

User Guide of the Code for Urban Water Quality Prediction

Introduction

The code released by us comprises three parts, including drawing code, model code and model comparison code. Copyright by [Urban Computing Team](#), Microsoft Research.

- Code of Model: implements our algorithm namely stMTMV, which is stored as “model_stMTMVL.m”.
- Code of evaluation: calculates MSE and RMSE of our method for model evaluation, including “eval_stMTMVL_mse.m” and “eval_stMTMVL_rmse.m”.
- Code for launching: you can use “run_stMTMV.m” to launch the method.

If you use codes above, please do cite our paper^{[1][2]}.

Reference

The model code is based on MALSAR, which is published by Jiayu Zhou on the website <http://www.MALSAR.org>. Please cite it^[3] before using it, and add the library to your MATLAB path first, or you cannot run our code successfully. You can follow the install guide on the homepage of MALSAR. The following figure is a simple guide for setting MATLAB path. For convenience, we also put the MALSAR folder in “reference code”.

Simple guide:

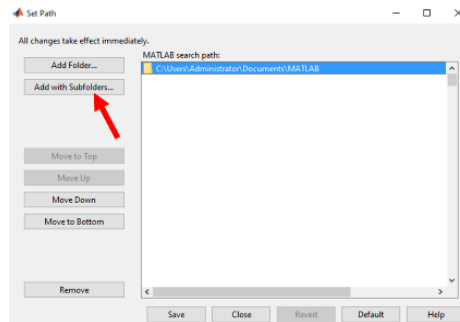


Fig.1 Add with subfolders to include all the file under the path

[1] [Urban Water Quality Prediction based on Multi-task Multi-view Learning](#), Ye Liu, Yu Zheng, Yuxuan Liang, Shuming Liu, David S. Rosenblum, in Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016, June 23, 2016.

[2] Zheng, Yu, et al. "Urban computing: concepts, methodologies, and applications." ACM Transactions on Intelligent Systems and Technology (TIST) 5.3 (2014): 38.

[3] Zhou, Jiayu, Jianhui Chen, and Jieping Ye. "Malsar: Multi-task learning via structural regularization." Arizona State University (2011).

Code of model

Introduction:

The model code of stMTMV method is implemented by Least_stMTMVL.m” and “model_stMTMVL.m”. The formal one is modified by “Least_L21.m” in MALSAR which excludes the parameters like pLambda, pGamma etc. And the following illustration is about the latter one.

Input:

X_{train}, X_{test} : denotes the input matrix of the training set and testing set. Their dimension are $\{sampleNums * featureNums\} * taskNums$. The contents in $\{\}$ should be put into a cell in MATLAB.

Y_{train}, Y_{test} : denotes the output matrix of the training set and testing set. Their dimension are $\{sampleNums * 1\} * taskNums$.

$simM$: denotes the similarity matrix(correlation between stations) for tasks.

$lambda$: spatio-temporal view consistency parameter, needs to be tune.

$gamma$: graph Laplacian regularization parameter, needs to be tune.

$theta$: L2, 1-norm regularization parameter, needs to be tune.

$STViewIndex$: index for seperating spatial and temporal view. The setting number can be 1: temporal view, 2: spatial view, 3: S-T view without alignment, 4:S-T view with alignment. This time we use 4 to combine the spatial and temporal view.

$ViewSegIndex$: segment index for temporal and spatial view.

Output:

w : denotes the weight matrix, with the dimension of $featureNums * taskNums$

P.S: The output value called fval (function value vector) is never used in this method.

Code for launching

Introduction:

To validate our stMTMV model, we deliver the launch code and compare it with some baselines.

Input:

X_{train} : includes each tasks' features as the training set. Its dimension is $\text{taskNums} * \{\text{sameleNums} * \text{featureNums}\}$. The contents in {} should be put into a cell in MATLAB.

X_{test} : include each tasks' features as the testing set, whose format is the same as X_{train} .

Y_{train} and Y_{test} : respectively denote Y in training set and testing set.

$paras_{stMTMV}$: consists of 3 parameters in stMTMV method, and need to be tune.

sim : denotes the similarity matrix which depends on the pipe attributes, namely denotes the correlation between each task defined in the paper.

$ViewSegIndex_{stMTMV}$: denotes the segment index between temporal view and spatial view.

Output:

$rmse_of_method$: denotes the rmse evaluated of our method.

Result:

Model Comparison	1 hour	2 hour	3 hour	4 hour
stMTMV	9.33e-2	9.66e-2	9.80e-2	9.90e-2

Code of evaluation

Introduction:

To calculate MSE and RMSE, we deliver this part of code.

Input:

X : includes each tasks' features as the testing set. Its dimension is $\text{taskNums} * \{\text{sameleNums} * \text{featureNums}\}$. The contents in {} should be put into a cell in MATLAB.

Y : denotes Y in testing set.

W : the weight W is obtained by our stMTMV method in the launch code.

Output:

$rmse$ or mse : respectively denote the rmse and mse evaluated of our method.

Contact

Contact: Yu Zheng, Lead Researcher at Microsoft Research

Email: yuzheng@microsoft.com

Homepages: <http://research.microsoft.com/en-us/people/yuzheng/>

Contact: Yuxuan Liang, Research intern supervised by Yu Zheng at Microsoft Research

Email: v-yuxlia@microsoft.com

Homepages: <https://yuxuanliang.com>

Contact: Ye Liu, Ph.D in School of Computing, National University of Singapore

Email: liuye.cis@gmail.com

Homepages: <https://sites.google.com/site/lewisliuyepku/home>