

PBMROC.3.0.0 and CvBMROC.2.0.0

Background

These two programs are the recent implementations of the conventional binormal model (CvBMROC.x.x.x, which supersedes ROCFIT, LABROC4 (and previous versions), or the single modality fits produced by ROCKIT.x.x) and the proper binormal model (PBMROC.x.x.x, which supersedes proproc.x.x and its previous implementations).

They are two-parameter ROC-models based on a multinomial likelihood function based on the categorization of data into their truth runs, they are based on maximum-likelihood estimation (1). Two-parameter models are frequently used in ROC analysis because they allow the creation of smooth curves without leaving “too much freedom” to the curve shape. In the following the parameters will be called par1 (a for CVBM and d_a for PBM) and par2 (b for CVBM and c for PBM) (2).

The two programs in their current implementations share most of their numerical core (3), and their differences are:

- Different preliminary analysis because the models tend to become degenerate (here meaning that the ROC curves are made of straight line segments) for different forms of the input data(4)
- Different initial estimates routines (because the log-likelihood functions look different)
- Different FPF(par1,par2,var) and TPF(par1, par2, var) (2)

The version number follows the general major (big difference) minor (some change in interface likely to produce backward compatibility issues with calls and/or input/output files) and version (bug removed, should work otherwise exactly in the same way).

Warnings and other

- Use it at your own risk (we will never assume any risks for usage) and we make no guarantees or warranties. We have tested the code extensively (3), but there are for sure still bugs (probably of the kind that rarely manifests itself or it is linked to unusual datasets or computational configurations) and possibly (small possibility) algorithmic issues (like for every other program).
- When in doubt ask <lpesce@uchicago.edu>, even a small doubt. We are more than happy to make sure that the analysis you do is meaningful and we want to be sure that the code is working.
- Please, don't share this software code with anyone. If anyone asks, please refer him or her to us. In general we are not sharing this code with many exactly because we want to be able to follow very carefully how the code is used and whether it works. It is free software with free support so there is little reason for not registering.
- Constructive feedback (like how you would like it to work) is more than welcome. Negative feedback is usually of no use and since we are working hard at it, it is likely to make us unwilling to help you.

How to Use the programs

Currently XROC.x.x.x is a DOS prompt program for Windows (we support Vista and Windows 7, but try to accommodate XP and 2000) and a command line program for Linux (with gcc-4.1+ installed should work) and Mac OS X (we are supporting 10.5 and 10.6).

This is how to run it (basic instructions, if you have other requirements, feel free to contact us; if you plan to use this as part of a script of a system call from an environment such as Perl or Matlab, you will need to adapt these instructions and we might be able to help):

1. Save the program
 - WINDOWS: (file: PBMROC.3.0.0.exe) to some folder of your choice
 - LINUX/UNIX/OS X (file PBMROC.3.0.0) to some folder or directory
2. Open terminal or console
 - Windows: Open the “DOS prompt” program
 - OS X: Open the “Terminal” application in “Utilities”
 - Linux: Open the terminal of your choice (e.g., rxvt, vt100)
3. Change folder (“cd” command) to the folder (directory) where the program is.
 - WINDOWS: For example if you save the file in “My documents\PROPROC\”. When you open the “DOS prompt” you will have to type “cd My Documents”, followed by “cd PROPROC”.
 - Other systems work equivalently, but the slash is forward “/” not backwards “\”.
4. Run the program:
 - WINDOWS: type (in the “DOS prompt”) “PBMROC.3.0.0.exe <input_file”, where input_file is the name of the file where your input is written (in labroc4 list format, to be described in a couple of lines).
 - LINUX/OS X: type in the console “./PBMROC.3.0.0 <input_file”. (the “./” might not be necessary, depending upon what version of Linux you are using and how it is set)
5. If you want to use an input file in another folder, please do so if you know how to do it, otherwise please copy it together with PBMROC.x.x.x. If you do be mindful about where you end up writing the many output files

If you try any of this unsuccessfully, feel free to contact us (lpesce@uchicago.edu).

The input file is built as follows

first line -> Title

second line -> whether positivity is for large or small values which means that positive cases tend to be larger or smaller than the negative cases (type "Large" or "small" on this line)

<values for negative cases one per line>

*

<values for positive cases one per line>

*

Example

Proproc example file

Large

1

2

6

8

4

2

3

*

4

5

6

9

10

12

*

(1, 2, 6, 8, 4, 2, 3) is the set of the actually-negative cases and (4, 5, 6, 9, 10, 12) is the set of the actually-positive cases for this dataset.

Note:

When using categorical data you need to convert it to this format. For example if you have 10 negative and 3 positive cases in the lowest category, you should type “1” 10 times in the section where the values for negative cases are and 3 times in the section where the positive are. E.g., for the following categorical data:

Category ->	I	II	III	IV	V
Actually-negative	4	3	2	0	1
Actually-positive	0	1	3	2	4

The input file becomes:

Example – conversion of categorical

Proc example file conversion categorical to list format

Large

1
1
1
1
2
2
2
3
3
5
*
2
3
3
3
4
4
5
5
5
5
*

Output files

The output files have the same name for the different models and the names do not change when a different input file is used. This is the simplest way to have the program work effectively from a script, in our opinion, and allows us to keep the interface as simple as possible. If you disagree you are welcome to express your opinion.

- MLE_fit.out, contains the output:
 - Wilcoxon statistics value and standard error (DeLong and DeLong) – it is usually a good reference against which to compare the model values to determine whether there are fitting issues.
 - d_a (a) and c (b): the estimated parameters for the curve according to the model used (i.e., CvBM or PBM).
 - AUC, the Area under the curve (also known as A_z for CvBM)
 - The variances and covariances from MLE estimates (parameters and AUC)
 - AUC variance from an approximate rule. Usually the rule is within 10% of the MLE, if not, you need to contact us, as there might be an estimation failure.
- Categorization.out. log file about which categories have been eliminated, if any, and what are the final categories used (note that categories are eliminated if and only if they are devoid of ROC information, unless there are more than 400 categories)
- roc_plot.out -> points to plot the curve and the input points, plus the theoretical input points (the value of the TPF and FPF obtained from the estimation of the cutoffs for the input points.)
- Some other files that are only needed for specialized usage, you are welcome to ask about them if you are interested.

References

1. Metz CE, Herman BA, Shen JH. Maximum likelihood estimation of receiver operating characteristic (ROC) curves from continuously-distributed data. *Stat Med.* 1998;17(9):1033-53.
2. Metz CE, Pan X. "Proper" Binormal ROC Curves: theory and Maximum-Likelihood Estimation. *J Math Psychol.* 1999;43(1):1-33.
3. Pesce LL, Metz CE. Reliable and computationally efficient maximum-likelihood estimation of "proper" binormal ROC curves. *Acad Radiol.* 2007;14(7):814-29.
4. Metz CE. Some practical issues of experimental design and data analysis in radiological ROC studies. *Invest Radiol.* 1989;24(3):234-45.