Practical 2: Classification of seal images

CS5014 Machine Learning

Due date: Mon 27th Mar (Week 10) 21:00
50% of the coursework grade

Aims

The main aim of this practical is to gain experience in working with real experimental, imperfect, and limited data which has not been analysed before. You will read and process and clean the data from the dataset in a suitable way. You will then create a classification model to predict output classes based on a set of inputs and evaluate its performance. It is particularly important that the evaluation is carefully described along with a discussion of the limitations of the data and of the method used.

Dataset

You will analyse two datasets containing features extracted from small images (60 × 60 pixels). The small images were cropped from larger aerial images obtained during seasonal surveys of islands in the North Sea. Your task is to classify these images based of what type of seal pup is contained within them. Several example images are available on studres, but the task will focus on already extracted features.

The dataset contains two directories. The directory called `binary` contains data for a binary classification task (to be solved first as the minimum basic requirement). The directory called `multi` contains data for a multi-class classification task (to be tackled after the binary classification task is finished as a more advanced requirement).

You can access the dataset directly from any lab machines at the path `/data/cs5014/P2/`, if you plan to work on a lab machine (physically or remotely). You should not copy these to your home directory because they are too large, simply access them directly from any lab machine. In case you need to download the data to work from home, you can download the zipped dataset from the University's OneDrive.[1]

Each directory (and the corresponding zip file) contains three files:

- X_train.csv
- Y_train.csv
- X_test.csv

`X_train.csv` contains a dataset of comma-separated values where the rows represents individual images and the columns are the features extracted from that image. The data are structured as follows: the first 900 columns correspond to a histogram of oriented gradients (HoG) extracted from the image (10 × 10 px cells, 9 orientations, 2 × 2 blocks).[2] The next 16

[1] OneDrive link

[2] Dalal and Triggs, "Histograms of oriented gradients for human detection", CVPR'05, link

columns are drawn from a normal distribution ($\mu = 0.5, \sigma = 2$). The last $3 \times 16 = 48$ columns correspond to three colour histograms extracted from the same image, one for each channel (red, green, blue), with 16 bins per channel.

`Y_train.csv` contains corresponding class ID (output) for each sample row from `X_train.csv`. These can be used as ground truth for the supervised learning task.

`X_test.csv` contains data in the same format as `X_train.csv` and serves as the test dataset. We have not provided the corresponding outputs for this data so you can not use these data for training. You can also not evaluate how well you do on these data (since you do not know the correct labels). Any validation and evaluation of your model will therefore have to be performed on the data contained in the provided training set, following best practices discussed in lectures. You will write a program to train the model on training data and predict the classes on the test data (one ID for each row of `X_test.csv`). This output file should be called `Y_test.csv` and use the same format as `Y_train.csv`: one label for each line in `X_test.csv`. You should submit the `Y_test.csv` file for each task as part of your submission.

Task

You are asked to predict the class of an image based on the extracted features contained in the test dataset after training a machine learning model using the training dataset provided. As in the first practical, the solution is expected to consist of several steps:

1. loading the data,
2. (optional) cleaning the data and creating new input features from the given dataset,
3. analysing and visualising the data,
4. preparing the inputs and choosing a suitable subset of features,
5. selecting and training a classification model,
6. selecting and training another classification model,
7. evaluating and comparing the performance of the models, and
8. a critical discussion of the results, your approach, the methods used and the dataset provided.

Each of these steps should be clearly explained in the report. You may find some of the steps more relevant than others, e.g. you may choose to use a subset of features or all of them, as long as you provide a justification for either decision. In all cases, you should show that you understand the consequences of each decision on the performance of your model and provide evidence showing how altering the decisions alters the model performance.

Try to keep the report informative and focussed on the important details and insights – the report also demonstrates an understanding of what is important. There is a maximum page-limit of 15 pages, note that this is a limit not a target. If you have large amounts of (relevant!) data, you can move them to an appendix and refer from the main text. There is no baseline to compete against - these are new data. Some of you may wish to compare your results against those of your peers and discuss strategies and insights. There are many legitimate ways to approach this task; treat it as an open problem on which you can test everything covered in the module so far.

Deliverables

Hand in via MMS, by the deadline (please leave enough time to upload your submission):

- The source code of your application which works in the Python3 virtual environment set up as described in the lab setup slides.
- The predicted output file `Y_test.csv` for each classification task (binary and multiclass). The output files should be copied into separate directories ("binary") and ("multi") ready for inspection.
- A report in PDF format which contains details of each step of the process, justification for any decisions you take, and an evaluation of the final model. This should also contain evidence of functionality and any notable figures you have produced. There is a limit of 15 pages.

Please create a `.zip` file containing all of these and submit this to MMS. Do *not* include the dataset, your python virtual environment, or git repository.

Keep in mind that the report is the most important part of the submission – we want to understand why your model is a good model, and to understand its strengths and weaknesses, not just reported evaluation metrics. Does your model perform well on all classes? Did you compare balanced vs. regular accuracies? How did you process the data and set the hyperparameters and why?

Marking and Extensions

This practical will be marked according to the guidelines at https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_Descriptor Some examples of submissions in various bands are:

- A *basic implementation* **in the 11–13 grade band** is a submission which implements a classification model in a straight-forward way and contains some evaluation, but is lacking in quality and detail, for example only the binary classification task is solved, or is accompanied by a weaker report which does not evidence good understanding.
- An implementation **in the 14–16 range** should complete all parts of the specification including both the binary and multi-class classification tasks, should consist of clean and understandable code, and be accompanied by a good report which clearly describes the process and reasoning behind each step and contains a good discussion of the achieved results including graphs and evaluation measures.
- To achieve a grade of **17 and higher**, your solution should extend a solid basic solution *in a meaningful way*. Potential extensions include comparison of multiple algorithms with meaningful evaluation and discussion of these, which can include more advanced algorithms from course textbooks and research publications. Any applied algorithm must be accompanied by a critical comparison to any other algorithms you used, with a discussion of any differences. Excellent analysis is strictly required for this grade band.

Note that the goal is *solid machine learning methodology and understanding* rather than a collection of evaluation metrics – a good scientific approach and analysis are difficult, whereas running many different scikit-learn algorithms on the same data is easy. A basic solution can be based on a logistic regression model, as long as the methodology and evaluation are sound. Be thorough in your basic solution and see extensions as a means to strengthen your basic argument and methodology. Also note that:

- We will not focus on software engineering practice and advanced Python techniques when marking, but your code should be sensibly organised, commented, and easy to follow.
- Overlength penalty applies: Scheme A, 1 mark for work that is 10% over-length, then a further 1 mark per additional 10% over. See https://www.st-andrews.ac.uk/policy/academic-policies-assessment-examination-and-award-coursework-penalties/coursework-penalties.pdf
- Standard lateness penalties apply as outlined in the student handbook at https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html
- Guidelines for good academic practice are outlined in the student handbook at https://info.cs.st-andrews.ac.uk/student-handbook/academic/gap.html