Thursday – Machine Learning with Ed Morrissey

Kaggle – predictive modelling/competition – way to make money etc and also to learn

Scikit learn (python library) nicely structured, documented etc – makes machine learning more accessible

Training data – predictions (supervised is giving a training set – known I supposed)

Derive a set of rules from known dataset

Unsupervised – don’t know what you’re looking for, i.e clustering – dimensionality reduction etc.

Iris dataset – using this model to train a dataset to predict species ffrom a dataset of iris

Sklearn (load iris)

Classification – partition space into regions so if a point falls into one region you call it one thing. Kernel – different effct on how lines are drawn (radial is more circular) partitions the space differently

SVM is partitioning space.

Decisions trees (simple and popular) sequence of yes no decisions. Sequence of is this variable bigger than this – series of partitions. Ie. Is CD4 expression on ? If yes then – next. Set of rules, simplistic. Overfit – doesn’t predict well outside of training data

Sklearn.tree DecisionTreeClassifier

Clf = clf.fit(data, target) #create a classifier, derives a set of rules to decide rules between different types.

Clf.predict(data[:2, :]) predicts on first 2 lines

Overfitting refers to overtraining on the data that you’ve given it so can’t predict on new data

If you partition your data (test v train) can see if it can predict – see if it overfits

Train test split into a test size of 0.25 –

Fit only to training data then try to fit test data – predicition not as good as before.

Evaluate how much you’re overfitting.

Methods to minimise this (random forest) – using decisions trees in a particular way

Decision trees on subsets on data – bunch of trees randomly, don’t use all variables at one. Splitting it into chunks, forests of trees so use the chunks together to reduce overfitting

Clf\_forest.score

Scikit learn dataset – loads of different models – effect of different classifiers and their performance

How do you decide which to use?

Random forest is quick and good – split into training/test etc

Overlap with statistics a lot here

Next: Dimensionality reduction (unsupervised) – uncover strcture and plot it

1. PCA

You get as many PC out as you put in – orders them based on how much variance explained/amount of info

1. tSNE

Calculate distance between cells using gaussian – 2D arrangement that respects distances

1. UMAP

Similar to tSNE (faster) because of intiialise. Can give classes or add new data

Knn – K nearest neighbours – find k closest cells – join up to get an interconnected graph. Subtract expression take sum square them all to give Euclidean distance and find nearest cells.

How do you choose K (5) – if you have small distnt cell pops then better to use fewerk as otherwise will start bringin in cells that it shoulder

Spring/Seurat – force directed layouts. Paths in the data better reproduced

Louvain clustering/Phenograph – networks finds the clusters

Machine learning vs stats

Stats generally more rigorous

ML – good for big data, speedier, less interested in underlying data

ML easier to just have a go

Outliers (supervised) need training dataset that has typical data and variance – if not, then wont extrapolate well to more varied dataset

Deep neural networks

10x genomics – dataset pbmc -2017

Purified different cell populations (beads/antibodies)

10 different populations measured separately

Take these different populations and use them as classifiers

Read files, concatenate, plot, initialise different classfiers – fit/test overfit

Download new data