NAME

cml_drl - Dimensionality Reduction

VERSION

Version 0.2.466

SYNOPSIS

cml_drl input_matrix output_file [**-a** *train_test_split rbm_iters bp_iters rbm_size bp_size*] [**-b** *random_seed*] [**-c** *converge_num*] [**-d** *embedding_dimensionality*] [**-e** *eigentype*] [**-f** *map_file*] [**-g**] [**-h**] [**-i** *max_iters*] [**-k** *num_neighbors*] [**-l** *landmark_type landmark_value*] [**-m** *reduction_method*] [**-n** *notice_level*] [**-o** *low_dim_file stats_file*] [**-p** *epsilon_distance*] [**-r**] [**-s** *stat_calculation*] [**-t** *test_file*] [**-u**] [**-v**] [**-w** *pymol_output*] [**-x** *dimensions*] [**-y** *neighbor_string*] [**-z**]

DESCRIPTION

Perform dimensionality reduction on a set of coordinates using principal component analysis (PCA), multi-dimensional scaling (MDS) locally linear embedding (LLE), IsoMap, or an AutoEncoder (AE). The input file consis is a matrix (see matrix file formats below) with each row cooresponding to a data point. For MDS reduction, the input file is a symmetric distance matrix. An internal test matrix representing 400 samples from the 'swiss roll' can be used by specifying INT_SWISSROLL as the input file name. 5000 samples of the swissroll can be specified with INT_SWISSROLL5. Other internal test matrices are: INT_SWISSROLLN for textbook version of swiss roll, INT_THINSWISSROLL for thin swiss roll, INT_JAPAROLL for Japaness flag roll, INT_NONDEVROLL for Nondevelopable swiss roll, INT_TKNOT for trefoil knot, INT_SPHERE for a sphere, INT_TORUS for torus, and INT_BOX for the open box. (Add 5 at the end of string for 5000 samples). The output file uses an internal format that stores the map from the high-**d**imensional to the low-**d**imensional space and vice-**v**ersa. Analysis can be performed without writing the map to disk by specifying NO_OUTPUT as the output file name. Once a map has been obtained, coordinates can be mapped back and forth using the **-f** flag. In this case, the input_file contains the coordinates in one dimension and the *output_file* contains the mapped coordinates.

PARAMETERS

-a *train_test_split rbm_iters bp_iters rbm_size bp_size*

Set parameters for the autoencoder. *train_test_split* is the fraction of input data to be used for training (default is 0.8). *rbm_iters* (default=10) and *bp_iters* (default=30) are the number of iterations to be used for restricted boltzmann and back propagation steps. *rbm_size* (default=10) and *bp_size* (default=100) is the desired batch size for RBM and BP steps. *rbm_size* and *bp_size* should evenly divide the training set size.

-b *random_seed*

Set the random number seed (ya_sizet integer value).

-c *converge_num*

Specify convergence criteria for power method. This is given in terms of the cosine of the angle between eigen vectors in successive iterations. The default is 0 which results in the default epsilon depending on the matrix and method.

-d *embedding_dimensionality*

Specify the number of dimension to project down to. The default is 2. For the AutoEncoder, all layers are specified in a string (**-d** "3 8 4 2"

-e *eigentype*

Specify the type of eigen decomposition. Options are DQ, POWER, RRR, and SIMPLE. The default is RRR

-f *map_fi le*

Map from the high dimensional to the low dimensional space or vice versa using a previously calculated reduction as specified in *map_fi le*. The *input_matrix* can contain multiple points, but the number of columns must match either the high or low dimensionality. For LLE and ISOMAP, the forward and reverse maps are calculated using an inverse distance weighted average of coordinates from neighbors of the high and low dimensionalities used in the original reduction. The number of neighbors used for the mapping is specified with the **-k** option. Reconstruction errors for the mapping can be calculated using the **-r** option.

-g If linked to the Intel math library, this will use the faster, low accuracy math functions.

-h Print out the man page for help

-i *max_iters*

Specify the maximum number of iterations for iterative methods on each eigen vector. A warning is generated if any eigen vector does not converge within the specified number of iterations. The default is 10000.

-k *num_neighbors*

Specify the number of neighbors to be used for methods which require k-**n**earest neighbors. The default is 10. If k<1, then we use it as epsilon distance for neighbors.

-l *landmark_type landmark_value*

Use landmark points for IsoMap reduction. If *landmark_type* is FILE, *landmark_value* specifies a file to load a matrix from (see file formats below). If *landmark_type* is STRING, a string specification of landmarks is used (see **-x** for examples). If *landmark_type* is RANDOM, a landmark value is a fraction>0 and <1 that specifies the random fraction of landmarks to be used. Landmarks are specified with 0 as the first point.

-m *reduction_method*

Specify the dimensionality reduction method to be used. Options are PCA, MDS, LLE, ISOMAP, and AE. The default is PCA.

-n *notice_level*

Set the degree of program output. Use:

-n 0 No output
-n 10 Normal program output

-n 20 Parameters useful for reproducing the results
-n 30 All output

-o *low_dim_file stats_file*

Output the coordinates in the low dimensional space from the reduction to *low_dim_file* in text format. Statistics are output to the file *stats_file*, also in text format.

-p *epsilon_distance*

Specify the epsilon distance for neighbors to be used for ISOMAP method.

-r Calculate the reconstruction error for the mapping specified with **-f**. If used with **-x**, the reconstruction error is calculated for each of the specified *dimensions* (note that the **-x** option cannot be used with the autoencoder). If this option is used during learning, the reconstruction error is calculated from the *input_matrix*. For the autoencoder, this is the training set and the test set if the **-t** option is not specified. If the **-t** option is specified, it is the reconstruction error for the data used for training. For LLE and Isomap, the reconstruction error for the training data will always be 0 due to the mapping algorithm. Therefore it is advised to use a separate test set with these algorithms.

-s *stat_calculation*

Perform calculations on the statistics for a reduction. Options are NONE, DISTANCE, RECONSTRUCT and ALL. The default is NONE. The *dimensions* for which statistics are calculated is specified with **-x**

-t *test_file*

Specify explicitly the test set for autoencoder training. In this case no randomization of data occurs.

-u Perform calculations to estimate the intrinsic dimensionality using a variant of local PCA. In this case, PCA is performed on the manifold around each point using the nearest neighbors and the residual varience around each point is calculated. Statistics on the residual variance for each dimension are written to the output file. The top line is the mean residual variance. The next line is the minimum, followed by the maximum and the standard deviation for each dimension. If a single dimension is specified, either with **-d** or **-x** the output file consists of the residual variance around each datapoint. The time required for calculation will not change, even if fewer *dimensions* are specified. If **-z** is specified, the *output_file* is plotted.

-v For MDS, LLE, and IsoMap, the forward and reverse maps are obtained from a weighted average of the neighbors. By default, the weights are obtained from the inverse of the distances to the neighbors. If the **-v** flag is set, weights are obtained by solving a least-**s**quares problem as is perfromed in LLE.

-w *pymol_output*

Test output for pymol.

-x *dimensions*

Dimensions for which statistics should be calculated in matrix format. Examples are: **-x** "[0:3]" **-x** "[0:2:4]" **-x** "[0 1 5]". For eigenvalues, all *dimensions* are currently output regardless of the *dimen-*

sions specified. Additionally, the *dimensions* specified with **-x** currently have no affect on autoencoder statistics.

-y *neighbor_string*

Do not do the reduction, but instead just generate statistics on the number of connected manifolds. Statistics are generated for each number of neighbors in *neighbor_string* (specified in the same manner as for **-x**).

-z Make plots with gnu plot.

MATRIX FILE FORMATS

Five file formats are available for matrix and vector I/O depending on the type of matrix and format: full, symmetric, sparse, diagonal, pretty, and binary format.

The format for the full matrix is:

VecMat <*rows*> <*columns*>
<*data at row1, col1*>
<*data at row2, col1*>
<*data at row3, col1*>
...
<*data at row2, col_n*>
<*data at row1, col2*>
...

The format for a symmetric matrix is:

VecMatSym <*rows*>
<*data at row1, col1*>
<*data at row2, col1*>
<*data at row2, col2*>
<*data at row3, col1*>
<*data at row3, col2*>
<*data at row3, col3*>
...

for the lower triangle. The format for a diagonal matrix is:

VecMatDiag <*rows*>
<*data at row1, col1*>
<*data at row2, col2*>
<*data at row3, col3*>
...

for the diagonal. The format for a sparse matrix is:

VecMatSparse <*rows*> <*columns*>
<*col_number*rows+row_number*> <*value*>
<*col_number*rows+row_number*> <*value*>
...
END

The format for a pretty file is:

<data row1, col1> <data row1, col2> ... <data row1,col_n>
<data row2, col1> <data row2, col2> ... <data row2,col_n>
...

Binary files can also be loaded and saved, but are not architecture independent.

AUTHORS

W. Michael Brown, Shawn Martin, Haixia Jia, Jean-Paul Watson