clana Documentation

Release 0.3.0

Martin Thoma

Jun 22, 2022

CONTENTS:

1 How to use clana with MNIST **1**
- 1.1 Prerequesites . . . 1
- 1.2 Usage . . . 1

2 Label Format **3**
- 2.1 Specification . . . 3
- 2.2 Example . . . 3

3 Classification Dump Format **5**
- 3.1 Specification . . . 5
- 3.2 Example . . . 5

4 Ground Truth Format **7**
- 4.1 Specification . . . 7
- 4.2 Example . . . 7

5 clana.distribution **9**

6 clana.io **11**

7 clana.get_cm **13**

8 clana.get_cm_simple **15**

9 clana.utils **17**

10 clana.visualize_cm **19**

11 Indices and tables **21**

CHAPTER

ONE

HOW TO USE CLANA WITH MNIST

1.1 Prerequesites

Install `clana` and execute the example:

```
$ pip install clana
$ python mnist_example.py
```

This will generate the clana files.

1.2 Usage

1.2.1 distribution

```
$ clana distribution --gt gt-test.csv
11.35% 1 (1135 elements)
10.32% 2 (1032 elements)
10.28% 7 (1028 elements)
10.10% 3 (1010 elements)
10.09% 9 (1009 elements)
 9.82% 4 ( 982 elements)
 9.80% 0 ( 980 elements)
 9.74% 8 ( 974 elements)
 9.58% 6 ( 958 elements)
 8.92% 5 ( 892 elements)
```

1.2.2 get-cm

This is an intermediate step required for the visualization.

```
$ clana get-cm --predictions train-pred.csv --gt gt-train.csv --n 10
2019-07-02 21:53:40,547 - root - INFO - cm was written to 'cm.json'
```

1.2.3 visualize

```
$ clana visualize --cm cm.json
Score: 12634
2019-07-02 22:13:54,987 - root - INFO - n=10
2019-07-02 22:13:54,987 - root - INFO - ## Starting Score: 12634.00
2019-07-02 22:13:54,988 - root - INFO - Current: 12249.00 (best: 12249.00, hot_prob_
↪thresh=100.0000%, step=0, swap=False)
2019-07-02 22:13:54,988 - root - INFO - Current: 10457.00 (best: 10457.00, hot_prob_
↪thresh=100.0000%, step=1, swap=False)
2019-07-02 22:13:54,988 - root - INFO - Current: 10453.00 (best: 10453.00, hot_prob_
↪thresh=100.0000%, step=3, swap=False)
2019-07-02 22:13:54,988 - root - INFO - Current: 10340.00 (best: 10340.00, hot_prob_
↪thresh=100.0000%, step=6, swap=True)
2019-07-02 22:13:54,989 - root - INFO - Current: 10166.00 (best: 10166.00, hot_prob_
↪thresh=100.0000%, step=14, swap=True)
2019-07-02 22:13:54,989 - root - INFO - Current: 9644.00 (best: 9644.00, hot_prob_
↪thresh=100.0000%, step=17, swap=True)
2019-07-02 22:13:54,989 - root - INFO - Current: 9617.00 (best: 9617.00, hot_prob_
↪thresh=100.0000%, step=19, swap=True)
2019-07-02 22:13:54,990 - root - INFO - Current: 9528.00 (best: 9528.00, hot_prob_
↪thresh=100.0000%, step=38, swap=False)
2019-07-02 22:13:54,992 - root - INFO - Current: 9297.00 (best: 9297.00, hot_prob_
↪thresh=100.0000%, step=86, swap=True)
2019-07-02 22:13:54,993 - root - INFO - Current: 9092.00 (best: 9092.00, hot_prob_
↪thresh=100.0000%, step=109, swap=True)
2019-07-02 22:13:54,994 - root - INFO - Current: 9018.00 (best: 9018.00, hot_prob_
↪thresh=100.0000%, step=123, swap=True)
Score: 9018
Perm: [0, 6, 5, 3, 8, 1, 2, 7, 9, 4]
2019-07-02 22:13:55,029 - root - INFO - Classes: [0, 6, 5, 3, 8, 1, 2, 7, 9, 4]
Accuracy: 94.34%
2019-07-02 22:13:55,152 - root - INFO - Save figure at '/home/moose/confusion_matrix.tmp.
↪pdf'
2019-07-02 22:13:55,269 - root - INFO - Found threshold for local connection: 258
2019-07-02 22:13:55,269 - root - INFO - Found 9 clusters
2019-07-02 22:13:55,270 - root - INFO - silhouette_score=-0.0067092812311967
    1: [0]
    1: [6]
    1: [5]
    1: [3]
    1: [8]
    1: [1]
    1: [2]
    2: [7, 9]
    1: [4]
```

The following file formats are used within `clana`.

CHAPTER

TWO

LABEL FORMAT

The label file format is a text format. It is used to make sense of the prediction. The order matters.

2.1 Specification

- One label per line
- It is a CSV file with `;` as the delimiter and `"` as the quoting character.
- The first value is a short version of the label. It has to be unique over all short versions.
- The second value is a long version of the label. It has to be unique over all long versions.

2.2 Example

2.2.1 Computer Vision

```
car;car
cat;cat
dog;dog
mouse;mouse
```

mnist.csv:

```
0;0
1;1
2;2
3;3
4;4
5;5
6;6
7;7
8;8
9;9
```

2.2.2 Language Identification

```
German;de
English;en
French;fr
```

CHAPTER

THREE

CLASSIFICATION DUMP FORMAT

TODO: THIS IS WAY TOO BIG!

The classification dump format is a text format. It describes what the output of a classifier for some inputs.

3.1 Specification

The Classification Dump Format is a text format.

- Each line contains exactly one output of the classifier for one input.
- It is a CSV file with `;` as the delimiter and `"` as the quoting character.
- The first value is an identifier for the input. It is no longer than 60 characters.
- The second and following values are the outputs for each label. Each of those values is a number in `[0, 1]`.
- The outputs are in the same order as in the related `label.csv` file.

3.2 Example

```
identifier 1;0.1;0.3;0.6
ident 2;0.8;0.1;0.1
```

CHAPTER

FOUR

GROUND TRUTH FORMAT

The Ground Truth Format is a text file format. It is used to describe the ground truth of data.

4.1 Specification

- Each line contains the ground truth of exactly one element.
- It is a CSV file with `;` as the delimiter and `"` as the quoting character.
- The first value is an identifier for the input. It is no longer than 60 characters.
- The second and following values are the outputs for each label. Each of those values is a number in `[0, 1]`.
- The outputs are in the same order as in the related `label.csv` file.

4.2 Example

```
identifier 1;1;0;1
identifier 1;0.5;0;0.5
```

CHAPTER

FIVE

CLANA.DISTRIBUTION

CHAPTER

SIX

CLANA.IO

CHAPTER

SEVEN

CLANA.GET_CM

CHAPTER

EIGHT

CLANA.GET_CM_SIMPLE

CHAPTER

NINE

CLANA.UTILS

CHAPTER

TEN

CLANA.VISUALIZE_CM

CHAPTER

ELEVEN

INDICES AND TABLES

- genindex
- modindex
- search