LE49 Probabilistic Machine Learning

DRAFT Investigative projects 2019/2020. Compiled by Damon Wischik.

You should treat these project description as the starting point for an investigation. All projects involve some element of research risk, and there is no guarantee that it's possible to find what the project description suggests you might find. You should spend your time on the leads that are most interesting and productive — and you can still write an perfectly good report about a negative result.

The general marking guidelines for coursework are at `https://www.cl.cam.ac.uk/teaching/exams/acs_assessment.html`. Some more details:

- Your project report should be written up in the style of a NIPS or ICML conference paper, 8 pages plus one for references. You should aim to spend around 40 hours on this project.
- You will be graded on your report, not on your code. You must submit your code, but you needn't spend time tidying it.
- For a mark of 70% or higher, you will be expected to use appropriate evaluation criteria for your work. This might involve, for example, inventing metrics and stating them at the beginning of your report; running appropriate cross validation; or running tests to see if your findings are just noise.
- For a mark of 80% or higher, you will be expected to show significant insight or creativity.

Project 1 (Your project). You may propose your own project, or your own dataset for one of the projects listed here — but you must get approval from Dr Wischik, to confirm your proposal is suitable for this course.

Project 2 (Gaussian process model for flood data). The UK Environment Agency provides a real-time and historical feed of data from water monitoring stations throughout the UK. Pick a single station, e.g. Jesus Lock in Cambridge, and develop a Gaussian process model for water levels as a function of time. Pick a catchment area with multiple stations, and extend your model to describe the correlations between the stations. How much warning do the upstream stations provide of high water levels downstream?

Data is at `http://environment.data.gov.uk/flood-monitoring/doc/reference`.

Project 3 (Gaussian process classifiers and CNN uncertainty). Convolutional neural networks (CNNs) achieve state-of-the-art performance on image classification tasks, but provide no measure of confidence in their predictions. Train the Keras example CNN on MNIST. After training the model, use the outputs from the top level feature layer before the softmax classifier to train a Gaussian process classifier. Without re-training, use your model to classify the n-MNIST dataset, and also try adversarial perturbations using the CleverHans toolbox. Does the Gaussian process classifier report lower confidence for images outside its experience? How does the choice of kernel affect this? How should you evaluate a classifier that can report "I don't know"?

For this project you will need to learn about Gaussian process classification, sections 3.1–3.4 of Rasmussen's book. Make sure the classifier library you are using returns an appropriate measure of confidence, as per equation (3.24) in section 3.4. Keras example CNN available at `https://github.com/fchollet/keras/blob/master/examples/mnist_cnn.py`, and n-MNIST dataset at `http://csc.lsu.edu/~saikat/n-mnist/`

Project 4 (Graphical model for restaurant ranking). Yelp has made available a dataset of restaurants, patrons, and reviews. Star ratings are notoriously unreliable — but it may be that the ordering implied by each user's rankings is more helpful. Build a ranking model for restaurants, where the match outcomes are of the form "User u thinks that restaurant r

is better than r'". Experiment with other ways to obtain these outcomes, for example using only pairs of restaurants with one star difference in rating, or restaurants of the same type; also consider how you will treat draws. Evaluate the predictive performance of your model. Also, compare the ordering you obtain to the average star ratings.

Data is at `https://www.yelp.co.uk/dataset/challenge`

Project 5 (Cambridge bumps). Build a ranking model for the Cambridge bumps. You should consider how to model the distinctive features of the bumps, such as overbumps and gradual change in a boat's performance from year to year.

Data is at `http://www.cucbc.org/mays/results`

Project 6 (Multidimensional ranking with graphical models). Using the Yelp review dataset, apply a mixture model with $K = 2$ topics to the review text, treating the entirety of a single user's reviews as one document. This gives us a weighting $(\pi_u, 1-\pi_u)$ for each user u with respect to two topics. Fit a ranking model similar to TrueSkill, but in which each restaurant r has two dimensions (w_r^1, w_r^2), and a user ranks restaurants based on $\pi_u w_r^1 + (1-\pi_u) w_r^2$. This lets us score restaurants on two separate scales. Interpret these scales. What happens with more dimensions?

Data is at `https://www.yelp.co.uk/dataset/challenge`

Project 7 (Mixture models for image classification). Yelp has made available a dataset of photos taken by restaurant patrons. Feed these photos into the VGGNet image classifier, and for each photo and for each top-level convolution feature, extract the activation level of that feature pooled across the entire image. Treat each top-level feature as a 'word', and each image as a 'document', and use a categorical mixture model to identify clusters of image types. Experiment with activation thresholds, and with the layer of features you use. Display your groupings appropriately, and relate them to the picture labels.

Data is at `https://www.yelp.co.uk/dataset/challenge`

Project 8 (Mixture models for trip data). The Bay Area Travel Survey is a publicly available dataset of trips, including details of the person making the trip and of the trip's purpose. Using a subset of the data, consider each destination area to be a 'document' and each trip to be a 'word', and let the words consist of trip purpose and time, for example "shopping on a weekday morning", then use a document mixture model to identify types of destination. Next, using the remainder of the data, consider each person to be a 'document' and each trip to be a 'word', and let the word consist of trip time and destination type; then use a document mixture model to identify types of person. Interpret the person-types you find.

In practice, we might have detailed labelled trip data for a small subset of users, and passively collected unlabelled data for the bulk of users. This project's goal is to see if we can assign useful labels to the bulk of the data, using the labelled subset as a hint. Trip data is at `https://www.icpsr.umich.edu/icpsrweb/ICPSR/studies/34805` *and map data is at* `http://opendata.mtc.ca.gov/datasets/transportation-analysis-zones`