# Predicting Galaxy Morphology Classifications

Kairui Jin
Department of Electronic Engineering and Computer Science, University of Michigan
Ann Arbor, USA
krjin@umich.edu

***Abstract*—I implement several DSP technique: FFT, Wiener Filter, convolution to preprocess the galaxy images and use some machine learning theories to learn and predict hand-made galaxy morphology classifications. My data and success metric are taken from a recent Kaggle competition "Galaxy Zoo - The Galaxy Challenge". I focus on three sequential algorithm modules: image analysis, data compression, and machine learning. I find that my approach performs with moderate success, though not as accurately as the approaches used by top competitors.**

***Keywords—galaxy, wiener filter, decision tree, linear regression, PCA***

## I. INTRODUCTION

The goal of my project is to predict the morphology classifications of galaxy image data, as outlined by the recent Kaggle competition Galaxy Zoo - The Galaxy Challenge. As stated on the competition website:

"With each passing day telescopes around and above the Earth capture more and more images of distant galaxies. As better and bigger telescopes continue to collect these images, the datasets begin to explode in size. In order to better understand how the different shapes (or morphologies) of galaxies relate to the physics that create them, such images need to be sorted and classified. Galaxies in this [data] set have already been classified once through the help of hundreds of thousands of volunteers, who collectively classified the shapes of these images by eye in a successful citizen science crowdsourcing project. However, this approach becomes less feasible as data sets grow to contain of hundreds of millions (or even billions) of galaxies. This competition asks you to analyze the JPG images of galaxies to find automated metrics that reproduce the probability distributions derived from human classifications. For each galaxy, determine the probability that it belongs in a particular class."

In particular, I seek to learn from labeled galaxy images in order to predict the probability distribution over morphology classifications that best models the distributions derived from crowd sourced classifications. I note that each crowd sourced classification is actually a path in a decision tree and, thus, the problem is equivalent to determining the probability distribution of expert decisions at each branch in the decision tree given the galaxy image data.

The paper is organized as follows: Section 2 I present the background and previous literatures related. Section 3 describes the different modules and classification method. The experimental results and analysis are present in Section 4. In section 5, some conclusions are drawn.

## II. BACKGROUND

My research is a continuation of the results of the Galaxy Zoo project [1]. This project compiled the crowd sourced classifications of 1 million galaxy images drawn from the Sloan Digital Sky Survey. The goal of our project is to learn from these hand-made classifications and automate the classification of galaxy images using machine learning techniques.

The literature contains numerous approaches to the galaxy classification problem. For example, Storrie-Lombardi et al. [2], were able to successfully separate galaxies into several classes using a feed-forward neural network. They reported 64% classification accuracy. Owens, Griffiths, and Ratnatunga [3] utilized oblique decision trees to perform classification on the same dataset. They report an overall accuracy of 63% using 5-fold cross-validation. Bazell and Aha [4] achieved 78.55% accuracy using ensembles of classifiers for 800 galaxies. Naive-Bayes classifier, neural network, and decision tree induction algorithms were used to divide galaxies into 6 classes. 14 features were considered for each galaxy.

An alternative to morphological classification is to introduce the Gini coefficient, as demonstrated by Abraham et al. [5]. Madgwick [6] investigated two statistical techniques to determine how accurately morphology can be estimated from the optical spectrum of galaxy. He reported 70% accuracy for 'early' galaxies and 83% accuracy for 'late' galaxies. Here, 'early' and 'late' refer to broad classes of galaxy morphologies.

In this paper, a method for performing automated morphological galaxy classification is present based on digital signal processing and computer vision techniques and machine learning algorithms. I first preprocess the image data by

cropping out the background and filter images, then use principal component analysis to further reduce the dimensionality. Then I apply linear regression on the projection coefficients of the principal components to predict the distribution of morphology classifications.

## III. PROPOSED APPROACH

The method of classification I present consists of three modules: Preprocessing Module, Data Compression Module and Regression Module. The method works as follows. It takes as input the galaxy images that are cropped, centered and rotated in the Preprocessing module. Then in the Data Compression Module, I reduce the dimension of the data based on the principal component analysis. The projection of the images onto the principal components gives the data input parameters for the regression module. Lastly, I construct the decision tree with 11 benchmarks. For each benchmark, four different regression algorithms are implemented. The next three sections would describe the details of these three modules.

### *A. Image Analysis*

In this step, I retrieve the raw images from the dataset and then transform in such a way that we discard a great deal of non-essential data. I reduce the dimensionality of the data by cropping out background pixels which are not a part of the subject galaxy (recall that the images provided in the dataset have been centered on the target galaxy so we may consider only the pixels near the center of the image to classify the images). I also convert RGB values to grayscale. This reduced the input dimensions by a factor of 1/3. Afterwards, I remove the noise by adaptive filtering. I choose wiener filter to an image adaptively, tailoring itself to the local image variance. This approach often produces better results than linear filtering. The adaptive filter is more selective than a comparable linear filter, preserving edges and other high-frequency parts of an image. The wiener filter could also handle all preliminary computations and implements the filter for an input image. Through comparing to other filter, I figure out that wiener filter works best when the noise is constant-power ("white") additive noise, such as Gaussian noise.

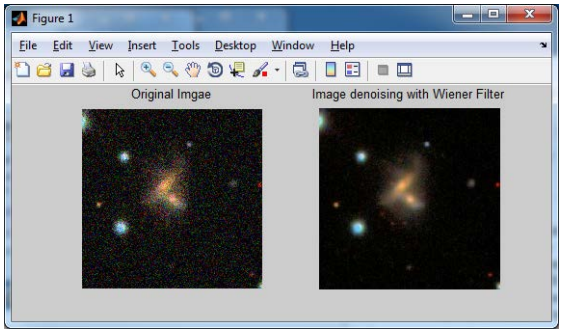


Figure 1: Image Denoising operation

After removing the noise, we were able to further reduce the dimensionality of the data by cropping out background pixels that were not part of the subject galaxy. The subjects in each galaxy image had been centered prior to distribution for the competition and so we were then able to bypass the task of centering each subject. We found, through visual inspection of several dozen images, that the salient features of each galaxy were typically contained within the central 200_200 pixels (of 424_424 pixels total). Thus, to ignore irrelevant data in the images, I cropped out those pixels which fell outside of the central 200_200 bound. I then scaled the cropped images down to 40_40 pixels. The choice here was mainly a technical one, as memory constraints prevented us from allowing the scaled images to be much larger while still holding many of these images in memory.

In order to compute the desired rotation angles, we took the following approach. First, we scaled the 424×424 images down to 80×80. This step was taken to reduce the computational complexity of image rotation. Second, we generated an 6400 × 2 matrix, X, of 2-dimensional offsets corresponding to the offset of each pixel from the center of the image. We then created a 6400×1 weight vector, W, where

$$W_i = \frac{\exp\left(-(X_{i,1}^2 + X_{i,2}^2)/200\right)}{\sqrt{X_{i,1}^2 + X_{i,2}^2}}$$

Note the resemblance of this weight to the normal distribution. Our choice of 200 for the denominator in the exponent implies that, all other factors equal, we expect about 95% of the weight to lie within 40 pixels from the center. The choice to consider the central 40×40 pixels corresponds, roughly, to our choice to maintain the central 200×200 pixels in the non-normalized case (compare 80×80 with 424×424). The factor in the denominator simply corrects for vector length, such that distance from the center does not affect weight beyond the Gaussian factor in the numerator. We then generate the weighted vectors, $\widehat{X_i}$, where

$$\widehat{X_i} = X_i * W_i$$

We may then take the first eigenvector v, of $\widehat{X_i}$ to determine the direction of greatest variance which, in turn, corresponds to the major axis of the galaxy. We may then rotate the image by -arctan(v1/v2) to align the major axis with the x-axis, as desired. We then crop out those pixels which fall outside of the central 40×40 bound. Again, this roughly corresponds to the bounds observed for the non-normalized case. Examples of color desaturation and image rotation and scaling can be seen in Figure 2.

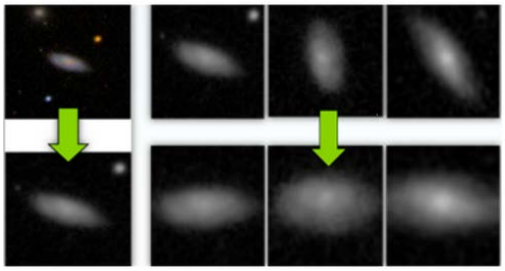

Figure 2: Image rotation normalization

### B. *Data Compression*

Once the raw image data has been transformed, as outlined above, we vectorize the resulting training images and apply principal component analysis to further reduce the dimensionality of the data. In particular, we compute the mean over input vectors, compute the covariance matrix of the mean-subtracted vectors, and then retrieve the eigenvectors of the covariance matrix which have the greatest eigenvalues. Figure 3 below shows the drop-off for the first 50 eigenvalues on the training set. We see that maintaining the first 20 eigenvectors captures approximately 99% of the information and so we decided to maintain only the mean vector and the first 20 eigenvectors.

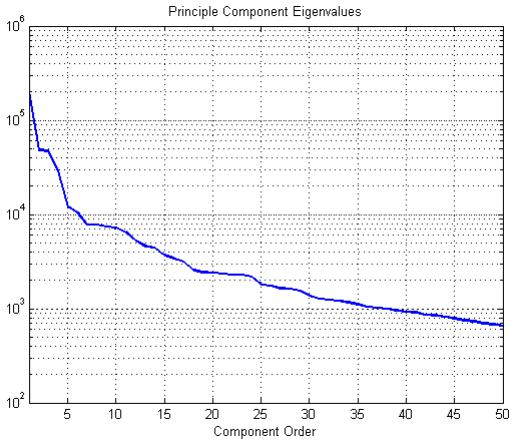


Figure 3. Eigenvalues of principal components.

### C. *Machine Learning*

Having transformed the original images into 40×40 images and then into 1600-dimensional vectors, we may subtract the mean computed in the data compression module and then project the centered vectors onto the 20 principal components. In this way, we transform each image into a vector of 20 projection coefficient which we may use, along with the labeled solutions, to train and evaluate the machine learning technique.

We apply this technique to each task separately. The tasks, including the number of responses for each, are listed in Table 1.

| Task | Question | Responses | Next |
|---|---|---|---|
| 01 | *Is the galaxy simply smooth and rounded, with no sign of a disk?* | smooth<br>features or disk<br>star or artifact | 07<br>02<br>**end** |
| 02 | *Could this be a disk viewed edge-on?* | yes<br>no | 09<br>03 |
| 03 | *Is there a sign of a bar feature through the centre of the galaxy?* | yes<br>no | 04<br>04 |
| 04 | *Is there any sign of a spiral arm pattern?* | yes<br>no | 10<br>05 |
| 05 | *How prominent is the central bulge, compared with the rest of the galaxy?* | no bulge<br>just noticeable<br>obvious<br>dominant | 06<br>06<br>06<br>06 |
| 06 | *Is there anything odd?* | yes<br>no | 08<br>**end** |
| 07 | *How rounded is it?* | completely round<br>in between<br>cigar-shaped | 06<br>06<br>06 |
| 08 | *Is the odd feature a ring, or is the galaxy disturbed or irregular?* | ring<br>lens or arc<br>disturbed<br>irregular<br>other<br>merger<br>dust lane | **end**<br>**end**<br>**end**<br>**end**<br>**end**<br>**end**<br>**end** |
| 09 | *Does the galaxy have a bulge at its centre? If so, what shape?* | rounded<br>boxy<br>no bulge | 06<br>06<br>06 |
| 10 | *How tightly wound do the spiral arms appear?* | tight<br>medium<br>loose | 11<br>11<br>11 |
| 11 | *How many spiral arms are there?* | 1<br>2<br>3<br>4<br>more than four<br>can't tell | 05<br>05<br>05<br>05<br>05<br>05 |

Table 1. The decision tree, comprising 11 tasks and 37 responses.[8]

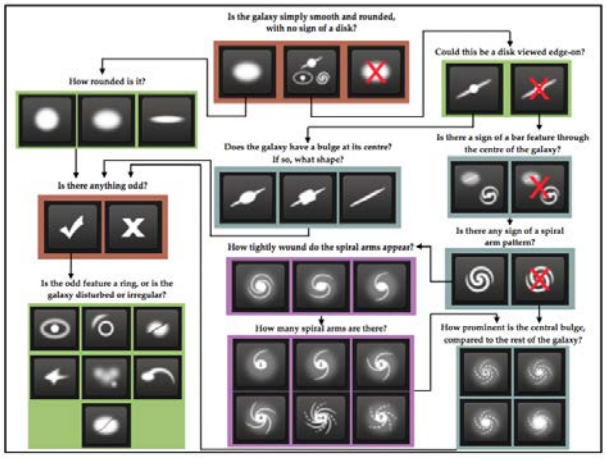


Figure 4. Flowchart of the classification tasks, beginning at the top center. [8]

## IV. EXPERIMENTS, RESULTS AND ANALYSIS

### A. Data

Our data is obtained from Kaggle's website [8]. It contains a collection of 61578 galaxy images along with a file containing solution vectors for each image. We have artificially split this data set into a training set, validation set, and test set, with a 50/25/25 share, respectively. We note that the images within the data set have been centered on the subject galaxies and so detecting galaxies within these images in not part of the problem scope. The solution vectors for each image have been taken as the sample average of manual galaxy classifications. Each sample is the result of a specific path down a decision tree, described in Table 1 and shown in Figure 1 below, taken directly from the Kaggle competition web page[9]. Each galaxy has been classified by multiple individuals, resulting in multiple paths along the decision tree. These paths generate probabilities for reaching each node. For example, if 25% of respondents chose response 1 of task 1, then those respondents would follow the tree to task 7. If 50% of respondents then chose response 1 of task 7, then the entry in the solution vector for task 7.1 would be 25% * 50% = 12.5%. The marginal distribution at each branch can easily be computed by normalizing over all responses for that branch. This method of cumulatively multiplying probabilities applies for every task branch, as mapped by the figure 1. The exception to this process is task 6, where responses 6.1 and 6.2 have been normalized to sum to 1. Since task 1 is at the top of the decision tree, the sum of probabilities over responses will sum to 1. For the remaining tasks, the sum of probabilities over responses will typically sum to less than 1.

### B. Experiments and Results

The performance of each algorithm is based on the root mean squared error (RMSE) of the difference between the solution matrix and the predicted solutions. That is, for N observations and L target vector dimensions, we compute

$$RMSE = \sqrt{\frac{1}{NL}\sum_{i=1}^{N}\sum_{j=1}^{L}(Y_{i,j} - \widehat{Y_{i,j}})^2}$$

This matches the evaluation criteria of the Kaggle competition. For linear regression, we trained the weights once with non-rotated pre-processed training images and once with rotated pre-processed training images. In the non-rotated case, we achieved an RMSE of 0.1541 over the validation set. In the rotated case, we achieved an RMSE of 0.1437.

### C. Analysis

We found that wiener filter and rotation normalization during the image analysis module greatly improved performance. The mean and first 8 eigenvector images for the non-normalized and normalized settings are shown in Figure 4 and Figure 5, respectively. Looking at the qualitative properties of these images, we may infer two things. First, it appears that our initial choice to focus on the central 200×200 pixels was a good one, since the salient regions of the images seem to end at or near the boundary of the image. Second, we can see why linear regression might perform poorly in the non-rotated case. Consider "PC2" and "PC3" in Figure 5. We might expect the projections of an elliptical galaxy image onto these eigenvectors to be sinusoidal as a function of rotation angle of the image. Thus, a pure linear model might not be able to capture such a relationship.

I believe that, overall, my methods were moderately successful. I compare to a simple benchmark and the winning solution from the competition. The so-called 'central pixel' benchmark simply trains a linear model on the RGB values of the center pixel of the training image set. This simple approach manages to achieve an RMSE of 0.16235. The winning solution [12], using convolutional neural networks and a number of other techniques was able to achieve an RMSE of 0.07567. It is notable that the best-known solution and the simple central pixel benchmark differ in RMSE by less than 0.1 (corresponding to 10%). If our best benchmark solution of 0.1289 is accurate for the full test set, then our solution ranks in the 55th percentile, hence our appraisal of moderate success.

## V. CONCLUSION

Between our own results and the results of the Kaggle competition, we found that neural network techniques performed best for this problem. Given more time and better understanding of how to implement the algorithms in such a way as to avoid excessive memory requirements, we would like to explore more neural network architectures and train/test on the complete data set. Competition winners also avoided the loss compression techniques of our image analysis module. It would be worthwhile to explore the performance gains by retaining image color and resolution. Given the information content of the 20 principal components, we might be able to approach optimal accuracy while simultaneously reducing computational cost. The initial motivation for the Kaggle competition was to automate the process of galaxy classification. It would be interesting to research a confidence-based model, as opposed to attempting to match crowd-sourced distribution. Such a system might employ one of our comparatively fast techniques to generate an initial assignment and confidence level. Low confidence levels may trigger the algorithm to delegate the galaxy image to a more sophisticated and computationally-intensive algorithm to increase precision. Such a design would certainly be practical for processing data sets with billions of galaxy images.

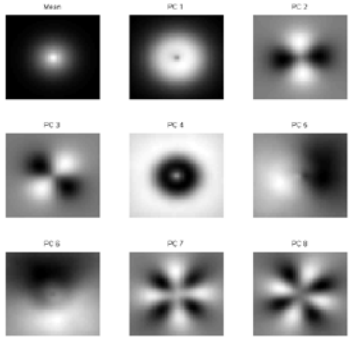

Figure 5. Non-rotated mean and eigenvalues

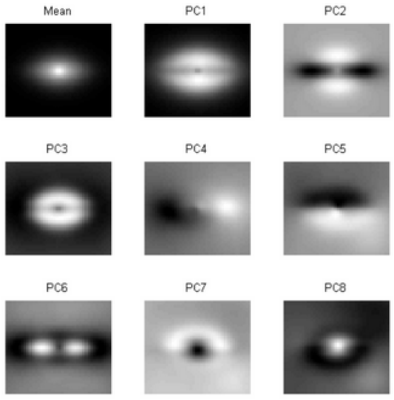


Figure 6: Rotated mean and eigenvalues

## REFERENCES

[1] Buta R. J., 2013, Galaxy Morphology, Oswalt T. D., Keel W. C., eds., Springer

[2] Storrie-Lombardi, M.C., Lahav, O., Sodre, L., Storrie-Lombardi, L.J. Morphological Classification of Galaxies by Arti_cial Neural Networks. Monthly Notices of the Royal Astronomical Society, 259(8), 1992

[3] Owens, E.A., Gri_ths, R.E., Ratnatunga K.U. Using Oblique Decision Trees for the Morphological Classi_cation of Galaxies. Monthly Notices of the Royal Astronomical Society, 281(153), 1996

[4] Bazell, D., Aha, D.W. Ensembles of Classi_ers for Morphological Galaxy Classi_cation. The Astrophysical Journal, 548:219-233, 2001

[5] Abraham R. G., van den Bergh S., Nair P. A new approach to Galaxy Morphology, 2003, ApJ, 588, 218

[6] Madgwick, D.S. Correlating galaxy morphologies and spectra in the 2dF Galaxy Redshift Survey. Monthly Notices of the Royal Astronomical Society, 338:197-207, 2003

[7] de la Calleja, J., Fuentes, O., Machine learning and image analysis for morphological galaxy classi_cation. Monthly Notices of the Royal Astronomical Society, 349(87), 2004

[8] Kyle W. Willett, Galaxy Zoo 2: detailed morphological classifications for 304,122 galaxies from the Sloan Digital Sky Survey, August 2013.

[9] Kaggle.com, http://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/data, April 7, 2014

[10] Kaggle.com, http://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge/details/the-galaxy- zoo-decision-tree, April 29, 2014

[11] Stanford Unsupervised Feature Learning and Deep Learning Wiki, http://udl.stanford.edu/wiki/index.php/Softmax Regression, April 29, 2014

[12] Multinomial Logistic Regression, http://en.wikipedia.org/wiki/Multinomial logistic regression, April 29, 2014

[13] Sedielem's Competition Solution, http://benanne.github.io/2014/04/05/galaxy-zoo.html, April 29, 2014