Project
Learning and Intelligent Systems
SS 2015

LAS Group, Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `http://las.ethz.ch/courses/lis-s15/`
Email questions to:
`lis2015@lists.inf.ethz.ch`

Project 4, May 5th, 2015
Classification with Missing Labels

You should not use any other data other than those that we provide you. You are also not allowed to hand-label the given data. You can make at most 200 submissions on the validation dataset and 10 on the test dataset.

1 Introduction

In this project you will classify images into 8 different classes. We have processed the images and extracted a total of 641 features, which you will use to build your classifiers.

2 Input specification

You are given the following four files:

- `train.csv` — The features of the training data.
- `train_y.csv` — The labels of the training data. Importantly, the labels could be missing for some data points, see further details below.
- `validate.csv` — The features of the validation data.
- `test.csv` — The features of the testing data.

Format of feature files. The files containing the features have one data point per line and the features of that data point are delimited by commas.

Format of label file. The file `train_y.csv` has one number per line. There is one such line for each corresponding data point in `train.csv`, that is, the files `train.csv` and `train_y.csv` have the same number of lines.

- When the label of the training data is available, the corresponding line will contain the class label represented by a number in `{0,1,2,3,4,5,6,7}`.
- When the label of the training data is missing, the corresponding line will contain number -1.

3 Output specification

For a given data point (in test set or in validation set), your goal is to predict the probabilities with which it belongs to one of the 8 classes. Let us denote the output of these probabilities for a data point as:

$$p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7$$

The number p_i for $i \in \{0, 1, \ldots, 7\}$ denotes the probability you assign that this data point belongs to class i. These 8 numbers in your prediction output should satisfy following:

- Each one of these numbers should be in $[0, 1]$, *i.e.*, $0 \leq p_i \leq 1,\ \forall i \in \{0, 1, \ldots, 7\}$.
- Sum of these 8 numbers should not exceed 1, *i.e.*, $\sum_{i=0}^{7} p_i \leq 1$.

You should produce files that contain 8 numbers per line for the probabilistic prediction of the corresponding data point in `validate.csv` or `test.csv`. As an example, your prediction files will contain lines in following format:

```
0.15,0.6,0.05,0.18,0.01,0.005,0.005,0
0.79,0.03,0.03,0.03,0.03,0.03,0.03,0.03
0.75,0.03,0.07,0.03,0.03,0.03,0.03,0.03
```

You have to provide two files of predictions — one for the validation dataset, and one for the testing dataset. NOTE: The prediction file corresponding to validation set will have same number of lines as in `validate.csv`. The prediction file corresponding to test set will have same number of lines as in `test.csv`.

4 Evaluation and Grading

Let us consider a data point (in `test.csv` or `validate.csv`) for which you are outputting the predictions. Let us denote the true class label of this data point by $y \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. Let us denote your prediction probabilities as $\mathbf{p} = [p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7]$. Then, the probability that you assigned to the true class is given by p_y. Your predictions are evaluated by the negative-log-likelihood of your prediction probability for the true class, as defined by the following loss function:

$$\ell(y, \mathbf{p}) = -\log\big(\max(0.0001, p_y)\big).$$

Note that $\log$ denotes logarithm to the base e. The $\max(0.0001, p_y)$ is used to avoid penalizing you badly for very wrong predictions (*e.g.*, outputting $p_y = 0$). Hence, the maximum penalty that you can receive for one data point is $-\log(0.0001)$, *i.e.,* 9.210340.

If there are n data points in the file, your final score will be computed as the average loss over these n points. This will give you a score for *validation set* and for *test set*. We will compare the score of your submission to two baseline solutions: a weak one (called "baseline easy") and a strong one (called "baseline hard"). The grade is computed as the *maximum* of the following two percentages.

- Perc_A — Equal to 50% if you are performing at least as good as the easy baseline on the *validation set* and 0% otherwise. Hence, by looking at the ranking you can immediately know if you will receive at least 50% of the grade.
- Perc_B — Let the scores of the easy baseline and the hard baseline on the *test set* be BE and BH respectively. If we denote the score that you reach on the *test set* as E, then you will obtain a score of

  $$\text{Perc}_B = \left(1 - \frac{\text{E} - \text{BH}}{\text{BE} - \text{BH}}\right) \times 50\% + 50\%.$$

  If you perform better than the hard baseline, you will receive $\text{Perc}_B = 100\%$.

4.1 Report

You are requested to upload a ZIP archive containing the team report *and* the code. We have included a template for LaTeX in the file `report.tex`. Please keep the reports brief (under 2 pages). If you do not want to use LaTeX, please use the same sections as in `report.tex`. Reports are uploaded on the same page as the test set submissions.

4.2 Deadline

The submission system will be open until **Sunday, 31.05.2015, 23:59:59**.