To run this code, the users first have to install “Eigen” which is a C++ package. For Ubuntu users it can be easily done by running the following command:

```
sudo apt-get install libeigen3-dev
```

Add the include path of Eigen. You can do this by running the following command:

```
sudo ln -s /usr/include/eigen3/Eigen /usr/local/include/Eigen
```

or

```
sudo mv /usr/include/eigen3/Eigen /usr/local/include/Eigen
```

To speed up the code, we use “openmp” for parallel computing. You need to run the following command to generate the executable file.

```
g++ -fopenmp COX_L21_main_strong.cpp -o name_executable
```

Now you are ready to run the experiment, this algorithm has 13 arguments:

- file name of the training dataset in source domain.
- file name of the training dataset in target domain.
- file name of the testing dataset in target domain.
- number of instances of the training dataset in source domain.
- number of instances of the training dataset in target domain.
- number of instances of the testing dataset in target domain.
- number of features
- maximum iteration
- weight of target dataset
- multiplier of L2 norm
- number of \lambda you want to search
- m: the smallest searching \lambda ‘s multiplier ($\lambda_{min} = m\times\lambda_{max}$)

Note: The training and testing files are both in ".csv" format. Where each instance is represented as a row in file and the last two columns are survival times and censored indicators, respectively. Please refer to “Source_train.csv” to check the format.

You can run the command code as a toy example:

```
./name_executable Source_train.csv Target_train.csv Target_test.csv 76 76 39 552 100 2 0 0.0001 100 0.05
```

And the prediction results are stored in “Source_train.csv_record_new.txt” is generated by running the above command code. In “Source_train.csv_record_new.txt”, each column corresponds to a \lambda, and each column has 56 elements:

Row.1	lambda
Row.2	The value of objective function
Row.3	The value of the smooth part of the objective function
Row.4	Number of features left after run the strong rule
Row.5	C-index

Row.6	Number of non-zero coefficients
Row.7—56	The index of top 50 selected features

For better understanding of the code, you are suggested to read the paper

Yan Li, Lu Wang, Jie Wang, Jieping Ye, and Chandan K. Reddy, "**Transfer Learning for Survival Analysis via Efficient L2,1-norm Regularized Cox Regression**", *In Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Barcelona, Spain, December 2016.