Star-Galaxy Separation in the Era of Precision Cosmology

Michael Baumer, Noah Kurinsky, and Max Zimet
Stanford University, Department of Physics, Stanford, CA 94305

November 14, 2014

Introduction

To anyone who has seen the beautiful images of the intricate structure in nearby galaxies, distinguishing between stars and galaxies might seem like an easy problem. However, modern cosmological surveys such as the Dark Energy Survey (DES) [3] are primarily interested in observing as many *distant* galaxies as possible, as these provide the most useful data for constraining cosmology (the history of structure formation in the universe). However, at such vast distances, both stars and galaxies begin to look like low-resolution point sources, making it difficult to isolate a sample of galaxies (containing interesting cosmological information) from intervening dim stars in our own galaxy.

This challenge, known as "star-galaxy separation" is a crucial step in any cosmological survey. Performing this classification more accurately can greatly improve the precision of scientific insights extracted from massive galaxy surveys like DES. The most widely-used methods for star-galaxy separation include `class_star`, which is a legacy classifier with a limited feature set, and `spread_model`, a linear discriminant based on weighted object size [1]. The performance of both of these methods is insufficient for the demands of modern precision cosmology, which is why the application of machine learning techniques to this problem has attracted recent interest in the cosmology community [6]. Since data on the real-world performance of such methods has yet to be published in the literature, we will use these two classifiers as benchmarks for assessing the performance of our models in this paper.

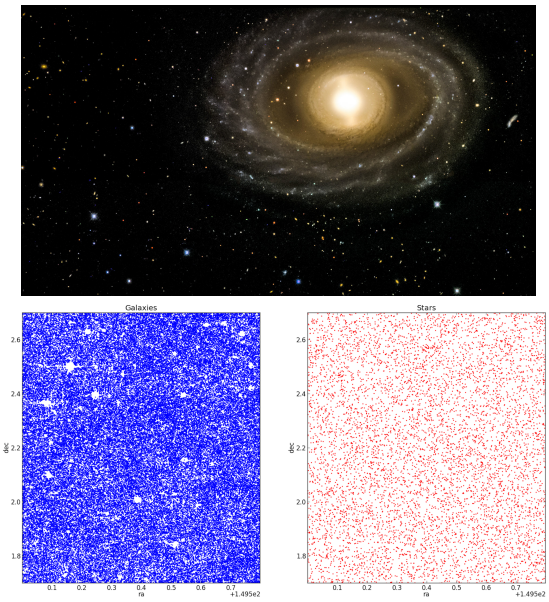
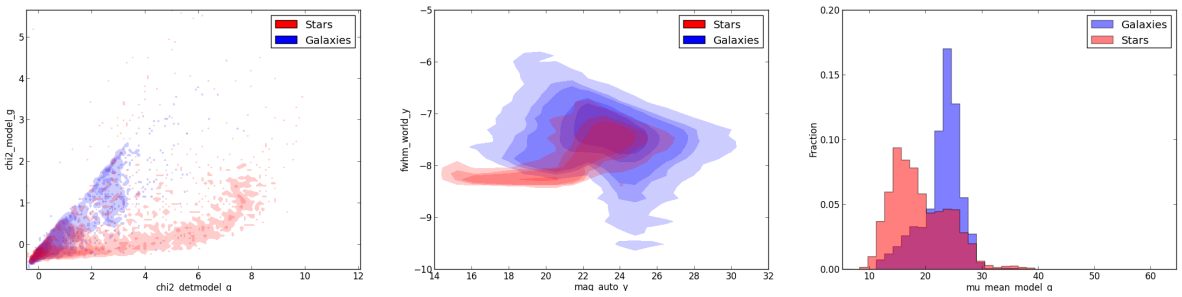


Figure 1: In the top false-color image from DES, it is easy to tell that the large object in the foreground, NCG 1398, is a galaxy, but what about all the point sources behind it? In the bottom images of our actual data, we see sky maps of stars (right) and galaxies (left). The voids seen in the map of galaxies are regions of the sky which are blocked by bright nearby stars.

Data and Feature Selection

We use a matched catalog of objects observed by both DES, a ground-based observatory, and the Hubble Space Telescope (HST) [4] over 1 square degree of the sky. The true classifications (star or galaxy) are binary labels computed from the more-detailed HST data. The catalog contains 222,000 sources, of which 70% (∼150,000) are used for training and 30% (∼70,000) are (randomly) reserved for cross-validation.

From the 226 variables available in the catalog, we selected 45 variables which appeared to have discriminating power based on plots like those shown in Figure 2. Features include the surface brightnesses, astronomical magnitudes, and sizes of all objects as observed through 5 different color filters (near ultraviolet to near infrared). We also use goodness-of-fit statistics for stellar and galactic luminosity profiles



(a) $\log \chi^2$ for a star-like luminosity profile vs. a galaxy-like profile.

(b) $\log$(size) (full-width at half-maximum) vs. infrared magnitude

(c) Histograms of mean surface brightness

Figure 2: A selection of catalog variables (before preprocessing) selected as input features to our models.

(which describe the fall-off of brightness from centers of objects).

As a preprocessing step, we transformed all variables to have a mean of 0 and variance of 1. For certain variables - namely chi-squares and object sizes - we decided to use their logarithm rather than their raw value, to increase their discriminating power and make each feature more Gaussian (see panels (a) and (b) in Figure 2).

In performing feature selection, it was important to us that features were included in physically-motivated groups. After making the plot of learning vs. included features shown in Figure 3, we implemented a backwards search, using a linear SVM, to determine the relative power of each of the variables. We were surprised to find that our less promising parameters (χ^2 and surface brightness) were often more powerful than magnitudes in certain color bands, and considered removing some magnitudes from our features. We ultimately decided against this, however, given that we wanted to select features in a uniform manner for all color bands to maintain a physically-motivated feature set.

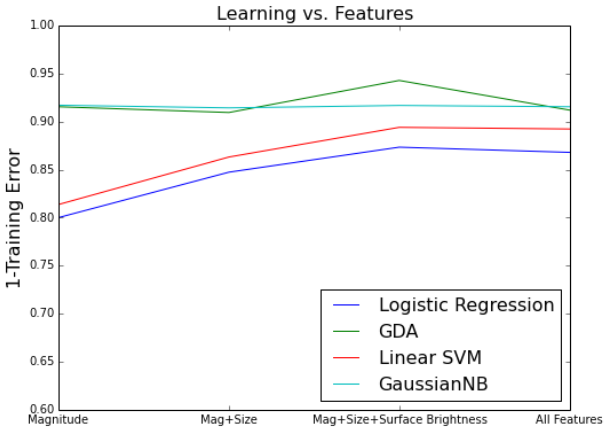


Figure 3: A plot of training success vs. features used for four of our models.

Method	$1 - \hat{\epsilon}_{train}$	$1 - \hat{\epsilon}_{test}$
GDA + SMOTE	66.8%	91.4%
GNB + SMOTE	70.5%	91.4%
LR	86.8%	86.4%
LinSVM	89.4%	89.0%
GaussianSVM	95.4%	94.2%

Table 1: Training and test error for multiple supervised learning methods.

Methods and Results

To determine which method would obtain the best discrimination, we initially ran a handful of basic algorithms with default parameters and compared results between them. Given that we had continuous inputs and a binary output, we employed logistic regression (LR), Gaussian discriminant analysis (GDA), linear- and Gaussian-kernel SVMs, and Gaussian Naive Bayes (GNB) as classifiers. We implemented our models in Python, using the `sklearn` module [5].

We opted to employ the ℓ^1 regularized linear SVM classifier `LinearSVC` (ℓ^2 regularization gave comparable results), the Naive Bayes algorithm `GaussianNB`, the logistic regression method `SGDClassifier(loss="log")`, and the GDA algorithm, using class weighting where implemented (namely, for SVM and LR) to compensate for the much larger number of galaxies than stars in our training sample. The results of these initial classi-

Method	GDA + SMOTE		GNB + SMOTE		LR		LinSVM		GaussianSVM	
	G	S	G	S	G	S	G	S	G	S
True G	95.5%	4.5%	95.0%	5.0%	87.3%	12.7%	90.2%	9.8%	95.2%	4.8%
True S	59.8%	40.2 %	52.3%	47.7 %	19.7%	80.3%	19.8%	80.2%	17.8%	82.2%

Table 2: Confusion matrices produced on our test set for different supervised learning techniques, showing the true/false positive rates for galaxies in the top row and the true/false negative rates for stars in the bottom row of each entry.

fications can be seen in table 1, and the confusion matrices from these runs can be seen in table 2.

The statistics in Table 1 show that we have sufficient training data for the LR, GDA, and SVM methods, since our test error is very similar to our training error. We decided to proceed further from here with Naive Bayes/GDA and SVM, in order to continue our search for a successful generative model and optimize our best discriminative algorithm.

Gaussian Naive Bayes and GDA with Bootstrapping and SMOTE We first describe the generative algorithms we applied. Eq. (1) shows the probability distribution assumption for the Gaussian Naive Bayes classifier, where x_i is the value of the i-th feature and y is the class (star or galaxy) of the object under consideration.

$$p(x|y) = \prod_{i=1}^{n} p(x_i|y), \quad x_i|y \sim \mathcal{N}(\mu_{i,y}, \sigma^2_{i,y}) \qquad (1)$$

This makes stronger assumptions than the quadratic Gaussian Discriminant Analysis model, whose assumed probability distributions are described in Eq. (2):

$$x|y \sim \mathcal{N}(\mu_y, \Sigma_y). \qquad (2)$$

In fact, this shows that the GNB model is equivalent to the quadratic GDA model, if we require in the latter that the covariance matrices Σ_y be diagonal.

To correct for the effects of our imbalanced data set, we applied the bootstrapping technique, whereby we sampled from our set of galaxies, and trained on this sample, plus the set of all of our stars; the intent was to take 50 such samples and average the results of these different classifiers. However, each individual classifier still performed poorly, misclassifying most stars, unless we heavily undersampled from our galaxies (that is, we chose far fewer galaxies than stars to be in our new data set), in which case the classifiers misclassified most galaxies.

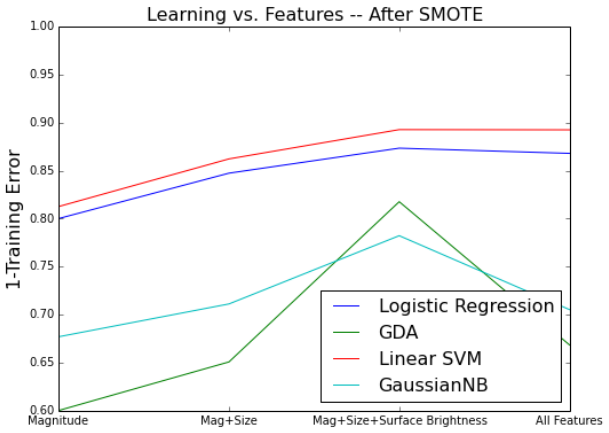


Figure 4: Training success vs. features used for data with equal numbers of stars and galaxies, produced using the SMOTE algorithm.

We also performed a synthetic oversampling of stars using the Synthetic Minority Over-sampling TechniquE (SMOTE) algorithm [2]. The idea of SMOTE is to create new training examples which look like the stars in our data set. Specifically, given a training example with features x that is a star, we choose, uniformly at random, one of the 5 nearest stars in the training set (where "nearest" means we use the Euclidean ℓ^2 norm on the space of our normalized features). Denoting this chosen neighbor by x', we then create a new training example at random location along the line segment connecting x and x'.

After oversampling our population of stars using the SMOTE algorithm to have equal numbers of stars and galaxies, the training errors for Gaussian Naive Bayes and GDA suffer, as they are no longer able to succeed by classifying almost everything as a galaxy. Given that these generative models fail to classify a balanced dataset well, we conclude that our data does not satisfy their respective assumptions sufficiently to warrant their use. This is not very surprising, as generative learning models make significant assumptions. As we will see next, the discriminative algorithms we

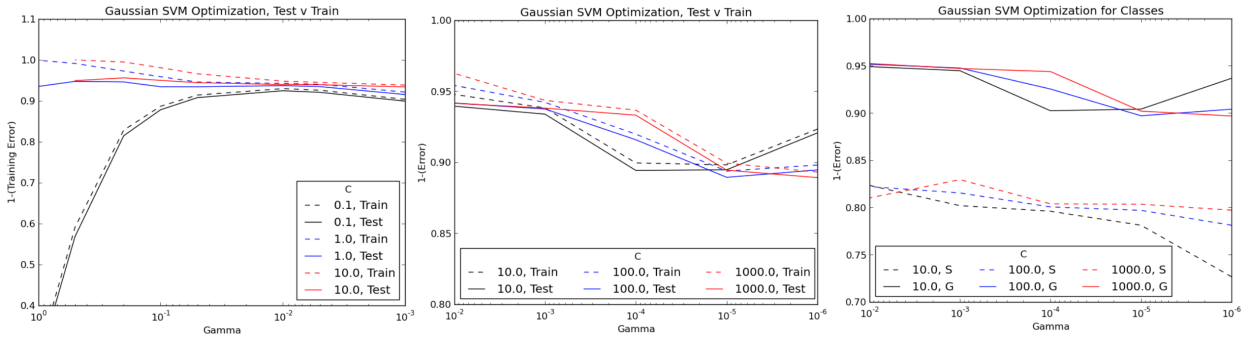


Figure 5: Results of two rounds of Gaussian SVM optimizations, varying the parameters C and γ logarithmically. The leftmost plot shows $C \in \{0.1, 10.0\}$ and $\gamma \in \{0, 10^{-3}\}$, while the middle and right plots show $C \in \{10.0, 1000.0\}$ and $\gamma \in \{0.01, 10^{-5}\}$. The left two plots show training versus test success percentages, while the rightmost plot compares success for stars and galaxies over the range of our better performing grid search.

used performed much better.

Grid-Search Optimized SVM with Gaussian Kernel, L1-Regularization Given the good performance of the Linear SVM and a preliminary run of the SVM with Gaussian Kernel, we decided to try to optimize the ℓ^1-regularized SVM by performing a grid search across the parameter space $C \in \{10^{-3}, 10^{3}\}$ and $\gamma \in \{10^{-5}, 1\}$ spaced logarithmically in powers of 10[1]. We employed the `SVC` method of `sklearn`, a python wrapper for `libSVM`, which provides automatic class weighting to counteract our unbalanced training sample. We used the barley batch server to run these grid samples in parallel, and had `SVC` record the training error, test error, and confusion matrices for each model. The resulting scores from these optimizations can be seen in Figure 5.

Our model choice was based on of the requirement for lowest generalization error, highest test success, and best separation for stars, as a classifier which classified every point as a galaxy would have a good overall test performance but perform no source separation, and thus perform poorly on stars. In Figure 5, this was performed functionally by selecting, first, the model with the highest test success, then among those the models closest to their corresponding training success rates, to minimize overfitting. Then, among this subset of models, we chose the model which performed best separating stars while not compromising galaxy separation performance.

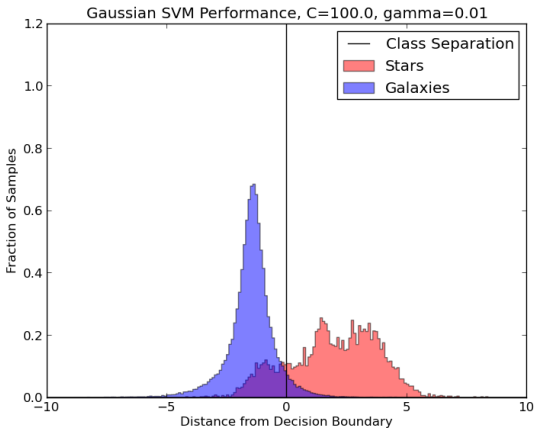


Figure 6: Histogram of the margin from the optimal separation plane (as determined by our Gaussian SVM) of each training example; a negative margin corresponds to a galaxy, positive to a star.

The sum total of these considerations leads us to select the model with $C = 100.0$, $\gamma = 0.01$, which has the confusion matrix shown in table 2 and an overall test success rate of 94.2%. Out of all methods tested, this SVM had the highest generalized success rate and best performance separating stars, at 82% success. The distance of each example from the separating hyperplane of the SVM can be seen plotted as a histogram separately for stars and galaxies in Figure

[1] At the poster session, our TAs suggested importance sampling or random sampling as alternatives to a grid search. Given the average run time of between 1 to 10 hours per optimization, and the fact that importance sampling is an iterative process, we opted for grid search to reduce computing time. It can be seen in our figures that our optimization is fairly convex, so the grid search appears to be sufficiently accurate. See also csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf.

6. The factor which most limits the star separation performance seems to be the multimodal distribution of stars in our feature space, one of which is highly overlapping with galaxies. It is interesting to note the highly gaussian distribution of galaxy margins from the plane, indicating that these were better modeled, and overall had more continuous properties.

Discussion

Since we determined that the generative models we attempted to use were not appropriate for our dataset, we are left with three successful discriminative models: logistic regression, Linear SVM, and Gaussian SVM. We used a Receiver Operating Characteristic (ROC) curve to characterize the performance of our benchmark models, `class_star` and `spread_model`, since they both give continuous output scores. Since our three models all give binary outputs, they appear as points on the ROC curve, as shown in Figure 7. We see that all three of our models significantly outperform the two benchmarks!

As noted in the previous section, the multimodal distribution of stars was the limiting factor in our ability to separate these object classes. Regardless of the model, we rarely saw a "true star rate" greater than about 80%, suggesting this class of stars comprises on average 20% of observed objects, and may be worth modeling. Given the high dimensionality of our features, it was not obvious what feature may have set these stars apart, but it is clear that they are confounded with similar galaxies, though galaxies do not show similar substructure.

Conclusions and Future Work

We have shown that machine learning techniques are remarkably successful in addressing the challenges of star-galaxy separation for modern cosmology. Though the assumptions of our generative models–GDA and Naive Bayes–were not borne out in the data, causing them to perform poorly, we had success with logistic regression and SVMs, and the largest challenge was in feature standardization and SVM optimization. Our best model, the Gaussian SVM, achieved very good performance, classifying 95.2% of true galaxies correctly, while achieving 82.2% accuracy in classifying true stars, surpassing both of our benchmark classifiers.

The distribution of stars near the SVM decision boundary indicates that there might be a class of stars that is not properly modeled by our current feature set. This motivates the future work of seeking out other astronomical catalogs with more or different features to enable better modeling of our stellar population. In addition, though we chose to focus on algorithms discussed in this course, deep learning also has great potential for improving star-galaxy separation. Such algorithms are the focus of some bleeding-edge cosmology research [6], though their performance on current-generation survey data has yet to be published.

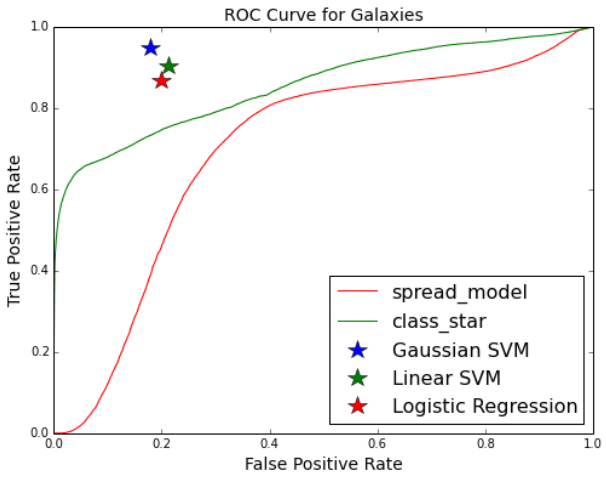


Figure 7: An ROC curve for our best three supervised learning methods compared to benchmark classifiers from the literature.

References

[1] Bertin & Arnouts (1996). SExtractor: Software for Source Extraction. *A&AS*: 117, 393.

[2] Chawla et al (2002). SMOTE: Synthetic Minority Over-sampling Technique. *JAIR*: 16, pp. 321-357.

[3] DES Collaboration (2005). The Dark Energy Survey. arXiv:astro-ph/0510346.

[4] Koekemoer et al (2007). The COSMOS Survey: Hubble Space Telescope Advanced Camera for Surveys Observations and Data Processing. *ApJS*: 172, 196.

[5] Pedregosa et al (2011). Scikit-learn: Machine Learning in Python. *JMLR*: 12, pp. 2825-2830.

[6] Soumagnac et al (2013). Star/galaxy separation at faint magnitudes: Application to a simulated Dark Energy Survey. arXiv:1306.5236