

Documentation of nnrank1.1

Nnrank1.1 is a machine learning software and a neural network software for ordinal regression and classification. It supports training, testing, and ensemble.

Installation:

unzip nnrank.tar.gz on Linux.

(1) Four programs: nnrank, nnrank_predict, nnclass, nnclass_predict.

Ordinal regression:

nnrank: training and testing

nnrank_predict: prediction

Classification

nnclass: training and testing

nnclass_predict: prediction

(2) data: contains 20 pairs of training and test datasets generated from Abalone dataset.

(3) class.sh: a script to test nnclass.

(4) rank.sh: a script to test nnrank.

Input data format:

InputDimension

ClassNum class1 class2 ...

DataPointNum

class input1 input2

Class is represented by integer starting from 0.

For ordinal regression, classes are ordered by the value of intergers.

For an example, see: ./data/abalone_train_5.1 as follows.

10

5 0 1 2 3 4

1000
0 0.00000 0.00000 1.00000 0.24500 0.19000 0.06000 0.08600 0.04200 0.01400 0.02500
.....

Explanation :

input dimension is 10.

number of classes is 5, represented by 0, 1, 2, 3, 4.

number of data points: 1000

the label of the first data point is: 0,

feature values of the first data point is:

0 0.00000 0.00000 1.00000 0.24500 0.19000 0.06000 0.08600 0.04200 0.01400 0.02500

Usage of programs:

(1) Ordinal regression training

nnrank options training_file

Options:

-t test file

-m file to save trained model

-h number of hidden units (default: 10)

-e number of epochs (default: 500)

-s random seed (default: 999)

-r learning rate (default: 0.01)

-b number of batch (default: total number of data points)

This parameter controls the learning mode. If it is set to the total number of data points (n), neural network update weights per example. Otherwise, it updates weights for every n / m examples.

m is the number of batch.

-a adaptive parameter to adjust learning rate (default: 50)

This parameter instructs neural network to halve learning rate if the training error continuously goes up for a specified number of epochs.

(2) Ordinal regression prediction

nnrank_predict options model_file test_file

Options:

-o output file (default: standard output)

-d decision threshold (default: 0.5)

This program computes accuracy on the test dataset.
It does not report detailed prediction results.

(3) Ordinal regression prediction using ensembles

```
nnrank_ens -d threshold model_definition_file test_dataset
```

threshold: default 0.5.

model_definition_file is used to specify the trained neural network models.

(for an example: rank_model.def)

line 1: num_of_models num_of_classes

line 2: path to a model file

line 3: path to a model file

....

Run rank_ensemble.sh to test it.

(4) classification training

```
nnclass options training file
```

Options:

-t test file

-m file to save trained model

-h number of hidden units (default: 10)

-e number of epochs (default: 500)

-s random seed (default: 999)

-r learning rate (default: 0.01)

-b number of batch (default: total number of data points)

-a adaptive parameter to adjust learning rate (default: 50)

(5) classification prediction

```
nnclass_predict options model_file test_file
```

Options:

-o output file (default: standard output)

-d decision threshold (default: 0.5)

(6) classification prediction using ensembles

```
nnclass_ens model_definition_file test_dataset
```

threshold: default 0.5.

model_definition_file is used to specify the trained neural network models.
(for an example: class_model.def)
line 1: num_of_models num_of_classes
line 2: path to a model file
line 3: path to a model file
....

Run class_ensemble.sh to test it.

History:

NNRank 1.0: 10/2006

NNRank 1.1: 4/2007

I thank Yuping Shen help identify a bug in the code (AMBIBUITY due to division by 0).

Reference:

J. Cheng and G. Pollastri. A Neural Network Approach to Ordinal Regression. International Joint Conference on Neural Networks (IJCNN), Hongkong, 2008.

Contact:

Dr. Jianlin Cheng
Department of Computer Science
University of Missouri
Email: chengji@missouri.edu
