Graph Matching using Interference of Coined Quantum Walks

David Emms, Edwin Hancock and Richard Wilson
University of York
E-mail: demms@cs.york.ac.uk

Abstract

In this paper we consider how coined quantum walks can be applied to exact graph matching. The matching problem is abstracted using an auxiliary structure that connects pairs of vertices from the graphs to be matched by way of auxiliary vertices. We locate matches using coined quantum walks on this structure. We have tested the algorithm on graphs derived from the NCI molecule database and found it to significantly reduce the space of possible matchings thereby allowing the graphs to be matched directly. We also perform a sensitivity analysis on the algorithm in order to examine its behaviour in the presence of noise.

1. Introduction

Quantum algorithms have recently attracted considerable attention in the theoretical computer science community. This is primarily because they offer considerable speed-up over classical algorithms. For instance, Grover’s [6] search method is polynomially faster than its classical counterpart, and Shor’s factorisation method is exponentially faster than classical methods. However, quantum algorithms also have a richer structure than their classical counterparts since they use qubits rather than bits as the basic representational unit [9]. Consequentially, an n qubit quantum computer would manipulate a state in $\mathbb{C}^{2^n}$ as opposed to $\mathbb{Z}_2^n$, which is the case classically. For instance, this structure is exploited in Shor’s algorithm where the Fourier transform is use to locate prime factors.

It is this issue of richer representations that is the subject of this paper. We are interested in how the idea of quantum walks can be applied to the problem of graph matching. From a practical perspective, there have been a number of useful applications of random walks. One of the most important of these is the analysis of routing problems in network and circuit theory. Of more recent interest is the use of ideas from random walks to define the page-rank index for internet search engines such as Googlebot [3].

In the pattern recognition community there have been several attempts to use random walks for graph matching. These include the work of Robles-Kelly and Hancock [11, 10] which has used both a standard spectral method [11] and a more sophisticated one based on ideas from graph seriation [10] to convert graphs to strings, so that string matching methods may be used. Gori, Maggini and Sarti [5] on the other hand, have used ideas borrowed from page-rank to associate a spectral index with graph nodes and have then used standard subgraph isomorphism methods for matching the resulting attributed graphs.

Quantum walks have been introduced as quantum counterparts of random walks [7] and possess a number of interesting properties not exhibited by classical random walks. The paths of the coined quantum walk have been used to define a matrix representation of graphs that is able to lift the cospectrality of certain classes of graphs that are typically hard to distinguish [4].

In this paper, we present a novel auxiliary graph structure, based on a pair of graphs to be matched, and simulate a coined quantum walk on this structure. The auxiliary structure contains vertices connecting each pair of vertices from the two graphs. It is on these vertices that the two walks interfere, and by identifying where this interference is exact we are able to identify matches between the graphs. To test the algorithms effectiveness at finding isomorphisms, we carry out experiments using graphs from the US National Cancer Institute database of molecules [8]. In addition, we carry out a sensitivity analysis in order to investigate its behaviour in the presence of structural errors.

2. The Quantum Walk

Let $G = (V_G, E_G)$ be a graph with vertex set $V_G$ and edge set, $E_G = \{(u, v) | u, v \in V_G, u \text{ adjacent to } v\}$. The degree of a vertex $u \in V_G$, denoted $d(u)$, is the number of vertices adjacent to $u$. Quantum processes are reversible, and in order to make the walk reversible a particular state of the walk must give both the current location of the walk and its previous location [1]. To this effect each edge $(u, v) \in E$ is replaced by a pair of directed arcs $(u, v)$ and $(v, u)$ and the set of these arcs is denoted $A_G$. The basis states for the
quantum walk are vectors in a Hilbert space, $\mathcal{H} \cong \mathbb{C}^{|V_G|}$, and are denoted $|uv\rangle$ where $(u, v) \in A_G$. Such a state is interpreted as the walk being at vertex $v$ having been at $u$. A general state for the walk is thus written

$$|\psi\rangle = \sum_{(u,v) \in A_G} \alpha_{uv}|uv\rangle$$

where $\alpha_{uv} \in \mathbb{C}$.

The probability that a walk is in a particular state is given by the rule, $P(|uv\rangle) = \alpha_{uv}^*\alpha_{uv}$ where $x^*$ is the complex conjugate of $x$. For the purpose of this work, the amplitudes will be real, albeit negative as well as positive. The fact that states can have negative amplitudes is of key importance as it allows various paths to cancel out (destructive interference) and this is utilized by our algorithm.

The evolution is linear and conserves probabilities. Consider a state $|\psi\rangle = |xy\rangle$ where the degree of $y$, $d(y) = r + 1$, so that as well as $x$, $y$ is adjacent to the vertices $z_1, z_2, \ldots, z_r$. The walk respects the connectivity structure of the graph and transitions are only allowed between adjacent vertices, hence, one step of the walk is such that

$$|xy\rangle \rightarrow a|yx\rangle + b \sum_{i=1}^{r} |yz_i\rangle \quad a, b \in \mathbb{C}.$$ 

Two separate amplitudes, $a$ and $b$, can be used since the transitions from $|xy\rangle$ to $|yx\rangle$ and $|yz_i\rangle$ can be distinguished without reference to any (arbitrary) labellings of the vertices or edges. Since probability must be conserved, $a^2 + rb^2 = 1$. It is usual to use the ‘Grover diffusion operators’ [6] which are such that $a = 2/d(y) - 1$ and $b = 2/d(y)$ (for transitions about vertex $y$) since these provide the transition that is furthest from the identity.

For a general graph (with edges replaced by arcs) the matrix, $U(G)$, governing the evolution of the walk can be written as

$$U(G)_{(i,j), (k,l)} := \begin{cases} \frac{2}{d(j)} - \delta_{i,j}, & \text{if } j = k; \\ 0, & \text{otherwise}. \end{cases}$$

for all $(i, j), (k, l) \in A_G$.

### 3. The Algorithm

Given a pair of graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ we would like to find a mapping, $\phi : V_G \to V_H$ such that $\{\phi(u), \phi(v)\} \in E_H$ if and only if $(u, v) \in E_G$. Our algorithm works by taking the two graphs that are to be matched, connecting all pairs of vertices (one from each graph) by way of one intermediate vertex for each pair and simulating a quantum walk on this graph. The intermediate vertices provide a site on which quantum interference can take place between the two walks, the final step is to simulate this interference to give a set of quantum amplitudes indicating possible matches between pairs of vertices.

More precisely, we form a new graph $\Gamma = (V_\Gamma, E_\Gamma)$ by taking the union of $G$ and $H$ and then joining every vertex in $V_G$ to every vertex in $V_H$ by way of $nm$ intermediate ‘auxiliary’ vertices, one for each connection (fig. 1). The auxiliary vertex linking $g_i$ to $h_j$ will be denoted $v_{(g_i, h_j)}$. The auxiliary graph is in some ways similar to the association graph [2], however, information about the structure of the two graphs comes from including the original graphs themselves rather than through connecting the auxiliary vertices as is done in the association graph.

$$V_\Gamma = V_G \cup V_H \cup V_A$$

where

$$V_A = \{v_{(g_i, h_j)} | g_i \in V_G, h_j \in V_H\}$$

and

$$E_\Gamma = E_G \cup E_H \cup E_A$$

where

$$E_A = \{\{g_i, v_{(g_i, h_j)}\}, \{h_j, v_{(g_i, h_j)}\} | g_i \in V_G, h_j \in V_H\}$$

![Figure 1. The vertices $g_1, g_2, g_3 \in V_G$ and $h_1, h_2, h_3 \in V_H$ connected by way of auxiliary vertices.](image)

The walk is evolved in discrete steps according to the rule $|\psi^{t+1}\rangle = U(\Gamma)|\psi^t\rangle$ from a starting state $|\psi^0\rangle$ with amplitudes, $\alpha_{xy}^0 = \begin{cases} 1 & \text{if } (x, y) \subset V_G \text{ or } (x, y) \subset V_H; \\ 0 & \text{otherwise}. \end{cases}$

After the walk has been evolved for a given number of steps a final interference step, $R$, is simulated giving the state $|\psi'\rangle = R|\psi^t\rangle$. This is such that, for all pairs of vertices, $g \in V_G$ and $h \in V_H$, the difference between the amplitudes of the corresponding states $|gv_{(g, h)}\rangle$ and $|hv_{(g, h)}\rangle$ is effectively calculated as the amplitude of the state $|hv_{(g, h)}\rangle$. For these pairs of states, in the basis $\{|gv_{(g, h)}\}, |hv_{(g, h)}\rangle\}$, the step is given by

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$
Consider two graphs, $G$ and $H$, such that there is an isomorphism $\phi : V_G \rightarrow V_H$ between them. If $g \in V_G$ and $h \in V_H$ are two vertices such that $\phi(g) = h$ then, as a result of the symmetry of the auxiliary graph and the starting state used, $\alpha^f_{\phi g \phi h} = \alpha^f_{\phi h \phi g}$ for all times, $t$. Consequently, in the final state, the amplitude $\alpha^f_{\phi g \phi h} = 0$ whenever $\phi(g) = h$. Thus we use the amplitudes of these states to identify possible isomorphisms between the two graphs.

4. Structural Errors

![Figure 2. The distribution of amplitudes for 'non-corresponding' vertices (left) and 'corresponding' vertices (right) for pairs of graphs on 15 vertices, differing by 2 edges, for 50 pairs of graphs. A Gaussian distribution has been fitted to the non-matching vertices and Gaussian (solid line) and double-exponential (dashed line) distributions for the matching vertices.](image)

As described above, if the two graphs are isomorphic then the amplitudes for corresponding pairs of vertices will all be zero. In many situations, however, there will exist structural errors and so no complete isomorphism will exist. In this case it is not guaranteed that the amplitudes for the 'best' pairings of vertices will be non-zero.

In order to investigate the robustness of the method in the presence of such errors we generated a random graph together with a partner that differed from it by a set number of edges. We found that the amplitudes of 'non-corresponding' vertices could be modelled as a Gaussian with standard deviation, $\sigma_f$ and zero mean. The distribution of 'corresponding' vertices on the other hand is much better modelled by the more strongly peaked double-exponential distribution with standard deviation $\sigma_t$ and zero mean

$$p(\alpha|t) = \frac{1}{\sigma_t \sqrt{2}} e^{-\frac{\sqrt{\pi \alpha}}{\sigma_t}},$$

where $\sigma_t < \sigma_f$ (fig. 2). Since the distribution for corresponding vertices is far more strongly peaked, we hope to still be able to reconstruct the match even in the presence of noise.

![Figure 3. The proportion of graphs that are isomorphic, non-isomorphic or undecided as a function of the number of vertices.](image)

For a given pair of vertices with amplitude for a match, $\alpha$, Bayes’ rule can be used to give the probability that this match is correct. The matching process could then be completed making use of structural constraints if necessary. We begin by modelling the probability distribution for $\alpha$ as the sum of the two distributions. We have

$$p(\alpha) = \frac{p(t)}{\sigma_t \sqrt{2}} e^{-\frac{\sqrt{\pi \alpha}}{\sigma_t}} + \frac{p(f)}{\sqrt{2\pi \sigma_f}} e^{-\frac{\alpha}{2 \sigma_f^2}}.$$

where

$$p(t) = 1 - p(f) = \frac{1}{|V_G|},$$

By applying Bayes’ rule, we have a the probability of a true match given $\alpha$

$$p(t|\alpha) = \frac{p(\alpha)p(t)}{p(\alpha)} = \frac{1}{\sigma_t |V_G| \sqrt{2\pi \sigma_f}} e^{-\frac{\sqrt{\pi |\alpha|}}{\sigma_f}} + \frac{1}{\sigma_t |V_G| \sqrt{2\pi \sigma_f}} e^{-\frac{\sqrt{\pi |\alpha|}}{\sigma_f}} + \frac{1}{\sqrt{2\pi \sigma_f}} e^{-\frac{\alpha^2}{2 \sigma_f^2}}.$$

Thus, given a set of amplitudes for the pairings of the vertices of $G$ with the vertices of $H$, we are able to calculate the probability that each pairing forms part of a correct match.

5. Experiments

We present experiments carried out on graphs representing the structure of a subset of the molecules from the NCI database of molecules. In the database, a particular molecule is represented as a graph with vertex attributes giving the type of atom and edges representing bonds. We disregard the type of atoms and only make use of the bond structure thereby giving us a set of non-attributed graphs, a number of which are isomorphic, the goal being to identify these. For a particular pair of graphs we use our algorithm to...
prune the space of possible matches and check the matches returned. If more than 1000 possible matches remain then we class the question of whether two graphs are isomorphic as undecided.

For the graphs tested fig. 3 shows the fraction of isomorphic graphs, non-isomorphic graphs and those that are undecided as a function of the number of vertices. We see that the algorithm is able to to reconstruct the match or identify that the graphs are non-isomorphic for these graphs with a low percentage undecided, this percentage can be lowered further by allowing more than 1000 matches to be checked.

In the second experiment, for each graph we carried out a random permutation of its vertices and then attempted to recover this permutation using our algorithm. In order to analyse the performance of the algorithm we consider the average number of matches that needed to be checked–once the search space had been pruned–in order to reconstruct the permutation, and recorded the fraction of graphs for which we did not recover the permutation (fig. 4 and table 1). As can be seen, the quantum walk is able to significantly prune the size of the search space, and hence make the problem of finding matches for graphs on large numbers of vertices significantly easier.

6. Conclusion

In this paper we have described an auxiliary graph that can be used for the purpose of graph matching. By simulating a discrete quantum walk on this structure quantum interference can be used to compare the two graphs. The walk gives rise to a set of amplitudes corresponding to possible pairings of the vertices of the two graphs. If the graphs are isomorphic then the states for which the interference is exact are used to significantly prune the space of possible mappings between the graphs allowing us to recover the isomorphism. We have tested the algorithm on graphs representing molecular structures and found that it to reduce the space of matches sufficiently in order for us to match the graphs directly. We have analysed how the algorithm behaves in the presence of structural errors and as further work would like to test its performance in such situations.

References


\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
$|V|$ & $> 1000$ matches & $|V|$ & $> 1000$ matches \\
\hline
$\leq 18$ & 0 & 26 & 0.42 \\
19 & 0.25 & 27 & 0.4 \\
20 & 0.08 & 28 & 0.67 \\
21 & 0.17 & 29 & 0.42 \\
22 & 0.17 & 30 & 0.71 \\
23 & 0.17 & 31 & 0.58 \\
24 & 0.42 & 32 & 0.60 \\
25 & 0.25 & \\
\hline
\end{tabular}
\caption{The fraction of the graphs tested for which more than 1000 matches still remain after the interference step of the walk.}
\end{table}