Machine Learning Tutorial

Dr. Michelle Lochner

In this tutorial you will use your knowledge of supernovae and machine learning to automatically classify types of supernovae based only on their photometric light curves. This is a simplified version of some work we did here: http://arxiv.org/abs/1603.00882v3. I've provided you with some skeleton code to get you started but you'll have to make your own decisions when doing the machine learning component. You'll need to work in pairs and I expect three files to be submitted: one PDF with your solutions, a jupyter notebook (also called an ipython notebook) with all your code and a text file with predictions for the test data (see below).

1 Setup

Retrieve the data from the github repository https://github.com/MichelleLochner/ml_tutorial by cloning or downloading the zip file. You will find several files and folders. One is a jupyter notebook you can use as a template for your tutorial solutions (`tutorial_template.ipynb`). You can run this from the command line with
`jupyter notebook tutorial_template.ipynb`.
Make sure the first cell executes without error (in other words, all libraries should import). If not, complain to Siphelo. The next step is to execute the cell that loads a single file into an Astropy table, that can then be manipulated by sncosmo.

2 Feature Extraction

In this section, you will convert raw photometric supernova light curve data into a handful of features that can be interpreted by a machine learning algorithm.

Question 1a: Briefly describe how sncosmo's `fit_lc` function works.

Question 1b: Briefly describe what each of the 5 parameters in the SALT2 model is.

Follow the code in the tutorial template notebook to see how sncosmo can be used to fit the SALT2 model to a light curve. Then, write your own code to fit SALT2 to all 1000 light curves in the training data folder. At the end you should have a numpy array of shape [1000, 5]. Save this to disk and make sure you don't rerun it every time you run the notebook! This part will take a long time (30-60 minutes depending on computer speed).

Question 2: Run `plot_lc` and make plots for data files 3, 4 and 5. Why do some of these fit badly?

Question 3: Plot histograms for the x1 and c parameters, colouring the Ia's and the non-Ia's with different colours. Do you notice any differences between the two types?

3 Machine Learning with Scikit-learn

You need to run 3 machine learning algorithms on your set of SALT2 features, at least ONE of which must NOT have been covered in the class (i.e. you may use a maximum of two of: k-nearest neighbours, neural networks and ensemble methods with decision trees).

We will be using the library `scikit-learn`, referred to as `sklearn` inside python. The scikit-learn website has a great deal of documentation and tutorials to get you started (this is a good start http://scikit-learn.org/stable/tutorial/basic/tutorial.html). Every machine learning classifier in scikit-learn is defined as a class that has several functions such as `fit` and `predict`. Because all classifiers behave the same, scikit-learn makes it easy to try out different classifiers.

The true labels of the supernovae are given in `training_labels.txt`, where the first column is the ID of the supernova, the second column is the type. See Table 1 for the codes. Using this and the SALT2 features you

Supernova type	Code
Ia	1
II	2
Ibc	3

Table 1: The encoding for supernova type used in the labels file

extracted in the previous section, run 3 different machine learning classifiers on the SALT2 features of the training data. Make sure you choose some subset of data to evaluate your algorithm on. Keep it aside and don't use it when training the algorithm (have a look at `train_test_split`).

Question 4: For each of the three algorithms you are using, answer the following:

(a) Does the algorithm you're using require the features to be rescaled (`http://scikit-learn.org/stable/modules/preprocessing.html`)? If so, what did you use to rescale them?

(b) What values of the hyperparameters of the algorithm did you use? How did you select them (consider investigating scikit-learn's `GridSearchCV` function)?

(c) What fraction of the data are you using to evaluate the algorithms?

4 Evaluating the Classification

Use the `predict_proba` function to compute the probability of belonging to each class for each object in your validation set. Evaluate the performance of each algorithm using `roc_curve` and compute the AUC (area-under-curve) for each case.

Question 5: On one figure, plot the ROC curve for each of your classifiers and include the name of the classifier and the calculated AUC in the legend.

5 Test data analysis

Unzip the test data folder. This contains another 1000 supernovae but this time, you don't have the labels for the data. Repeat the above procedure for feature extraction to end up with an array of SALT2 features for the test data (remember this will take a while, save these to disk). Choose your best classifier and the best values for its hyperparameters, and run it on this feature set (run `predict_proba`). Output the probabilities along with the id of each supernova to file. The file should be an ascii text file of 1000 rows (one for each supernova) and 4 columns: ID, Ia probability, II probability, Ibc probability.

6 Submitting your solutions

Send me an email with three attached files (make sure you name them with the surnames of everyone in your team):

1. A PDF with your answers, written using latex, including all the plots you made (with captions) and the written answers to the above questions.
2. A jupyter notebook with all your code
3. A text file with the probabilities for the unlabeled test data

I'll evaluate all your algorithms on the test data (since I have the answers) and give a prize to the team with the best performing algorithm!

References

http://scikit-learn.org/stable/
http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
http://arxiv.org/abs/1603.00882
https://www.coursera.org/learn/machine-learning
https://github.com/rasbt/python-machine-learning-book
https://github.com/jakevdp/sklearn_tutorial
http://ipython-books.github.io/featured-04/