Spectral Graph Theory Lecture 22

Testing Isomorphism of Graphs with Distinct Eigenvalues

Daniel A. Spielman November 13, 2009

22.1 Introduction

I will present an algorithm of Leighton and Miller [**?**] 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 [**?**], 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. The fastest-known algorithm for it takes time time $2^{\sqrt{O(n \log n)}}$ (See [**?**, **?**, **?**]). However, it seems easy in almost all practical instances. Today's lecture and one next week will give you some idea as to why.

22.2 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 Π with entries given by

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

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

$$(\Pi \boldsymbol{v})(a) = \boldsymbol{v}(\pi(b)).$$

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 Π such that

$$\Pi A \Pi^T = B.$$

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

22.3 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 next week. 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

$$A = V \Lambda V^T,$$

where Λ is the diagonal matrix of eigenvalues of A and V is an orthnormal matrix holding its eigenvectors. If B has the same eigenvalues, we can write

$$B = U \Lambda U^T.$$

If Π is the matrix of an isomorphism from G to H, then

$$\Pi V \Lambda V^T \Pi^T = U \Lambda U^T.$$

As each entry of Λ is distinct, this implies

$$\Pi V = US,$$

where S is a diagonal matrix with ± 1 entries on its diagonal. Our algorithm for testing isomorphism will determine all such matrices S. That is, we will find diagonal ± 1 matrices S such that the set of rows of US is the same as the set of rows of V. Equivalentaly, we search for $s_1, \dots, s_n \in \{\pm 1\}^n$ for which $S = \mathrm{diag}(s_1, \dots, s_n)$ satisfies this condition. In this case, for each vertex $a \in G$ there will be a vertex $b \in H$ for which

$$\boldsymbol{v}_1(a), \dots, \boldsymbol{v}_n(a) = s_1 \boldsymbol{u}_1(b), \dots, s_n \boldsymbol{v}_n(b).$$

I will say that an eigenvector $\boldsymbol{v}_i$ is *helpful* if for all $a \neq b \in V$, $|\boldsymbol{v}_i(a)| \neq |\boldsymbol{v}_i(b)|$. In this case, it is very easy to test if G and H are isomorphic, because this helpful vector gives us a cannonical name for every vertex. If Π is an isomorphism from G to H, then $\Pi \boldsymbol{v}_i$ must be an eigenvector of B. We can then determine the sign s_i by considering the vertex of largest absolute value in $\boldsymbol{v}_i$ and $\boldsymbol{u}_i$. In particular, we must choose s_i so that these entries are the same. We could then find the isomorphism, if it exists, by mapping a to the vertex b for which $\boldsymbol{v}_i(a) = s_i \boldsymbol{u}_i(b)$.

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 $\boldsymbol{v}_i(a) \neq \boldsymbol{v}_i(b)$ for all $a \neq b$ and there is a cannonical way to choose a sign for the vector $\boldsymbol{v}_i$. For example, if the sum of the entries in $\boldsymbol{v}_i$ is not zero, we can choose its sign to make the sum positive. In fact, unless $\boldsymbol{v}_i$ and $-\boldsymbol{v}_i$ have exactly the same set of values, there is a cannonical choice of the sign for this vector.

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

22.4 All the Isomorphisms

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. If G and H are isomorphic, then the number of isomorphisms from G to H will be equal to the number of automorphisms of G. In particular, if Π is a permutation matrix such that $\Pi A \Pi^T = B$ and if Θ is a permutation matrix such that $\Theta A \Theta^T = A$, then

$$\Pi \Theta A \Theta^T \Pi^T = (\Pi\Theta) A (\Pi\Theta)^T = B.$$

If Π and Θ are an automorphisms of G, then we can write

$$\Pi = V S_1 V^T \quad \text{and} \quad \Theta = V S_2 V^T.$$

We then have that

$$\Pi\Theta = V S_1 V^T V S_2 V^T = V S_1 S_2 V^T$$

is also an automorphism of G. So, the group of automorphisms of G is isomorphic to a subgroup of the diagonal matrices with ± 1 entries under multiplication.

The form of our final solution will be as follows. We will determine a set of eigenvectors, $T \{1, \ldots, n\}$, for which the signs $(s_i)_{i \in T}$ may be set arbitrarily. For every $j \not\in T$, s_j will be set to a function of the signs of vectors in T. In particular, there will be a constant $\alpha_j \in \pm 1$ and a subset T_j such that

$$s_j = \alpha_j \prod_{i \in T_j} s_i.$$

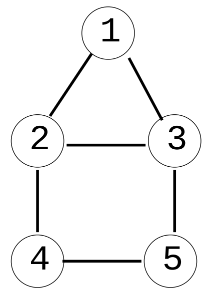
All settings of signs that obey these rules will result in isomorphisms. In particular, there will be $2^{|T|}$ isomorphisms.

22.5 Example

An example will be very helpful. Consider the house graph:

It has five distinct eigenvalues, with the following eigenvectors (to two digits of precsion, and not necessarily scaled to be unit vectors)

$$\boldsymbol{v}_1 = \begin{pmatrix} 0 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \quad \boldsymbol{v}_2 = \begin{pmatrix} -.74 \\ .43 \\ .43 \\ -.2 \\ -.2 \end{pmatrix} \quad \boldsymbol{v}_3 = \begin{pmatrix} 0 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} \quad \boldsymbol{v}_4 = \begin{pmatrix} -.52 \\ -.18 \\ -.18 \\ .58 \\ .58 \end{pmatrix} \quad \boldsymbol{v}_5 = \begin{pmatrix} .43 \\ .53 \\ .53 \\ .36 \\ .36 \end{pmatrix}.$$

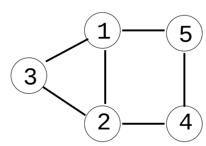


We see that none of these vectors is *helpful*, which must be the case as this graph has an automorphism. If we apply this automorphism to these vectors, we get the vector set

$$\boldsymbol{u}_1 = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \quad \boldsymbol{u}_2 = \begin{pmatrix} -.74 \\ .43 \\ .43 \\ -.2 \\ -.2 \end{pmatrix} \quad \boldsymbol{u}_3 = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \quad \boldsymbol{u}_4 = \begin{pmatrix} -.52 \\ -.18 \\ -.18 \\ .58 \\ .58 \end{pmatrix} \quad \boldsymbol{u}_5 = \begin{pmatrix} .43 \\ .53 \\ .53 \\ .36 \\ .36 \end{pmatrix}.$$

The signs of vectors $\boldsymbol{u}_3$ and $\boldsymbol{v}_3$ are different, as are the signs of vectors $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$.

If we were given the same graph, but with different vertex labels, these eigenvectors would help us establish the isomorphism. For example, the graph could be presented as



with eigenvectors

$$\boldsymbol{u}_1 = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \\ -1 \end{pmatrix} \quad \boldsymbol{u}_2 = \begin{pmatrix} .43 \\ .43 \\ -.74 \\ -.2 \\ -.2 \end{pmatrix} \quad \boldsymbol{u}_3 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \\ -1 \end{pmatrix} \quad \boldsymbol{u}_4 = \begin{pmatrix} -.18 \\ -.18 \\ -.52 \\ .58 \\ .58 \end{pmatrix} \quad \boldsymbol{u}_5 = \begin{pmatrix} .53 \\ .53 \\ .43 \\ .36 \\ .36 \end{pmatrix}.$$

As $\boldsymbol{u}_5(3)$ and $\boldsymbol{v}_5(1)$ are the only vertices that have value .43 in the fifth eigenvector, we could conclude that any isomorphism must map one of these to the other. Still using the fifth eigenvector, we could conclude that any isomorphism must map vertices $\{2, 3\}$ in G to vertices $\{1, 2\}$ in H, and vertices $\{4, 5\}$ in G to vertices $\{4, 5\}$ in H.

22.6 Equivalence Classes of Vertices

We say that a vertices $a, b \in V$ are *equivalent* if there exists an automorphism of G that maps a to b. Our algorithm will begin by determining that certain pairs of vertices are not equivalent. That is, it breaks the vertices into classes in such a way that we are guaranteed that vertices in different classes are not equivalent. It will use the same division in G as in H.

We may begin by dividing vertices according to the absolute values of their entries in eigenvectors. That is, if $|\boldsymbol{v}_i(a)| \neq |\boldsymbol{v}_i(b)|$, then we may place vertices a and b in different classes, as there can be no automorphism that maps a to b. We may label these classes by the absolute values achieved. For example, we may label a class by $\alpha_1, \dots, \alpha_n$ if it contains the vertices a for which

$$|\boldsymbol{v}_i(a)| = \alpha_i.$$

We may do the same in H, using the same set of labels. Of course, if we get different classes in H, or the classes in H have different sizes then they do in G, then we will have determined that G and H are non-isomorphic.

22.7 Refining Unbalanced Classes

It is possible to refine the vertex classes further. Let C be a class containing 3 vertices and imagine that $\boldsymbol{v}_i$ is an eigenvector that takes the values $(\beta, -\beta, -\beta)$ on C. Even though the absolute value of $\boldsymbol{v}_i$ on C is constant, it is clear that one of these vertices is different from the other two, as it is the only vertex of its sign. Generally, if an eigenvector $\boldsymbol{v}_i$ takes values $\pm\beta$ on a class C but the number of positive values is different from the number of negative values, then we may split this class in two pieces. No automorphism of the graph will map a vertex from one part to the other. We may label these classes consistently in G and H by recording the index of the eigenvector used to split, and by recording which new class had the minority of the values. If a vector $\boldsymbol{v}_i$ has the same number of positive as negative values on C, then we way that $\boldsymbol{v}_i$ is *balanced* with respect to C.

We can extend this idea further by using products of eigenvectors. For example, imagine that C is a class with 6 vertices, and that eigenvectors $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ have the following values on C:

$$\begin{pmatrix} \beta \\ \beta \\ \beta \\ -\beta \\ -\beta - \beta \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \gamma \\ -\gamma \\ -\gamma \\ -\gamma \\ \gamma \\ \gamma \end{pmatrix}.$$

While both vectors are balanced with respect to C, their product is not. The product of $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ takes one value twice and another value four times. Again, no automorphism of G will mix these two classes. So, we can split these classes.

Of course, we can carry out this procedure with products of more eigenvectors. We say that a class C is balanced with respect to the set T of eigenvectors if none of these eigenvectors is zero on any element of C, if $\prod_{i \in T} \boldsymbol{v}_i$ is uniform in absolute value on C, and takes the same number of positive as negative values on C.

22.8 Balancing with all products

We will split classes until they are balanced with respect to *every* product of eigenvectors. If this results in every vertex being in its own class, then that is wonderful: it means that G has no non-trivial automorphisms and that the classes determine the isomorphism from G to H, if it exists. If some class C is bigger, it will turn out that all vertices in C are in fact equivalent.

References