Spectral Graph Theory Lecture 8

Testing Isomorphism of Graphs with Distinct Eigenvalues

Daniel A. Spielman September 24, 2018

8.1 Introduction

I will present an algorithm of Leighton and Miller [LM82] for testing isomorphism of graphs in which all eigenvalues have multiplicity 1. This algorithm was never published, as the results were technically subsumed by those in a paper of Babai, Grigoriev and Mount [BGM82], which gave a polynomial time algorithm for testing isomorphism of graphs in which all eigenvalues have multiplicity bounded by a constant.

I present the weaker result in the interest of simplicity.

Testing isomorphism of graphs is a notorious problem. Until very recently, the fastest-known algorithm for it took time time $2^{\sqrt{O(n \log n)}}$ (See [Bab81, BL83, ZKT85]). Babai [Bab16] recently announced a breakthrough that reduces the complexity to $2^{(\log n)^{O(1)}}$.

However, testing graph isomorphism seems easy in almost all practical instances. Today's lecture and one next week will give you some idea as to why.

8.2 Graph Isomorphism

Recall that two graphs $G = (V, E)$ and $H = (V, F)$ are isomorphic if there exists a permutation π of V such that

$$(a, b) \in E \iff (\pi(a), \pi(b)) \in F.$$

Of course, we can express this relation in terms of matrices associated with the graphs. It doesn't matter much which matrices we use. So for this lecture we will use the adjacency matrices.

Every permutation may be realized by a *permutation matrix*. For the permutation π, this is the matrix $\mathbf{\Pi}$ with entries given by

$$\mathbf{\Pi}(a, b) = \begin{cases} 1 & \text{if } \pi(a) = b \\ 0 & \text{otherwise.} \end{cases}$$

For a vector $\boldsymbol{\psi}$, we see[1] that

$$(\mathbf{\Pi}\boldsymbol{\psi})(a) = \boldsymbol{\psi}(\pi(a)).$$

[1] I hope I got that right. It's very easy to confuse the permutation and its inverse.

Let A be the adjacency matrix of G and let B be the adjacency matrix of H. We see that G and H are isomorphic if and only if there exists a permutation matrix $\mathbf{\Pi}$ such that

$$\mathbf{\Pi} A \mathbf{\Pi}^T = B.$$

8.3 Using Eigenvalues and Eigenvectors

If G and H are isomorphic, then A and B must have the same eigenvalues. However, there are many pairs of graphs that are non-isomorphic but which have the same eigenvalues. We will see some tricky ones next lecture. But, for now, we note that if A and B have different eigenvalues, then we know that the corresponding graphs are non-isomorphic, and we don't have to worry about them.

For the rest of this lecture, we will assume that A and B have the same eigenvalues, and that each of these eigenvalues has multiplicity 1. We will begin our study of this situation by considering some cases in which testing isomorphism is easy.

Recall that we can write

$$A = \boldsymbol{\Psi} \Lambda \boldsymbol{\Psi}^T,$$

where Λ is the diagonal matrix of eigenvalues of A and $\boldsymbol{\Psi}$ is an orthonormal matrix holding its eigenvectors. If B has the same eigenvalues, we can write

$$B = \boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T.$$

If $\mathbf{\Pi}$ is the matrix of an isomorphism from G to H, then

$$\mathbf{\Pi} \boldsymbol{\Psi} \Lambda \boldsymbol{\Psi}^T \mathbf{\Pi}^T = \boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T.$$

As each entry of Λ is distinct, this looks like it would imply $\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi}$. But, the eigenvectors (columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$) are only determined up to sign. So, it just implies

$$\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi} S,$$

where S is a diagonal matrix with ± 1 entries on its diagonal.

Lemma 8.3.1. *Let $\boldsymbol{A} = \boldsymbol{\Psi} \Lambda \boldsymbol{\Psi}^T$ and $\boldsymbol{B} = \boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T$ where Λ is a diagonal matrix with distinct entries and $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ are orthogonal matrices. A permutation matrix $\mathbf{\Pi}$ satisfies $\mathbf{\Pi} \boldsymbol{A} \mathbf{\Pi}^T = \boldsymbol{B}$ if and only if there exists a diagonal ± 1 matrix $\boldsymbol{S}$ for which*

$$\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi} \boldsymbol{S}.$$

Proof. Let $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ be the columns of $\boldsymbol{\Psi}$ and let $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n$ be the columns of $\boldsymbol{\Phi}$. Assuming there is a $\mathbf{\Pi}$ for which $\mathbf{\Pi} \boldsymbol{A} \mathbf{\Pi}^T = \boldsymbol{B}$,

$$\boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T = \sum_{i=1}^{n} \boldsymbol{\phi}_i \lambda_i \boldsymbol{\phi}_i^T = \sum_{i=1}^{n} (\mathbf{\Pi} \boldsymbol{\psi}_i) \lambda_i (\boldsymbol{\psi}_i^T \mathbf{\Pi}^T),$$

which implies that for all i

$$\boldsymbol{\phi}_i \boldsymbol{\phi}_i^T = (\boldsymbol{\Pi}\boldsymbol{\psi}_i)(\boldsymbol{\Pi}\boldsymbol{\psi}_i)^T.$$

This in turn implies that

$$\boldsymbol{\phi}_i = \pm \boldsymbol{\Pi}\boldsymbol{\psi}_i.$$

To go the other direction, assume $\boldsymbol{\Pi}\,\boldsymbol{\Psi} = \boldsymbol{\Phi}\boldsymbol{S}$. Then,

$$\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{\Pi}^T = \boldsymbol{\Pi}\,\boldsymbol{\Psi}\boldsymbol{\Lambda}\,\boldsymbol{\Psi}^T\boldsymbol{\Pi}^T = \boldsymbol{\Phi}\boldsymbol{S}\boldsymbol{\Lambda}\boldsymbol{S}\,\boldsymbol{\Phi}^T = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{S}\boldsymbol{S}\,\boldsymbol{\Phi}^T = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T = \boldsymbol{B},$$

as $\boldsymbol{S}$ and $\boldsymbol{\Lambda}$ are diagonal and thus commute, and $\boldsymbol{S}^2 = \boldsymbol{I}$. □

Our algorithm for testing isomorphism will determine all such matrices $\boldsymbol{S}$. Let $\mathcal{S}$ be the set of all diagonal ± 1 matrices. We will find diagonal matrices $\boldsymbol{S} \in \mathcal{S}$ such that the set of rows of $\boldsymbol{\Phi}\boldsymbol{S}$ is the same as the set of rows of $\boldsymbol{\Psi}$. As the rows of $\boldsymbol{\Psi}$ are indexed by vertices $a \in V$, we will write the row indexed by a as the row-vector

$$\boldsymbol{v}_a \stackrel{\text{def}}{=} (\boldsymbol{\psi}_1(a), \ldots, \boldsymbol{\psi}_n(a)).$$

Similarly denote the rows of $\boldsymbol{\Phi}$ by vectors $\boldsymbol{u}_a$. In this notation, we are searching for matrices $\boldsymbol{S} \in \mathcal{S}$ for which the set of vectors $\{\boldsymbol{v}_a\}_{a \in V}$ is identical to the set of vectors $\{\boldsymbol{u}_a \boldsymbol{S}\}_{a \in V}$ We have thus transformed the graph isomorphism problem into a problem about vectors:

8.4 An easy case

I will say that an eigenvector $\boldsymbol{\psi}_i$ is *helpful* if for all $a \neq b \in V$, $|\boldsymbol{\psi}_i(a)| \neq |\boldsymbol{\psi}_i(b)|$. In this case, it is very easy to test if G and H are isomorphic, because this helpful vector gives us a canonical name for every vertex. If $\boldsymbol{\Pi}$ is an isomorphism from G to H, then $\boldsymbol{\Pi}\boldsymbol{\psi}_i$ must be an eigenvector of B. In fact, is must be $\pm\boldsymbol{\phi}_i$. If the sets of absolute values of entries of $\boldsymbol{\psi}_i$ and $\boldsymbol{\phi}_i$ are the same, then we may find the permutation that maps $\boldsymbol{A}$ to $\boldsymbol{B}$ by mapping every vertex a to the vertex b for which $|\boldsymbol{\psi}_i(a)| = |\boldsymbol{\phi}_i(b)|$.

The reason that I put absolute values in the definition of helpful, rather than just taking values, is that eigenvectors are only determined up to sign. On the other hand, a single eigenvector determines the isomorphism if $\boldsymbol{\psi}_i(a) \neq \boldsymbol{\psi}_i(b)$ for all $a \neq b$ and there is a canonical way to choose a sign for the vector $\boldsymbol{\psi}_i$. For example, if the sum of the entries in $\boldsymbol{\psi}_i$ is not zero, we can choose its sign to make the sum positive. In fact, unless $\boldsymbol{\psi}_i$ and $-\boldsymbol{\psi}_i$ have exactly the same set of values, there is a canonical choice of the sign for this vector.

Even if there is no canonical choice of sign for this vector, it leaves at most two choices for the isomorphism.

8.5 All the Automorphisms

The graph isomorphism problem is complicated by the fact that there can be many isomorphisms from one graph to another. So, any algorithm for finding isomorphisms must be able to find many

of them.

Recall that an *automorphism* of a graph is an isomorphism from the graph to itself. These form a group which we denote $\mathsf{aut}(G)$: if $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ are automorphisms of $\boldsymbol{A}$ then so is $\boldsymbol{\Pi}\boldsymbol{\Gamma}$. Let $\mathcal{A} \subseteq \mathcal{S}$ denote the corresponding set of diagonal ± 1 matrices. The set $\mathcal{A}$ is in fact a group and is isomorphic to $\mathsf{aut}(G)$.

Here is a way to make this isomorphism very concrete: Lemma 8.3.1 implies that the $\boldsymbol{\Pi} \in \mathsf{aut}(G)$ and the $\boldsymbol{S} \in \mathcal{A}$ are related by

$$\boldsymbol{\Pi} = \boldsymbol{\Psi}\boldsymbol{S}\boldsymbol{\Psi}^T \quad \text{and} \quad \boldsymbol{S} = \boldsymbol{\Psi}^T\boldsymbol{\Pi}\boldsymbol{\Psi}.$$

As diagonal matrices commute, we have that for every $\boldsymbol{\Pi}_1$ and $\boldsymbol{\Pi}_2$ in $\mathsf{aut}(G)$ and for $\boldsymbol{S}_1 = \boldsymbol{\Psi}^T\boldsymbol{\Pi}_1\boldsymbol{\Psi}$ and $\boldsymbol{S}_2 = \boldsymbol{\Psi}^T\boldsymbol{\Pi}_2\boldsymbol{\Psi}$,

$$\boldsymbol{\Pi}_1\boldsymbol{\Pi}_2 = \boldsymbol{\Psi}\boldsymbol{S}_1\boldsymbol{\Psi}^T\boldsymbol{\Psi}\boldsymbol{S}_2\boldsymbol{\Psi}^T = \boldsymbol{\Psi}\boldsymbol{S}_1\boldsymbol{S}_2\boldsymbol{\Psi}^T = \boldsymbol{\Psi}\boldsymbol{S}_2\boldsymbol{S}_1\boldsymbol{\Psi}^T = \boldsymbol{\Psi}\boldsymbol{S}_2\boldsymbol{\Psi}^T\boldsymbol{\Psi}\boldsymbol{S}_1\boldsymbol{\Psi}^T = \boldsymbol{\Pi}_2\boldsymbol{\Pi}_1.$$

Thus, the automorphism group of a graph with distinct eigenvalues is commutative, and it is isomorphic to a subgroup of $\mathcal{S}$.

It might be easier to think about these subgroups by realizing that they are isomorphic to subspaces of $(\mathbf{Z}/2\mathbf{Z})^n$. Let $f : \mathcal{S} \to (\mathbf{Z}/2\mathbf{Z})^n$ be the function that maps the group of diagonal matrices with ± 1 entries to vectors $\boldsymbol{t}$ modulo 2 by setting $\boldsymbol{t}(i)$ so that $\boldsymbol{S}(i,i) = (-1)^{\boldsymbol{t}(i)}$. You should check that this is a group homomorphism: $f(\boldsymbol{S}_1\boldsymbol{S}_2) = f(\boldsymbol{S}_1) + f(\boldsymbol{S}_2)$. You should also confirm that f is invertible.

For today's lecture, we will focus on the problem of finding the group of automorphisms of a graph with distinct eigenvalues. We will probably save the slight extension to finding isomorphisms for homework. Note that we will not try to list all the isomorphisms, as there could be many. Rather, we will give a basis of the corresponding subspace of $(\mathbf{Z}/2\mathbf{Z})^n$.

8.6 Equivalence Classes of Vertices

Recall that the *orbit* of an element under the action of a group is the set of elements to which it is mapped by the elements of the group. Concretely, the orbit of a vertex a in the graph is the set of vertices to which it can be mapped by automorphisms. We will discover the orbits by realizing that the orbit of a vertex a is the set of b for which $\boldsymbol{v}_a\boldsymbol{S} = \boldsymbol{v}_b$ for some $\boldsymbol{S} \in \mathcal{A}$.

The set of orbits of vertices forms a partition of the vertices. We say that a partition of the vertices is *valid* if every orbit is contained entirely within one set in the partition. That is, each class of the partition is a union of orbits. Our algorithm will proceed by constructing a valid partition of the vertices and then splitting classes in the partition until each is exactly an orbit.

Recall that a set is *stabilized* by a group if the set is unchanged when the group acts on all of its members. We will say that a group $\mathcal{G} \subseteq \mathcal{S}$ stabilizes a set of vertices C if it stabilizes the set of vectors $\{\boldsymbol{v}_a\}_{a \in C}$. Thus, $\mathcal{A}$ is the group that stabilizes V.

An orbit is stabilized by $\mathcal{A}$, and so are unions of orbits and thus classes of valid partitions. We would like to construct the subgroup of $\mathcal{S}$ that stabilizes each orbit C_j. *However, I do not yet see how to do that directly.* Instead, we will construct a particular valid partition of the vertices, and find for each class in the partition C_j the subgroup of $\mathcal{A}_j \subseteq \mathcal{S}$ that stabilizes C_j, where here we are considering the actions of matrices $\boldsymbol{S} \in \mathcal{S}$ on vectors $\boldsymbol{v}_a$. In fact, $\mathcal{A}_j$ will act transitively[2] on the class C_j. As $\mathcal{A}$ stabilizes every orbit, and thus every union of orbits, it is a subgroup of $\mathcal{A}_j$. In fact, $\mathcal{A}$ is exactly the intersection of all the groups $\mathcal{A}_j$.

We now observe that we can use linear algebra to efficiently construct $\mathcal{A}$ from the groups $\mathcal{A}_j$ by exploiting the isomorphism between $\boldsymbol{S}$ and $(\mathbf{Z}/2)^n$. Each subgroup $\mathcal{A}_j$ is isomorphic to a subgroup of $(\mathbf{Z}/2)^n$. Each subgroup of $(\mathbf{Z}/2)^n$ is precisely a vector space modulo 2, and thus may be described by a basis. It will eventually become clear that by "compute $\mathcal{A}_j$" we mean to compute such a basis. From the basis, we may compute a basis of the nullspace. The subgroup of $(\mathbf{Z}/2)^n$ corresponding to $\mathcal{A}$ is then the nullspace of the span of the nullspaces of the subspaces corresponding to the $\mathcal{A}_j$. We can compute all these using Gaussian elimination.

8.7 The first partition

We may begin by dividing vertices according to the absolute values of their entries in eigenvectors. That is, if $|\boldsymbol{\psi}_i(a)| \neq |\boldsymbol{\psi}_i(b)|$ for some i, then we may place vertices a and b in different classes, as there can be no $\boldsymbol{S} \in \mathcal{S}$ for which $\boldsymbol{v}_a\boldsymbol{S} = \boldsymbol{v}_b$. The partition that we obtain this way is thus valid, and is the starting point of our algorithm.

8.8 Unbalanced vectors

We say that an eigenvector $\boldsymbol{\psi}_i$ is *unbalanced* if there is some value x for which

$$|\{a : \boldsymbol{\psi}_i(a) = x\}| \neq |\{a : \boldsymbol{\psi}_i(a) = -x\}|\,.$$

Such vectors cannot change sign in an automorphism. That is, $\boldsymbol{S}(i,i)$ must equal 1. The reason is that an automorphism with $\boldsymbol{S}(i,i) = -1$ must induce a bijection between the two sets above, but this is impossible if their sizes are different.

Thus, an unbalanced vector tells us that all vertices for which $\boldsymbol{\psi}_i(a) = x$ are in different orbits from those for which $\boldsymbol{\psi}_i(a) = -x$. This lets us refine classes.

We now extend this idea in two ways. First, we say that $\boldsymbol{\psi}_i$ is *unbalanced* on a class C if there is some value x for which

$$|\{a \in C : \boldsymbol{\psi}_i(a) = x\}| \neq |\{a \in C : \boldsymbol{\psi}_i(a) = -x\}|\,.$$

By the same reasoning, we can infer that the sign of $\boldsymbol{S}(i,i)$ must be fixed to 1. Assuming, as will be the case, that C is a class in a valid partition and thus a union of orbits, we are now able to

[2]That is, for every a and b in C_j, there is an $\boldsymbol{S} \in \mathcal{A}_j$ for which $\boldsymbol{v}_a\boldsymbol{S} = \boldsymbol{b}_b$.

split C into two smaller classes

$$C_0 = \{a \in C : \boldsymbol{\psi}_i(a) = x\} \quad \text{and} \quad C_1 = \{a \in C : \boldsymbol{\psi}_i(a) = -x\}\,.$$

The partition we obtain by splitting C into C_1 and C_2 is thus also valid. Of course, it is only useful if both sets are non-empty.

Finally, we consider vectors formed from products of eigenvectors. For $R \subseteq \{1, \ldots, n\}$, define $\boldsymbol{\psi}_R$ to be the component-wise product of the $\boldsymbol{\psi}_i$ for $i \in R$:

$$\boldsymbol{\psi}_R(a) = \prod_{i \in R} \boldsymbol{\psi}_i(a).$$

We say that the vector $\boldsymbol{\psi}_R$ is unbalanced on class C if there is some value x for which

$$|\{a \in C : \boldsymbol{\psi}_R(a) = x\}| \neq |\{a \in C : \boldsymbol{\psi}_R(a) = -x\}|\,.$$

An unbalanced vector of this form again tells us that the vertices in the two sets belong to different orbits. So, if both sets are nonempty we can use such a vector to split the class C in two to obtain a more refined valid partition. It also provides some relations between the entries of $\boldsymbol{S}$, but we will not exploit those.

We say that a vector is *balanced* if it is not unbalanced.

We say that a subset of the vertices $C \subseteq V$ is *balanced* if *every* non-constant product of eigenvectors is balanced on C. Thus, orbits are balanced. Our algorithm will partition the vertices into balanced classes.

My confusion over this lecture stemmed from thinking that all balanced classes must be orbits. But, I don't know if this is true.

Question: Is every balanced class an orbit of $\mathcal{A}$?

8.9 The structure of the balanced classes

Let C_j be a balanced class. By definition, the product of every subset of eigenvectors is either constant or balanced on C_j. We say that a subset of eigenvectors Q is *independent* on C_j if all products of subsets of eigenvectors in Q are balanced on C_j (except for the empty product). In particular, none of these eigenvectors is zero or constant on C_j. Construct a matrix $M_{C_j,Q}$ whose rows are indexed by vertices in $a \in C_j$, whose columns are indexed by subsets $R \subseteq Q$, and whose entries are given by

$$M_{C_j,Q}(a, R) = \mathsf{sgn}(\boldsymbol{\psi}_R(a)), \text{where I recall } \mathsf{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0, \text{and} \\ 0 & \text{if } x = 0. \end{cases}$$

Lemma 8.9.1. *If Q is independent on C then the columns of $M_{C,Q}$ are orthogonal.*

Proof. Let R_1 and R_2 index two columns of $M_{C,Q}$. That is, R_1 and R_2 are two different subsets of Q. Let R_0 be their symmetric difference. We have

$$M_{C,Q}(a,R_1)M_{C,Q}(a,R_2) = \mathsf{sgn}(\boldsymbol{\psi}_{R_1}(a))\mathsf{sgn}(\boldsymbol{\psi}_{R_2}(a)) = \prod_{i\in R_1}\mathsf{sgn}(\boldsymbol{\psi}_i(a))\prod_{i\in R_2}\mathsf{sgn}(\boldsymbol{\psi}_i(a)) = \prod_{i\in R_0}\mathsf{sgn}(\boldsymbol{\psi}_i(a)) = \mathsf{sgn}(\boldsymbol{\psi}_{R_0}(a)) = M_{C,Q}(a,R_0).$$

As all the nonempty products of subsets of eigenvectors in Q are balanced on C, $M_{C,Q}(a,R_0)$ is positive for half the $a \in C$ and negative for the other half. So,

$$M_{C,Q}(:,R_1)^T M_{C,Q}(:,R_2) = \sum_{a\in C} M_{C,Q}(a,R_1)M_{C,Q}(a,R_2) = \sum_{a\in C} M_{C,Q}(a,R_0) = 0.$$

□

Lemma 8.9.2. *If C is a balanced class of vertices and Q is a maximal set of eigenvectors that are independent on C, then for every a and b in C there is an $i \in Q$ for which $\boldsymbol{\psi}_i(a) \neq \boldsymbol{\psi}_i(b)$.*

Proof. Assume by way of contradiction that this does not hold. There must be some eigenvector i for which $\boldsymbol{\psi}_i(a) \neq \boldsymbol{\psi}_i(b)$. We will show that if we added i to Q, the product of every subset would still be balanced. As we already know this for subsets of Q, we just have to prove it for subsets of the form $R \cup \{i\}$, where $R \subseteq Q$. As $\boldsymbol{\psi}_h(a) = \boldsymbol{\psi}_h(b)$ for every $h \in Q$, $\boldsymbol{\psi}_R(a) = \boldsymbol{\psi}_R(b)$. This implies $\boldsymbol{\psi}_{R\cup\{i\}}(a) \neq \boldsymbol{\psi}_{R\cup\{i\}}(b)$. Thus, $\boldsymbol{\psi}_{R\cup\{i\}}$ is not uniform on C, and so it must be balanced on C. □

Lemma 8.9.3. *If C is a balanced class of vertices and Q is a maximal set of eigenvectors that are independent on C, then the rows of $M_{C,Q}$ are orthogonal.*

Proof. Let a and b be in C. From Lemma 8.9.2 we know that there is an $i \in Q$ for which $\boldsymbol{\psi}_i(a) = -\boldsymbol{\psi}_i(b)$. To prove that the rows $M_{C,Q}(a,:)$ and $M_{C,Q}(b,:)$ are orthogonal, we compute their inner product:

$$\begin{aligned}
\sum_{R\subseteq Q}\mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) &= \sum_{R\subseteq Q-\{i\}}\mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) + \mathsf{sgn}(\boldsymbol{\psi}_{R\cup\{i\}}(a)\boldsymbol{\psi}_{R\cup\{i\}}(b)) \\
&= \sum_{R\subseteq Q-\{i\}}\mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) + \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_i(a)\boldsymbol{\psi}_R(b)\boldsymbol{\psi}_i(b)) \\
&= \sum_{R\subseteq Q-\{i\}}\mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) + \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b))\mathsf{sgn}(\boldsymbol{\psi}_i(a)\boldsymbol{\psi}_i(b)) \\
&= \sum_{R\subseteq Q-\{i\}}\mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) - \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) \\
&= 0.
\end{aligned}$$

□

Corollary 8.9.4. *Let C be a balanced subset of vertices. Then the size of C is a power of 2. If Q is an independent set of eigenvectors on C, then $|Q| \leq \log_2 |C|$.*

Proof. Let C be an orbit and let Q be a maximal set of eigenvectors that are independent on C. As the rows and columns of $M_{C,Q}$ are both orthogonal, $M_{C,Q}$ must be square. This implies that $|C| = 2^{|Q|}$. If we drop the assumption that Q is maximal, we still know that all the columns of $M_{C,Q}$ are orthogonal. This matrix has $2^{|Q|}$ columns. As they are vectors in $|C|$ dimensions, there can be at most $|C|$ of them. □

We can now describe the structure of a balanced subset of vertices C. We call a maximal set of eigenvectors that are independent on C a *base* for C. Every other eigenvector j is either constant on C or becomes constant when multiplied by the product of some subset R of eigenvectors in Q. In either case, we can write

$$\boldsymbol{\psi}_j(a) = \gamma \prod_{i \in R} \boldsymbol{\psi}_i(a) \quad \text{for all } a \in C, \tag{8.1}$$

for some constant γ.

Let $\boldsymbol{v}_a(Q)$ denote the vector $(\boldsymbol{v}_a(i))_{i \in Q}$—the restriction of the vector $\boldsymbol{v}_a$ to the coordinates in Q. I claim that every one of the $2^{|Q|}$ $\pm$ sign patterns of length $|Q|$ must appear in exactly one of the vectors $\boldsymbol{v}_q(Q)$. The reason is that there are $|C| = 2^{|Q|}$ of these vectors, and we established in Lemma 8.9.2 that $\boldsymbol{v}_a(Q) \neq \boldsymbol{v}_b(Q)$ for all $a \neq b$ in Q. Thus, for every diagonal $\pm$ matrix $\boldsymbol{S}_Q$ of dimension $|Q|$, we have

$$\{\boldsymbol{v}_a(Q)\boldsymbol{S}_Q : a \in C\} = \{\boldsymbol{v}_a(Q) : a \in C\}.$$

That is, this set of vectors is stabilized by ± 1 diagonal matrices.

As equation (8.1) gives a formula for the value taken on C by every eigenvector not in Q in terms of the eigenvectors in Q, we have described the structure of the subgroup of $\boldsymbol{S}$ that stabilizes C: the diagonals corresponding to Q are unconstrained, and every other diagonal is some product of these. This structure is something that you are used to seeing in subspaces. Apply f to map this subgroup of $\boldsymbol{S}$ to $(\mathbf{Z}/2)^n$, and let $\boldsymbol{B}$ be a n-by-$\log_2(|C|)$ matrix containing a basis of the subspace in its columns. Any independent subset of $\log_2(|C|)$ rows of $\boldsymbol{B}$ will form a basis of the row-space, and is isomorphic to a base for C of the eigenvectors.

8.10 Algorithms

Let C_j be a balanced class. We just saw how to compute $\mathcal{A}_j$, assuming that we know C_j and a base Q for it. Of course, by "compute" we mean computing a basis of $f(\mathcal{A}_j)$. We now show how to find a base for a balanced class C_j. We do this by building up a set Q of eigenvectors that are independent on C_j. To do this, we go through the eigenvectors in order. For each eigenvector $\boldsymbol{\psi}_i$, we must determine whether or not its values on C_j can be expressed as a product of eigenvectors already present in Q. If it can be, then we record this product as part of the structure of $\mathcal{A}_j$. If not, we add i to Q.

The eigenvector $\boldsymbol{\psi}_i$ is a product of eigenvectors in Q on C_j if and only if there is a constant γ and $y_h \in \{0, 1\}$ for $h \in Q$ such that

$$\boldsymbol{\psi}_i(a) = \gamma \prod_{h \in Q} (\boldsymbol{\psi}_h(a))^{y_h},$$

for all vertices $a \in C_j$. This happens if and only if

$$\mathsf{sgn}(\boldsymbol{\psi}_i(a)) = \prod_{h \in Q} \mathsf{sgn}(\boldsymbol{\psi}_h(a))^{y_h}.$$

We can tell whether or not these equations have a solution using linear algebra modulo 2. Let $\boldsymbol{B}$ be the matrix over $\mathbf{Z}/2$ such that

$$\boldsymbol{\psi}_i(a) = (-1)^{\boldsymbol{B}(i,a)}.$$

Then, the above equations become

$$\boldsymbol{B}(i,a) = \sum_{h \in Q} y_h \boldsymbol{B}(h,a) \quad \text{for all } a \in C_j.$$

Thus, we can solve for the coefficients y_h in polynomial time, if they exist. If they do not, we add i to Q.

Once we have determined a base Q and how to express on C_j the values of every other eigenvector as a product of eigenvectors in Q, we have determine A_j.

It remains to explain how we partition the vertices into balanced classes. Consider applying the above procedure to a class C_j that is not balanced. We will discover that C_j is not balanced by finding a product of eigenvectors that is neither constant nor balanced on C_j. Every time we add an eigenvector $\boldsymbol{\psi}_i$ to Q, we will examine *every* product of vectors in Q to check if any are unbalanced on C_j. We can do this efficiently, because there are at most $2^{|Q|} \leq |C_j|$ such products to consider. As we have added $\boldsymbol{\psi}_i$ to Q, none of the products of vectors in Q can be constant on C_j. If we find a product that it not balanced on C_j, then it must also be non-constant, and thus provide a way of splitting class C_j into two.

We can now summarize the entire algorithm. We first compute the partition by absolute values of entries described in section 8.7. We then go through the classes of the partition one-by-one. For each, we use the above procedure until we have either split it in two or we have determined that it is balanced and we have computed its automorphism group. If we do split the class in two, we refine the partition and start over. As the total number of times we split classes is at most n, this algorithm runs in polynomial time.

After we have computed a partition into balanced classes and have computer their automorphisms groups, we combine them to find the automorphisms group of the entire graph as described at the end of section 8.6.

References

[Bab81] László Babai. Moderately exponential bound for graph isomorphism. In *Fundamentals of Computation Theory*, number 117 in Lecture Notes in Math, pages 34–50. Springer-Verlag, Berlin-Heidelberg-New York, 1981.

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697. ACM, 2016.

[BGM82] László Babai, D Yu Grigoryev, and David M Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 310–324. ACM, 1982.

[BL83] László Babai and Eugene M Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 171–183. ACM, 1983.

[LM82] F. Tom Leighton and Gary Miller. Certificates for graphs with distinct eigenvalues. Manuscript, 1982.

[ZKT85] V. M. Zemlyachenko, N. M. Kornienko, and R. I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29:1426–1481, 1985.