COMP116 – Work Sheet Seven

Associated Module Learning Outcomes

1. Basic understanding of the role of Linear algebra (including eigenvalues and eigenvectors) in computation problems such as web page ranking.

Matrices and Spectral Techniques

Q1: Basic Matrix Methods

This question concerns general properties that hold when combining two $n \times n$ matrices, $\mathbf{P}$ and $\mathbf{Q}$. For each of the following claims, state whether it is always **true** (do **not** give a proof) or is sometimes **false**. In the latter case give a counterexample (that is a matrix or matrices satisfying the precondition described but not satisfying the property asserted).

a. If $\mathbf{P}$ and $\mathbf{Q}$ are $n \times n$ matrices both of which are *non-singular* (recall Lecture notes and course textbook page 383) then:

   i. $\mathbf{P} + \mathbf{Q}$ is **always** non-singular.

   ii. $\mathbf{P} - \mathbf{Q}$ is **always** non-singular.

   iii. $\mathbf{P} \cdot \mathbf{Q}$ is **always** non-singular.

b. If $\mathbf{P}$ is an $n \times n$ matrix then:

   i. If some **row**, $\mathbf{r}_i$ is equal to (the transpose of) some **column** $\mathbf{c}_j$ then $\mathbf{P}$ is singular.

   ii. If $p_{ii} = 0$ for some $1 \leq i \leq n$ then $\mathbf{P}$ is **always** singular.

   iii. If $p_{ii} \neq 0$ for every $1 \leq i \leq n$ then $\mathbf{P}$ is **always** non-singular.

Q2: Computing eigenvalues and eigenvectors

[**Suggestion:** A good resource for processing large (up to 32×32) matrices online may be found at `http://www.bluebit.gr/matrix-calculator/`. This will allow matrix-vector (as used in the Power Method) computations and scalar product: $\mathbf{x} \cdot \mathbf{y}$ for n-vectors $\mathbf{x},\ \mathbf{y}$ is just the matrix product $\mathbf{x}\mathbf{y}^\top$, i.e a $1 \times n$-matrix ($\mathbf{x}$) multiplied by an $n \times 1$-matrix ($\mathbf{y}^\top$).]

With the exception of **ranking problems** (e.g. Google's ordering of pages by relevance to a given query) a number of applications are concerned not with finding **all** eigenvalues but specific cases. Very often significant information about a structure can be found using only the **largest** eigenvalue.

A classical such result of this form dates from 1967:[1] this states that for any graph $G(V, E)$ the number of colours needed so that the nodes can be assigned colours in such a way that no two nodes joined by an edge get the same colour, is at most $\lambda_1(G) + 1$ where $\lambda_1(G)$ is the largest eigenvalue of the (symmetric) $(0, 1)$-matrix of G. This "optimal colouring problem" (which, among other applications, is important in timetable construction and scheduling problems) is in general extremely difficult, so being able to estimate a good solution provides a useful approach.

One technique for computing an eigenvector and dominant eigenvalue is described in the course text (pages 402–405) where the following algorithm is used,

Algorithm 1 Building an approximation to a dominant eigenvector.

```
Input A $n \times n$ Real-valued matrix;
Output: (approximation) to a dominant eigenvector of $\mathbf{A}$
$k := 0$; { Counter for number of iterations}
$\mathbf{x}_0 := \mathbf{1}$; {Initial "guess"}
repeat
    $k := k + 1$;
    $\mathbf{x}_k \ := \ \mathbf{A} \cdot \mathbf{x}_{k-1}$;
until $k > MAX$ {Preset number of iterations to do}
return $\mathbf{x}_k$;
```

When Algorithm 1 is applied to the $(0, 1)$-matrix, $\mathbf{G}$ resulting from an **undirected** graph, G, it will always manage to find a dominant eigenvector, $\mathbf{x}$. This

[1] H. S. Wilf, The Eigenvalues of a Graph and Its Chromatic Number, Journal of the London Mathematical Society, s1-42(1):330–332, 1967

vector can then be used to approximate the dominant eigenvalue by computing

$$\frac{(\mathbf{Gx}) \cdot \mathbf{x}}{\mathbf{x} \cdot \mathbf{x}}$$

Here $\mathbf{v} \cdot \mathbf{w}$ is the **scalar** (or "dot") product of 2 n-vectors $\mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{n} v_i w_i$. (discussed in Part 2 of the course; see also course text, pages 69 and 405).

Consider the graph in Figure 1

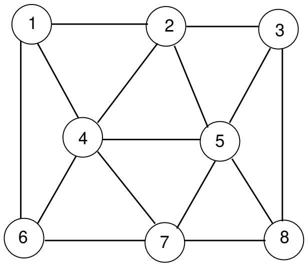

Figure 1: 8 node undirected graph

1. Recalling Wilf's result that the number of colours needed is at most $\lambda_1(G) + 1$ do you think $\lambda_1(G) < 3$?

2. Construct the 8×8 adjacency matrix for this graph and using the **power method** (using $MAX = 3$ will be sufficiently accurate) find an approximation to a dominant eigenvector.

3. For the dominant eigenvector identified in (2), find the associated eigenvalue. Does this confirm or refute your answer to (1)?

4. Let λ_8 be the **smallest** eigenvalue (where "smallest" in this context allows $\lambda < 0$: remember that all symmetric matrices have eigenvalues in $\mathbb{R}$ so the ordering complications from $\mathbb{C}$ do not occur). With respect to the node colouring problem, what information does the value $1 - (\lambda_1/\lambda_8)$ seem to provide?

Now consider the graph in Figure 2

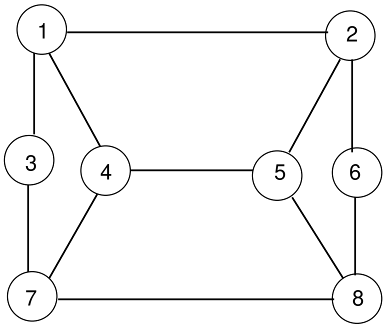

Figure 2: 8 node undirected graph

1. Find a dominant eigenvector for this graph using the **power method** from Algorithm 1 (again $MAX = 3$ is sufficient).
2. Find (an approximation to) the dominant eigenvalue for this eigenvector.
3. The on-line resource referred to provides an option to compute all of the eigenvalues. What property do you notice about the eigenvalues relating to the adjacency matrix of the graph in Figure 2? Does a similar property hold for the graph used in Figure 1?
4. Suppose G is a graph whose eigenvalues show similar behaviour to that of Figure 2 but that G has an **odd** number of nodes (this is possible). What specific value **must** be an eigenvalue in this case? What can be deduced about the adjacency matrix of such G?
5. Find (for both examples) the result of computing

$$\frac{1}{2}\sum_{i=1}^{n} \lambda_i^2$$

After making allowances for rounding errors in the arithmetic, what relationship do these values seem to have to the graphs themselves?