Draft version May 19, 2016
Preprint typeset using LaTeX style AASTeX6 v. 1.0

GALAXY ZOO EXPRESS: INTEGRATING HUMAN AND MACHINE INTELLIGENCE IN MORPHOLOGY CLASSIFICATION TASKS

Melanie Beck, Claudia Scarlata, Lucy Fortson
Minnesota Institute for Astrophysics, University of Minnesota, Minneapolis, MN 55454

Chris Lintott
Department of Physics, University of Oxford, Oxford OX1 3RH

ABSTRACT

Abstract goes here.

Keywords: editorials, notices — miscellaneous — catalogs — surveys

1. INTRODUCTION

In the context of Computer Science and/or Astrophysics?

Galaxy evolution. We want to study all the things and sometimes you need some morphologies to really probe how dat galaxy be living its life. So you gotta get dem morphologies. Bolster this with a couple of examples of major insights which have stemmed directly from morphological studies. Make morphology fucking important!
Once the audience believes it's important AF, inform them on the various ways we acquire said morphologies along with pros and cons.

1. Old Skool style: Visual classification (Edwin Hubble, graduate students, Galaxy Zoo)

2. Automated methods: Simplistic methods like Gini-M20, C-A, MID, etc. Cite all the papers (Lotz, Concelise, Freeman, etc.).

3. Automated methods (part 2): Sophisticated methods like machine learning shits (Huertas-Company, Deilman, etc.)

Now that the audience has been briefed on collection methods, explain to them why NONE of them will work when Euclid, WFIRST, and LSST go online:

1. visual classifications: too many to visually classify

2. auto1: accuracy isn't great and you only know morphologies in a statistical sense; can' t do fine detail

3. auto2: need really large training sets; can't apply neural net trained on SDSS to LSST (resolution / pixel size / other image issues)

NOW the audience understands the full problem. Outline our solution for a HYBRID method utilizing both human eyes for the detail work and machines for the broad/bulk classification. Put in the context of Galaxy Zoo (but explain overall method should be modular enough to be used with other formats for visual classification)

With all that said, start the paper! Section blah will be the components of the method. Section blah will be detail about post-processing visual classifications. Section blah will be about the machine algorithm. Section blah will be testing the method in various circumstances. Section blah will be results. Section blah will be Discussion/Conclusions. What sections do we want?

Outline 1:
overview of method the "collaborative filter"
post-processing of human classification (SWAP)
machine classification algorithm
simulation
describe GZ2 classification data describe how the simulation(s) were run – case studies (i.e. smooth/not)
simulation results?

2. OVERVIEW OF THE METHOD?

Figure 1 shows a simple schematic of a single collaborative 'filter' which demostrates the flow of classification data in our hybrid system. A human classifier is shown an image pulled randomly from a pool of images

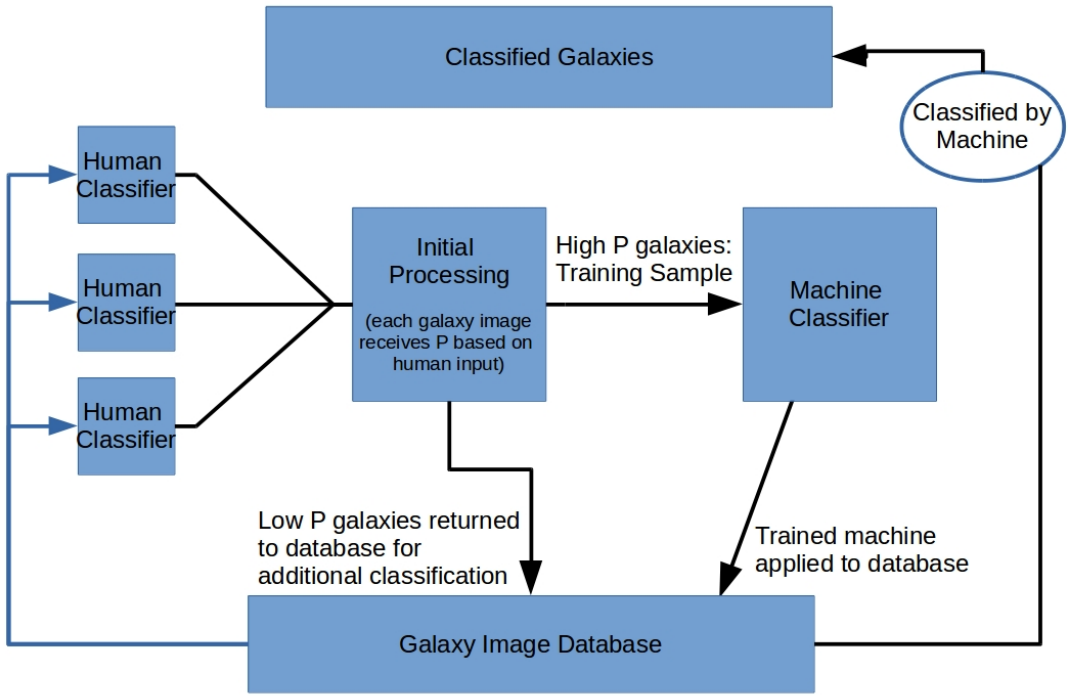

Figure 1. Schematic of our hybrid system. Human classifiers are shown images of galaxies via the Galaxy Zoo web interface. These classifications are recorded and processed according to section XXX. As a result of the processing, those subjects whose probabilties cross the classification thresholds are passed to the machine classifier as a training sample. The trained machine is then applied to the remaining subjects in the database (test sample). Those subjects which the machine classifies with high confidence are removed from the sample and considered fully classified. The rest remain in the database to be seen by human classifiers.

to be classified. She provides her input in the form of a simple yes/noanswer to a *single* question. Simultaneously, several other human classifiers are responding to their randomly assigned galaxy image. These binary responses are then used to both assess the quality of the human classifier's vote as well as the probability of their assigned galaxy to contain the characterstic of choice. If any galaxy obtains a probability that crosses either the upper threshold (thus having a very high probability of exhibiting the feature of interest) or the lower threshold (thus having a very high probability of NOT exhibiting the feature of interest) it is then passed to the machine classifier. Those galaxy images which do not cross either threshold are sent back to the pool to be further examined by human eyes. Those that move on to the machine classifer become a well defined training sample on which the machine learns. The fully trained machine is then applied to the remaining galaxy images in the pool. Those that the machine classifies with high confidence are set aside; the rest remain in the pool for additional scrutiny by human classifiers. In practice we have only run a single collaborative filter at a time, however these could be run in series or parallel; each with its own machine classifier. We now examine each stage of the process in detail.

Add a part about skipping sections because there are going to be ALOT of details.

3. GALAXY ZOO 2 CLASSIFICATION DATA

Our simulations utilize original classifications made by volunteers during the GZ2 project. These data are described in detail in (Willett et al. 2013) though we provide a brief overview here. The GZ2 subject sample was designed to consist of the brightest 25% (r band magnitude < 17) of resolved galaxies residing in the SDSS North Galactic Cap region from Data Release 7 and included both subjects with spectroscopic and photometric redshifts out to $z < 0.25$. In total, 285,962 subjects were classified in the GZ2 Main Sample catalogs (reference website?). Of these, 243,500 have spectroscopic redshifts while 42,462 have only photometric redshifts.

Subjects were shown as color composite images via a web-based interface wherein volunteers answered a series of questions pertaining to the morphology of the subject. In terms of GZ2, a *classification* is defined as the

total amount of information about a subject obtained by completing all tasks in the decision tree. A *task* represents a segment of the tree consisting of a *question* and possible *responses*. With the exception of the first task, subsequent tasks were dependent on volunteer responses from the previous task creating the decision tree as shown in Fig **??**. In total, the data consist of over 14 million classifications from 83,943 individual volunteers.

Our first simulated run considers only the first task in the decision tree: 'Is the galaxy simply smooth and rounded, with no sign of a disk?', to which possible responses include 'smooth', 'feature or disk', and 'star or artifact'. Because all volunteers see the first task, our simulations are run with as many as 14,144,941 classifications. These classifications consist of volunteer ID, subject ID, timestamp of the classification, and the volunteer's vote.

4. POST-PROCESSING OF HUMAN CLASSIFICATIONS

In this section we compare and contrast volunteer classification post-processing methods between standard GZ projects and what is implement on our simulations. In order to reach concensus for all tasks, Galaxy Zoo decision trees require a large number of independent classifications for each subject, i.e. approximately forty individual volunteers classify each subject in every Galaxy Zoo project including GZ2, Galaxy Zoo: Hubble (**?**), and Galaxy Zoo: CANDELS (**?**). Once a project reaches completion, GZ team scientists down-weight inconsistent and unreliable volunteers while the vast majority of volunteers are treated equally and no volunteers are upweighted. While this process reduces input from malicious users and 'bots' from contributing to the classification, it doesn't reward consistent and correct volunteers. Additionally, waiting until project completion doesn't allow for efficient utilization of super-users, those volunteers who are exceptional at classification tasks. [Do I need to cite something here?]

Instead, GZ:EXPRESS employs software adapted from the Space Warps Zoonivere project (Marshall et al. 2016) which searched for and successfully found several gravitational lens candidates in the CFHTLS survey. Dubbed SWAP (Space Warps Analysis Pipeline), the software predicted the probability that an image contained a gravitational lens given volunteers' classifations as well as their past experience. While full details found in Marshall et al. (2016), however, we briefly outline the method here.

The software assigns each volunteer an *agent* which interpret's that volunteer's classifications. Each agent assigns a 2 by 2 confusion matrix to their volunteer which encodes that volunteer's probability of correctly identifying feature 'A' given that the subject actually exhibits feature A and the data (i.e., the volunteer's classification history). The confusion matrix also encodes that volunteer's probabiliy of correctly identifying the absense of feature A given that the subject does not exhibit feature A and the data. **wtf won't bmatrix work?!**. The agent updates these probabilities every time a volunteer makes a classification by estimating those probabilities as

$$P(\text{“}X\text{”}|X,d) \approx \frac{N_{\text{“}X\text{”}}}{N_X} \tag{1}$$

where "X" is the classification made by the volunteer and X denotes the true classification of the subject. Each subject begins with a prior probability that it exhibits feature A: $P(A) = p_0$. When a volunteer makes a classification C, Bayes' Theorem is used to derive how the agent should update the subject's prior probability into a posterior:

$$P(A|C) = \frac{P(C|A)P(A)}{P(C|A)P(A) + P(C|NOT)P(NOT)} \tag{2}$$

where this value can then be calculated using the elements of the agent's confusion matrix. Marshall et al. (2016) show that perfect volunteers (i.e., those with $P(\text{“}A\text{”}|A) = 1.0$ and $P(\text{“}NOT\text{”}|NOT) = 1.0$ would calculate the posterior probability of the subject to be 1.0 which is not surprising (perfect classifiers are perfect!). However, they also show that *obtuse* classifiers (those with $P(\text{“}A\text{”}|A) = 0.0$ and $P(\text{“}NOT\text{”}|NOT) = 0.0$ also produce a posterior probability of 1.0; demonstrating that obtuse volunteers are just as helpful as perfect volunteers.

As the project progresses, each subject moves from its prior probability and is nudged to higher or lower probability depending on volunteer classifications. Eventually most subjects cross a "classification" threshold, i.e. they have either been *accepted* or *rejected* as having the FoI. At this point, the software retires the subject from being seen by volunteers. We now describe how this software is used in practice, incorporated into GZ:EXPRESS, and bridged with various machine classifiers.

4.1. *XXX in the simulation*

In order to use Marshall et al. (2016) software we make a few modifications including generalizing which features the software recognizes (not just gravitational lenses), as well as mimicking "real" time by allowing the software to run on a regular timestep. For our particular simulation we choose a $\Delta t = 1$ day. At each timestep, the software pulls all volunteer classifications between that and the previous timestep. For each vote, SWAP assigns and/or updates that volunteers's agent's confusion matrix as well as the probability that image

contains the feature of interest (FoI).

Before the simulation can be run, a number of SWAP parameters must first be chosen, including the initial confusion matrix assigned to each volunteer, the classification thresholds for the subject, and the prior probability for the FoI.

Each user's vote contributes some fraction to the probability that a galaxy image contains a characteristic of choice. The prior probability for all galaxies is determined by an educated guess for the relative frequency of that characteristic (elliptical galaxies comprise approx. 30% of the local universe). As users vote yes/no, this probability is updated; decreasing if the user votes nay and increasing if yay. An important aspect of this software is that care must be taken when choosing the classification thresholds. The original SWAP set the rejection and acceptanace thresholds equidistant in log-space because their prior was significantly small to begin with (lenses are expected to be very rare; the probability for finding one in a random sky image was estimated at p = 2e-4). Spiral arms or elliptical galaxies are not rare galaxy characteristics; thus determining proper thresholds for retiring an image from the pool must considered lots and I actually haven't done that whatsoever. I just picked some numbers that seemed pretty sweet.

As we show in Fig XXX, the value of the thresholds significantly changes the number of subject images which then move on to the next stage.

Anyway. In SWAP they were called "rejection" and "acceptance" thresholds. Here we interpret these differently – we aren't rejecting anything, simply acknowledging that it's probability strongly suggests that it does not have the characteristic of interest. Thus, we instead consider these "classification" thresholds. Any subject image which crosses one of these thresholds is then passed on to the next stage.

5. MACHINE CLASSIFIER

Supervised learning is the machine learning task of inference from labeled traning data. The training data consist of a set of training examples, consisting of an input vector and a desired output (or label). In general, a supervised learning algorithm analyzes the training data and produces an inferred function that can then mapped to new examples. An optimized algorithm will correctly determine class labels for unseen data. In the case of a collaborative filter, the task requires designation of an image to either have a feature of interest or not and is thus a discrete, binary task. *can I dig up any references which show support for binary tasks being easier for machines to learn?* As such, there are a wide variety of algorithms which are suitable and we discuss our initial choices later on.

5.1. *Training and Validation Samples*

Any supervised learning algorithm requires some form of input and associated label upon which the classifier will attempt to learn the appropriate characteristics which identify a subject as, in our case, either having the feature of interest or not. In order to learn a model which will generalize well to unseen examples, one should provide the largest, most representative training sample possible. In our case, however, this is not immediately possible as the size of the training sample is highly dependent on the rate of human classifications. During the initial stages of the project, human classifiers have not yet provided adequate information to generate sufficiently large training examples. The size of the training sample contains exactly zero subjects at the first timestep but continually grows as human classifications are processed. With a timestep of T and the SWAP parameters XXX discussed above, we achieve a training sample of about 10K subjects within a 7 days of user classifications. Figure XXX shows how the sample size grows as a function of time for various timesteps and parameters of such and things.

5.2. *Feature Representation and Pre-Processing*

Machine learning algorithms require a feature vector for each training example. This vector is composed of D individual numeric quantities associated with the subject which the machine will use to discern that subject from others in the training sample. These features can be composed of any set of parameters which inform or corrolate with individual galaxy morphology. Good examples include various color metrics, Sersic index, and B/T ratio, among others. Individually, these metrics are naive determinations of overall galaxy morphology. Though we plan to incorporate these and other characteristics in the future, our current feature set draws on ZEST (Scarlata et al. 2007) and is composed of well-known morphological parameters including Concentration, Asymmetry, Gini, M20 and ellipticity. Thus, our feature vector is constructed from aggregated measurements of each galaxy's light profile. The details of these measurements can be found in the Appendix. Altogether, these features describe a five dimensional feature space in which the machine attempts to learn.

Should this paragraph be here? Probably not. But it should be somewhere. This is quite distinct from *deep learning* methods which learn on every pixel of an image through a series of non-linear transformations as explored by Huertas-Company and Company and Dieleman. Though deep learning methods promise new heights of classification accuracy, there are some drawbacks. Most notably, the interpration of these methods in the context of physical quantities is lacking.

Because of the complex suite of layered non-linear transformations, it is difficult to backstrapolate what qualities of the image were most successful at understanding that galaxy's morphology. Moreover, connecting that to physical mechanisms within the galaxy remains to be seen. On the other hand, much work has been invested in connecting aggregate features such as CAS, G-M20 which are well demonstrated to correlate with specific galaxy stuffs like SFH and ... whatnots.. and described below.

Before we feed the algorithm with these feature vectors we first perform two pre-processing steps. First, we clean the data as there are some very few number of cases where our algorithm failed to recover appropriate values for the Petrosian radius, C, A, G, or M20. Our code represents these failures as infs or nans and we thus remove these subjects from all samples. The second transformation puts each of the features on equal footing. Taken at face value, each of the five morphology parameters resides in a different range of values: M20 is nearly always negative as it is logarithmic, while Asymmetry and Gini are always between 0 and 1. In order for the machine classifier to treat all features equally we scale each feature along columns. If a row represents an individual subject, then a column represents the same feature for all subjects. We normalize each subject's features in the standard way:

$$z_{feature} = \frac{f_i - \mu}{\sigma} \tag{3}$$

Where f_i is the ith subject's feature value, μ is the mean of the entire feature sample, and σ is the standard deviation of the entire feature sample. This scales each feature to values between 0 and 1.

I might actually change this, however. It may be that you don't WANT a machine to treat each feature equally – some features may better separate classes within the feature space.

5.3. *Algorithms*

In this section we briefly describe a handful of the supervised classification algorithms we employ.

5.3.1. *K-nearest neighbors*

One of the simplest algorithms to conceptualize and use is the K-nearest neighbors (KNN) classifier. This classifier works simply by considering the K neighbors nearest to the test point in Euclidean space. As we have only a binary classification challenge, the test point is classified according to the majority of its K neighbors. In the event of a tie, the label if the closest neighbor to the test point determines its class. Though simplistic, this algorithm is surprisingly powerful in that it performs well in higher-dimensional space and is relatively fast to train. Obviously, the most important parameter to consider is the number of nearest neighbors to assign to each test point. But this is pretty much the only knob to turn in this method which is another benefit – ease of interpretation.

We use scikit-learn's implementation of K-Nearest Neighbor Classifier (KNC) and optimize the K parameter.

5.3.2. *Random forests*

Random forests (RFs) are an ensemble classifier in that they take an average of several individual classifiers, in this case, decision trees. The training data is bootstrapped (sampled with replacement) and a decision tree is performed on each sub-sample. The resulting classifications of each decision tree are combined. A decision tree

This model requires optimization of several parameters including the number of trees, the depth of each tree and BLANK.

Again we use scikit-learn's Random Forest implementation.

Discuss how this affects the accuracy of the machine classifier.

Discuss the validation set – Nair / GZ2 / Expert user classification set

5.4. *Cross-validation*

Cross-validation goes here?
Another fundamental characteristic of any machine learning algorithm is the various parameters used to guide the training of the machine. For example, the number of k neighbors in a k-nearest neighbors algorithm; or the depth, d, of a tree in a Random Forest algorithm. These parameters cannot be chosen a priori as they must be tuned to reflect the nature of the classification and to optimize the accuracy of the output. Ideally, one would train a machine classifier with every possible combination of parameters and test the resulting accuracy of each combination against another sample withheld from the training set so as not to bias the results. For this, we choose a subset of the GZ subject sample designated as a validation sample which we describe in detail in the next section.

At each time step, the machine classifier will have some some number, N, of training galaxies. The machine is then trained using several parameter sets (enough to sample the parameter space) and each trained machine is applied to the validation set. The resulting accuracy, completeness, and contamination are recorded. This process repeats at each time step until the machine has achieved a large enough training sample thus allowing the machine to reach its maximum accuracy. Only once the machine has reached its peak performance is it fi-

nally applied to the test set (remaining galaxy images in the pool).

5.5. *Machine Output?*

Along with a prediction, the machine also returns an associated probability (discussed below). If this probability, or confidence, is above some threshold (the machine threshold, not to be confused with the post-processing thresholds), then those galaxy images are considered classified and

5.6. *modularity?*

we used a super simple algorithm for this but guess what? If you structure your code a certain way, we can modularize and insert your favorite machine. Additionally, I'd really like it to be able to run multiple machines simultaneously.

APPENDIX

A. MEASURING MORPHOLOGICAL PARAMETERS ON SDSS CUTOUTS

So we did a LOT of work to measure all that shit.

Concetration measures the ...

$$C = 5\log(r_{80}/r_{20}) \tag{A1}$$

where r_{80} and r_{20} are the radii containing 80% and 20% of the galaxy light respectively. Large values of this ratio tend to indicate disky galaxies, while smaller values correlate with early-type ellipticals.

Asymmetry quantifies the degree of rotational symmetry in the galaxy light distribution (not necessarily the physical shape of the galaxy as this parameter is not highly sensitive to low surface brightness features).

$$A = \frac{\sum_{x,y}|I - I_{180}|}{2\sum|I|} - B_{180} \tag{A2}$$

where I is the galaxy flux in each pixel (x, y), I_{180} is the image rotated by 180 degrees about the galaxy's central pixel, and B_{180} is the average asymmetry of the background.

The Gini coefficient, G, describes how uniformly distributed a galaxy's flux is. If G is 0, the flux is distrubuted homogeneously among all galaxy pixels.; while if G is 1, all of the light is contained within a single pixel. This term correlates with C, however, unlike concentration, G does not require that the flux be concentrated within the central region of the galaxy. We calculate G by first ordering the pixels by increasing flux value, and then computing

$$G = \frac{1}{|\bar{X}|n(n-1)}\sum_{i}^{n}(2i - n - 1)|X_i| \tag{A3}$$

where n is the number of pixels assigned to the galaxy, and $\bar{X}$ is the mean pixel value.

M_{20} is the second order moment of the brightest 20% of the galaxy flux.

$$M_{tot} = \sum_{i}^{n} f_i[(x_i - x_c)^2 + (y_i - y_c)^2] \tag{A4}$$

$$M_{20} = \log_{10}(\frac{\sum_i M_i}{M_{tot}}), \quad \text{while} \sum_i fi < 0.2 f_{tot} \tag{A5}$$

REFERENCES

Marshall, P. J., Verma, A., More, A., et al. 2016, MNRAS, 455, 1171

Scarlata, C., Carollo, C. M., Lilly, S., et al. 2007, ApJS, 172, 406

Willett, K. W., Lintott, C. J., Bamford, S. P., et al. 2013, MNRAS, 435, 2835