How to Use the SSVM Toolbox- R Version

A. Introduction

The usage of SSVM-R version toolbox is same as the Matlab one which contents the following functions:

- ssvm_train()
  Training and building a SSVM or SSVR predicting model. Please note that for classification problems, the one-against-one multi-classification algorithm is also available.
- ssvm_predict()
  By utilizing the predicting model built by ssvm_train(), ssvm_predict() can give the predicted result of a testing data set.
- hibiscus()
  Choosing the best combination of the weight parameter C and γ in the Gaussian kernel. Please note that this function can only be used in the Gaussian kernel

B. Training a model

Example:

model=ssvm_train(label, instance, s=0, t=0, r=1, c=100, g=0.1, v=5)

Input:

- label: The label of training data.
- Instance: The instance of training data.
- Other control parameters:
  - ✓ s: Learning method (default: 0).
    0- SSVM
    1- SSVR
  - ✓ t: The kernel type (default: 0).
    0- linear
    1- Gaussian
    others- Can be specified by users
  - ✓ c: The weight parameter C of SVMs (default: 100).
  - ✓ g: The gamma in Gaussian kernel function (default: 0.1).
  - ✓ e: The epsilon-insensitive value in epsilon-SVR (default: 0.1).
  - ✓ r: The ratio of subset size to the full data size when using Reduced SVM (default: 1).
  - ✓ v: The number of cross-validation folds (default: 5).

Output:

- w: The normal vector of separating (or response) hyperplane.
- b: The bias term.

- RS: The reduced set.
- Err.Training: The training error.
- Err.Validation: The validation error.
- Err.Final: If the validation fold equals to 1, Err.Final returns training error rate in SSVM and R-squared, relative 2-norm, and mean absolute training error in SSVR.

C. Predicting data

Example:

ERR = ssvm_predict(label, instance, model)

Input:

- label: The label of testing data.
- Instance: The instance of testing data.
- Model: The model built by ssvm_train().

Output:

- ErrRate: The testing error.
- PredictedLabel: The predicted label of testing data.

D. Choosing the best parameters for Gaussian kernel using the uniform design

Example:

Result = hibiscus(label, instance, s=0, v=5, r=1, Design=c(9, 5))

Input:

- label: The label of training data.
- Instance: The instance of training data.
- Other control parameters:
  - ✓ s: The learning method (default: 0).
    - 0- SSVM
    - 1- SSVR
  - ✓ e: The epsilon-insensitive value in epsilon-SVR (default: 0.1).
  - ✓ r: The ratio of subset size to the full data size when using Reduced SVM (default: 1).
  - ✓ v: The number of cross-validation folds (default: 5).
  - ✓ Design: The 2-stage of uniform design, can be 9-5 or 13-9 (default: 9-5).

Output:

- TErr: The training error.

- VErr: The validation error.
- Best_C: The best weight parameter C.
- Best_Gamma: The best γ in Gaussian kernel.
- Points: Trying points.

E. An example

```
result = hibiscus(y, A, s=0, v=5, r=1, Design=c(13,9))
model=ssvm_train(y, A, s=0, t=1, r=1,c=result[["Best_C"]], v=5, g=result[["Best_Gamma"]])
ERR = ssvm_predict(y, A, model)
```