COMP90051 Statistical Machine Learning

Project: Character Recognition from Historic Documents

Due: Wednesday 5^{th} October 2016

Weight: 30% of final mark

Introduction

Optical Character Recognition (OCR) is an important machine learning application which converts an image of a printed document into text. This is a useful tool for digitising vast collections of books such as in libraries, meaning that text can be used efficiently for efficient searching and other applications. While OCR for modern documents is largely a solved problem, it remains very challenging case for historical documents. Early printing presses required manual selection and placement of each of the characters by the printer, and features wandering baselines,[1] ink splodges, use of odd fonts and caligraphic capitals, and the use of characters that are no longer in use in modern texts. See the figure below for an example. Added to this scanning artefacts such as ghosting of the reverse page and the curvature of the page towards the spine are often a problem. Together these issues mean that off the shelf application of modern OCR technologies produce mostly gibberish, and consequently is of little practical use.

In this project you will develop character classifiers for several historical documents, each produced shortly after the advent of the printing press. Note that these documents are in different languages, use different fonts, and have other idiosyncrasies specific to their author and printer. Your job is to identify for a given bitmap image of a character the identity of that character.

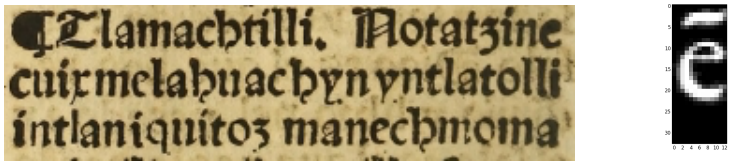

Figure 1: Example snippet, showing three lines from a historical document. Your job is to predict each of the characters identities (¶, L, l, a, ...), given the bitmap images for each character. Shown to the right is a bitmap for a single character after processing, which forms one of the training examples.

We hope that you will enjoy the project. To make it more engaging we will run this task as a kaggle in-class competition. Kaggle is one of the most popular online platforms for predictive modelling and analytics tasks. You will be competing with other teams in the class. The following sections give more details on data format, the use of kaggle, and marking scheme. Your assessment will be based on your team's rank, the score that you achieve, and your report. The marking scheme is designed so that you will pass if you put in effort. So fear not and embrace the power of machine learning.

Data Format

You will have access to several files, primarily train.csv, test.csv, and sampleSubmission.csv. These files will be available from the competition website (see the next section). File train.csv contains about 50,000 images along with their character label.[2] Each row corresponds to an character bitmap image and a small number of other features, and the

[1] The baseline is the vertical location of the bottom of each character, although note some characters like g and y have components that go below the baseline.

[2] There are some errors in the labelling of the data, as in most cases with real data which is never completely perfect. Don't get hung up on this, as it's only affects a small fraction of instances.

columns have a meaning as shown below:

Column(s)	Name	Meaning
0	Id	An integer identifier which is unique within a file
1	Character	An integer identifying the character (see Table 2 for their meanings)
2–4	Location	Integers identifying the book, page, and line in which the character is located
5–8	Bounding box	Left, bottom, right and top offset into the line image where the character is located
9–438	Pixmap	Pixels of the 13 by 33 image stored in greyscale, with each pixel an integer value in range $[0, 255]$. The matrix has flattened into a vector; for viewing, reshape these elements to the original image size (e.g., using `imshow` in matplotlib.) Note that images have been preprocessed by removing noise, mapping to greyscale, and rescaled to a uniform size.

Next, file test.csv contains 8715 records with the same fields as above, but with all Character values set to a placeholder value of 0, because this is what you need to predict. The Location and Bounding box fields are to support indexing a separate folder with the images for each line. You may elect to use this additional context beyond the bounds of the Pixmap supplied, should this be useful in predicting the character. The book identifier is another field you may find useful, given that the books different in several respects: particularly the font and language.

Train and test data comprise two non-overlapping sets of characters, even though in both files there will be Ids 0, 1, and so on. Finally, sampleSubmission.csv shows the beginning of an example submission file. Once you have done predictions of the characters in the test file, you should create a submission file in CSV format with the following structure.

```
Id,Character
0,5
1,17
2,3
...
```

The first line should be a header, exactly as shown. There should be 8715 lines in total, each with a unique ID. The IDs of predictions should match the IDs of entries in the test file, in the same order.[3]

As we provide no explicit validation set, you may want to reserve part of the training partition for this purpose during model development. Your job is to develop an algorithm that can automatically capture the nuances of the problem, in order to generalise well to unseen data (estimated here over the test set.)

Kaggle In-class Competition

Link: `https://inclass.kaggle.com/c/comp90051-2016`
Team registration form: `https://goo.gl/forms/D9S4xPPYrPxWYiim1`

Please do the following by the end of the first week after receiving this assignment:

- Setup an account on Kaggle with username and email both being your unimelb student email — only unimelb emails can access to the competition;

[3]Note that the sequence of Ids do not form a consecutive range of integers.

- Form your team of student peers;
- Connect with your team mates on Kaggle as a Kaggle team, using a team nam in `sml-2016-[team-name]` format. Only submit via the team; and
- Register your team using the Google forms link above.

You should only make submissions using the team name, individual submissions are not allowed and may attract penalties. Note that teams will be limited to 5 submissions per day.

The real labels for the test data are hidden from you, but were made available to kaggle. Each time a submission is made, half of the predictions will be used to compute your public accuracy score and determine your rank in public leaderboard. This information will become available from the competition page almost immediately. At the same time, the other half of predictions is used to compute a private accuracy and rank in private leaderboard, and this information will be hidden from you. At the end of the competition, only private scores and private ranks will be used for assessment. This type of scoring is a common practice and was introduced to discourage overfitting to public leaderboard. A good model should generalize and work well on new data, which in this case is represented by the portion of data with the hidden accuracy.

The evaluation score used in this competition is the accuracy over all classes, defined as the number of instances labelled correctly as a fraction of the total number of instances. Before the end of the competition each team will need to choose 5 best submissions for scoring. These do not have to be the latest submissions. Kaggle will compute a private accuracy for the chosen submissions only. The best out of the 5 will then be automatically selected and this private score and the corresponding private leaderboard ranking will be used for marking.

Report

Each team will submit a report with the description, analysis, and comparative assessment (where applicable) of the method or methods used. There is no fixed template for the report, but it should start with a very brief introduction of the problem and notation used. Then the report should describe the approaches that you have attempted along with the motivation for trying them. Reflect on why the method(s) performed or didn't perform well. If you tried different models, compare the methods to each other in the context of this competition. If you used any feature transformations or generated new features, you should also describe them in the report along with the expected effect from using such features and effect observed after implementation and evaluation. In comparing methods, you may want to use an evaluation besides measuring accuracy, in order to better understand the kinds of mistakes being made (e.g., with rare classes.)

Your description of the algorithms should be clear and concise. You should write it at a level that a postgraduate student can read and understand without difficulty. If you use any existing algorithms, you do not have to rewrite the complete description, but must provide a summary that shows your understanding and references to the relevant literature. In the report, we will be very interested in seeing evidence of your thought processes and reasoning for choosing one algorithm over another.

The report should be submitted as a PDF, and be no more than four A4 pages of content, including all plots, tables and references[4] (single column, font size of 11 or more and margins at least 1 cm, much like this document). You do not need to include a cover page. If a report is longer than four pages in length, we will only read and assess the report up to page four and ignore further pages.

[4]Plots can be useful for model selection, assessing convergence, displaying results and model interpretation, among other things. For instance, plotting the parameters of your model as 'images' can often give insights into what the model has learned.

Submission and Assessment

In summary, each team is required to make the following submissions for this project:

- One or more submission files with predictions for test data (at kaggle). This submission must be of the expected format as described above, and produce a place somewhere on the leaderboard. Invalid submissions do not attract marks for the competition portion of grading;
- Report in PDF format (via LMS);
- Source code used in this project as a single ZIP archive (via LMS). Your code can be in any of the following languages C, C++, C#, Python, R or Matlab. If there is another language you like to use, please contact us first. If the language requires compiling, a makefile or script must be provide to build the executables. We may or may not run your code, but we will definitely read it.

The project will be marked out of 30. No late submission of Kaggle portion will be accepted; late submissions of reports will incur a deduction of 4 marks per day, or part thereof. Based on our experimentation with the project task and the design of the marking scheme below, we expect that all reasonable efforts at the project will achieve a passing grade or higher. So relax and have fun!

Kaggle competition (15 marks) This mark takes into account both achieved accuracy, as well as your team's standing in the class. Assuming N is the number of teams, and R is your team's rank,[5] the mark you get for the competition part is

$$10 \times \frac{\max\{\min(acc, 0.90) - 0.25, 0\}}{0.65} + 5 \times \frac{N - R}{N - 1}$$

where $(x)_+$ returns x where $x > 0$ and 0 otherwise. The first term constitutes up to 10 marks, and rewards high accuracy systems with a maximum score for excellent systems with $\geq$ 90% accuracy, and zero score to those with scores $\leq$ 25% which are barely better than random guessing. The second term, worth 5 marks, is based on your rank and is designed to encourage competition and innovation. External teams of unenrolled students (auditing the subject) may participate, but their entries will be removed before computing the final rankings and the above expression, and will not affect registered students' grades. Note that invalid submissions will come last and will attract a mark of 0 for the score, so please ensure your output conforms to the specified requirements, and have at least some kind of valid submission early on!

Report (15 marks) The report will be marked using the rubric in Table 1.

Plagiarism policy

You are reminded that all submitted project work in this subject is to be your own individual work. Automated similarity checking software will be used to compare submissions. It is University policy that cheating by students in any form is not permitted, and that work submitted for assessment purposes must be the independent work of the student(s) concerned. For more details, please see the policy at `http://academichonesty.unimelb.edu.au/policy.html`.

[5] Ties are handled so that you are not penalised by the tie. For example, if the team accuracy scores are $A > \{B, C\} > D$, we assign the ranks $R_A = 1$, $R_B = R_C = 2$ and $R_D = 4$.

Critical Analysis (8 marks)	Report Clarity and Structure (7 marks)
7–8 marks Final approach is well motivated and its advantages/disadvantages clearly discussed; thorough and insightful analysis of why the final approach works/not work for provided training data; insightful discussion and analysis of other approaches and why they were not used	*6–7 marks* Very clear and accessible description of all that has been done, a postgraduate student can pick up the report and read with no difficulty
5–6 marks Final approach is reasonably motivated and its advantages/disadvantages somewhat discussed; good analysis of why the final approach works/not work for provided training data; some discussion and analysis of other approaches and why they were not used	*4–5 marks* Clear description for the most part, with some minor deficiencies/loose ends
3–4 marks Advantages/disadvantages discussed; limited analysis of why the final approach works/not work for provided training data; limited discussion and analysis of other approaches and why they were not used	*2–3 marks* Generally clear description, but there are notable gaps and/or unclear sections.
1–2 marks Final approach is barely or not motivated and its advantages/disadvantages are not discussed; no analysis of why the final approach works/not work for provided training data; little or no discussion and analysis of other approaches and why they were not used	*1 mark* The report is unclear on the whole and the reader can barely discern what has been done

Table 1: Report marking rubric.

0	&	10	1	20	;	30	I	40	T	50	è	60	r̃	70	g	80	r	90	¶
1	(	11	2	21	?	31	J	41	V	51	ì	61	t̃	71	h	81	s	91	ä
2	)	12	3	22	A	32	L	42	Y	52	ã	62	ũ	72	i	82	t	92	æ
3	*	13	4	23	B	33	M	43	[	53	d̃	63	]	73	j	83	u	93	ç
4	+	14	5	24	C	34	N	44	í	54	ẽ	64	a	74	l	84	v	94	ñ
5	,	15	6	25	D	35	O	45	q́	55	ĩ	65	b	75	m	85	x	95	ę
6	-	16	7	26	E	36	P	46	•	56	ñ	66	c	76	n	86	y	96	œ
7	.	17	8	27	F	37	Q	47	p̨	57	õ	67	d	77	o	87	z	97	ſ
8	/	18	9	28	G	38	R	48	q̨	58	p̃	68	e	78	p	88	~		
9	0	19	:	29	H	39	S	49	à	59	q̃	69	f	79	q	89	§		

Table 2: Mapping between label numbers and character symbols. Note that • denotes an unreadable character.