 个没连接到蛋白质上。参考文献 14 来查看详细信息。

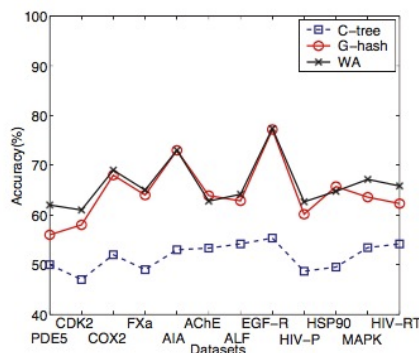
1.1.2 NCI/NIH AIDS 抗体数据库

NCI/NIH 艾滋病抗体数据库包含 42390 个化学结构, 总共有 63 种分子, 最常见的是, C, O, N, S。数据集包括三种边, 单边, 双边, 芳香边。我们选取了所有结构作为数据库, 并随机选取 1000 个化学结构作为查询集合。

1.2 利用分类做节点相似性度量

我们用不同的相似性度量方法比较 k -NN 分类器在 Jorissen 数据集和 BindingDB 数据集上的分类精确度。对于 WA 算法，我们采用小波匹配核函数来获得核矩阵，然后计算距离矩阵来获得最近的 k 个最相似的图。对于 G-Hash，我们根据之前描述的算法计算核函数，然后找出最相似的 k 个图。对于 C-tree，我们直接找回最相似的子图。我们用标准的 5-fold 交叉验证来获得分类精确度， $(TP + TN)/S$ ，其中 TP 表示真正结果的数目， TN 表示真负的数目， S 是全部的数据图数目。在本实验中，我们取 $k = 5$ 。

精确性对比，实验结果如图 1 所示。可以看出，G-Hash 在所有数据集中都比 C-tree 表现优秀，至少有 8% 的提高。G-Hash 和 C-tree 的平均差异大概在 13%。而 WA 算法较 G-Hash 表现更为优异，大概有 2% 的提升。主要原因是我们简化了距离度量公式。从实验来看，可以明显看出基于核函数的相似性度量方法比基于编辑距离的相似性方法更有优势。因为 k -NN 分类器的精确度和 k 有关，所以我们也研究了不同 k 下算法表现。实验表明精确度对比实验结果不受参数 k 影响。



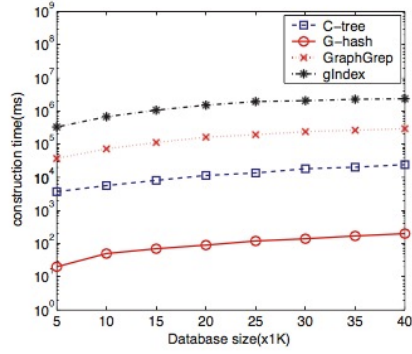
图表 1: 精确性对比图

1.3 伸缩性

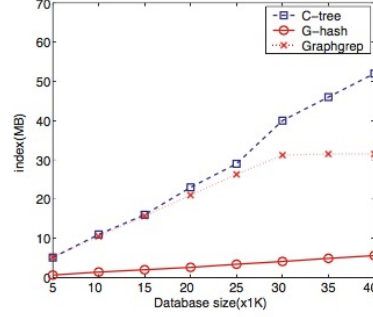
1.3.1 索引构建时间

我们探究了 G-Hash, WA 和 C-tree, GraphGrep 还有 gIndex 在 NCI/NIH 艾滋病抗体数据库上的表现。我们比较了不同算法下索引构建时间和索引大小。实验结果如图 2,3 所示。

可以看出，无论构建时间，还是索引大小，G-Hash 算法均优于其他算法。



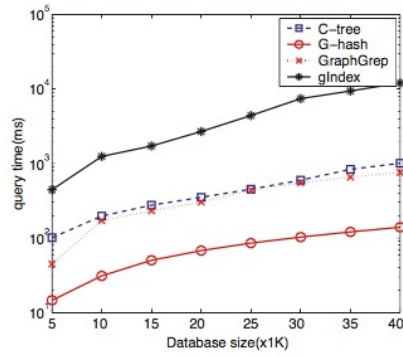
图表 2: 索引构造时间对比图



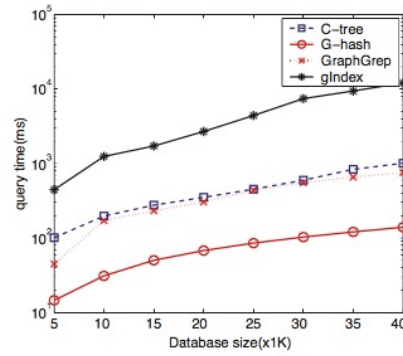
图表 3: 索引大小对比图

1.4 查询时间

我们探究了不同数据库大小对查询时间的影响，实验结果如图 4所示。



图表 4: 查询时间和数据库大小关系对比图



图表 5: 查询时间和 k 值关系对比图

显然,G-Hash 比其他算法表现均好,当数据库达到 4000 时,C-tree,GraphGrep 和 gIndex 的查询时间是 G-Hash 的 8 倍, 10 倍和 100 倍。

最后我们比较了不同 k 值下查询时间，如图 5所示。

实验表明 G-Hash 的查询时间与 k 值的大小没有影响。