A Meta-Learner Framework to Estimate Individualized Treatment Effects for Survival Outcomes

Na Bo Yue Wei Lang Zeng Chaeryon Kang Ying Ding

1. Guidance

All simulation files are stored at "Simulations" folder. All real data analysis files are stored at "RealData-Analysis" folder. We will describe each folder below. Some browsers may block the html file. Please try a different browser if it cannot be opened.

2. Simulation

2.1 File description

All simulation files are stored at "Simulations" folder. Within this folder:

- "intermediate_result_100simulations.RData" contains the intermediate results used for reproducing tables and figures simulations. You will need to load these intermediate results if you run codes in "reproduce_simulation_tables_plots.html". Please refer to section "Reproduce simulation tables and figure" below for running R codes in this html file.
- "prepared_simulation_data_example.RData" contains one training data (train1.RData) and test (testdat.RData) data prepared by us for simulations when data is generated from Weibull models under balanced design and scenario 1. "Tlearner_result.csv" and "Xlearner_result.csv" are simulation results of D-T and D-X by submitting Python script "DNNSurv_Tlearner_1run_simulation.py" and "DNNSurv_Xlearner_1run_simulation.py" in "Simulations" folder, using our prepared training and test data. Please refer to the html file "Example_code_simulation_1run.html" generated by R markdown described in section "Example of one run of simulation when data is generated from Weibull models under balanced design and scenario 1" below for running D-T and D-X in Python. Our Python codes are implemented in Tensorfolow1.
- "Simulations" folder contains all functions needed to run R-T, R-X, B-T, B-X, D-T, D-X, weibull true model, logistic true model and data generation process. Here are more detailed descriptions:

1. baft_TXlearners.R includes functions for running T- and X-learners for bayesian accelerated failure models (BAFT).
2. baft_TXlearner_fun_util.R is sourced to predict survival probabilities in baft_TXlearners.R.
3. rsf_TXlearners.R includes functions for running T- and X-learners for random survival forests (RSF).
4. DNNSurv_Tlearner_1run_simulation.py includes the function for running T-learner for Cox-based deep neural network models for survival outcomes (DNNSurv) for one simulation.

5. DNNSurv_Tlearner_100runs_simulation.py includes the function for running T-learner for Cox-based deep neural network models for survival outcomes (DNNSurv) for 100 simulations.

6. DNNSurv_Xlearner_1run_simulation.py includes the function for running X-learner for Cox-based deep neural network models for survival outcomes (DNNSurv) for one simulation.

7. DNNSurv_Xlearner_100runs_simulation.py includes the function for running X-learner for Cox-based deep neural network models for survival outcomes (DNNSurv) for 100 simulations.

8. DNNSurve_TXlearner_fun_util.py is sourced in DNNSurv_Tlearner_simlulation.py and DNNSurv_Xlearner_simlula for calculating the baseline hazards.

9. simulation_design_survival_StandardLogistic.R includes the function for data generation when true potential survival times are generated from log-logistic models.

10. simulation_design_survival_Weibull.R includes the function for data generation when true potential survival times are generated from Weibull models.

11. StandardLogistic_TrueModel.R includes the function for running the log-logistic model which is considered as the true model when true potential survival times are generated from the log-logistic model.

12. weibull_true.R includes the function for running the Weibull model which is considered as the true model when true potential survival times are generated from the Weibull model.

2.2 Example of one run of simulation when data is generated from Weibull models under balanced design and scenario 1

File "Example_code_simulation_1run.html" generated by R markdown gives an example of running one simulation when data is generated from Weibull models under balanced design and scenario 1. You can open and run codes in it by following the instructions in it. It also contains instructions of running D-T and D-X in Python. Other simulation settings can be run in the same way.

2.3 Reproduce simulation tables and figures

File "reproduce_simulation_tables_plots.html" contains codes for running all simulations in the manuscript. It also contains instructions of running D-T and D-X in Python. Please follow the instructions in it and use server to run 100 times simulations. This file also gives codes of reproducing simulation tables and figures in the manuscript. You can open and knit or run it by following the instructions in it.

3. Real data analysis

3.1 File description

All real data analysis files are stored at "RealDataAnalysis" folder. Within this folder:

- "intermediate_result.RData" contains the intermediate results used for reproducing tables and figures. You will need to load these intermediate results if you run codes in "RealDataResults_manuscript.html". Please refer to section "Reproduce tables and figures in real data analysis" below for running codes in this html file.
- We used R and Python interface "r-reticulate" for running neural networks in Rstudio. r-reticulate is an R and Python interface (virtual environment) for R user to use Python for some works, e.g. calling Python from R, translating R and Python objects, checking the path where Python packages are installed, etc. Please install R package "reticulate" using the code: install.packages("reticulate").

- We use Keras from Python to build survival neural network in Rstudio. Please following the official guide (https://tensorflow.rstudio.com/install/) for installing Tensorflow and Keras in Rstudio.
- Notes for building neural networks through r-reticulate interface:

1. Python modules are installed under virtual environment. In the official guide, line "install_tensorflow(envname ="r-tensorflow")" means your tensorflow is installed under virtual environment "r-tensorflow", which is automatically created when you run this code. You can give a different name for it (e.g. we name it as "r-reticulate".). The same rationale for installing Keras.
2. If you never have Python or Anaconda installed on your computer, it will prompt a question: "Would you like to install Miniconda? [Y/n]:". Please type Y.
3. The most important point: Python and all Python packages have to be installed under the same virtual environment.

- "RealDataAnalysis" folder contains partially shared AREDS and AREDS2 data. We do not share full data, but we share outcomes, treatment information and three SNPs for generating tables.
- "RealDataAnalysis" folder contains all functions needed to run R-T, R-X, B-T, B-X, D-T and D-X. Here are more detailed descriptions:

1. baft_TXlearner.R includes functions for running T- and X-learners for bayesian accelerated failure models (BAFT).
2. baft_fun_util.R is sourced to predict survival probabilities in baft_TXlearner.R.
3. DNNSurv_TXlearner.R includes the functions for running T- and X-learner for Cox-based deep neural network models for survival outcomes (DNNSurv). This requires to source TensorFlow from Python.
4. DNNSurv_fun_util.R is sourced in DNNSurv_TXlearner.R for calculating the baseline hazards.
5. rsf_TXlearner.R includes the functions for running T- and X-learners for random survival forests (RSF).

3.2 Reproduce tables and figures in real data analysis

"RealDataResults_manuscript.html" contains codes for reproducing plots and tables for real data analysis. Please follow the instructions in it. It also contains instructions about calling Python in R. Since we do not share full data, we provide example codes of running R-T, R-X, B-T, B-X, D-T, D-X and CSF in real data. These codes are commented out.