

EigenValuesAndEigenVectors

In Hama, we use [Jacobi](#) to calculate all eigenvalues and eigenvectors of a real symmetric matrix.

Description

Let A be a symmetric matrix, and $G = G(i,j)$ be a Givens rotation matrix. Then:

$$A' = G^T A G,$$

is symmetric and similar to A .

As they are similar, A and A' have the same Frobenius norm $\|A\|_F$ (the sum of squares of all components), however we can choose θ such that $A'(ij) = 0$, in which case A' has a larger sum of squares on the diagonal:

$$A'(ij) = \cos(2\theta) * A(ij) + (1/2) * \sin(2\theta) * (A(ii) - A(jj))$$

Set this equal to 0, and rearrange:

$$\tan(2\theta) = (2 * A(ij)) / (A(ii) - A(jj))$$

In order to optimize this effect, $A(ij)$ should be the largest off-diagonal component, called the pivot.

The Jacobi eigenvalue method repeatedly finds the pivot and performs rotations until the matrix becomes almost diagonal. Then the elements in the diagonal are approximations of the (real) eigenvalues of A .

Methods

The implementation of Jacobi eigenvalue method in Hama are described as below:

1. Find the pivot: Get the largest off-diagonal element of the matrix.

We use a M/R job to find the pivot p and its index in the matrix(pivot_row, pivot_col). These are all done in PivotMapper and PivotReducer.

- 1.#2 Compute the rotation parameters.

1. Rotate the matrix using the rotation matrix.

As the rotation matrix is a 2X2 matrix, each rotation just effects 4 elements of the matrix. The rotations can not effect each other, so we can do the rotations parallely.

The Pseudocode described as below:

1. for $i := 1$ to $k1$ do rotate(i, k, i, l) endfor
2. for $i := k+1$ to $l1$ do rotate(k, i, i, l) endfor
3. for $i := l+1$ to n do rotate(k, i, l, i) endfor

Rotate(k, l, i, j) ☹️ S is the matrix, c & s is the parameters of the rotation matrix computed in step 2)

$$S(kl) = c * S(kl) - s * S(ij)$$

$$S(ij) = s * S(kl) + c * S(ij)$$

The Implementation of M/R job to rotate the matrix is a tricky. We do not rotate the matrix actually in Mapper & Reducer. We do this rotation during we scan over the matrix table. Plz check the code in RotationInputFormat.

- 1.#4 Rotate the eigenvectors matrix.

1. Repeats until the matrix becomes almost diagonal, or the loops reach the limit specified by the user.