Homework 1

Greg Nelson, Jeremy Morris and Anup Bhawalkar

February 1, 2005

1 Hashing

We use a hashing procedure to store the non-zero entries in a large sparse matrix. The procedure is the following:

1. Create an array of size P (make sure P is prime)

2. Generate a unique key. If we want to store the entry a_{ij}, we'll use $iN + j + 1$ as the key.

3. Here there are two situations:

   (a) Store the value of a_{ij} in the location corresponding to the key we generated in the previous step.

   (b) If there is already something stored in this location (we call this a *collision*), calculate some increment value and increment until an empty location is full.

The size of the array P has to be prime for the reason of how collisions are dealt with. If we pick a number that is not prime, when we increment, we may only hit a few of the entries in the array. This could be disastrous since we might think that the array is full when we've only inspected a small number of entries. For example, take an array of size 60 and an increment of 36, if we start at position 1 and iterate through the array 6000 times we only hit locations: $1, 13, 25, 37, 49$.

2 Fill In

a. The graph of A:

b. The matrix after Gaussian Elimination looks like:

$$\begin{bmatrix}
x & x & & & x & x & & \\
& x & x & & 0 & x & x & \\
& & x & x & 0 & 0 & x & x \\
& & & x & 0 & 0 & 0 & x \\
& & & & x & x & 0 & 0 \\
& & & & & x & x & 0 \\
& & & & & & x & x \\
& & & & & & & x
\end{bmatrix}$$

Where x =original entries, 0 =Fill-in. Notice that we have a fill-in of 9.

c. Here we relabel the entries:

d. The resulting matrix looks like:

$$\begin{bmatrix}
x & x & x & & & & & \\
x & x & x & x & & & & \\
x & x & x & x & x & & & \\
& x & x & x & x & x & & \\
& & x & x & x & x & x & \\
& & & x & x & x & x & x \\
& & & & x & x & x & x \\
& & & & & x & x & x
\end{bmatrix}$$

3 Complete Graphs

A graph with n nodes is complete if every node is connected to every other node by an edge. We have an edge from node i to node j if

$$a_{ij} = a_{ji} \neq 0$$

Since the graph of the matrix is complete, we have

for $i = 1 : n$
  for $j = 1 : n$
    $a_{ij} = a_{ji} \neq 0$
  end
end

where n is the number of rows or columns in a matrix. Hence, a matrix whose graph is complete has no sparsity, all entries are non-zero.

4 Connected Graphs

The main property, that we are concerned with, of disconnected graphs is that the related adjacency matrix will be block diagonal. This means that when solving the related system of equations, the problem can be divided into two parts. We know that the solution of an $n \times n$ system is $O(\frac{n^3}{6})$, so solving a block diagonal system would be $O(\frac{n^3}{24})$. This is a significant reduction.

5 Lexicographical Ordering

For the lexicographical ordering of an $m \times n$ array, there is a fill-in of $(m-1)(n-1)^2$. To derive this formula, we simply count the fill-in. One way to count fill-in is by setting up the graph, removing nodes in the order in which they are labeled, and counting the number of new edges that are introduced during the process.

The graph has m rows and n columns. Removing the first n nodes (the first row) introduces $(n-1)^2$ new edges. Removing the next n nodes (the second row) introduces $(n-1)^2$ new edges. This pattern continues until you remove the final n nodes (the last row), which introduces zero new edges. Hence, we have a total of $(m-1) * (n-1)^2$ new edges introduced via this process. This corresponds to the total fill-in incurred during the process of Gaussian elimination.

6 Permutation Matrices

Permutation matrices are obtained by permuting the rows and/or columns of an identity matrix. Consider the matrix:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

By interchanging row 2 and 3 we obtain:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

which is a permutation matrix.

Now, the identity matrix is an orthogonal matrix. The rows and columns of an orthogonal matrix form an orthonormal basis of $\mathbb{R}^n$, i.e. row and column vectros are the basis vectors for $\mathbb{R}^n$ euclidean space. Hence rows of identity matrix form a euclidean space as well as columns form a euclidean space. Interchange of rows or columns from identity matrix will not affect the orthonormality of basis vectors. Hence the columns and rows of permutation matrix will form an orthonormal basis then we can say that the perumtation matrix P is orthogonal and $P^{-1} = P^T$.

To show that right multiplying with a permutation matrix exchanges columns, we expand the multiplication into :

$$(AP)_{ij} = \sum_{k=1}^{n} a_{ik} p_{kj}$$

We know that in the j^{th} column of P there is some index value l where $p_{lj} = 1$ and all other entries of the j^{th} column are 0. This means that our sum becomes :

$$(AP)_{ij} = a_{il} \cdot 1 = A_{il}$$

This shows that the j^{th} column is now in the l^{th} column. By symmetry we know that $P_{jl} = 1$ and all other entries in the l^{th} column are zero and we see that right multiplying with a permutation matrix will exchange columns.

A similar argument works to show that left multiplying with a permutation matrix will exchange rows.

7 What we are doing

If we look at the graph of a matrix while doing Gaussian elimination, we notice that everytime we do an elimination, we remove a node from the graph and connect all its neighbors. Any new edges that are created in this process contribute to fill-in. We can see an example of this when we have the matrix:

$$\begin{bmatrix} x & x & x & x \\ x & x & x & \\ x & x & x & \\ x & & & x \end{bmatrix}$$

Which corresponds to the graph :

Then we see that when we do Gaussian elimination on the first column we get the matrix :

$$\begin{bmatrix} x & x & x & x \\ & x & x & 0 \\ & x & x & 0 \\ & 0 & 0 & x \end{bmatrix}$$

Where 0 represents fill-in. Which corresponds to the graph :

Notice that we have connected all neighbors of node 1 where there was not already an edge.

8 Stability

$$e(h) = \lim_{h \to 0} \frac{1 - \gamma^{1/h}}{1 - \gamma} h^{p+1} \tag{1}$$

$$e(h) = \lim_{h \to 0} \frac{1}{1 - \gamma} h^{p+1} - \lim_{h \to 0} \frac{\gamma^{1/h}}{1 - \gamma} h^{p+1} \tag{2}$$

Now we consider one particular case where $\gamma = 1 + h$ and $h = 1/n$. If we plug into (2), we get :

$$\begin{aligned} e(h) &= \lim_{h\to 0} \frac{h^{p+1}}{-h} h^{p+1} - \lim_{h\to 0} \frac{(1+h)^{1/h}}{-h} h^{p+1} && (3) \\ &= 0 + \lim_{h\to 0} (1+h)^{1/h} h^p && (4) \\ &= \lim_{h\to 0} (1+h)^{1/h} \lim_{h\to 0} h^p && (5) \\ &= e \cdot 0 && (6) \\ e(h) &= 0 && (7) \end{aligned}$$

This is an intersting result. The value of $\gamma > 1$, so the error should grow exponentially, but instead, the error goes to 0 in the limit as $h \to 0$. Also, the order of error is h^p instead of h^{p+1}.