# Multi-space Logistic Markov Embedding

- **Overview**

  Multi-space Logistic Markov Embedding (Multi-LME) is a software developed by Shuo Chen (shuochen@cs.cornell.edu) from Dept. of Computer Science, Cornell University. It learns from sequence data to embed the elements that constitute the sequences into multiple spaces. We originally used it in music playlists modeling. Please see the references for more details. This program is granted free of charge for research and education purposes. However you must obtain a license from the authors to use it for commercial purposes. Since it is free, there is no warranty for it.

- **Download**

  You can get the code from here.

- **Build**

  The software is implemented in C with the support of Open MPI 1.6. A Makefile that uses gcc and mpicc compilers is also included. To build the software, a simple "make" in the same directory as the source files will do. It creates three binaries MLogisticEmbed, MLogisticEmbed_MPI and MLogisticPred in the same directory. The building has been tested on various systems including GNU/Linux 2.6.9, 2.6.32, 2.6.18, 3.2.0 and Mac OS X Mountain Lion.

- **Usage**

  MLogisticEmbed and MLogisticEmbed_MPI take a training playlist dataset as input and produces a multi-space embedding/model file for the songs. MLogisticPred takes a testing playlist dataset and an embedding/model file as input and print to stdout the average log-likelihood on the test set.

  Format of the playlist data:
  The first line of the data file is the IDs for the songs, separated by a space. The second line are the number of appearances of each song in the file, also separated by a space. In fact these two lines are not essential in the program, you can replace it with any integer placeholders. Starting from the third line are the playlists, with each song represented by its integer ID in this file (from 0 to the total number of songs minus one). Note that in the playlist data file, each line is ended with a space.

  We provide sample files, which are the datasets we used for our papers. You can download them at http://lme.joachims.org.

  Training:

  MLogisticEmbed is used in the following format for training with single process (sequentially solving each embedding in different spaces):

  ```
  MLogisticEmbed [options] training_file model_file
  ```

  where training_file is the input training playlist set, model_file is the model to output.

  Similarly, MLogisticEmbed_MPI is used for training embeddings in different spaces in parallel. To run it with mult-core setting, one can use

  ```
  mpirun -np x MLogisticEmbed_MPI [options] training_file model_file
  ```

  where x is the number of processes you want to launch. Usually it should be no more than the number of cores your CPU has. When running in a distributed environment, one needs to support with a host file:

  ```
  mpirun -np x --hostfile myhostfile MLogisticEmbed_MPI [options] training_file model_file
  ```

  where myhostfile may look like:

machine-0 slots=2 max-slots=2
machine-1 slots=2 max-slots=2
machine-2 slots=2 max-slots=2
.....

It specifies what machines can be used and how many processes can each of them host. For more details, please refer to the manual of Open MPI.

Available options are:

| | | |
|---|---|---|
| -d | float | Dimensionality of the embedding (default 2) |
| -i | float | Learning rate (default 10.0) |
| -e | float | Error allowed for termination (default 0.00001) |
| -w | int | Number of last few iterations that the program tracks to determine termination (default 10) |
| -a | float | Adaptively increase the learning rate by this number if the improvement of the training likelihood is less than 0.005 (default 1.1) |
| -b | float | Adaptively decrease the learning rate by this number if the training likelihood deteriorates (default 2.0) |
| -p | [0, 1] | Whether to enable the popularity term (default 1, see [3] for more details) |
| -K | int | Number of spaces/clusters to divide the songs into (default 1). When running with MPI, please make sure it is no less than number of processes you want to launch. |
| -C | string | The file name of a preclustering file produced by other software. If this option is missing, the program uses the built-in preclustering method, with number of clusters specified by -K flag. If it is provided, the program uses the result in the file. The file should contain one line of integers separated by spaces. The number of integers should equal the number of songs in the training file. The integers specify which cluster the songs belong to. They range from 0 to number of clusters minus one. When using this preclustering file, please also make sure that the number of clusters is no less than the number of precosses you want to launch. |
| -G | float | The percentage of songs used as internal songs in the built-in preclustering method (default 0.08). Note that is has no effect when -C is used. |

Testing:

Testing only runs with a single process. It is simply as

```
MLogisticPred testing_file model_file
```

where testing_file is the input testing playlist set, model_file is the model obtained from training.

- **Visualization**

  We also provide a simple python script plot.py to visualize the embeddings with portals in multiple spaces. The usage is:

  ```
  python plot.py model_file
  ```

  Note that one needs to install Numpy and Matplotlib in order to run the script.

- **Example**

  The following three command lines show how to launch 4 processes in MPI to training a 2-dimensional model with 10 spaces/clusters, then test for average log-likelihood on the test set, finally visualize the trained model:

  ```
  mpirun -np 4 MLogisticEmbed_MPI -d 2 -K 10 train.txt model.ebd
  MLogisticPred test.txt model.ebd
  python plot.py model.ebd
  ```

- **Bug Report**

Please contact the author if you spot any bug in the software.

- **References**

If you use the datasets, please cite the following papers:

[1] Shuo Chen, Joshua L. Moore, Douglas Turnbull, Thorsten Joachims, Playlist Prediction via Metric Embedding, ACM Conference on Knowledge Discovery and Data Mining (KDD), 2012.

[2] Joshua L. Moore, Shuo Chen, Thorsten Joachims, Douglas Turnbull, Learning to Embed Songs and Tags for Playlists Prediction, International Society for Music Information Retrieval (ISMIR), 2012.

[3] Shuo Chen, Jiexun Xu, Thorsten Joachims, Multi-space Probabilistic Sequence Modeling, ACM Conference on Knowledge Discovery and Data Mining (KDD), 2013.