global

University of Sussex
Department of Physics & Astronomy

Using Machine Learning Algorithms to Identify Fast and Slow Rotating Galaxies

Joshua Fenech

Submitted in part fulfilment of the requirements for the degree of
Physics with Astrophysics

Abstract

Text of the Abstract.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Objectives

Motivation and Objectives here.

1.2 Contributions

Contributions here.

1.3 Statement of Originality

Statement here.

Chapter 2

Background Theory

2.1 Introduction

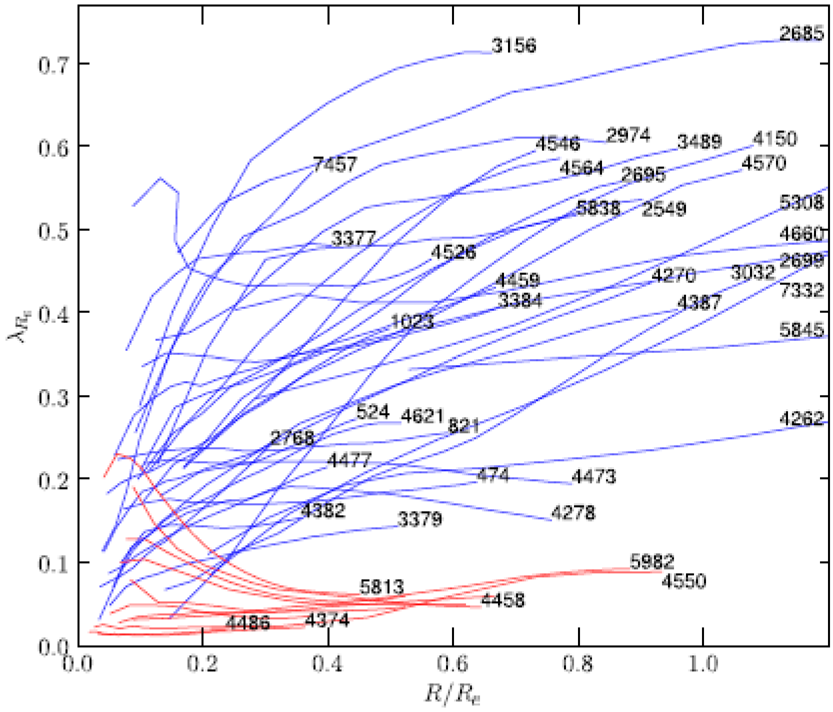
Galaxy morphology has traditionally been classified based on the Hubble sequence, originating from the identification galaxy features from photographic plates. However, this classification is based on visual distinctions and fails to accurately represent early-type galaxies (E's and S0's) and it was argued by [**Cappellari2011**] and [**Emsellem2011**] that a more telling classification would be based on the spin parameter due to the intrinsic qualitative change in velocity structure exhibited by galaxies, with a threshold of separating slow ($\lambda < 0.1$) and fast (≥ 0.1) rotators, as can be seen in figure 2.1. This threshold was later updated to include the ellipticity ϵ by defining slow rotators and fast rotators have λ_{Re} lower and larger than $k_{FS} \times \sqrt{\epsilon}$, respectively, where $k_{FS} = 0.31$ for measurements made within an effective radius R_e [**Emsellem2011**][p1]. This new criterion is nearly independent of viewing angle. λ_R is defined as[**sauron9**]:

$$\lambda_R = \frac{\sum_{i=1}^{N_p} F_i R_i |V_i|}{\sum_{i=1}^{N_p} F_i R_i \sqrt{V_i^2 + \sigma_i^2}} \tag{2.1}$$

where F_i, R_i, V_i and $_i$ are the flux, circular radius, velocity and velocity dispersion of the ith spatial bin, the sum running on the N_p bins. λ_{Re} indicates spin parameter calculated within 1 effective radius R_e. This is superior over a velocity dispersion classification, V/σ, which fails when confronted by galaxies with kinematically decouple cores (KDC), "whose angular momentum vector is misaligned with respect to that of the bulk of the galaxy" [**mo_bosch_white_2010**]. Furthermore, λ_R 'can quantify galaxy morphology via the kinematic properties of galaxies[**Cortese2016**][p1 beyond early types.

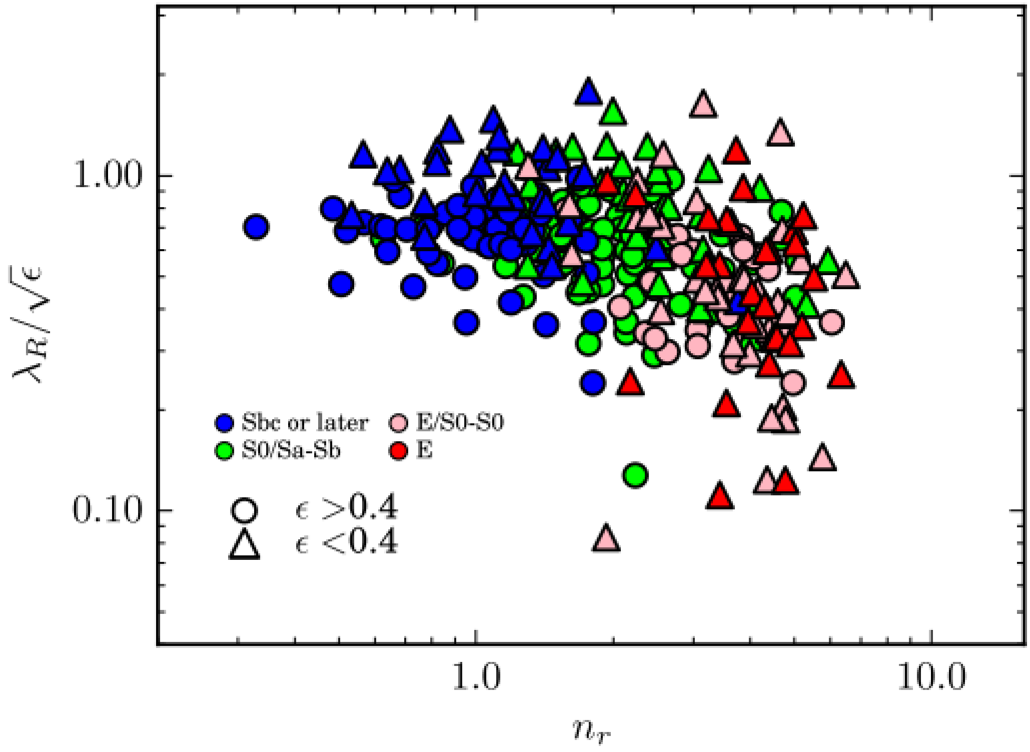
(Although this paper suggests that the correlation breaks down for early types, others...DON'T INCLUDE THIS MAYBE...).

Figure 2.1: Radial λ_R profiles for the 48 E and S0 galaxies of the SAURON sample. Profiles of slow and fast rotators are coloured in red and blue, respectively. NGC numbers are indicated for all fast rotators and most slow. rotators [**Emsellem2011**]



As can be seen in figure 2.2, there appears a weak correlation for galaxies in the SAMI survey for early-type galaxies, suggesting that based on seŕsic index alone there is some prospect for applying statistical methods to infer morphology. The spin parameter is costly to determine due to its reliance on integral field spectroscopy and has therefore only been found for a small number of galaxies: 260 from the ATLAS3D survey and 446 from SAMI. This compares with over 500 million galaxies with photometric data from the Sloan Digital Sky Survey (SDSS) alone [**SDSS**]. Although other classifications do not rely on this parameter, it is still of value in relation to other properties. It is therefore of great value to find an alternative means of

Figure 2.2: Galaxies from ATLAS$_{3D}$ colour coded by optical morphology. There appears a weak correlation for early types that grows more pronounced for late types (Sbc or later), using seŕsic index as a weak proxy for morphology. [**Cortese2016**][p12]



identifying the rotation. Although the motivation for this study is to extend identifying the spin parameter to all morphological classes, this paper will focus on the early-type galaxies taken from [**Emsellem2011**], defined as Hubble type E/S0's, T-type $T < 3.5$ (Es) and $T \geq 3.5$ (S0s).

2.1.1 Slow vs Fast Rotators

Differentiating between fast and slow rotators based on galaxy properties alone is difficult due to the lack of distinct boundaries for each characteristic. The parameters that will be considered that have some correlation with morphology are

GOING TO PUT GRAPHS OF PARAMETERS HERE

In [**Emsellem2008**], it was found that fast rotators tend to be relatively low luminosity with $M_B \lesssim -20.5$ and well aligned photometric and kinemetric axes, while slow rotators tend to be brighter and more massive galaxies, that exhibit either no rotation or KDC's.

In order to evaluate the rotation and its effect on observable parameters, it is necessary to consider the current understanding of galaxy morphology. The traditional means of classifying galaxies was based on the Hubble tuning fork that split galaxies into spiral and elliptical galaxies based on their visual morphological appearance. HUBBLE TUNING FORK IMAGE HERE There are several reasons for this distinction. Spirals are generally younger galaxies with a net angular momentum and hence more likely to form discs on a plane coincident with this. FUCKIN REFERENCE There have been a variety of methods of quantifying these classifications, but generally consists of identifying the different components of the galaxy, being the disk and the bulge, and their relative importance. Several (MORE DETAIL HERE) properties of galaxies correlate with their classification, and the subject of this part of the paper will be to briefly describe how this is so (IMPROVE THE ABOVE PARAGRAPH).

2.2 Measured Parameters

The parameters used in modelling the data were:

2.2.1 Variations of the Spin Parameter

The ATLAS3D paper measured λ to 1 effective radius, R_e and to half the effective radius, $R_e/2$, where

$$I(R_e = I_0/e) \tag{2.2}$$

whereas the SAMI paper only measured this for R_e[**Cortese2016**].

2.2.2 Sérsic Index of the Single Fit and Bulge Component, n and n_b

The Sérsic profile models the light intensity over the surface of the galaxy in terms of an exponential function as a function of the distance from the centre, R, and the Sérsic index n:

$$I(R) = I_e exp\{-b_n[(R/R_e)^{1/n} - 1]\} \tag{2.3}$$

The range of the Sérsic index covers the full range from steep (i.e. concetrated n»1) to shallow (n$\lesssim$ 1) surface brightness profiles Galaxies

$$\lambda_{Re} = (0.31 \pm 0.01) \tag{2.4}$$

2.3 ATLAS^{3D}

This survey combined According to [**Cappellari2011**] this survey focused on a 'volume-limited ($1.16 \times 10^5 Mpc^3$) sample of 260 early-type (elliptical E and lenticular S0) galaxies (ETGs)...The sample consists of nearby (D $<$ 42 Mpc, $|\delta - 29°| < 35°$, $|b| > 15°$) morphologically selected ETG's extracted from a parent sample of 871 galaxies (8 per cent E, 22 per cent S0 and 70 per cent spirals) brighter than $M_K < -21.5$mag (stellar mass $M_\star \gtrsim 6 \times 10^9 M_\odot$).' ETG's were defined as having de Vaucouleurs T type T $>$ -3.5 and T ≤ -3.5 for S0 and E galaxies respectively, which correlates with the Hubble classes lenticular and elliptical.

Chapter 3

Methods

3.1 The scikit-learn library

It was chosen to implement algorithms from the scikit-learn module of the python language since it "exposes a wide variety of machine learning algorithms, both supervised and unsupervised, using a consistent, task-oriented interface, thus enabling easy comparison of methods for a given application" [**Pedregosa2012**]. This allowed several different algorithms to be implemented within the same environment and gain meaningful results with minimal prior coding and analysis. The algorithms initially chosen was decision trees (DT's) since these allow both regression and classification analysis, and are known as 'white boxes' due to their relatively transparent process whereby the mechanics of training could be evaluated more readily.

3.2 Decision Trees

Different machine learning algorithms use different statistical tests in order to evaluate data and make inferences. DT's emulate a logical classification procedure similar to that used in biology to identify species. Starting from the full dataset a series of binary if-then tests are consequentially performed and data split into 2 branches continuously until a final statistical criterion is satisfied: the sklearn decision tree uses the gini impurity to evaluate the success. The algorithm does this by splitting the data into two branches by choosing the split that minimise the gini index, defined as:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{3.1}$$

arbitrarily choosing a value that splits the parameter space in 2, forming a node and 2 branches. The

Sklearn can use the Gini impurity or the entropy as the determining how to split the tree. The default method of Gini impurity was used here.

$$Q_{left}(\theta) = (x, y)|x_j <= t_m \tag{3.2}$$

$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta) \tag{3.3}$$

The impurity at m is computed using an impurity function H(), the choice of which depends on the task being solved (classification or regression)

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta)) \tag{3.4}$$

Select the parameters that minimises the impurity

$$\theta^* = \operatorname{argmin}_\theta G(Q, \theta) \tag{3.5}$$

Recurse for subsets $Q_{left}(\theta^*)$ and $Q_{right}(\theta^*)$ until the maximum allowable depth is reached, $N_m < \min_{samples}$ or $N_m = 1$. **[sklearn]**

The impurity measure used is the gini impurity:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{3.6}$$

This process aims to form 2 child sets that are more statistically correlated with each other than the parent. The backend of the SKlearn modules were however not evaluated directly but implemented naïvely. The parameters were initialised to their defaults as described in the documentation[**sklearn**]. Using the modules themselves required elementary use of python to pass the module arrays of necessary values. Implementation proceeded in 2 stages once the data had been suitably formatted. The data was arbitrarily split into 2 groups, a training and a test set, based solely on their position within the results: they were ordered according to their LEDA classification and occupied different locations of the sky, and so should not have any bias. For classification, the training set was passed to the module as a list with each item holding the feature value (i.e. Sérsic index) or a list of feature values if more than one feature used, and target variable (i.e. fast/slow rotator classification, spin parameter value). A function was output that incorporated the learned rules which was then applied to the test data set resulting in an array of predicted values that could be measured against the known values. Initially, the algorithm was run as a classification problem due to the more simple analysis of success and errors. This is because it allowed the success to be evaluated in a more elementary fashion, without recourse to analysing error distributions. Decision tree and random forest methods were then applied and the parameters adjusted manually to achieve optimal results. The parameters included:

3.2.1 Parameters

SKLearn allows several parameters to be adjusted manually in order to maximise the efficiency of the models, and these are particular to each. For decision trees, these are as follows:

Parameter	Options	Description
Criterion	Gini or Entropy	Uses the gini impurity as outlined above or entropy for information gain. The gini impurity default was used
Splitter	Best or Random	Chooses the best split or the best random split to make at each node based on the criterion.
max_features	int	Considers the maximum number of features of the dataset to consider when recursing for the best split.
max_depth	int or None	The maximum depth of the tree, i.e. how many splits the tree makes. The default of None will recurse until all leaves are pure, i.e. contain 1 class, or contain less than the min_samples_split.
min_samples_split	int,float (default=2)	The minimum number of samples required to be at a leaf node, where the leaf node represents the split into classifications.
max_leaf_nodes	int, none (default=None)	The maximum number of leaf nodes allowed, with the tree choosing nodes which best minimises the gini impurity.
min_weight_fraction_leaf	float or None	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node. Samples have equal weight when sample_weight is not provided. No weights were supplied since and so were equally weighted.
min_impurity_split	float (default=1e-7)	Threshold for early stopping in tree growth. A node will split if its impurity is above the threshold, otherwise it splitting will cease and the node forms

3.3 Data & Formatting

3.3.1 ATLAS^{3D}

Spectroscopic data (namely Sérsic index of the single fit and bulge component, n and n_b respectively, and D/T, the Disk-to-Total light ratio) was extracted from [**Krajnovic2013**] whilst the kinematic parameters λ_{Re}, $\lambda_{Re/2}$ and the Fast/Slow rotation classification were extracted from [**Emsellem2011**]. The data from both sources was combined using a pandas dataframe. The classifier was first trained using the spin parameter λ_{Re} with the FS rotation classification as target variable as a test measure, which successfully predicted 100% of the test set. The algorithm was then trained using n and D/T individually and simultaneously.

Chapter 4

Results

4.1 ATLAS^{3D}

Initially, the algorithm was trained using the λ_{Re} to predict the FS classification since they were related directly according to refeq:2.4 (NEED TO FIX REFERENCING EQUATIONS). Promisingly, the classifier was 100% successful in its predictions. Training was then performed using the Seŕsic index of the single fit, n:

Chapter 5

Conclusion

5.1 Summary of Thesis Achievements

Summary.

5.2 Applications

Applications.

5.3 Future Work

Future Work.