COMP90051 Statistical Machine Learning

Project 2 Description

Due date: 17:00 AEST (UTC +10), Wednesday 27th May
Weight: 30% (forming a combined hurdle with Project 1)

A central goal to machine learning is generalization: from a small observed sample of data from a distribution (the "training set") can we learn a good model which describes instances from that distribution (as measured using a "test set"). Most often we assume that the training and test distribution are the same, and that instances are drawn independently. That is, the data is *"i.i.d."*. This is the assumption behind the models and learning algorithms we have covered in the subject.

In practise, however, data is often a lot messier: The intended test scenario is often different to the training (or development) settings, that is, these datasets follow different distributions. Consider, for example, a handwriting recognition system trained to recognise characters written by a dozen different individuals, that is then deployed to recognise handwriting from another unseen person. If this test user has a different writing style, uses a different pen, or various other quirks that make their writing different from the training users, the predictive performance of the system will degrade. In general, when labelled testing samples do not match well with the sampling on which the learner has been trained, the system will not be able to generalize well to these new samples. This raises the challenge of developing machine learning methods which generalize well to both similar ("in-domain") and dissimilar ("out-of-domain") instances to those seen in training.

This problem goes by various names, such as *domain mismatch* or *covariate shift*. *Transfer learning* or its specialization, *domain adaptation*, are methods for addressing the problem. These are not techniques that we have looked at in class, and this project is your opportunity for you to learn for yourself about this important body of work. You will be learning straight from the research literature, and thus developing key skills that will help you in tracking leading developments in machine learning research.

In this project, you will work *individually* to

1. reproduce a research method on domain adaptation;
2. evaluate your method on a supplied dataset;
3. find another idea, and then implement and evaluate this on the dataset.

This will require reading several research papers, using these papers to guide your approach, e.g., through reimplementing algorithms and evaluation methodologies, or simply finding inspiration. More generally this task is aimed to develop your skills in finding relevant research, reading scientific literature, and synthesising these ideas in program implementation (and figuring out which details you can ignore!).

Part 1: Frustratingly Easy Domain Adaptation

Start by reading the following paper, which we will refer to as "FEDA" hereafter:

> Daumé III, Hal. "Frustratingly easy domain adaptation." *Proc. 45th Annual Meeting of the Association for Computational Linguistics, 2007*.

You will need to read the paper carefully, as this project requires you to implement this method for domain adaptation and evaluate your implementation on a supplied dataset.

To begin, download the dataset which consists of examination records from students at U.K. schools over a 3-year period. The data is comprised of three files: MALE.CSV, FEMALE.CSV, MIXED.CSV which correspond to three different types of school: single-sex male, single-sex female, and mixed gender, respectively. These will serve as our "domains". Each file includes a row for each student, with a number of features about the student and their school, and their exam score. For a full description of the dataset and the feature definitions, please see README.MD.

You will be developing a regression model to predict the exam score. To evaluate your methods, you will need to first split the data into partitions for training, development and testing. Next, to mimic a data-impoverished domain adaptation scenario, you will need to restrict the amount of labelled data in a target domain. To do so, you will use a form of 3-fold cross-validation, where the folds are the domains. In each replicate you will reduce the amount of data for one domain (100 instances each for training and development sets), which is the "target" domain, and this domain will be used for evaluation. The other two domains are treated as out-of-domain data. Your aim is to achieve low predictive error on held-out data in the target domain.

Task 1.1: Develop and evaluate baseline methods

In the FEDA paper, the authors consider six domain adaption baseline approaches: SRCONLY, TGTONLY, ALL, WEIGHTED, PRED and LININT. Your task is to implement and evaluate these baselines on then Schools dataset and report their mean squared error (MSE).
You will need to decide the best means of encoding the input features for learning and evaluate these methods with at least two different types of machine learning model. E.g., you might consider one neural network model, and one statistical learning model. Hyper-parameters should be set appropriately to achieve low generalisation error.

Task 1.2: Implement and evaluate FEDA

Implement the FEDA feature augmentation method according to the description in the FEDA paper, section 3. You do not need to implement the kernelized version. You should implement your method to be sufficiently modular to allow multiple different learning algorithms to be used, i.e., the learning approaches from Task 1.1. Note that the FEDA paper links to a Perl script implementing the method, however we ask that you implement this yourself.

Evaluate your method on the Schools dataset, using at least two different learning algorithms. Fit all hyperparameters carefully, and report sensitivity to the hyperparameter settings. Consider how the hyperparameters affect the treatment of weights in the "general" component, versus weights specific to out-of-domain or target domain components. Report your results in a similar form to the FEDA paper Table 2. You would expect to have six rows, corresponding to the 3 different domains x 2 classes of model.

You should also perform a secondary experiment to determine the effect of the amount of training data in the target domain on test error. That is, evaluate with several sizes of in-domain training sets, not just 100 instances as used in your main experiments. Report your results and discuss your findings.

Part 2: Domain Adaptation Extension

Your next task is to research other techniques for domain adaptation and related problems in transfer learning. You may wish to start by with the survey paper:

> Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2009): 1345-1359.

as well as the papers *cited by* the FEDA paper, and papers *that cite* the FEDA paper.[1] To find the latter, you can search for the paper title in Google Scholar (or similar, e.g., semantic scholar) and then follow the *"cited by"* link. This will allow you to find more modern papers in the area. There's been considerable progress in the last decade (including work by myself, see Li et al, 2018, cited below.) You can also search in Google Scholar (or similar) using keywords from the papers you have read. Except to scan over at least 10 papers and read 2-3 handful closely.

Your task will be to find a paper with a technique relevant to our domain adaptation problem, and implement the approach, or something inspired by what you've read, even if it does not match the approach exactly. Feel free to engage your creativity, if you have some great ideas of your own, although you will need to relate your method to techniques in the machine learning literature. Finally, evaluate your method on the Schools data, reporting your results using tables and/or figures as appropriate.

You will find many papers substantially more complicated than the FEDA paper, to the extent that they may be difficult to even understand, let alone implement. You are free to adapt implementation code you find online, although we would encourage you to keep to code

[1] For more pointers, see https://github.com/artix41/awesome-transfer-learning which includes a massive list of resources, including several tutorials, surveys and papers.

developed by the authors of the research paper in question (e.g., linked from the paper), or code that is part of a larger, recognised project. You should acknowledge wherever you have used other peoples' code, and link to their codebase and/or paper.

Note that the Schools data has been used by prior research in transfer learning, which you may find useful to guide your research. E.g.:

> Evgeniou, Theodoros, Charles A. Micchelli, and Massimiliano Pontil. "Learning multiple tasks with kernel methods." *Journal of machine learning research* 6.Apr (2005): 615-637.

Although this paper used the data differently to us, by treating each school as a separate task, rather than using the schools' gender mix.

Reporting your Findings

Finally, you need to write a short report in the style of a short academic paper. You will need to describe your problem, the details of your approach and how this relates to prior work in the academic literature, report your results and your conclusions. You should include citations to related work, and sufficient details that readers unfamiliar with the cited work can still understand the central aspects of your work. As an example of short paper structure and writing style, see:

> Li, Yitong, Timothy Baldwin, and Trevor Cohn. "What's in a Domain? Learning Domain-Robust Text Representations using Adversarial Training." In *Proc. Annual Conference of the North American Chapter of the Association for Computational Linguistics.* 2018.

Submissions should follow the formatting style of NeurIPS:
https://www.overleaf.coadbm/latex/templates/neurips-2019/tprktwxmqmgk
A Word template exists for NeurIPS (a.k.a. NIPS) 2015, which you are welcome to use, although I would encourage the use of LaTeX, which is better suited to academic writing. Your report should be a maximum of 3 pages. References do not count towards the page limit (they can spill onto page 4), although all your tables and figures will need to fit in the first 3 pages. You do not need to include an abstract.

Project Submission

Submissions should be made using the LMS and should include both your source code (as a .zip file) and your PDF report. Your code should be clearly organised with respect to each individual part, such that they can be easily marked. There should also be an overall description file indicating the development environment, e.g., python version and any other packages that is used in your project. You are welcome to use Python, Matlab or R, as well as scripting tools. Please ask if you wish to use other languages. We will not be assessing the code for style or extent of commenting, but we will use it to resolve questions we have with your report so it must be accessible. Please do not upload your data, just the code.

Evaluation

Your submissions will be evaluated on the following criteria:

Report writing [8 marks], looking for:

- clarity of writing;
- coherence of report structure;
- appropriate use of tables, figures, illustrations and formulae; and
- use of citations and references[2]

Report content [22 marks]

- *Part 1* [10 marks = 3 + 7 for sub-tasks 1 and 2, respectively], looking for:
  - exposition of method;
  - justification of decisions;
  - clear presentation of results; and
  - clear analysis of findings
- *Part 2* [12 marks], looking for the above criteria, plus:
  - motivation of method;
  - grounding against related work;
  - correctness of technique; and
  - ambition of technique

A submission that only attempts Part 1 could achieve a maximum of 18/30 marks, or 60%. Note that there are no explicit marks for the accuracy of your technique, although we will look at your reported results to judge the correctness of your method and the validity of your analysis. The emphasis here is on producing an excellent piece of empirical research, rather than simply performing well on a leaderboard.

Please submit your submissions as a PDF with format SID_USERNAME_proj2report.pdf, where SID is your student number and USERNAME is your Unimelb username. Submissions that fail to follow this simple instruction will attract a penalty.

Further Rules. You may discuss the academic papers you read at a high-level with others, but do not collaborate on solutions. You may consult online resources to build your conceptual knowledge, but you must not use online code for Part 1.[3] You are welcome to use standard APIs in python, as used in the notebooks from the workshops, or their equivalents in R or Matlab. You should use matplotlib, ggplot or similar for plotting graphs. Late submissions will be accepted to 4 calendar days with -3 penalty per day.

[2] You're free to choose a citation style. See various examples at https://www.overleaf.com/learn/latex/Natbib_bibliography_styles.

[3] The use of online code is permitted in Part 2, but you must acknowledge the source (e.g., providing the URL, and a citation, if appropriate) in both your report and your code.