AMTL MATRIX LIBRARY

AMTL Matrix library provides an AMTL_Matrix object to work with several available BLAS libraries such as EJML, UJMP, JAMA and JBLAS. As long as available BLAS library is specified, matrix operations can be used by creating one AMTL_Matrix object. We can also compare the performance of different BLAS libraries by just changing the value of one variable instead of implementing the whole operation by using different libraries.

BlasID	Blas Library
0	EJML
1	UJMP
2	JAMA
3	JBLAS

1. AMTL_Matrix Class

AMTL_Matrix class implements seriablizable to be able to use it in applications where we need to communicate with different machines. Attributes and methods of this class are given below.

Attributes	Explanation
public int NumRows	Number of rows of the matrix
public int NumColumns	Number of columns of the matrix
public int BlasID	An integer number specifies the BLAS library
public Object M	Object which will be type cast as the object of the specified BLAS library

Methods	Explanation
public AMTL_Matrix(int NumRows, int NumColumns, int BlasID)	Constructor to create a matrix or a vector of zeros for a given Blas library type
public AMTL_Matrix(double[][] Input, int BlasID)	Constructor to create a matrix or a vector from an array
public AMTL_Matrix(AMTL_Matrix Input)	Constructor to create a matrix or a vector same as an AMTL_Matrix type of object.
public AMTL_Matrix(Object Input, int BlasID)	Constructor to create a matrix or a vector from a specific Blas library object.
public int getNumRows()	Method to return the number of rows
public int getNumColumns()	Method to return the number of columns
public double getDouble(int row, int column)	Method to return the value of the element at the specified index
public double setDouble(int row, int column, double val)	Method to assign a value to the element at the specified index

2. MatrixOps Class

This class contains matrix operations by using an AMTL_Matrix type object. Methods of this class are given below.

Methods	**Explanation**
public static void ADD(AMTL_Matrix obj1, AMTL_Matrix obj2, AMTL_Matrix obj_result)	Method for matrix or vector addition. Matrix contained in obj_result is the sum of the matrices contained in obj1 and obj2. Blas type of all matrices should be same.
public static void ReverseSign(AMTL_Matrix obj)	Method to Reverse signs of each element in the matrix contained by obj.
public static void Scale(AMTL_Matrix, double val)	Method to scale the each element of the matrix contained by obj.
public static void MULT(AMTL_Matrix obj1, AMTL_Matrix obj2, AMTL_Matrix obj_result)	Method for matrix multiplication. Matrix contained in obj_result is the multiplication of the matrices contained in obj1 and obj2. Blas type of all matrices should be same.
public static void Transpose(AMTL_Matrix obj)	Method to take the transpose of the matrix contained in obj. Matrix in obj is changed with its transpose.
public static int getRank(AMTL_Matrix obj)	Method to return the rank of the matrix contained in obj.
public static void SVD(AMTL_Matrix obj,AMTL_Matrix obj_U,AMTL_Matrix obj_V,AMTL_Matrix obj_S)	Method to calculate SVD of the matrix in obj and saves U,V, and S matrices in obj_U, obj_V and obj_S.

3. Norms Class

This class contains some of the matrix and vector norms, and singular value thresholding which is used in proximal operator of trace norm.

Methods	**Explanation**
public static void L2_Norm(AMTL_Matrix obj)	Method to return ℓ_2 norm of the vector contained in obj.
public static void L1_Norm(AMTL_Matrix obj)	Method to return ℓ_1 norm of the vector contained in obj.
public static void Frobenius_Norm(AMTL_Matrix obj)	Method to calculate Frobenius norm of the matrix contained in obj.
public static void Trace_Norm(AMTL_Matrix obj)	Method to calculate trace norm of the matrix contained in obj.
public static void SingularValueThresholding(AMTL_Matrix obj, double threshold)	Method to apply singular value thresholding to the matrix contained in obj. SVD of the matrix is calculated and the matrix reconstructed as $\mathbf{U}\left(\Sigma - threshold\mathbf{I}\right)_{+}\mathbf{V}^{T}$