Star Galaxy Separation: Documentation

V. Belokurov, N.W. Evans, S. Koposov, L.Smith

March 4, 2019

1 Introduction

A critical component of photometric surveys like the Sloan Digital Sky Survey (SDSS) and the Large Synoptic Survey Telescope (LSST) is star/galaxy separation. At faint magnitudes, the separation between point-like and extended sources is fuzzy, which makes star/galaxy separation an awkward task. This problem is even harder for enormous surveys like the LSST due to their huge data volume.

There are a host of scientific problems that require efficient star/galaxy separation. In precision cosmology, measurements of primordial non-Gaussianities require pure samples of galaxies. Biasing can be mimicked by systematic effects adding power on large scales – for example by contamination from stars in the Milky Way Galaxy. Similarly, in galactic astronomy, the study of faint, low surface brightness tidal features around nearby galaxies requires pure samples of stars, uncontaminated by small, unresolved galaxies. Hence, the search for classification methods that are both accurate and efficient is highly relevant to the science goals of LSST. We would like to be able to convert the number of LSST epochs into a limiting depth at which we experience say 10 per cent or 20 per cent contamination, so as to build samples of known purity and completeness.

In SDSS, the photometric pipeline performs a morphological star/galaxy separation, using a diagnostic like the difference between the psfMag (obtained by fitiing PSF model) and cmodelMag (obtained from best-fitting exponential or de Vaucouleurs profile). The quality of this separation is obviously related to the seeing and sky brightness.

Automated classification methods have also been used to solve star/galaxy separation at faint magnitudes. For example, in Sextractor (Bertin & Anouts 1996), star/galaxy separation is achieved on most images using a neural network trained with simulated images. Recent years have seen increasing efforts in applying machine learning methods as distinct as artificial neural networks (e.g., Support Vector Machines, Random Forests)

Ideally, a star/galaxy classifier should have the following three desirables: (i) Calibrated star/galaxy probabilities that match the actual frequencies, (ii) the ability to include priors (i.e. priors on colour/magnitude, on right ascension and declination), (iii) the ability to deal with repeated exposures of variable quality. Here, we describe a novel approach that fulfills these requirements.

2 A New Algorithm and Its Implementation

We introduce the following terminology. The data are denoted by $I_{i,f}$, which is the i-th image observed of s source in a given filter f and by $\hat{I}_{i,f}$, which is the normalised i-th image – i.e. the image with the information about absolute fluxes erased. The latter only carries morphological information.

The PSF on the i–the image is given by $PSF_{i,f}(x,y)$, while the magnitude vector m_f is model magnitude vector from the stack. The goal is to provide $P(\mathrm{star}|\{\hat{I}_{i,f}\},\alpha,\delta,\{m_f\})$. The correct way to compute this is via the equation

$$\frac{P(\{\hat{I}_{i,f}\}|\mathrm{star},\alpha,\delta,\{m_f\})P(\mathrm{star}|\alpha,\delta,\{m_f\})}{P(\{\hat{I}_{i,f}\}|\mathrm{star},\alpha,\delta,\{m_f\})P(\mathrm{star}|\alpha,\delta,\{m_f\}+P(\{\hat{I}_{i,f}\}|\mathrm{gal},\alpha,\delta,\{m_f\})P(\mathrm{gal}|\alpha,\delta,\{m_f\})}$$

The calculation of this requires a number of ingredients. First, we need the likelihood of the data for star and galaxy, given the broad band photometry and position, namely $P(\{\hat{I}_{i,f}\}|\mathrm{star},\alpha,\delta,\{m_f\})$ and $P(\{\hat{I}_{i,f}\}|\mathrm{galaxy},\alpha,\delta,\{m_f\})$. Then, we need the priors, that is the probability of of being a

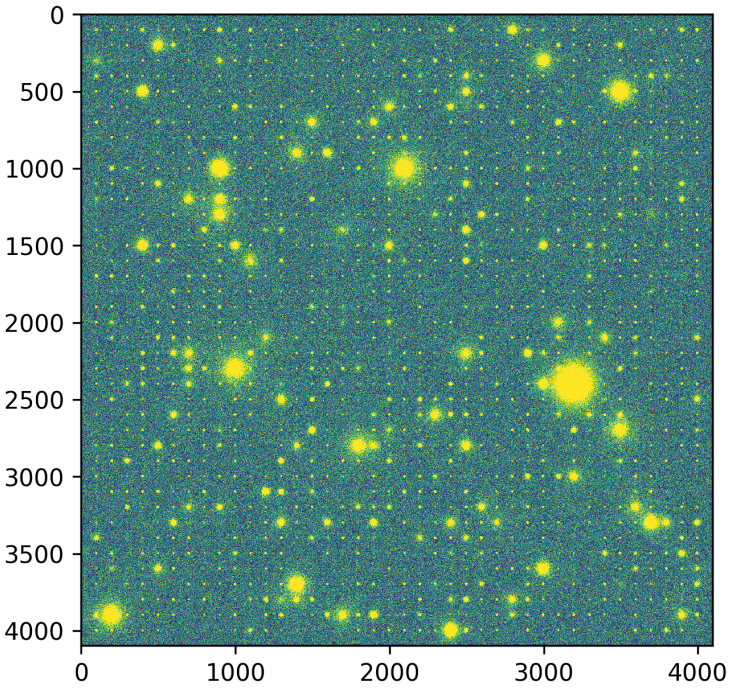

Figure 1: Sample generated image including stellar and galaxy profiles confolved with a psf and background, Poisson and readout noise.

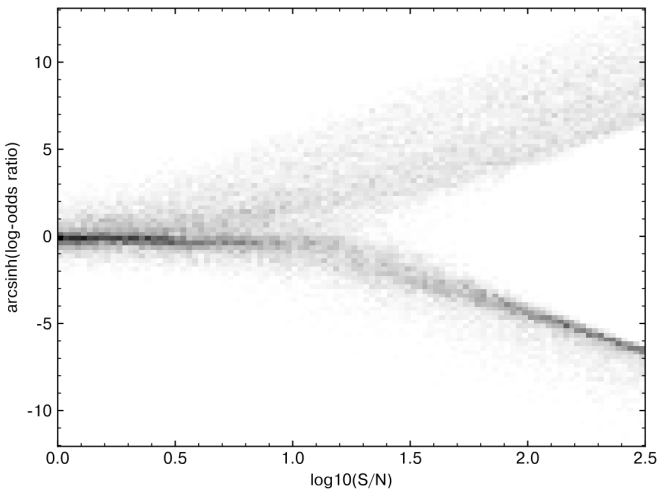

Figure 2: The log of the ratio of star/galaxy odds as a function of source signal to noise. We are able to reasonably reliably distinguish stars from galaxies even at S/N < 10.

star or galaxy conditional on position and broad band (model magnitudes) $P(\text{star}|\alpha, \delta, \{m_f\})$ and $P(\text{gal}|\alpha, \delta, \{m_f\})$

The key ingredients are the likelihoods, The likelihood of the stellar model is trivial, namely $P(\{\hat{I}_{i,f}\}|\text{star}, \alpha, \delta, \{m_f\})$ if we know the PSFs of each frame. Then it will be essentially just a sum of χ-squares of the best fits for all the images/bands that we have (if we do not marginalize over position). Also, the conditionality on the position and magnitude vector is irrelevant for stars.

For galaxies, matters are more complicated. The problem is that the parameter space of surface brightness models for galaxies is vast. We assume that the galaxy profile can be fitted by some analytical model with parameters ϕ_f (the parameter is a function of the filter band f). For the purpose of this discussion, the parameters could be Sersic index, Sersic size and the center of the object. If we have this analytical parametrisation of the light profiles, we can evaluate the actual likelihood of the object being a galaxy

$$P(\{\hat{I}_{i,f}\}|\text{gal}, \alpha, \delta, \{m_f\}) = \int P(\{\hat{I}_{i,f}\}|\text{gal}, \alpha, \delta, \{m_f\}, \{\phi_f\}) P(\{\phi_f\}|\text{gal}, \alpha, \delta, \{m_f\}) d\{\phi_f\} \quad (1)$$

For this, we have to marginalise over all possible values of galactic shape parameters. The fundamental ingredient of this equation is the likelihood of the data given specific values of ϕ_f

$$P(\{\hat{I}_{i,f}\}|\text{gal}, \alpha, \delta, \{m_f\}, \{\phi_f\})$$

This is easily computable as this is essentially a likelihood of the data given specific analytic surface brightness profile (convolved with the PSF).

The harder portion is the prior $P(\{\phi_f\}|\alpha, \delta, \{m_f\})$ which tells us the probability distribution over possible surface brightness profiles given position on the sky and broad band magnitude vector. It is clear that this probability should not depend on position but only on the broad band vector $P(\{\phi_f\}|\{m_f\})$ This is the hyper-prior that we need.. i.e. it tells us what's the expected distribution over Sersic sizes and Sersic indices given the broad band magnitude and color. This needs to be obtained from the actual data.

Also to perform the integration in Eq. (1), we can sum over a grid of models. That is, we have a grid of models with different Sersic indices and sizes, and we fit each one to the given set images of a source. That gives us the likelihoods. Then, we need it to combine it with our prior on the Sersic sizes and indices, given the magnitude and color of the source and then sum that to get total likelihood that the source is a galaxy.

The final ingredients that are left are the priors $P(\text{star}|\alpha, \delta, \{m_f\})$ and $P(\text{gal}|\alpha, \delta, \{m_f\})$ on star or galaxy, given positions and broad band magnitudes. These are essentially the stellar-locus/galactic model and galaxy luminosity function/galaxy color-locus priors. These either can be data driven or theoretically motivated. Although this analysis could be in principle done by completely ignoring the information about the $\{m_f\}$ magnitude vector. In this case, the analysis is purely morphological. It is likely to be very suboptimal, and probably some conditioning at least on a flux in a single band is worthwhile i.e. r.

Code has been written in C to quickly perform the galaxy/point source fitting on realistic simulated image cutouts given a model grid.

Images are generated in Python with user defined star/galaxy ratio, pixel scale, source density and image size. Point sources and Sersic galaxy models are included, convolved with PSFs and the images include a realistic background, Poisson and readout noise (see Figure 1). Source identification is then performed by SExtractor, and from this image cutouts are produced using the SExtractor segmentation map. Ultimately source lists, psf models and ideally image cutouts will be provided by the LSST pipeline, so the performance of the above stage, barring the realism of the images, is of little importance.

Image cutouts are fed into the main model fitting code, which is written in a mixture of C and Python. The code then uses a 20x20 grid of pre-generated galaxy models (spanning a range of Sersic indices and galaxy sizes) convolved with the PSF to evaluate likelihoods of each source given each galaxy model. Combined with the priors on galaxy shapes this provides a star/galaxy probability for each source. The required computation time for 64x64 pixels cutouts is 10 CPU ms per object.

The code is able to reliably recover the input star/galaxy fraction to within a few percent, even in cases of fairly noisy data, as illustrated in Figure 2. Furthermore the code is able to approximately recover the hyper-prior on the galaxy light profile parameter distribution directly from the grid of evaluated model likelihoods.

The codes, including Jupyter notebooks for demonstration, are available at https://github.com/lsmithast/stargalaxy_hierarchical.

3 Summary

This document contains a new proposal for a Star-Galaxy classifier. It makes explicit the working of the algorithm, as well providing the set of priors used. The GitHub repository contains a working code, together with notebooks that assess its performance.

One clear missing component of the code as it currently stands is the treatment of extended source ellipticity. The extension of the grid of galaxy models to flattened models, which require two additional shape parameters: ellipticity and position angle that need to be marginalized away. Models with mildly elliptical profiles have been tested and the results are promising, although more work is needed.

Another clear potential problem is blended objects. Our code currently includes no treatment of object blending, but we propose to use the models for all the sources as provided by LSST DM stack to subtract away the blends and then separately model each object of interest.