Introductory Applied Machine Learning – Course Outline

Eleanor Platt, David Sterratt, Pavlos Andreadis,Nigel Goddard

Dealing with Data - Preprocessing

- Attribute-value representation (including representing images)
- Differences between categorical, ordinal and numerical attributes

- Standardisation (normalisation) of data
- Differences between generative and discriminative classifiers

Generalisation & Evaluation

- Train on training data, choose model with validation, report performance on test
- N-fold cross validation is a method of selecting the model and hyperparameters
- There are many evaluation metrics to assess the performance of the model
- *Overfitting* occurs when model too complex & *underfitting* when model too simple

Optimisation & Regularisation

- Machine learning problems can be written as optimisation problems
- Problem could be to *maximise* the (log-)likelihood or *minimise* error/loss
- Algorithms like gradient descent can be used to iteratively update parameters, but can result in finding sub-optimal solutions, e.g. local optima if problem non-convex
- Regularisation (e.g. penalty to loss function) prevents extreme parameter values

Ethics: Stakeholders, Fairness, Accountability and Transparency

Supervised Learning

- Trying to predict a specific quantity or class
- Have training examples with *labels*

- Can measure the evaluation metric directly – we compare model’s predictions with the true labels

Classification

- Aim: to predict the class/label of a given data point
- Binary classification (2 classes) vs multi-class (>2 classes)
- Have notion of decision boundaries between classes
- Example: predicting whether an email is “spam” or “not spam”

Decision Trees

- Split on attributes until subset at leaf is pure or depth budget reached
- Some splits are more *informative* than others – use information gain (which uses entropy) to work out which attribute to split on
- To classify new point: trace down tree until a leaf node reached, predict class that is the majority class of training samples in subset

Naive Bayes

- Uses Naive Bayes assumption (naive = *conditional* independence & use Bayes’ theorem to compute P(c|x) from class models and prior)
- Naive Bayes works well with for incomplete data
- To classify a new point: compute the probabilities P(c|x) for each class c, and choose the class that maximises this probability

Logistic Regression

- Maximise the log-likelihood function
- Apply the logistic function to the output to squash to the range [0,1] for probability - output is probability of belonging to certain class
- To classify a new point: plug feature values into model and observe class probability – choose class with highest probability

Support Vector Machines

- Linearly separable versus non-separable data
- Find decision boundary which separates the hyperplane with the maximum margin – optimisation problem with analytic solution
- Support vectors are points that lie on the margins
- Kernel method: project data into high dim space to find linear DB
- To classify a new point: plug feature values into model, see which side of the decision boundary the point lies and predict this class

k Nearest Neighbours

- Do not ‘learn’ a model as such – there is no optimisation
- Choose value of k through hyperparameter tuning on validation set
- To classify a new point: compute distance from test point to every training example and predict the most common label amongst the k closest training instances
- Different methods for resolving ties

Neural Networks

Regression

- Aim: to predict a value of a particular data point
- Labels are real-value quantities (continuous variables)
- Loss function encapsulates how close the predictions are to the true values
- Example: predicting a person’s 5km run time based on their height and age

Decision (Regression) Trees

- Requires different definition of entropy to classification decision trees because there are no distinct classes
- To predict value of new point: trace down tree until leaf node reached, predicted output is the average of the training sample labels in subset

Linear Regression

- Fit a linear model to training data by optimising the weights and bias
- Model always linear in parameters – features can be transformed
- Can maximise log-likelihood or minimise MSE – these have same analytical solution for the optimal weights
- To predict value of new point: plug feature values into linear model

Support Vector Machines

- Underlying idea same as for linear regression, but use epsilon-insensitive error instead of MSE
- Vary similar to SVMs for classification: learn linear model (not a decision boundary) to maximise the margin
- Can still use kernel method for non-linear SVMs
- To predict value of new point: plug feature values into the model learned by the SVM to get the predicted output

k Nearest Neighbours

- Do not ‘learn’ a model as such – there is no optimisation
- Choose value of k through hyperparameter tuning on validation set
- To predict value of new point: compute distance from test point to every training example and output the average of the label values of the k closest training instances
- No notion of a ‘tie’ for regression

Neural Networks

Unsupervised Learning

- Do not have any labels associated with the data
- Trying to look for patterns and structure in the data instead of prediction

- No notion of ‘accuracy’ so evaluation is usually qualitative
- We consider how to group data-points and how to reduce their dimensions in IAML

Clustering

- Discover sub-populations / groups within the data
- How many distinct groups are there? What points belong to what group?

K-Means

- Initialise K cluster centres randomly in the data, assign each training sample to the nearest centre, recompute the centre of each cluster by taking the mean value of the points assigned to it, reassign points to the nearest new cluster, repeat until no changes…
- K-means minimises the intra-cluster distance
- Choose value of K through hyperparameter tuning on validation set

Hierarchical Clustering

Gaussian Mixture Models

Dimensionality Reduction

- Dimensionality of data = number of features measured (e.g. number of pixels or words) – subject to curse of dimensionality
- How to reduce the dimension of the data while retaining its important properties?

Principal Components Analysis

- Find a new (lower dimensional) set of axes to represent the data on
- 1st direction is the one of greatest variability in the data, 2nd is perpendicular to the first and of greatest variability, etc.
- Can find axes (principal components) by considering eigenvectors of the covariance matrix
- Reduce dimensionality by projecting data to principal components