Machine Learning with real-world data

This course is designed to teach the principles of using machine learning (ML) to solve real world tasks.

There are a huge number of ML methods, with new variants being invented every day. But the ways that one works with these various methods are very similar. Hence we are teaching this course to give students an introduction to the methodological and experimental issues involved in working with ML. To a large extent, the principles of good experimental practice are the same whether one is using a well-known and simple approach or a deep-learning method that was developed last month. We're going to concentrate on the simple methods in this course, so that students can fully understand what's going on without having a very detailed description of a highly complex algorithm. But the principles we cover also apply to experiments with state-of-the-art methods.

Abstractly, we can think of ML as involving the following three aspects:

1. task
2. data
3. algorithm

In conjunction with these three aspects, we consider three processes:

1. data acquisition and preparation
2. feature extraction
3. evaluation

Task

Tasks are usually an abstraction from a real problem, or a piece of a larger architecture (in research, the larger architecture is often hypothetical). End-user systems may be quite different from experimental systems. Most research publications concern standard tasks: sentiment classification of movie reviews, document classification, image captioning etc, etc. Taking real problems and turning them into ML tasks is a complex art: it is no accident that many of the most high-profile AI results involve games (where the task is straightforward to define). In some cases, a subtle change in task makes a huge difference to research progress: for instance, it became easier to make progress in **statistical machine translation (SMT)** when the task was (implicitly) redefined as producing the closest match to a reference translation as measured by the **BLEU score**.

Data

Data is used to train and evaluate the ML system, ideally using the following three-way split (discussed in http://www.cl.cam.ac.uk/teaching/1920/MLRD/

handbook/dataset-splits.html):

1. Train
2. Development (also known as the validation or tuning set): this is used for ongoing experiments or for tuning system parameters.
3. Evaluation (also known as test data): ideally this is unseen by experimenter and only used once, for the final experiment.

ML systems may have various degrees of **supervision**:

1. Supervised: training data labelled with desired outcome
2. Unsupervised
3. Semi-supervised, moderately supervised etc

Data way be annotated in various ways:

1. Manually annotated (possibly with expert annotators, possibly **crowd-sourced** using a service such as Amazon Mechanical Turk,)
2. 'found' annotation (e.g., star ratings for movie reviews)

Data has to be acquired somehow. For instance, for language processing, the data will be text (a **corpus**), perhaps scraped from the web, possibly with additional material, such as translations into another language (**parallel text**, used in machine translation), images and so on.

Acquiring and pre-processing data is often the most time-consuming part of a real ML project. Factors to consider include:

1. What type of data can be used for the task?
2. Where should the data come from (note that there are complex and difficult questions here, such as how to avoid bias).
3. How much data is needed?
4. Is annotation needed for training or evaluation? (Annotation methodology is discussed in Session 6.)
5. What sort of preprocessing does the data require? For instance, data scraped from the web may require removal of markup, and detection and deletion of duplicate documents.

Features

Once data has been acquired in a suitable form, **features** are extracted for use by the chosen ML algorithm. The features may be as simple as the words in a text (**bag of words**), but some other types of features require considerable processing to extract, possibly using an existing ML system. At this point, external resources of various sorts may be used: for instance, Wikipedia may be used as a source of basic **world knowledge**.

Algorithm

This is the only part of ML which is described in the standard ML textbooks. An algorithm should be chosen which is appropriate for the task, given the available data and features. A fast and dumb algorithm is often better than a slow and sophisticated one, especially if large amounts of data are available for training. For any sort of practical application, robustness to unexpected data and consistency across datasets may be more important than obtaining the highest score on the evaluation dataset.

Evaluation

Evaluation allows one to see how well a chosen algorithm trained on some dataset performs on the given task. There are many different metrics and approaches for different tasks. Evaluation may be done by humans (possibly using **crowdsourcing**) but for standard tasks there are standardised metrics and test sets, enabling comparisons between research strategies.

In real world ML, the 'best' performance will depend on the details of the task. For instance, spam filtering has variants where the spam is discarded without the user being able to see it at any point and other where the spam is collected so that the user can check it. The first case requires very high confidence in the identification of spam: it is better to let through a considerable amount of spam rather than to reject one piece of real email (assuming the email address is used for important messages).

Another consideration is whether the behaviour of the system can be interpreted. It may be better to use a system with slightly inferior performance if it is possible to track the contribution of the different features since this allows a form of debugging. This is a strong reason to experiment with simple algorithms when initially investigating a task.

Evaluation is discussed in detail in several sessions in this course.

Standard tasks and datasets

A large proportion of research on ML is done using standardized tasks of various sorts. The task is defined, standard test and training data is used, and the evaluation is fixed. The researchers are only concerned with feature extraction and with the algorithm. In other cases, the feature sets are provided as well. This is very helpful for research on algorithms.