Interfaces of multidimensional dense matrix

Constructors
CDenseMDMatrix
Functions to work with dimensions
SetDimensions GetDimensions GetClasses GetDimensionsNumber
Functions to get data
GetValue GetVectorValue
Functions to set data
SetValue SetVectorValue
Overloaded operators
operator+ operator- operator* operator=
Other functions
Normalize IsNormalized ClearData Clear

Constructors

CDenseMDMatrix *(void)*

Basic constructor. Creates an empty matrix.

CDenseMDMatrix *(const CDenseMDMatrix &_source)*

Copy constructor. Creates matrix with the same dimensions as in *_source* and copies all data from there.

Functions to work with dimensions

bool SetDimensions *(unsigned _nType, unsigned _nClasses)*

Sets one-dimensional distribution of type *_nType* and numbers of classes *_nClasses*. *_nType* is one of predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All old data will be erased. Matrix will be initialized with zero values. Returns *false* on error.

bool SetDimensions *(unsigned _nType1, unsigned _nClasses1, unsigned _nType2, unsigned _nClasses2)*

Sets two-dimensional distribution of types *_nType1* and *_nType2* and numbers of classes *_nClasses1* and *_nClasses2*. *_nType1* and *_nType2* are from predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All types must be unique. All old data will be erased. Matrix will be initialized with zero values. Returns *false* on error.

bool SetDimensions *(unsigned _nType1, unsigned _nClasses1, unsigned _nType2, unsigned _nClasses2, unsigned _nType3, unsigned _nClasses3)*

Sets three-dimensional distribution of types *_nType1*, *nType2* and *_nType3* and numbers of classes *_nClasses1*, *_nClasses2* and *_nClasses3*. *_nType1*, *nType2* and *_nType3* are from predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All types must be unique. All old data will be erased. Matrix will be initialized with zero values. Returns *false* on error.

bool SetDimensions *(const std::vector<unsigned> &_vTypes, const std::vector<unsigned> &_vClasses)*

Sets types *_vTypes* of dimensions and numbers of classes *_vClasses*. *_vTypes* is the vector of predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All types must be unique. All old data will be erased. Matrix will be initialized with zero values. Returns *false* on error.

std::vector<unsigned> GetDimensions *()*

Returns vector with all currently defined dimensions types.

std::vector<unsigned> GetClasses *()*

Returns vector with current numbers of classes.

unsigned GetDimensionsNumber *()*

Returns current number of dimensions.

Functions to get data

double GetValue *(unsigned _nDim, unsigned _nCoord)*

Returns value with specified coordinate *_nCoord* of specified dimension *_nDim*. It is possible to use this function if matrix has more than one dimension: if number of dimensions does not conform to the matrix, the sum of values by remaining dimensions will be returned. Returns *-1* on error.

double GetValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nDim2, unsigned _nCoord2)*

Returns value according to specified coordinates and dimensions. It is possible to use this function if matrix has more than two dimensions: if number of dimensions does not conform to the matrix, the sum of values by remaining dimensions will be returned. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *-1* on error.

double GetValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nDim2, unsigned _nCoord2, unsigned _nDim3, unsigned _nCoord3)*

Returns value according to specified coordinates and dimensions. It is possible to use this function if matrix has more than three dimensions: if number of dimensions does not conform to the matrix, the sum of values by remaining dimensions will be returned. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *-1* on error.

double GetValue *(const std::vector<unsigned>& _vDims, const std::vector<unsigned>& _vCoords)*

Returns value according to specified coordinates and dimensions. It is possible to use this function if matrix has more dimensions than was defined in *_vDims*: if number of dimensions does not conform to the matrix, the sum of values by remaining dimensions will be returned. Sequence of dimensions may not match the sequence, which was defined in the matrix. Number of dimensions *_vDims* and coordinates *_vCoords* must be the same. Returns *-1* on error.

double GetValue *(const std::vector<unsigned>& _vCoords)*

Returns value by specified coordinates according to the full defined set of dimensions. Returns *-1* on error.

bool GetVectorValue *(unsigned _nDim, std::vector<double>& _vResult)*

Returns vector *_vResult* according to specified dimension. If number of dimensions in the matrix is more than one, then the sum of values by remaining dimensions will be returned. Returns *false* on error.

bool GetVectorValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nDim2, std::vector<double>& _vResult)*

Returns vector of values *_vResult* according to specified dimensions and coordinate. If number of dimensions in the matrix is more than two, then the sum of values by remaining dimensions will be returned. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool GetVectorValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nDim2, unsigned _nCoord2, unsigned _nDim3, std::vector<double>& _vResult)*

Returns vector of values *_vResult* according to specified dimensions and coordinates. If number of dimensions in the matrix is more than three, then the sum of values by remaining dimensions will be returned. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool GetVectorValue *(const std::vector<unsigned>& _vDims, const std::vector<unsigned>& _vCoords, std::vector<double>& _vResult)*

Returns vector of values *_vResult* according to specified dimensions and coordinates. If number of dimensions in the matrix is more than it was specified in *_vDims*, then the sum of values by remaining dimensions will be returned. Sequence of dimensions may not match the sequence, which was defined in the matrix. Number of coordinates *_vCoords* must be one less than the number of dimensions *_vDims*. Returns *false* on error.

bool GetVectorValue *(const std::vector<unsigned>& _vCoords, std::vector<double>& _vResult)*

Returns vector of values *_vResult* by specified coordinates according to the full defined set of dimensions. Returns *false* on error.

Functions to set data

bool SetValue *(unsigned _nCoord, double _dValue)*

Sets value *_dValue* with coordinate *_nCoord* into one-dimensional matrix. Sets the value only if the matrix has one dimension. Returns *false* on error.

bool SetValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nCoord2, double _dValue)*

Sets value *_dValue* according to specified coordinates and dimensions into two-dimensional matrix. Sets the value only if the matrix has two dimensions. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool SetValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nDim2, unsigned _nCoord2, unsigned _nCoord3, double _dValue)*

Sets value *_dValue* according to specified coordinates and dimensions into three-dimensional matrix. Sets the value only if the matrix has three dimensions. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool SetValue *(const std::vector<unsigned>& _vDims, const std::vector<unsigned>& _vCoords, double _dValue)*
Sets value *_dValue* according to specified coordinates and dimensions. Sets the value only if the number of dimensions is the same as in the matrix. Number of dimensions *_vDims* and coordinates *_vCoords* must be the same. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool SetValue *(const std::vector<unsigned>& _vCoords, double _dValue)*
Sets value *_dValue* according to specified coordinates for full set of dimensions. Sets the value only if the number of coordinates is the same as the number of dimensions in the matrix. Number of coordinates *_vCoords* must be equal to a number of dimensions in matrix. Returns *false* on error.

bool SetVectorValue *(const std::vector<double>& _vValue)*
Sets vector of values *_vValue* in one-dimensional matrix. Sets the values only if the matrix has one dimension. Returns *false* on error.

bool SetVectorValue *(unsigned _nDim, unsigned _nCoord, const std::vector<double>& _vValue)*
Sets vector of values *_vValue* according to specified dimension and coordinate in two-dimensional matrix. Sets the values only if the matrix has two dimensions. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool SetVectorValue *(unsigned _nDim1, unsigned _nCoord1, unsigned _nDim2, unsigned _nCoord2, const std::vector<double>& _vValue)*
Sets vector of values *_vValue* according to specified dimensions and coordinates in three-dimensional matrix. Sets the values only if the matrix has three dimensions. Sequence of dimensions may not match the sequence, which was defined in the matrix. Returns *false* on error.

bool SetVectorValue *(const std::vector<unsigned>& _vDims, const std::vector<unsigned>& _vCoords, const std::vector<double>& _vValue)*
Sets vector of values *_vValue* according to specified dimensions and coordinates. Sets values only if the number of dimensions in *_vDims* is one less than in the matrix. Number of dimensions *_vDims* and coordinates *_vCoords* must be equal. Returns *false* on error.

bool SetVectorValue *(const std::vector<unsigned>& _vCoords, const std::vector<double>& _vValue)*
Sets vector of values *_vValue* according to specified coordinates for full set of dimensions, which were defined in the matrix. Sets values only if the number of coordinates is one less than number of dimensions in the matrix. Returns *false* on error.

Overloaded operators

CDenseMDMatrix operator+ *(const CDenseMDMatrix& _matrix)*
Performs addition of the matrix with the same dimensions. If dimensions are not the same, then an empty matrix with 0 dimensions will be returned.

CDenseMDMatrix operator- *(const CDenseMDMatrix& _matrix)*
Subtracts matrix with the same dimensions. If dimensions are not the same, then an empty matrix with 0 defined dimensions will be returned.

CDenseMDMatrix operator* *(double _dFactor)*
Performs multiplication of the matrix by a coefficient *_dFactor*.

CDenseMDMatrix& operator= *(const CDenseMDMatrix& _matrix)*
Sets dimenions and data from the *_matrix* to a left matrix.

Other functions

void Normalize *()*
Normalizes the matrix so that the sum of all elements equals to 1.

bool IsNormalized *()*
Returns *true* if the matrix is normalized.

void ClearData *()*
Sets all data in matrix equal to 0.

void Clear *()*
Removes all data and information about dimensions from the matrix.