Practical 2: Learning visual attributes

CS5014 Machine Learning

Due date: Fri 16th April (Week 10) 21:00
60% of the coursework grade
MMS is the definitive source for deadline and credit details

Aims

The aim of this practical is to apply machine learning on a recent, open-ended research problem. Compared to Practical 1, you will have a lot more freedom to choose the most appropriate approach, but you will need to justify and explain your decisions. The papers provided explain some of the background, but do not have to be understood in depth. A successful submission will demonstrate the understanding of:

- how to select and train a suitable classification model;
- how to evaluate and compare the performance of different models; and
- how to explain and justify your work in a technical report.

Dataset

The dataset is based on a subset of the GQA dataset for learning attributes and relations.[1] The GQA dataset consists of images where objects are annotated in terms of bounding boxes and relevant attributes and relations. Each row in our dataset corresponds to an object in one of the images and contains the following fields:

- the image ID
- the object ID
- position (x, y) and size (width, height) of the object's bounding box
- colour of the object (categorical)
- texture of the object (categorical)
- colour histogram extracted from the bounding box. This comprises 9 values for each of the three components in the CIELAB colour space (intensity, red-green, blue-yellow), for a total of 27 values
- a histogram of oriented gradients (HOG) extracted from the bounding box,[2] for a total of 288 values

[1] D. A. Hudson and C. D. Manning, "GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering," CVPR 2019, pp. 6693-6702

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection". CVPR 2005

- a set of complex cell responses based on oriented Gabor filters generated by the BIMP model,[3] for a total of 126 values

The histograms and cell responses are calculated in 9 different areas of the bounding box to encode spatial characteristics of objects. Please see the file "features.png" for a rough illustration of the features.

Your task is to correctly identify the colour and texture for these objects based on other features. Several example images are available on studres, but the task will focus on already extracted features – you are not expected to process any images. On studres, you will find two files.

`data_train.csv` contains the training set where rows represent individual objects, and columns represent features extracted from that object's bounding box. The columns labelled 'colour' and 'texture' are target variables that you should predict from the remaining features. All training and analysis should be done on this dataset. You are free to further split it into training and validation sets, use crossvalidation, or simply use all of it for training.

`data_test.csv` contains data in very similar format as `data_train.csv` and serves as the test dataset. Note that there are no columns for 'colour' and 'texture' here 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 label for each row of `data_test.csv`). Your program should produce two files containing your predictions for colour and texture, called `colour_test.csv` and `texture_test.csv`, respectively. You should submit these two files as part of your solution.

Main task - Classification

You are asked to predict the texture and colour of an object based on the extracted features contained in the test dataset after training a machine learning model using the training dataset provided. You should perform all tasks using Python3 and sklearn, as covered during the lab sessions. Your code should be correct, easy to follow, re-useable, and easy to extend, as in the previous practical.

Unlike the first practical, 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.

Good quality solutions to the main task can warrant a mark of up to 16.

Report

Your report should clearly explain and justify all the steps you undertake. 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. We specifically ask you to address the following questions in your report:

1. How did you process the data, including any splitting, scaling and selection. Justify any such decisions.

[3] K. Terzić, J.M.F. Rodrigues and J.M.H. du Buf, "BIMP: A Real-Time Biological Model of Multi-Scale Keypoint Detection in V1", Neurocomputing, Vol. 150, pp. 227-237, Feb 2015

2. Which machine learning algorithm did you use and why is it suitable for the task at hand?

3. What are the hyperparameters of your model, what do they represent, and how did you set them?

4. What type of regularisation is employed by your model?

5. Which optimisation strategy did you use and how does it work? What are its strength and weaknesses and why did you choose it over some other approach?

6. Which evaluation metrics did you use and why are they suitable for the task at hand? How well did your algorithm perform? How can you be sure that the performance you report is a repeatable result and not a statistical fluke?

7. Are some classes easier than others and why do you think that is?

8. Compare your solution to a simple logistic regression classifier from sklearn and explain any differences.

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 an advisory pagelimit of 10 pages for explaining the main task, or 15 pages if you attempted advanced tasks listed below. Note that this is a limit not a target and that too much text may hinder readability. If you have large amounts of (relevant!) data, you can move them to an appendix and refer from the main text.

Advanced tasks

Once the main task is completed to a high level, you may want to attempt some of the following advanced tasks for higher marks.

- You should compare your main model to another advanced model (i.e. not logistic regression), which should be a very different algorithm. Your solution should contain a detailed comparison of the two algorithms and an insightful explanation of how they differ and why they perform differently (or not). You should explain how the new algorithm works in detail.

- You should also provide a mathematical description of your main algorithm, clearly explaining all the main elements: the loss function, the gradient of the loss function, the optimisation strategy, how any nonlinear behaviour is generated, and any important assumptions made by your main algorithm.

- You should explore the importance of different features for each of the two tasks (colour and texture). You should consider both your understanding of the features and any experiments you conducted to show this. Was there anything surprising?

Please note that attempting advanced tasks does not guarantee a high mark. An excellent solution to the main task is strictly required for a mark of 17 and above.

Deliverables

Hand in via MMS, by the deadline of 9pm on Friday of Week 10 (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 W01 lab slides. This must be in the form of human-readable .py files, not the binary .ipynb notebook format!
- The predicted output files `colour_test.csv` and `texture_test.csv` for the two classification tasks. Please use this naming so we can test more easily.
- 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 an advisory limit of 15 pages and irrelevant content may be penalised.

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. Your file should not be more than a few MB in size.

In this practical, 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 look at 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 with significant weaknesses. The report may demonstrate a lack of understanding of some topics, and not all the questions are answered well.
- An implementation **in the 14–16 range** should complete the main classification task for both colour and texture. It should consist of clean and understandable code, and be accompanied by an insightful report which clearly describes the process and reasoning behind each step. The report should contain good and detailed evaluation, accompanied by any figures you find useful. The answers to specific questions listed above should demonstrate understanding and be mostly correct.
- To achieve a grade **in the 17–18 range**, you will need an excellent solution to the main task, including excellent code, insightful report, with no major mistakes. You will also need to produce a high-quality solution to one of the advanced tasks with a good explanation in the report.
- To achieve a grade **in the 19-20 range**, you will need to complete all tasks to a very high standard, and your solution and report should show thorough understanding of all techniques you have used, and evidence extensive further reading.

Note that there is no benefit in running many advanced algorithms. A basic solution can be based on any classification model, as long as the methodology and evaluation are sound. Be thorough in your basic solution and see advanced tasks as a means to strengthen your basic argument and methodology.

Policies

- Standard lateness penalties apply as outlined in the student handbook at `https://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html`. Always check that your upload was successful and alert `dopgt-cs@st-andrews.ac.uk` immediately if you notice an upload went wrong.
- Guidelines for good academic practice are outlined in the student handbook at `https://info.cs.st-andrews.ac.uk/student-handbook/academic/gap.html`