Stephen Woodbury 4/30/17 CMPS 102
I have read and agree to the collaboration policy. Stephen Woodbury.
Collaborators: none

Assignment 2_1 : Divide and Conquer

Part 1 : Proving Recursive Form of Walsh Hadamard Matrix

Claim: $H_n = (1 / 2^{(1/2)}) * \{[H_{(n-1)}\ H_{(n-1)}], [H_{(n-1)}\ -H_{(n-1)}]\}$

Pf(By Induction): *Base Cases:*

I) $H_1 = (1 / 2^{(1/2)}) * \{[H_0\ H_0], [H_0\ -H_0]\} = (1 / 2^{(1/2)}) * \{[1\ 1], [1\ -1]\}$.
This result abides by all requirements of a Walsh Hadamard matrix
1) Size: H_1 size should be $2^n * 2^n$, 4 elements, which is true.
2) Constant Match Up: The constant in front of a WH matrix should be $1/2^{(n/2)}$. We see this is the case as when n=1, our constant is $1/2^{(1/2)}$.
3) Element Match Up: Our elements of H_1 should abide by the formula : $H_n[i,j] = [1/2^{(n/2)}]*[(-1)^{(ioj)}]$. This is is seen to apply as we can see : $H_1[0,0]=1$, $H_1[0,1]=1$, $H_1[1,0]=1$, $H_1[1,1]=-1$.

II) $H_2 = (1/2^{(1/2)}) * \{[H_1\ H_1], [H_1\ -H_1]\}$
$= (1/2^{(1)}) * \{[H_0\ H_0\ H_0\ H_0], [H_0\ -H_0\ H_0\ -H_0], [H_0\ H_0\ -H_0\ -H_0], [H_0\ -H_0\ -H_0\ H_0]\}$.
$= (1/2^{(1)}) * \{[1\ 1\ 1\ 1], [1\ -1\ 1\ -1], [1\ 1\ -1\ -1], [1\ -1\ -1\ 1]\}$.
Note: Constant changed as each H1 has a $(1/2^{(1/2)})$ multiplied to its respective matrix. By matrice nature, we can pull that constant out to multiply it with the already standing constant.
This result abides by all requirements of a Walsh Hadamard matrix
1) Size: H_2 Size should be $2^2 * 2^2$, 16 elements, which is true.
2) Constant Match Up: The constant in front of a WH matrix should be $1/2^{(n/2)}$. We see this is the case as when n=2, our constant is $1/2^{(2/2)}$.
3) Element Match Up: Our elements of H_2 should abide by the formula : $H_n[i,j] = [1/2^{(n/2)}]*[(-1)^{(ioj)}]$. This is is seen to apply as we can see : $H_2[0,0]=1$, $H_2[0,1]=1$, $H_2[0,2]=1$, $H_2[0,3]=1$, $H_2[1,0]=1$, $H_2[1,1]=-1$, $H_2[1,2]=1$, $H_2[1,3]=-1$, $H_2[2,0]=1$, $H_2[2,1]=1$, $H_2[2,2]=-1$, $H_2[2,3]=-1$, $H_2[3,0]=1$, $H_2[3,1]=-1$, $H_2[3,2]=-1$, $H_2[3,3]=1$.

Induction Step: Assume $H_k=1/2^{(1/2)})*\{[H_{(k-1)}\ H_{(k-1)}],[H_{(k-1)}\ -H_{(k-1)}]\}$ for k>=1. This is our Induction Hypothesis.
Induction Conclusion : $H_{(k+1)} = (1 / 2^{(1/2)}) * \{[H_k\ H_k], [H_k\ -H_k]\}$ for k>=1.
Analyze $H_{(k+1)}$. According to our formula:
$H_{(k+1)} = (1 / 2^{(1/2)}) * \{[H_k\ H_k], [H_k\ -H_k]\}$, each H_k is a WH Matrix
Analyze the conditions needed to make a $H_{(k+1)}$ a WH Matrix.
1) Size: $H_{(k+1)}$ Size should be $2^{k+1}*2^{k+1}$, We know that by our IH that each H_k has 2^k*2^k elements. Total elements in $H_{(k+1)}$ is $4*(2^k*2^k)=2^2(2^k*2^k)=(2*2^k*2*2^k)=2^{k+1}*2^{k+1}$ elements as wanted.
2) Constant Match Up: Constant in front of $H_{(k+1)}$ should be $1/2^{((k+1)/2)}$. Constant $(1/2^{(k/2)})$ in front of each H_k by IH. By Matrix Nature, Can pull constant out from every element of $H_{(k+1)}$ and multiply to $H_{(k+1)}$'s constant from the formula. $(1/2^{(1/2)})*(1/2^{(k/2)})=(1/2^{((k+1)/2)})$ as wanted.
3) Element Match Up: Every element in $H_{(k+1)}$ should abide by the WH formula: $H_{(k+1)}[i,j] = [1/2^{((k+1)/2)}]*[(-1)^{(ioj)}]$. We know that every $H_{(k)}$ is a proper WH matrix by our IH. Also by our IH, we find that $(-1)^{[ioj]}= (1)^{[io(j+n)]}=(-1)^{[(i+n)oj]}$ for the indices of the sub-matrix in the first quadrant of a larger WH matrix which ensures that the sub-matrices in Quadrant I,II,& III of $H_{(k+1)}$ are equivalent, which fits our formula. We also find that from the same indices from the first quadrant sub-matrix, that $(-1)^{[ioj]} = -(-1)^{[(i+k)o(j+k)]}$ Which explains the last quadrant of matrix $H_{(k+1)}$. The constant is accurate for the formula as shown on the Constant match up above.
So in conclusion, $H_{(k+1)}$ is a WH Matrix, completing our proof. QED

Part 2: Euclidean Norm for Every Column and Row:

The Euclidean Norm Formula is : $\|V\|=\sqrt{\sum_{k=1}^{n'}|v_k|^2}$

Vk is an element of vector V of size n'. We know that the formula for any element in a WH Matrix is : $(1/\sqrt{2^n})*(-1)^{(ioj)}$.

Treat any row or column in our WH matrix as a vector and find the Euclidean Norm: $\|V\|=\sqrt{\sum_{k=1}^{n'}\left|(1/\sqrt{2^n})*(-1)^{(ioj)}\right|^2}=\sqrt{\sum_{k=1}^{n'}(1/2^n)}=(\sqrt{n'}/\sqrt{2^n})$

Lastly, the size of a WH matrix is $2^n * 2^n$. Ie, this is the number of rows times the number of columns. This means that each column and row is of length 2^n. So, that means $n' = 2^n$.

$\|V\|=(\sqrt{n'}/\sqrt{2^n})=(\sqrt{2^n}/\sqrt{2^n})=1$ As wanted.

Part 3: Orthonormal Columns:

Claim: Columns of H_n form an orthonormal basis.

Pf: Part 1: One of the conditions of orthonormal is that the Euclidean Norm of all columns in our WH matrix is 1. This was proven in part 2.

Part 2: The other condition of orthonormal matrices is that the dot product of all columns is 0. We'll prove this through induction.

Base Case: $H_1 = (1 / 2^{(1/2)})*\{[\ H_0\ H_0\ ],[\ H_0\ -H_0]\}=(1 / 2^{(1/2)})*\{[\ 1\ 1\ ],[\ 1\ -1]\}$.

Dot product all variations of the columns: (1*1)+(1*-1)=0, (1*-1)+(1*1)=0.

Inductive Step: Assume H_k columns form orthonormal basis: Ind. Hypothesis.

[NTP: $H_{(k+1)}$ columns form orthonormal basis : ind. Conclusion.]

By part 1 above, we know $H_{k+1} = (1 / 2^{(1/2)})* \{[\ H_k\ H_k\ ],[\ H_k\ -H_k]\}$.

By our induction hypothesis, we know that any column dotted with any other column that wouldn't have been a clone will yield a dot product of 0. if it was H_k rather than $-H_k$, clone columns would occur between column x where $x<2^{(k+1)}/2$ and column x' where $x' = x+2^{(k+1)}/2$. We know that any column dotted with itself will always equal the magnitude of the vector. $-H_k$ ensures that no column is ever dotted with a clone of itself. It actually ensures that the first $2^{(k+1)}/2$ elements of our columns when dotted with their would be clones creates a dot product value that is directly negated by the next $2^{(k+1)}/2$ elements of our would be clone column's dot product. QED

Part 4: Dot Product between Matrix and Vector Algorithm:

Components to our Algorithm: 2^nx2^n WH Matrix M, 2^n Vector to be multiplied = V, 2^n Solution Vector = S. Matrix will be composed of double vectors, RxC.

Algorithm: 1) For (int i=0, i<2^n; i++)
2) For(int j=0; j<2^n/2; j++)
3) $S_j=S_j+(V_i*M_{ji})$;
4) if(i>=2^n/2)
5) $S_{(j+(2\text{^}n)/2)}=S_{(j+(2\text{^}n)/2)} + (-1)*S_j$;
6) else
7) $S_{(j+(2\text{^}n)/2)}=S_{(j+(2\text{^}n)/2)} + S_j$;

Time Complexity: We assume n' = # columns

1 O(n')
2 O(n'/2)
3 O(1)
4 O(1)
5 O(1)
6 no cost
7 O(1)

Total: O(n') * (O(n'/2) + 4*O(2)) = ***O(n'^2)***

I couldn't get run time of O(nlog(n)).