Approximate graph matching

Jeffrey Finkelstein
Computer Science Department, Boston University
September 24, 2015

1 Notation and definitions

For any positive integer $n$, let $[n]$ be the set $\{0, \ldots, n-1\}$.

1.1 Matrices

In this paper, vectors are column vectors. Unless otherwise specified, the norm of a matrix $\|M\|$ is the Frobenius norm, defined by $\|M\| = \sqrt{\sum_{i,j} M_{ij}^2}$. The all ones vector is denoted $\mathbf{1}$ and the all zeros vector is denoted $\mathbf{0}$. The all ones matrix is denoted $\mathbf{J}$ and the all zeros matrix is denoted $\mathbf{O}$. The $n \times n$ identity matrix is denoted $\mathbf{I}_n$, or simply $\mathbf{I}$ if its size is understood from context. The vectorization of an $m \times n$ matrix $A$, denoted $\text{vec}(A)$, is the $mn$-dimensional vector created by stacking the rows of $A$ as columns on top of one another. The trace of a square matrix $A$, denoted $\text{tr} A$, is the sum of its diagonal entries. The direct sum of square matrices $A$ and $B$, denoted $A \oplus B$, is the block matrix

$$
\begin{bmatrix}
A & O \\
O & B
\end{bmatrix}.
$$

For two matrices $M$ and $N$, entrywise inequality is denoted $M \geq N$.

A nonnegative real matrix $M$ is doubly stochastic if it satisfies the equations $M\mathbf{1} = \mathbf{1}$ and $M^T \mathbf{1} = \mathbf{1}$. If furthermore each entry in the matrix is either zero or one, the matrix is called a permutation matrix.

1.2 Graph matching

Suppose $A$ and $B$ are the adjacency matrices of two undirected graphs on $n$ vertices. The two graphs are isomorphic if there is a permutation matrix $P$ such
that \( PA = BP \). The \textit{graph isomorphism problem} is the problem of determining whether such a permutation exists, given \( A \) and \( B \). The natural optimization problem corresponding to that decision problem is called the \textit{graph matching problem} and can be expressed as

\[
\begin{align*}
\text{minimize} & \quad \|PA - BP\|^2 \\
\text{subject to} & \quad P1 = 1 \\
& \quad P^T1 = 1 \\
& \quad P \in \{0, 1\}^{n \times n}.
\end{align*}
\]

Since

\[
\|PA - BP\|^2 = \|PA\|^2 + \|BP\|^2 - 2\langle PA, BP \rangle
\]

\[
= \|PA\|^2 + \|BP\|^2 - 2 \text{tr}(AP^TBP)
\]

\[
= \|A\|^2 + \|B\|^2 - 2 \text{tr}(AP^TBP),
\] (1)

and since \( \|A\| \) and \( \|B\| \) are constants with respect to the variable \( P \), an equivalent formulation of the graph matching problem is

\[
\begin{align*}
\text{minimize} & \quad -\text{tr}(AP^TBP) \\
\text{subject to} & \quad P1 = 1 \\
& \quad P^T1 = 1 \\
& \quad P \in \{0, 1\}^{n \times n}.
\end{align*}
\]

A solution to the graph matching problem (in either of these two formulations) is a permutation matrix \( P \) that witnesses an isomorphism between the graphs whose adjacency matrices are \( A \) and \( B \).

Relaxing the constraint “\( P \in \{0, 1\}^{n \times n} \)” to the constraint “\( P \geq O \),” in other words, allowing \( P \) to be a doubly stochastic instead of a permutation matrix, yields the \textit{relaxed graph matching problem}. However, when \( P \) is a doubly stochastic matrix, Equation 1 does not necessarily hold. Hence the two formulations of the relaxed graph matching problem corresponding to the two formulations of the graph matching problem are not necessarily equivalent. For this reason, we give each formulation a different name.

The \textit{convex relaxed graph matching problem} is

\[
\begin{align*}
\text{minimize} & \quad \|DA - BD\|^2 \\
\text{subject to} & \quad D1 = 1 \\
& \quad DT1 = 1 \\
& \quad D \geq O,
\end{align*}
\]
and the indefinite relaxed graph matching problem is

\[
\begin{align*}
\text{minimize} & \quad -\text{tr}(AD^TBD) \\
\text{subject to} & \quad D1 = 1 \\
& \quad D^T1 = 1 \\
& \quad D \geq O.
\end{align*}
\]

In either problem, the domain over which the objective function is minimized is the set of doubly stochastic matrices, which is the convex hull of the set of permutation matrices. However, only in the former problem is the objective function convex (hence its name). TODO Explain why the objective function is convex. The latter problem is called “indefinite” because the Hessian matrix of the objective function has trace zero at each point in the domain, and is therefore an indefinite matrix, which indicates that the objective function is not convex. Computing the Hessian reveals that entry \((i'i', j'j')\) in the matrix is \(A_{ii'}B_{jj'} + A_{i'i}B_{j'j}\) independent of \(D\). Since \(A\) and \(B\) are symmetric matrices, this is simply \(2A_{ii'}B_{jj'}\). The trace of the matrix is therefore \(2\text{tr}(A \otimes B)\), a constant function with respect to \(D\) whose value is always zero if the graphs being considered do not have self-loops.)

Now Equation 1 holds even if \(P\) is just a unitary matrix (that is, a matrix satisfying \(P^TP = I\)), so another relaxation could be

\[
\begin{align*}
\text{minimize} & \quad \|UA - BU\|^2 \\
\text{subject to} & \quad U^TU = I \\
& \quad U \geq O,
\end{align*}
\]

or equivalently,

\[
\begin{align*}
\text{minimize} & \quad -\text{tr}(AU^TBU) \\
\text{subject to} & \quad U^TU = I \\
& \quad U \geq O.
\end{align*}
\]

The Hessian of a real-valued multivariate function \(f\) is the matrix of its second-order partial derivatives; specifically, entry \((i, j)\) in the Hessian matrix is \(\frac{\partial^2 f}{\partial x_i \partial x_j}\). Evaluating the Hessian matrix entrywise at a point in the domain produces a matrix with real entries. If the Hessian matrix evaluated at a critical point is nonsingular, then the signs of the eigenvalues of this matrix indicate whether the critical point is a local minimum, a local maximum, or a saddle point. (This is just the multivariate generalization of using the second derivative of a univariate function to determine whether a critical point is a local minimum or a local maximum.) If all the eigenvalues are positive (or equivalently, the matrix is positive definite), then the point is a minimum. If all the eigenvalues are negative (or equivalently, the matrix is negative definite), then the point is a maximum. If some are negative and some are positive (in other words, the matrix is indefinite), then the point is a saddle point. If a function has a saddle point, then it is not convex.

If the second-order partial derivatives are locally continuous at the critical point, then the Hessian matrix evaluated at that point is symmetric. The trace of a real symmetric matrix equals the sum of its eigenvalues. If the sum is zero, some eigenvalues must be positive and some must be negative. Therefore, a trace of zero implies a saddle point which implies the function is not convex.
Whereas the relaxed graph matching problems defined above for doubly stochas-
tic matrices have a quadratic objective function and linear constraints, these
relaxations have a quadratic objective function with quadratic constraints and
are therefore more difficult to solve.

2 Complexity of graph matching problems

In this section, we have renamed the two formulations of the graph match-
ing problem to Minimum Disagreement and Maximum Agreement, and
expressed the problems as formal optimization problems.

TODO Note that formally, optimization problems must have positive measure functions, but this doesn’t matter too much.

TODO Define budget problem for optimization problems.

Definition 2.1 (Minimum Disagreement).
Instance: adjacency matrices $A$ and $B$ of two undirected graphs on $n$
vertices.
Solution: $n \times n$ permutation matrix $P$.
Measure: $\|PA - BP\|^2$.
Type: minimization.

Definition 2.2 (Maximum Agreement).
Instance: adjacency matrices $A$ and $B$ of two undirected graphs on $n$
vertices.
Solution: $n \times n$ permutation matrix $P$.
Measure: $-\text{tr}(AP^TBP)$.
Type: minimization.

2.1 Graph matching

As stated in the previous section, Minimum Disagreement and Maximum
Agreement are equivalent, so it suffices to study only Maximum Agreement.

Theorem 2.3. The budget problems for Minimum Disagreement and Maximum
Agreement are equivalent under polynomial time many-one reductions.

Proof. As shown in Equation 1

$$\|PA - BP\|^2 = \|A\|^2 + \|B\|^2 - 2 \text{tr}(AP^TBP),$$

which is equivalent to

$$-\text{tr}(AP^TBP) = \frac{1}{2}(\|PA - BP\|^2 - \|A\|^2 - \|B\|^2).$$

By these two equalities,

$$\|PA - BP\|^2 \leq k \iff -\text{tr}(AP^TBP) \leq \frac{1}{2}(k - \|A\|^2 - \|B\|^2)$$
and
\[- \text{tr}(AP^T BP) \leq k \iff \|PA - BP\|^2 \leq \|A\|^2 + \|B\|^2 - 2k.\]
It is straightforward to apply these equivalences to construct the stated reductions.

These budget problems are NP-complete. This proof is based on the proof of the NP-completeness of the budget problem for Minimum Quadratic Assignment implicit in [20].

**Theorem 2.4.** The budget problem for Minimum Disagreement is NP-complete, as is the budget problem for Maximum Agreement.

**Proof.** By the previous theorem, it suffices to show that Maximum Agreement is NP-complete.

The problem is in NP: the witness is the permutation matrix $P$, and the measure can be computed (and compared to the budget) in polynomial time by matrix multiplication. The NP-hardness follows from a polynomial time many-one reduction from the Hamiltonian Cycle problem. Given a graph $G$ on $n$ vertices, the reduction is the mapping $G \mapsto (A, B, 2n)$, where $A$ is the adjacency matrix of $G$ and $B$ is the adjacency matrix of the cycle graph on $n$ vertices. Each of $A$, $B$, and $2n$ can be computed and written in polynomial time.

If $G$ has a Hamiltonian cycle, there is some permutation $\pi$ such that the undirected edge $(\pi(i), \pi(i + 1))$ is in $E$ for each $i \in [n]$, where addition is modulo $n$. Choosing $P$ to be the permutation matrix corresponding to $\pi$,

\[
\text{tr}(AP^T BP) = \langle PA, BP \rangle \\
= \langle PAP^T, B \rangle \\
= \sum_{i,j} (PAP^T)_{i,j} B_{i,j} \\
= \sum_{i,j} A_{\pi(i), \pi(j)} B_{i,j} \\
= 2 \sum_i A_{\pi(i), \pi(i+1)},
\]

where the last equality is due to the definition of $B$ and the factor of 2 is due to the fact that both $A$ and $B$ are symmetric matrices. Since $A$ is just the adjacency matrix of $G$ and the undirected edge $(\pi(i), \pi(i + 1))$ is in $E$ for each $i$ in $[n]$, the above sum is exactly $2n$. In particular, $- \text{tr}(AP^T BP) \leq -2n$, so $(A, B, -2n)$ is in the budget problem for Maximum Agreement.

Now suppose that $- \text{tr}(AP^T BP) \leq -2n$. The above equalities still hold, so $2 \sum_i A_{\pi(i), \pi(i+1)} \geq 2n$. Since the entries of $A$ are in $\{0, 1\}$, this inequality is only satisfied if each $A_{\pi(i), \pi(i+1)}$ is 1. In other words, for each $i$, the undirected edge $(\pi(i), \pi(i + 1))$ is in $E$, so $\pi$ defines a Hamiltonian cycle in $G$. Therefore we have shown a correct polynomial time many-one reduction from Hamiltonian Cycle to Maximum Agreement. \qed
Graph Isomorphism is the special case of the budget problem for Minimum Disagreement in which the budget is zero (in other words, it is the problem of finding a permutation causing no disagreements). It is also the special case of the budget problem for Maximum Agreement in which the budget is the number of edges in the graphs (in other words, it is the problem of finding a permutation that causes every possible agreement).

Theorem 2.5. Minimum Disagreement and Maximum Agreement are equivalent under TODO the strictest possible approximation-preserving reductions.

Maximum Agreement is a special case of Maximum Quadratic Assignment.

Definition 2.6 (Maximum Quadratic Assignment).
- Instance: $n \times n$ real matrices $A$ and $B$.
- Solution: $n \times n$ permutation matrix $P$.
- Measure: $-\text{tr}(AP^T BP)$.
- Type: minimization.

Since the budget problem for Maximum Agreement is NP-complete, so is the budget problem for Maximum Quadratic Assignment (I could not find this proven anywhere else). Since the budget problem is in NP, the optimization problem is in NPO. Although certain special cases are in APX (for example, see [2]), the general problem is APX-hard because some of its special cases are APX-hard [18]. The general problem has a polynomial time $O(\sqrt{n})$-approximation, but unless NP has a probabilistic quasipolynomial time simulation, there is no polylogarithmic approximation [18].

Maximum Agreement is a special case, so it may be more approximable. TODO What is the approximability of this problem?

2.2 Indefinite relaxed graph matching

Definition 2.7 (Relaxed Maximum Agreement).
- Instance: adjacency matrices $A$ and $B$ of two undirected graphs on $n$ vertices.
- Solution: $n \times n$ doubly stochastic matrix $D$.
- Measure: $-\text{tr}(AD^T BD)$.
- Type: minimization.

Relaxed Maximum Agreement is a special case of Minimum Quadratic Programming.

Definition 2.8 (Minimum Quadratic Programming).
- Instance: an $m \times n$ real matrix $A$, an $m \times 1$ real vector $b$, an $n \times 1$ real vector $c$, and an $n \times n$ real symmetric matrix $Q$.
- Solution: an $n \times 1$ real vector $x$, subject to the constraints $Ax \leq b$.
- Measure: $\frac{1}{2}x^T Qx + c^T x$.
- Type: minimization.
Why is the former optimization problem a special case of the latter? Consider the vector $x$ to be $\text{vec}(D)$. The feasible area can be defined by the linear constraints $D1 = 1$ and $D^T1 = 1$. By the definition of trace and tensor product, $\text{tr}(AD^TBD) = \text{vec}(D)^T(B \otimes A)\text{vec}(D)$. For the measure function, choose $Q = -2(B \otimes A)$ and $c = 0$. The matrix $Q$ is symmetric because the tensor product preserves symmetry.

The budget problem for Minimum Quadratic Programming is NP-complete [23], so the optimization problem is in NPO. Although special cases are approximable (for example, [22, 8]), in general no quasipolynomial approximation exists unless $\text{NP} \subseteq \text{QP}$, and no $O(1)$-approximation exists unless $\text{P} = \text{NP}$ [4].

The authors of [24] provide what, on first glance, appears to be a fully polynomial time approximation scheme for Relaxed Maximum Agreement. However, they provide no theoretical guarantee on the performance of their approximation algorithm, and it seems unlikely that any such conclusions could be drawn: the objective function is not convex, and hence the approximation guarantees provided by, for example, [12, Theorem 2.3], do not apply. TODO How approximable is this problem?

Even worse, I believe the objective function is never convex, regardless of input graphs. The quadratic form of the Hessian of the objective function, $x^THx$, equals $-2\text{tr}(AX \otimes BX)$, where $X$ is the $n \times n$ matrix containing the $n^2$ entries of the vector $x$. This value is negative if $x$ is a positive vector, so $H$ is not a positive definite matrix, and therefore the objective function has multiple local minima.

There is a polynomial time constant-factor approximation to the indefinite quadratic programming problem if the matrix of quadratic coefficients $Q$ has a constant number of negative eigenvalues [22]. By the definition of trace, tensor product, and eigenvalues of a matrix, $\lambda_{i,j}(A \otimes B) = \lambda_i(A)\lambda_j(B)$. Thus if both $A$ and $B$ have a constant number of negative eigenvalues, so does $A \otimes B$. [21, Theorem 1] gives a precise (though quite obtuse) characterization of graphs with a fixed number of negative eigenvalues; I have yet to fully understand it. TODO Might also be able to use [6], which is also somewhat complicated.

2.3 Convex relaxed graph matching

Definition 2.9 (Relaxed Minimum Disagreement).

Instance: adjacency matrices $A$ and $B$ of two undirected graphs on $n$ vertices.

Solution: $n \times n$ doubly stochastic matrix $D$.

Measure: $\|DA - BD\|^2$.

Type: minimization.

Relaxed Minimum Disagreement is a special case of the Convex Quadratic Programming problem (itself a special case of Minimum Quadratic Programming) TODO reference or explain why this is a special case.

Definition 2.10 (Convex Quadratic Programming).
Instance: an \( m \times n \) real matrix \( A \), an \( m \times 1 \) real vector \( b \), an \( n \times 1 \) real vector \( c \), and an \( n \times n \) real symmetric matrix \( Q \) such that the measure function is convex in \( x \).

Solution: an \( n \times 1 \) real vector \( x \), subject to the constraints \( Ax \leq b \).

Measure: \( \frac{1}{2} x^T Q x + c^T x \).

Type: minimization.

Convex Quadratic Programming is an inherently sequential problem because it admits a polynomial time computable function that produces optimal solutions, but it is a generalization of Linear Programming, which is itself inherently sequential.

**Theorem 2.11.** The budget problem for Convex Quadratic Programming is \( \mathcal{P} \)-complete.

**Proof.** Computing an exact solution to Convex Quadratic Programming can be done in polynomial time [9], so the problem is in \( \mathcal{P} \). Since a linear function is a convex function, and a linear function can be expressed as a quadratic function with \( Q = O \), Linear Programming is a special case of Convex Quadratic Programming, therefore the latter is \( \mathcal{P} \)-hard since the former is [10]. Therefore Convex Quadratic Programming is \( \mathcal{P} \)-complete. \( \Box \)

Since the budget problem for Convex Quadratic Programming is in \( \mathcal{P} \), Relaxed Minimum Disagreement is in \( \mathcal{PO} \).

TODO Is there a highly parallel approximation algorithm for Convex Quadratic Programming? For Relaxed Minimum Disagreement? Since the objective function is convex, we could use the Frank–Wolfe algorithm to solve this problem, which can be solved in a distributed manner at least (with a sparsity constraint) via [5]. Can we parallelize Frank–Wolfe by simply choosing a polynomial number of equally spaced starting points in the domain, then recursively shrinking the domain and recursing? (This may not work because finding the best direction may require linear programming, which is \( \mathcal{PO} \)-complete.)

### 2.4 Seeded graph matchings

TODO read and interpret [14, 17, 16].

The seeded graph matching problem is a generalization of the graph matching problem in which some pairs of vertices are required to be matched. The optimization problem corresponding to the seeded graph matching problem is called Seeded Minimum Disagreement. We can assume without loss of generality (by applying an additional permutation to the graphs) that if there are \( s \) seed pairs, then they are the vertices \( \{1, \ldots, s\} \) in both graphs. Thus, instead of finding an \( n \times n \) permutation matrix \( P \) that minimizes \( \|PA - BP\|^2 \) as in the graph matching problem, the goal is now to find a \( (n - s) \times (n - s) \) permutation matrix \( P \) that minimizes \( \|(I_s \oplus P)A - B(I_s \oplus P)\|^2 \).
\textbf{Definition 2.12 (Seeded Minimum Disagreement).}
Instance: symmetric $n \times n$ Boolean matrices $A$ and $B$, natural number $s$.
Solution: $(n - s) \times (n - s)$ permutation matrix $P$.
Measure: $\| (I_s \oplus P)A - B(I_s \oplus P) \|^2$.
Type: minimization.

\textbf{Definition 2.13 (Seeded Maximum Agreement).}
Instance: symmetric $n \times n$ Boolean matrices $A$ and $B$, natural number $s$.
Solution: $(n - s) \times (n - s)$ permutation matrix $P$.
Measure: $- \text{tr}(A(I_s \oplus P^T)B(I_s \oplus P))$.
Type: minimization.

These problems are at least as difficult as their unseeded counterparts, so their budget problems are NP-complete.

TODO How approximable are these problems? Can we use an approximation for the unseeded versions to solve the seeded versions?

\section{Seeded relaxed graph matchings}

\textbf{Definition 2.14 (Seeded Relaxed Minimum Disagreement).}
Instance: symmetric $n \times n$ Boolean matrices $A$ and $B$, $s \times s$ doubly stochastic matrix $C$.
Solution: $(n - s) \times (n - s)$ doubly stochastic matrix $D$.
Measure: $\| (C \oplus D)A - B(C \oplus D) \|^2$.
Type: minimization.

Just as \textit{Relaxed Minimum Disagreement} is in \textit{PO}, so too is \textit{Seeded Relaxed Minimum Disagreement}, for the same reason.

\textbf{Definition 2.15 (Seeded Relaxed Maximum Agreement).}
Instance: symmetric $n \times n$ Boolean matrices $A$ and $B$, $s \times s$ doubly stochastic matrix $C$.
Solution: $(n - s) \times (n - s)$ doubly stochastic matrix $D$.
Measure: $- \text{tr}(A(C^T \oplus D^T)B(C \oplus D))$.
Type: minimization.

TODO How approximable are these problems? Can we use an approximation for the unseeded versions to solve the seeded versions?

\section{Using relaxed matchings to recover matchings}

Much of the current research seems to involve computing a relaxed graph matching (using one of the two formulations), then projecting that doubly stochastic matrix onto a permutation matrix in order to get a graph matching. In \cite{15}, the authors prove that when doing this, it is best to solve the indefinite relaxed graph...
matching problem instead of the convex relaxed graph matching problem (that is, Relaxed Maximum Agreement instead of Relaxed Minimum Disagreement), because the former is almost always right and the latter almost always wrong; \cite{24} follows this strategy. Still, under certain conditions, solving the convex relaxed graph matching problem can yield the correct permutation matrix \cite{[1],[7]}. \cite{[13]} seems to be the only work simply finding a relaxed graph matching without projecting that doubly stochastic matrix back to a permutation matrix (they are really only finding a relaxed graph matching between a single graph and itself, but it should be possible, using \cite{[11]}, to enforce that when running such an algorithm on the disjoint union of two graphs $A$ and $B$, the solution matches some vertices of $A$ with some vertices of $B$).

A graph is compact if the polytope of its fractional automorphisms is integral (that is, if the vertices of the polytope of doubly stochastic matrices $S$ such that $SA = AS$ have all integer entries). Deciding whether two compact graphs are isomorphic can be done by computing a vertex of this polytope and checking whether it is integral. (See \cite{[3]} for more information.) In other words, for compact graphs, no projection is necessary for recovering an isomorphism from a doubly stochastic matrix found via a relaxed graph matching procedure: the matrix is either already a permutation matrix, or, if not, the graphs are not isomorphic.

\section{Amplifying disagreements}

One of our goals is to find a graph operation that, when applied to graphs that are nearly isomorphic, makes the graphs less isomorphic.

Suppose $G$ and $H$ are undirected, unweighted, nonempty graphs on $n$ vertices with adjacency matrices $A$ and $B$, respectively. Suppose $\pi$ is a bijection from $V(G)$ to $V(H)$ and $P$ is its corresponding permutation matrix. For each permutation $\pi$, the agreement between $G$ and $H$ with respect to $\pi$, denoted $\text{agr}_\pi(G, H)$, is defined by

$$\text{agr}_\pi(G, H) = \frac{\text{tr}(AP^TBP)}{\max(||A||^2, ||B||^2)}.$$ 

In other words, the agreement between $G$ and $H$ is the ratio of the number of edges correctly matched by $\pi$ to the total number of edges (in \cite{[19]}, the agreement is defined equivalently in these graph theoretic terms instead of our linear algebraic terms). The agreeability between $G$ and $H$, denoted $\text{agr}(G, H)$, is defined by

$$\text{agr}(G, H) = \max_{\pi \in S_n} \text{agr}_\pi(G, H).$$

When $G$ and $H$ admit a better matching, $\text{agr}(G, H)$ increases. The graphs $G$ and $H$ are isomorphic exactly when $\text{agr}(G, H) = 1$.

Now we hope to find a graph operation $\star$ such that $\text{agr}(G \star X, H \star Y) < \text{agr}(G, H)$ for some graphs $X$ and $Y$. One initial attempt is the tensor product, denoted $\otimes$. A second attempt may be graph powering; something more complicated like the zig-zag product might come later.
Definition 4.1 (Tensor products).

- If $A$ and $B$ are $n \times n$ matrices, then the tensor product of $A$ and $B$, denoted $A \otimes B$, is an $n^2 \times n^2$ matrix defined as follows. Entry $(i, j, i', j')$ of $A \otimes B$ is $a_{i,j}b_{i',j'}$.

- If $G$ and $H$ are graphs with adjacency matrices $A$ and $B$, respectively, then the tensor product of $G$ and $H$, denoted $G \otimes H$, is the graph whose adjacency matrix is $A \otimes B$.

- If $\pi$ and $\tau$ are permutations in $S_n$, then the tensor product of $\pi$ and $\tau$, denoted $\pi \otimes \tau$, is a permutation in $S_{n^2}$ defined by $(\pi \otimes \tau)(i, j) = (\pi(i), \tau(j))$.

Theorem 4.2. If $\pi$ and $\tau$ are permutations with corresponding permutation matrices $P$ and $Q$, respectively, then $\pi \otimes \tau$ is a permutation with corresponding permutation $P \otimes Q$.

Theorem 4.3. Suppose $A$ and $B$ are $n \times n$ Boolean matrices. We have $A = B$ if and only if $A \otimes A = B \otimes B$.

Proof. If $A = B$ then $A \otimes A = B \otimes B$ immediately from the definitions. Suppose $A \neq B$. Assume without loss of generality that the top left entry of $A$ is 0 and the top left entry of $B$ is 1. Then the top left entry of $A \otimes A$ is 0 and the top left entry of $B \otimes B$ is 1. Therefore $A \otimes A \neq B \otimes B$. \qed

Theorem 4.4. For any undirected, unweighted, nonempty graphs $G$ and $H$ on $n$ vertices, and for any bijection $\pi$ from $V(G)$ to $V(H)$,

$$\text{agr}_{\pi \otimes \pi}(G \otimes G, H \otimes H) = \text{agr}_\pi(G, H)^2.$$  

Proof. This follows directly from the properties of the tensor product. We consider the numerator and the denominator of the agreement separately. Suppose $A$ and $B$ are the adjacency matrices of $G$ and $H$, respectively, and suppose $P$ is the permutation matrix corresponding to $\pi$. For the numerator,

$$\text{tr}((A \otimes A)(P \otimes P)^T(B \otimes B)(P \otimes P))$$

$$= \text{tr}((A \otimes A)(P^T \otimes P^T)(B \otimes B)(P \otimes P))$$

$$= \text{tr}((AP^T \otimes AP^T)(BP \otimes BP))$$

$$= \text{tr}(AP^T BP \otimes AP^T BP)$$

$$= \text{tr}(AP^T BP)^2.$$  

For the denominator,

$$\max(\|A \otimes A\|^2, \|B \otimes B\|^2) = \max((\|A\|^2)^2, (\|B\|^2)^2) = \max(\|A\|^2, \|B\|^2)^2.$$  

Combine these two equalities with the definition of the agreement between $G$ and $H$ to produce the equality in the statement of the theorem. \qed

The next corollary follows from the previous theorem by induction on $\ell$.  

11
Corollary 4.5. Suppose \( \ell \) is a positive integer, \( G \) and \( H \) are undirected, unweighted, nonempty graphs on \( n \) vertices, and \( \pi \) is a bijection from \( V(G) \) to \( V(H) \).

Let \( G = \bigotimes_{i=1}^{\ell} G \), let \( H = \bigotimes_{i=1}^{\ell} H \), and let \( \Pi = \bigotimes_{i=1}^{\ell} \pi \). Now
\[
\text{agr}_\Pi(G, H) = \text{agr}_\pi(G, H)^\ell.
\]

This corollary provides a lower bound for the agreement of the larger graphs \( G \) and \( H \) given a lower bound for the agreement of the original graphs \( G \) and \( H \), though the bound is rather unhelpful because it decreases rapidly. However, it does show that the tensor product, when used in this way, preserves isomorphism.

Theorem 4.6. Suppose \( \ell \) is a positive integer and \( G \) and \( H \) are undirected, unweighted, nonempty graphs \( G \) and \( H \) on \( n \) vertices. Let \( G = \bigotimes_{i=1}^{\ell} G \) and \( H = \bigotimes_{i=1}^{\ell} H \).

\[
\text{agr}(G, H)^\ell \leq \text{agr}(G, H),
\]
with equality when \( G \) is isomorphic to \( H \). In particular, for any \( \epsilon \in [0, 1] \), if \( \epsilon \leq \text{agr}(G, H) \) then \( \epsilon^\ell \leq \text{agr}(G, H) \).

Proof. Let \( \pi \) be the bijection that maximizes \( \text{agr}_\pi(G, H) \) and let \( \Pi = \bigotimes_{i=1}^{\ell} \pi \). By the previous corollary,
\[
\text{agr}(G, H)^\ell = \text{agr}_\pi(G, H)^\ell = \text{agr}_\Pi(G, H).
\]

Since \( \Pi \) is just one bijection out of all possible bijections between the vertices of \( G \) and \( H \),
\[
\text{agr}_\Pi(G, H) \leq \max_{\tau} \text{agr}_\tau(G, H) = \text{agr}(G, H).
\]
Combine these equalities and inequalities to produce the inequality in the statement of the theorem. If \( G \) and \( H \) are isomorphic, then \( \text{agr}(G, H)^\ell = 1^\ell = 1 \), so
\[
1 = \text{agr}(G, H)^\ell \leq \text{agr}(G, H) \leq 1.
\]
Finally, if \( \epsilon \leq \text{agr}(G, H) \), then \( \epsilon^\ell \leq \text{agr}(G, H)^\ell \leq \text{agr}(G, H) \).

What we would like is an upper bound for the agreement of the larger graphs \( G \) and \( H \) given an upper bound for the agreement of the original graphs \( G \) and \( H \). Something like if \( \text{agr}(G, H) < \epsilon \), then \( \text{agr}(G, H) < \epsilon' < \epsilon \). **TODO** How to show this?

**TODO** Is it true that if \( \hat{P}(A \otimes A) = (B \otimes B)\hat{P} \) then \( \hat{P} = P \otimes P \) for some permutation matrix \( P \)? If so, then we can conclude that \( PA = BP \). In [19], the authors claim, citing a complicated 1971 math paper, that if \( G \otimes G \) is isomorphic to \( H \otimes H \), then \( G \) is isomorphic to \( H \).

**TODO** Is it easier to show that some graph product increases disagreement instead of showing that it decreases agreement?

**TODO** What if we use the relaxed versions of the problem here?
References


[23] Stephen A. Vavasis. “Quadratic programming is in NP”. In: Information Processing Letters 36.2 (1990), pp. 73–77. ISSN: 0020-0190. DOI: 10.1016/0020-0190(90)90100-C.