Submitted

Active Learning to Overcome Sample Selection Bias: Application to Photometric Variable Star Classification

Joseph W. Richards[1,2,*], Dan L. Starr[1], Henrik Brink[3], Adam A. Miller[1], Joshua S. Bloom[1], Nathaniel R. Butler[1], J. Berian James[1,3], James P. Long[2] and John Rice[2]

ABSTRACT

Despite the great promise of machine-learning algorithms to classify and predict astrophysical parameters for the vast numbers of astrophysical sources and transients observed in large-scale surveys, the peculiarities of the training data often manifest as strongly biased predictions on the data of interest. Typically, training sets are derived from historical surveys of brighter, more nearby objects than those from more extensive, deeper surveys (*testing data*). This *sample selection bias* can cause catastrophic errors in predictions on the testing data because a) standard assumptions for machine-learned model selection procedures break down and b) dense regions of testing space might be completely devoid of training data. We explore possible remedies to sample selection bias, including importance weighting (IW), co-training (CT), and active learning (AL). We argue that AL—where the data whose inclusion in the training set would most improve predictions on the testing set are queried for manual follow-up—is an effective approach and is appropriate for many astronomical applications. For a variable star classification problem on a well-studied set of stars from *Hipparcos* and OGLE, AL is the optimal method in terms of error rate on the testing data, beating the off-the-shelf classifier by 3.4% and the other proposed methods by at least 3.0%. To aid with manual labeling of variable stars, we developed a web interface which allows for easy light curve visualization and querying of external databases. Finally, we apply active learning to classify variable stars in the ASAS survey, finding dramatic improvement in our agreement with the ACVS

[1]Astronomy Department, University of California, Berkeley, CA, 94720-7450, USA

[2]Statistics Department, University of California, Berkeley, CA, 94720-7450, USA

[3]Dark Cosmology Centre, Juliane Maries Vej 30, 2100 Copenhagen Ø, Denmark

[*]E-mail: jwrichar@stat.berkeley.edu

catalog, from 65.5% to 79.5%, and a significant increase in the classifier's average confidence for the testing set, from 14.6% to 42.9%, after a few AL iterations.

Subject headings: stars: variables: general – methods: data analysis – methods: statistical – techniques: photometric

1. Introduction

Automated classification and parameter estimation procedures are crucial for the analysis of upcoming astronomical surveys. Planned missions such as Gaia (Perryman et al. 2001) and the Large Synoptic Survey Telescope (LSST; LSST Science Collaborations et al. 2009) will collect data for more than a billion objects, making it impossible for researchers to manually study significant subsets of the data. At the same time, these upcoming missions will probe never-before-seen regions of astrophysical parameter space and will do so with larger telescopes and more precise detectors. This makes the training of automated learners for these new surveys a difficult, non-trivial task.

Supervised machine learning methods (see Bloom & Richards (2011) for review) have shown great promise for the automatic estimation of astrophysical quantities of interest—*response* variables in the statistics parlance—from sets of *features* extracted from the observed data. These studies include areas as diverse as photometric redshift estimation (Collister & Lahav 2004, Wadadekar 2005, D'Abrusco et al. 2007, Carliles et al. 2010), stellar parameter estimation and classification (Tsalmantza et al. 2007, Smith et al. 2010), galaxy morphology classification (Ball et al. 2004, Huertas-Company et al. 2008), galaxy-star separation (Gao et al. 2008, Richards et al. 2009), supernova typing (Newling et al. 2011, Richards et al. 2011a) and variable star classification (Debosscher et al. 2007, Dubath et al. 2011, Richards et al. 2011b), among others.

These studies typically assume that the distribution of training data is representative of the set of data to be analyzed (the so-called *testing* data). In reality, in astronomy the distributions of training and testing data are usually substantially different. This *sample selection bias* can cause significant problems for an automated supervised method and must be addressed to ensure satisfactory performance for the testing data. For instance, standard cross-validation techniques assume that the training and testing distributions are exactly the same; when this is not the case, sub-optimal model selection can occur.

In this paper, we show the debilitating effects of sample selection bias on the problem of automated classification of variable stars from their observed light curves. Using a set of highly studied, well-classified variable star light curves from the *Hipparcos* (Perryman

et al. 1997) Space Astrometry Mission and the Optical Gravitational Lensing Experiment (OGLE, Udalski et al. 1999a) missions, we train a classifier to automatically predict the class of each variable star in the All Sky Automated Survey (ASAS, Pojmanski 1997, Pojmański 2001). We demonstrate that this classifier results in a high error rate, a substantial number of anomalies, and low average classifier confidence. These debilitating effects are also seen in existing catalogs such as the ACVS (Pojmanski 2000, Pojmanski et al. 2005), whose use of training data from OGLE plus from an early ASAS release yields a supervised classifier that is only confident on 24% of all sources. Upcoming surveys, whose automated prediction algorithms will be trained on data from older surveys or idealized models, will suffer from these same maladies if sample selection bias is not treated properly.

To overcome sample selection bias, we propose a few methods, including importance weighting, co-training, and active learning. On both the ASAS variable star classification problem and a simulated variable star data set, we find that active learning (AL) performs the best. AL is an iterative procedure, whereby on each iteration the testing data whose inclusion in the training set would most improve predictions over the entire testing set are queried for manual follow-up and added to the training set. AL is a semi-supervised method that leverages the known features of the testing data to make the best decision about which of these objects is most useful to the supervised learner. We argue that active learning is appropriate in many areas of astrophysics, where follow-up information can often be attained through spectroscopic observations, manual study, or citizen science projects (e.g., Lintott et al. 2008). Furthermore, AL is a principled method for selecting objects for expensive follow-up in circumstances where it is infeasible to perform an in-depth analysis on every object. In particular, projects such as Galaxy Zoo stand to benefit from the active learning approach for candidate object selection, especially when data sizes become prohibitively large to manually analyze each source.

The structure of the paper is as follows. In §2 we describe in detail the problem of sample selection bias, showing how it can arise in various astronomical settings and detailing its adverse effects in a variable star classification problem. In §3 we propose a few methods that can be used to mitigate the effects of sample selection bias. We describe active learning in detail, focusing on its implementation with Random Forest classification. Next, we test those methods in §4, showing that AL attains the best results in a simulated variable star classification experiment. In §5 we describe our online active learning variable star classification tool, `ALLSTARS`, which was developed to aid the manual study of objects in various photometric surveys. We present the result of applying active learning to classify ASAS variable stars in §6, showing drastic improvement over the off-the-shelf classifier. Finally, we end with some concluding remarks in §7.

2. Sample Selection Bias in Astronomical Surveys

A fundamental assumption for supervised machine learning methods is that the training and testing sets[1] are drawn independently from the same underlying distribution. However, in astrophysics this is rarely the case. Populations of well-understood, well-studied training objects are inherently biased toward intrinsically brighter and nearby sources and available data are typically from older, lower signal-to-noise detectors.

Indeed, in studies of variable stars, samples of more luminous, well-understood stars are often employed to train supervised algorithms to classify fainter stars observed by newer, deeper surveys. Examples of this abound in the literature. For instance, Debosscher et al. (2009) use a training set from OGLE, a ground-based survey from Las Campanas Observatory covering fields in the Magellanic Clouds and Galactic bulge, to classify higher-quality CoRoT (COnvection ROtation and planetary Transits, Auvergne et al. 2009) satellite data. Dubath et al. (2011) train a classification model using a subset of the *Hipparcos* periodic star catalog containing the most reliable labels from the literature and most confident period estimates. This systematic difference between the training and testing sets can cause supervised methods to perform poorly, especially for the types of object under-sampled by the training set.

In Debosscher et al. (2009), the authors recognize that a training set "should be constructed from data measured with the same instrument as the data to be classified" and claim that some misclassifications occur in their analysis due to systematic differences between the two surveys. Because the aims and specifications of each survey are different, their observed sources usually occupy different regions of feature space. See, for example, Fig. 1, where there is an obvious absence of the combined *Hipparcos* and OGLE training data in the high-frequency, high-amplitude regime where the density of the testing set of ASAS variables is high. Even if two surveys have similar specifications (e.g., cadence, filter, depth), they may be looking in different parts of the sky or with different sensitivities and thus will observe different demographics of the same sources, causing a systematic differences in the survey priors.

In other areas of astrophysics and cosmology it is common practice to construct supervised models using spectroscopic samples and apply those models to predict parameters of interest for objects that fall entirely outside the support of the distribution of the spectroscopic data. For example, photometric redshift estimation methods typically train a

[1]Throughout the paper, we call training data those objects with known response variable that are used to train the supervised model, and we call testing data the objects of interest whose unknown response is to be predicted by the model.

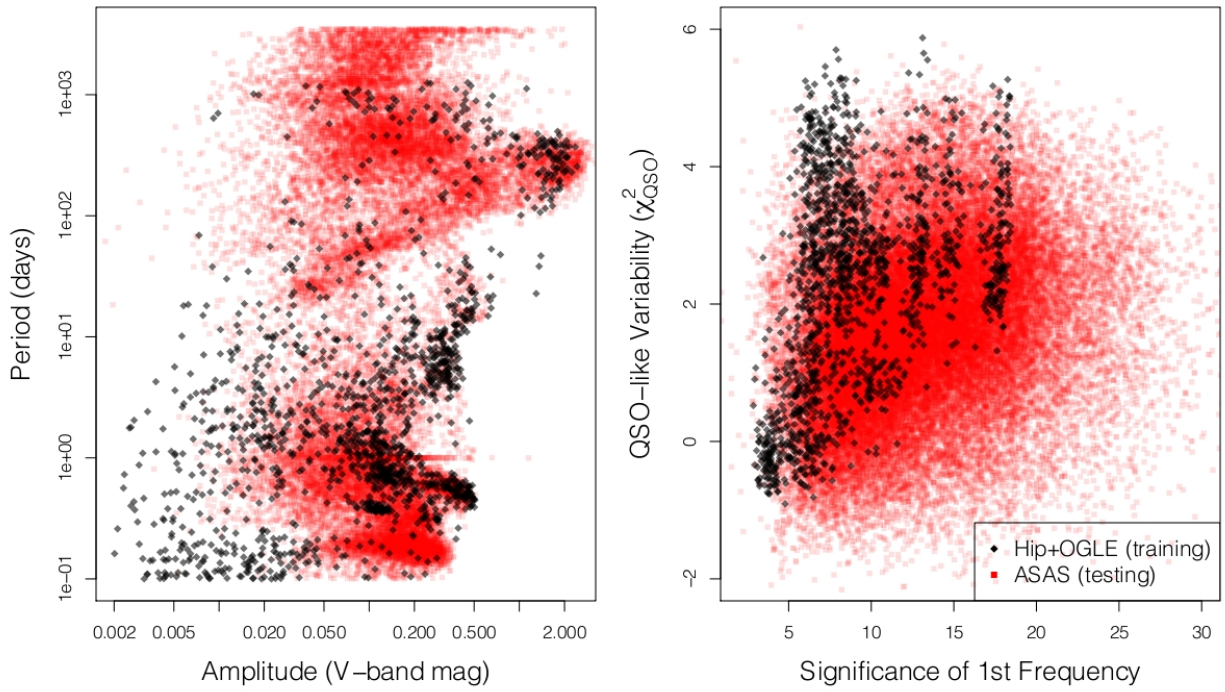

Fig. 1.— Sample selection bias for ASAS variable star (red □) classification using a training set of well-understood data from *Hipparcos* and OGLE (black ⋄). Left: Large distributional mismatch exists in the period-amplitude plane. Only those ASAS data whose statistical significance of the frequency estimate is larger than the median are plotted. ASAS testing data have high density in short-period, high-amplitude and long-period, moderate-amplitude regions, where there are little training data. Right: Testing data tend to have smaller values of the QSO-like variability metric—which measures how well the observed light curve fits a damped random walk QSO model (see Butler & Bloom 2011)—and larger values of the statistical significance of the first frequency (compared to a null, white-noise model; see Richards et al. 2011b).

regression model using a set of spectroscopically confirmed objects, whereby those models are extended to populations of galaxies that are fainter and (often) at higher redshift (papers that have studied this problem include Bonfield et al. 2010 and Sypniewski & Gerdes 2011). Several authors have proposed novel methods to mitigate the effects of non-representative photo-z training sets using physical association of galaxies (Matthews & Newman 2010; Quadri & Williams 2010) or calibration through cross-correlation (Schulz 2010). Another field where these issues occur is supernova typing, where classifiers are typically trained on spectroscopically confirmed templates and then applied to classify fainter testing data (Kessler et al. 2010; Newling et al. 2011). Recently, Richards et al. (2011a) studied the impact of the accuracy of a supervised supernova classification method on the particular spectroscopic strategy employed to obtain training sets, finding that deeper samples with fewer objects are preferred to surveys with shallower limits.

The situation we describe, where the training and testing samples are generated from different distributions, is referred to in the statistics and machine learning literature as *covariate shift* (Shimodaira 2000) or *sample selection bias* (Heckman 1979). This systematic difference can cause catastrophic prediction errors when the trained model is applied to new data. These problems arise for two reasons. First, under sample selection bias, standard generalization error estimation procedures, such as cross-validation, are biased, resulting in poor model selection. Off-the-shelf supervised methods are designed to choose the model that minimizes some error criterion integrated with respect to the training distribution; when the testing distribution is substantially different, this model is likely to be suboptimal for prediction on the testing data. In (§3.1) we describe a principled weighting scheme to alleviate this complication. Second, significant regions of parameter space may be ignored by the training data—such as in the variable star classification problem shown in Fig. 1—causing catastrophically bad extrapolation of the model onto those regions. In this case, any classifier trained only on the training data will produce poor class predictions for the ignored regions of parameter space: no weighting scheme on the training data can enforce good classifier performance in these regions. This suggests that the testing data need to be used, in a semi-supervised manner, to augment the training set. In this paper, we explore two different approaches to this problem: *co-training* (§3.2 and *self-training*), where testing instances with most certain class prediction are iteratively added to the training set, and *active learning* (§3.3), where testing instances whose labels, if known, would be of maximal benefit to the supervised method, are manually studied to ascertain the value of their response (e.g. class label, redshift, etc.), and subsequently included in the training set.

2.1. Example: Source Classification for ASAS

In this section, we demonstrate the effects of sample selection bias in classifying variable stars from the All Sky Automated Survey (ASAS). Particularly, we use an automated machine learning classifier to classify sources in the ASAS Catalogue of Variable Stars (ACVS, Pojmanski 2002). ACVS verson 1.1[2] consists of V-band light curves for 50,124 stars that have passed tests of variability as described in Pojmanski (2000). As a training set for this classification problem, we use only the confidently labeled *Hipparcos* and OGLE sources used in Debosscher et al. (2007) and Richards et al. (2011b). This data set consists of 1542 variable stars from 25 different science classes. The period-amplitude relationship of the instances in the training set of *Hipparcos* and OGLE data, and in the ACVS catalog are plotted in Figure 1, where sample selection bias is obvious.

As a part of ACVS, predicted classes are provided for a fraction of the stars. As described in Pojmanski (2002), ACVS obtains their classifications using a neural net type algorithm trained on set of visually labeled ASAS sources, confirmed OGLE cepheids Udalski et al. (1999b,c), and OGLE Bulge variable stars Wozniak et al. (2002). A filter is used to divide strictly periodic from less regular periodic sources. A neural net is trained on the period, amplitude, Fourier coefficients (first 4 harmonics), $J-H$ and $H-K$ colors and IR fluxes to predict the classes of the strictly periodic sources. Several ACVS objects either have multiple labels or are annotated as having low confidence classifications. For less regular periodic sources, location in the $J-H$ vs. $H-K$ plane is tested; if the object falls within an area of late-type irregular or semi-regular stars, it is assigned the label MISC, else it is inspected by eye. We find that 38,117 ACVS stars, representing 76% of the catalog, are either labeled as MISC, assigned multiple labels, or have low class confidence. The remaining 24% of stars have confident ACVS labels, and provide a set of classifications to compare our algorithms against. In Figure 2 we plot in color, in period-amplitude space, the classes of the training data and the ACVS classes of the ASAS data[3].

As our base model, we use a Random Forest classifier (Breiman 2001). Random Forest has recently been shown by Richards et al. (2011b) and Dubath et al. (2011) to attain accurate results in automated classification of variable stars. In this paper, we represent each variable star in our data set by the 59 light-curve features used by Richards et al.

---

[2]The ACVS catalog can be downloaded at http://www.astrouw.edu.pl/asas/data/ACVS.1.1.gz.

[3]Note that not all sources are actually periodic, meaning that some period estimates are nonsensical. However, we also use the statistical significance of the frequency estimate as an input feature into our classifier; thus the classifier learns to trust the only periodic features of those sources with high frequency significance, and to rely on only the non-periodic features of the low-significance data.

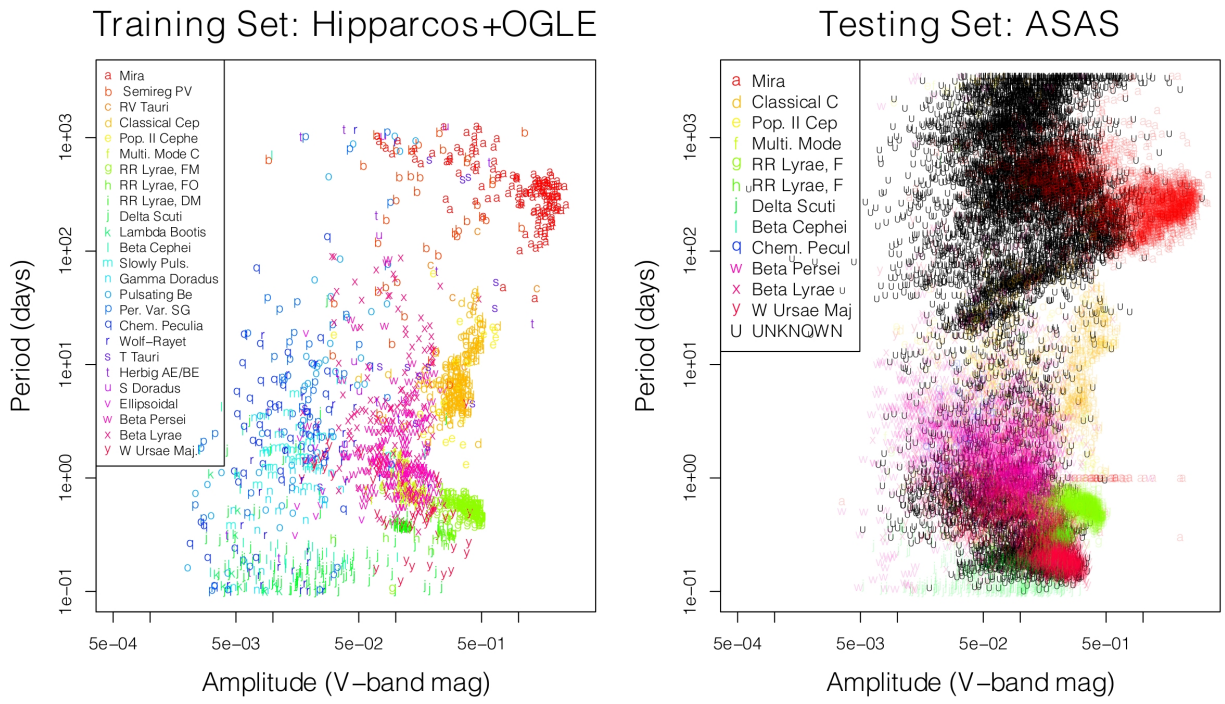

Fig. 2.— Left: Period-amplitude relationship for the 1542 training set sources from the *Hipparcos* and OGLE surveys. Symbols and colors denote the true science class of each object. Right: Same for an arbitrary sample of size 10,000 from the 50,124 ASAS testing objects, where symbols and colors denote the ACVS labels. Black 'U' denotes that the source is either labeled MISC, doubly-labeled, or has low confidence label by ACVS. Our goal is to use the training data set to predict the class label (and posterior class probabilities) for each ASAS object. Complicating this task is the significant distributional difference between the training and testing sets.

(2011b), as well as 5 additional light-curve features from Dubath et al. (2011). The Random Forest classifier is a supervised, non-parametric method that attempts to predict the science class of each star from its high-dimensional feature vector. It operates by constructing an ensemble of classification decision trees, and subsequently averaging the results. The key to the good performance of Random Forest is that its component trees are de-correlated by sub-selecting a small random number of features as splitting candidates in each non-terminal node of the tree. As a result, the average of the de-correlated trees attains highly decreased variance over each single tree, with no substantial increase in bias.[4].

By training a Random Forest classifier on the *Hipparcos* and OGLE data as in Richards et al. (2011b) and applying that classification model to predict the class label of each object in ACVS, we obtain a 65.5% correspondence with the ACVS labels for the 24% of objects that have a confident ACVS label. A table showing the correspondence of our predicted Random Forest classification labels with those of ACVS is plotted in Figure 3. The Random Forest algorithm classifies 90% and 79% of the Mira and RR Lyrae, FM stars identified by ACVS, but shows much lower correspondence for other classes, such as Delta Scuti, Population II Cepheid, and RR Lyrae, FO. Note that the Random Forest class taxonomy is finer than that used by ACVS, including twice as many classes; as such, the Random Forest has the ability to identify objects of rarer classes, such as T Tauri and Gamma Doradus stars.

However, there are serious problems that arise by running the analysis in this manner and ignoring the significant sample selection bias between the training and testing sets. In Figure 1 we saw that the distribution of the training set of *Hipparcos* and OGLE sources is wildly different than the distribution of ASAS sources; notably, regions of long-period, amplitude < 1 sources and regions of short-period, high-amplitude sources are densely populated in ASAS but contain little or no training data. As a consequence, a large proportion of the ASAS data set has no counterpart in the training set that closely matches its feature vector, meaning that it will likely be incorrectly identified by the Random Forest classifier as belonging to a physically different class of variable star. One telling statistic is that for only 14.6% of the ASAS objects does the Random Forest produce a posterior class probability of ≥ 0.5, meaning that the classifier is only confident on the class predictions for 15% of the entire ASAS ACVS catalog.

In Figure 3 we find that many ASAS sources (9109 of 50,124, or 18.2%) are identified by the Random Forest classifier as being of RR Lyrae, DM type, a relatively rare type of doubly pulsating variable star. This is far too many RR Lyrae, DM candidates; for comparison, Soszyński et al. (2011) find only 91 RR Lyrae, DM candidates in the entire OGLE-III catalog,

[4]For more details about the Random Forest variable star classifier used, see Richards et al. (2011b).

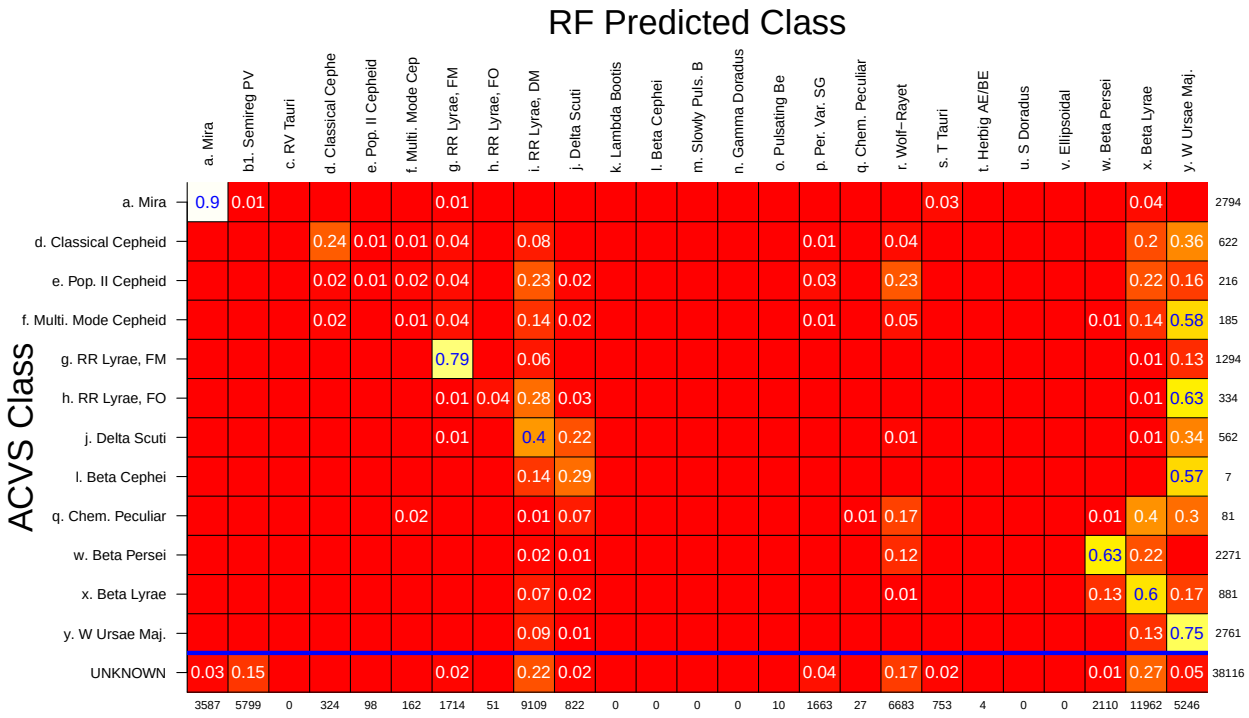

Fig. 3.— Off-the-shelf Random Forest classifications of the ASAS data set, using a training set of the 1542 *Hipparcos* & OGLE sources, compared to the ACVS classifications. Rows are normalized to sum to 100%, marginal counts are listed to the right and bottom of the table. The RF classifier finds a 65.5% correspondence with the ACVS labels, for the 12,007 objects with ACVS label, with many major discrepancies. Particularly, the RF detects a very small number of the ACVS Cepheids, Delta Scuti, and Chemically Peculiar stars. Also, the RF finds a gross overabundance of Double Mode RR Lyrae and Wolf-Rayet stars. These artifacts result from sample selection bias.

out of 16,836 total RR Lyrae candidates (0.5%). This artifact in our classification occurs because the RR Lyrae, DM objects have multiple pulsational modes, causing their data to poorly fold around a single period. Because ASAS photometry is less precise than that of *Hipparcos* or OGLE, its folded light curves are considerably more noisy. Consequently, for a large subset of ASAS sources that do not resemble any of the training data, the classifier's "best guess" is RR Lyrae, DM because training light curves of that class most resemble ASAS data. In addition, an artificially high number of Wolf Rayet and Beta Lyrae stars are found by the RF. This deficiency of the off-the-shelf classifier illustrates the need for other approaches.

3. Methods to Treat Sample Selection Bias

Above, sample selection bias was defined, its presence in astrophysical problems motivated, and its adverse effects exemplified with an example in variable star classification. In this section, we will introduce three different principled approaches of treating sample selection bias, and argue that active learning is the most appropriate of these methods for dealing with astronomical sample biases. Later, these methods will be compared using variable star data from the OGLE and *Hipparcos* missions.

3.1. Importance Weighting

Under sample selection bias, standard generalization error estimation procedures, such as cross-validation, are biased, resulting in poor model selection for supervised methods. To remedy this, importance weighting (IW) cross-validation is often used (see Sugiyama & Müller 2005, Huang et al. 2007, and Sugiyama et al. 2007). Under this approach, the training examples are weighted by an empirical estimate of the ratio of test-to-training-set feature densities during the training procedure. Specifically, when evaluating the statistical risk of the statistical model over the training data, the weights

$$w_i = \frac{\mathbf{P}_{\text{Test}}(\mathbf{x}_i, y_i)}{\mathbf{P}_{\text{Train}}(\mathbf{x}_i, y_i)} = \frac{\mathbf{P}_{\text{Test}}(\mathbf{x}_i)\mathbf{P}_{\text{Test}}(y_i|\mathbf{x}_i)}{\mathbf{P}_{\text{Train}}(\mathbf{x}_i)\mathbf{P}_{\text{Train}}(y_i|\mathbf{x}_i)} = \frac{\mathbf{P}_{\text{Test}}(\mathbf{x}_i)}{\mathbf{P}_{\text{Train}}(\mathbf{x}_i)} \tag{1}$$

are typically used, where $\mathbf{x}_i$ is the feature vector and y_i is the response variable (i.e., class) for training object i. To achieve the last inequality in Equation 1, it is assumed that $\mathbf{P}_{\text{Test}}(y_i|\mathbf{x}_i) = \mathbf{P}_{\text{Train}}(y_i|\mathbf{x}_i)$, i.e. that the probability of a specific response given a feature vector is the same for training and testing sets. In practice, this equality will probably not hold for the types of astrophysical data sets we are interested in: though the mapping

from features to response values may be the same for the training an testing sets, the prior distributions over the responses, y, are different, in general. Even in this situation, use of the ratio of feature densities—though imperfect—may still be useful, and is more tractable than using the joint feature-response densities[5]. Even so, in practice the training and testing feature densities are difficult to estimate (and their ratio is even harder to estimate) because they reside in high-dimensional feature spaces. To overcome this, Eqn. 1 can be estimated via distribution matching (Huang et al. 2007) or by fitting a probabilistic classifier to the classification problem of training vs. testing set and employing the output probability estimates (Zadrozny 2004).

Using the weights defined in Eqn. 1 when training a classifier induces an estimation procedure that gives higher importance to training set objects in regions of feature space that are relatively under-sampled by the training data, with respect to the testing density. This enforces a higher penalty for making errors in regions of feature space that are under-represented by the training set. This is sensible because, since the ultimate goal is to apply the model to predict the response of the testing data, we should attempt to do well at modeling the output in regions of feature space densely populated by testing data (and conversely ignore modeling regions devoid of testing data). For the ASAS example, importance weighting will give large weights to the training data in the region of Amplitude < 0.5 and Period > 100 and affix small weights to data in the high-amplitude clump centered around a 300-day period.

Though this approach is useful in some problems, importance weighting has been shown to be asymptotically sub-optimal when the statistical model is correctly specified[6] (Shimodaira 2000), and with flexible non-parametric models such as Random Forest we observe very little change in performance using IW (see §4). An additional, more debilitating drawback is that IW requires the support of the testing distribution be a *subset* of the support of the training distribution[7], which, in the types of supervised learning problems common in astrophysics, is rarely the case.

---

[5]Note that we could alternatively rewrite the joint density as $\mathbf{P}(y_i)\mathbf{P}(\mathbf{x}_i|y_i)$. It is unlikely that $\mathbf{P}_{\rm Test}(\mathbf{x}_i|y_i) = \mathbf{P}_{\rm Train}(\mathbf{x}_i|y_i)$ in most practical situations; however, if this were to hold then the importance weights would simply reduce to the ratio of response priors.

[6]In other words, IW produces worse results than the analogous unweighted method if the parametric form of $\mathbf{P}(y|\mathbf{x})$ is correct.

[7]Else the weights, defined as the ratio of test-to-training set feature densities, explode, and the theoretical properties of the method no longer hold.

3.2. Co-training

In astronomical problems, we typically have much more unlabeled than labeled data. This is due to both the pain-staking procedures by which labels must be accrued (e.g., by spectroscopic follow-up or manual assignment), and the fact that there are exponentially more dim, low signal-to-noise sources than bright, well-understood sources. Recently, supervised classification algorithms have been developed that use both labeled and unlabeled examples to make decisions. This class of models is referred to as *semi-supervised* because learning is performed both on the instances with known response values and on the feature distribution of instances with no known response. Semi-supervised methods such as *co-training* and *self-training* slowly augment the training set by iteratively adding the most confidently-classified test cases in the previous iteration.

Co-training was formalized by Blum & Mitchell (1998) as a method of building a classifier from scarce training data. In this method, two separate classifiers, h_1 and h_2, are built on different (disjoint) sets of features, $\mathbf{x}_1$ and $\mathbf{x}_2$. In an iteration, each classifier adds its p most confidently labeled test instances to the training set of the *other* classifier. This process continues either for N iterations or until all test data belong to the training set of both classifiers. The final class predictions are determined by multiplying the class probabilities of each classifier, i.e. $p(y|\mathbf{x}) = h_1(y|\mathbf{x}_1)h_2(y|\mathbf{x}_2)$. Co-training has shown impressive performance in situations where very few training examples are used to classify many test cases. Blum & Mitchell (1998) use co-training in a two-class problem, using 12 labeled web pages to classify a corpus of 1051 unlabeled pages, achieving a 5% error rate.

In the original co-training formulation, it was assumed that each object could be described by two different 'views' (i.e. feature sets) of the data that were both redundant (each view of the object gives similar information) and conditionally independent given the true class label. While this natural redundancy may be present in web page classification (e.g., the words on the web page and the words on pages linked to that page), it is not generally the case. Later papers by Goldman & Zhou (2000) and Nigam & Ghani (2000) argue that even when a natural feature division does not exist, arbitrary or random feature splits produce better results than self-training (Nigam & Ghani 2000), where a single classifier is built on all of the features whereby the most confidently classified testing instances are iteratively moved to the training set.

In the variable star classification paper of Debosscher et al. (2009), something akin to a single iteration of self-training was performed for CoRoT classification using OGLE training data, where candidate lists obtained with the first version of the classifier were used to select very probable class members amongst the testing set data for subsequent inclusion in the training set. This augmentation procedure led to inclusion of an extra 114 sources into the

training set.

Both co-training and self-training are reasonable approaches to problems that suffer from sample selection bias because they iteratively move testing data to the training set, thereby gradually decreasing the amount of bias that exists between the two sets. However, in any one step of the algorithm, only those data in a close neighborhood to existing training data will be confidently classified and made available to be moved to the training set. Thus, as the iterations proceed, the dominant classes in the training data diffuse into larger regions of feature space, potentially gaining undue influence over the testing data. In addition, co-training and self-training will never predict classes that are rare or unrepresented in the training data, even if they are prominent in the testing data. In §4 we apply both self-training and co-training to variable star classification, finding that these methods perform poorly in terms of overall error rate, especially for classes that are under-sampled by the training data.

3.3. Active Learning

An important feature to supervised problems in astronomy is that we often have the ability to selectively follow up on objects to ascertain their true nature. For example, for different problems this can be achieved by targeted spectroscopic study, visualization of (folded) light curves, or querying of other databases and catalogs. Consider astronomical source classification: while it is impractical to manually label all hundred-million plus objects that will be observed by Gaia and LSST, manual labeling of a small, judiciously chosen set of objects can greatly improve the accuracy of an automated supervised classifier. This is the approach of *active learning* (and in particular, pool-based active learning, Lewis & Gale 1994). Under pool-based AL for classification, an algorithm iteratively selects, out of the entire set of unlabeled data, the object (or set of objects) that would give the expected maximal performance gains of the classification model, if its true label(s) were known. The algorithm then queries the user to manually ascertain the science class of the object(s), whereby the supervised learner incorporates this information its subsequent training sets to improve upon the original classifier. For a thorough review of active learning, see Settles (2010).

Active learning has enjoyed wide use in machine learning, with impressive results in many areas of application, such as text classification, speech recognition, image and video classification, and medical imaging (Lewis & Gale 1994; Tong & Chang 2001; Tong & Koller 2002; Yan et al. 2003; Liu 2004; Tur et al. 2005). Begin with a training set $\mathcal{L}$ and testing set $\mathcal{U}$. On each active learning iteration, we manually find the class of the testing set source,

$\mathbf{x}' \in \mathcal{U}$, whose inclusion into $\mathcal{L}$ would most improve the classifier's performance on the testing data (according to some metric, see §3.3.1). These queried active learning samples tend to be data that reside in relatively dense regions of testing set feature space, $\mathbf{P}_{\text{Test}}(\mathbf{x})$, scarcely populated regions of training set feature space, $\mathbf{P}_{\text{Train}}(\mathbf{x})$, and in regions where the class identity is uncertain.

For an appropriate selection metric, a small number of active learning samples will suffice in making the labeled set feature distribution resemble the unlabeled set distribution. This approach is similar to the importance sampling approach of Zadrozny (2004), who show that if training set sources are resampled with respect to the appropriate (weighted) distribution, then the statistical risk of the classifier built on that data will minimize the statistical risk evaluated over all of the data. The drawback to that approach is that it needs a relatively large initial training sample and requires that for all non-zero regions of $\mathbf{P}_{\text{Test}}$, $\mathbf{P}_{\text{Train}}$ also be non-zero. On the other hand, the active learning approach to sample selection bias is to *expand* the training set in a way that makes it most closely resemble the testing set, and thus these problems are avoided.

3.3.1. Active Learning Query Function

Several strategies have been proposed to determine which testing data about which active learning will query the "human annotator." Most of these prescriptions attempt to select data whose label, if known, would maximally help the classifier. The simplest form of querying is *uncertainty sampling* (Lewis & Gale 1994), by which on each iteration, the training datum with highest label uncertainty (measured, e.g., by entropy or margin) is queried for manual identification. Though simple, this approach does not explicitly consider changes to the overall error rate of the classifier, and is prone to select outlying points that have little influence in the classification of the other testing data.

Since we have an explicit goal of minimizing the classification error rate over the entire set of testing data, it is sensible to consider this metric explicitly when queuing data for AL. This is the approach taken by the *expected error reduction* strategies (Roy & McCallum 2001), where on each iteration the algorithm queries the testing point whose inclusion into the training set would produce the smallest classification error rate (statistical risk) over the testing set. These methods operate by iteratively adding each testing point to the training set and retraining the classifier[8]. However, because the true labels of the training data are

[8]For many machine learning algorithms, fast incremental updating algorithms exist, making this approach tractable.

not known *a priori*, one must also iterate over the possible labels of the training data, and can only compute an estimate of the expected decrease in testing error rate by approximating the error under all possible labels of all testing data. For common astronomy data sets, with $\gtrsim 10^5$ objects, expected error reduction is impractical. A viable alternative is *variance reduction* (Cohn 1996), where the testing object that minimizes the classifier's variance is selected on each iteration. Since a classifier's error can be decomposed into variance plus squared-bias plus label noise[9], minimizing the variance amounts to minimizing the error rate; also, for many models, the variance can be written in closed form, circumventing any costly computations.

In this paper, we consider two different selection criteria. The first criterion is motivated by importance weighting and the second is motivated by selecting the sources whose inclusion into the training set would produce the most total change in the predicted class probabilities for the testing sources. To meet these criteria, we revisit the Random Forest classifier. For each of B bootstrap samples from the training set, we build a decision tree, θ_b, which predicts the class of each object from its feature vector, $\mathbf{x}$. The Random Forest class probability of class y is simply the empirical proportion,

$$\widehat{P}_{\mathrm{RF}}(y|\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B}\theta_b(y|\mathbf{x}) \tag{2}$$

of the B trees that predict class y. Additionally, the Random Forest provides a measure of the *proximity* of any two feature vectors with respect to the ensemble of decision trees, defined as

$$\rho(\mathbf{x}', \mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} I(\mathbf{x} \in T_b(\mathbf{x}')) \tag{3}$$

which is the proportion of trees for which the two objects $\mathbf{x}$ and $\mathbf{x}'$ fall in the same terminal node, where $I(\cdot)$ is a boolean indicator function. Here, we use the notation $T_b(\mathbf{x}')$ to denote the terminal node of feature vector $\mathbf{x}'$ in tree b.

Heuristically, sample selection bias causes problems in the building of a classifier principally because large density regions of testing data are not well represented by the training data. Our first AL selection procedure uses this heuristic argument to select the testing point, $\mathbf{x}' \in \mathcal{U}$, whose feature density is most under-sampled by the training data, as measured by

[9]Classifier variance measures the variability in a classifier with respect to the actual training set used, classifier bias is the amount of discrepancy between the true labels and the expected prediction of a classifier (averaged over all possible training sets), and label noise is the amount of error in the training set labels.

the ratio of the two densities. This amounts to choosing AL samples that maximize

$$S_1(\mathbf{x}') = \frac{\mathbf{P}_{\text{Test}}(\mathbf{x}')}{\mathbf{P}_{\text{Train}}(\mathbf{x}')} \approx \frac{\sum_{\mathbf{x}\in\mathcal{U}} \rho(\mathbf{x}', \mathbf{x})/N_{\text{Test}}}{\sum_{\mathbf{z}\in\mathcal{L}} \rho(\mathbf{x}', \mathbf{z})/N_{\text{Train}}} \tag{4}$$

where we estimate the training and testing set densities at $\mathbf{x}'$ by averaging the RF proximity measure over the set of training ($\mathcal{L}$) and testing ($\mathcal{U}$) sets, respectively. The expression $\sum_{\mathbf{x}\in\mathcal{U}} \rho(\mathbf{x}', \mathbf{x})/N_{\text{Test}}$, is the average, over the trees in the forest, of the proportion of testing data with which $\mathbf{x}'$ shares a terminal node. The estimate of the probability density at $\mathbf{x}'$ would need be normalized by the average volume of the terminal nodes of $\mathbf{x}'$; however, since Equation 4 considers the ratio of two such densities at $\mathbf{x}'$, the average volume terms cancel, giving the above expression.

Our second AL selection criterion is to choose the testing example, $\mathbf{x}' \in \mathcal{U}$, that maximizes the total amount of change in the predicted probabilities for the testing data. This is a reasonable metric because it says that we will only spend time manually annotating the testing data whose labels most affect the predicted classifications. To achieve this, we create a selection metric that attempts to choose the $\mathbf{x}'$ that maximizes the total change, summed over the testing set, of the RF probability vectors (as measured using the ℓ_1 norm). An approximate solution to this problem is to choose the testing data points that maximize

$$S_2(\mathbf{x}') = \frac{\sum_{\mathbf{x}\in\mathcal{U}} \rho(\mathbf{x}', \mathbf{x})(1 - \max_y \widehat{P}_{\text{RF}}(y|\mathbf{x}))}{\sum_{\mathbf{z}\in\mathcal{L}} \rho(\mathbf{x}', \mathbf{z}) + 1} \tag{5}$$

where the Random Forest probability, $\widehat{P}_{\text{RF}}(y|\mathbf{x})$, is defined in Equation 2. In Appendix A we work out the details of deriving Eqn. (5) from the stated goal of selecting testing points whose labels maximally affect the total change of the Random Forest predicted probabilities over $\mathcal{U}$.

The key elements to Eqns. 4–5 are (1) the testing set density, represented by $\sum_{\mathbf{x}\in\mathcal{U}} \rho(\mathbf{x}', \mathbf{x})$ is in the numerator, and (2) the training set density, represented by $\sum_{\mathbf{z}\in\mathcal{L}} \rho(\mathbf{x}', \mathbf{z})$, is in the denominator. This means that we will choose instances that are in close proximity to many testing points and are far from any training data, thereby reducing sample selection bias. In addition, S_2 is a weighted version of S_1 with the Random Forest prediction uncertainty, represented by $1 - \max_y \widehat{P}_{\text{RF}}(y|\mathbf{x})$, in the numerator. This means that S_2 gives higher weight to those testing points that are difficult to classify thereby causing the algorithm to focus more attention along class boundaries, which should lead to better performance of the classifier.

3.3.2. Batch-Mode Active Learning

In typical active learning applications, queries are chosen in serial. However, in most astronomical applications, it makes more sense to query several testing set objects at once, in *batch mode*; for instance, in a typical observing run multiple objects are queued for follow-up observation. In the variable star classification problem, we determine that the best use of users' time is to supply them with dozens of sources to label at one sitting.

The challenge with batch-mode AL is to determine how to best choose multiple testing instances at once. Selecting the top few candidates is typically suboptimal because those objects generally lie in the same region of feature space, as is obvious from analyzing the criteria in Eqns. 4-5. Heuristic methods have been devised that create diversity in the batch of AL samples for a particular classifier (e.g., Brinker 2003 for SVMs). In our use of AL, we sample batches of AL samples by treating the criterion function as a probability sampling density, i.e., $\mathbf{P}(\text{select } \mathbf{x}') \propto S_1(\mathbf{x}')$. In §4 we compare this density method, which we call AL-d, to a method that selects the top candidates on each AL iteration, which we refer to as AL-t.

3.3.3. Crowdsourcing Labels

Most active learning papers assume that labels can be found, without noise, for any queried data point. In typical astronomical applications, this will not be the case. For instance, after follow-up observations of an object, its true nature might still be difficult to ascertain and will often remain unknown. Indeed, in classifying variable stars, users will sometimes have difficulty in obtaining the true class of an object, especially for noisy or aperiodic sources. This causes two complications in the AL process:

1. some queried sources will still have an unknown label after manual classification, and
2. a few sources will be annotated with an *incorrect* label.

The first difficulty means that we expect to receive user labels for only a fraction of the queried sources; to avoid wasting costly user time, we attempt to select AL sources that users will have a higher probability of successfully labeling (in §3.3.4 we describe how this is achieved by using a cost function). To overcome the second complication, we use crowdsourcing, where several users are presented with the same set of AL sources. The idea behind crowdsourcing is that by using the combined set of information about each object from multiple users, we are able to suppress the noise in the manual labeling process.

A difficulty in crowdsourcing is in simultaneously predicting the best label and judging the accuracy of each annotator from a set of user responses. Users are likely to disagree on some objects, so determining a true label can often be tricky. However, because each annotator has a different skill level, we should give more credence to the labels of the more adept users in deciding on a label. In the active learning paper of Donmez et al. (2009), a novel, yet simple method called `IEThresh` was introduced to filter out the less-adept users in crowdsourcing labels. Their basic approach is to start each user with the same prior skill level. Then, as the AL iterations progress, users whose responses agree with the consensus votes of the crowd are given higher 'reward'. The skill level of each user is determined by the upper confidence interval (UI) of the mean reward of all their previous labels. For each subsequent iteration, only those users whose UIs are higher than ϵ times the UI of the best annotator are included in the vote for the class of that object. Even if a particular user's label is not used in a vote, their reward level can change, meaning that users are able to drift in and out of the decision-making process over time.

In §6, we use the `IEThresh` algorithm with $\epsilon = 0.85$ to crowdsource the ASAS labels. In addition, for a source to be included in the training set, we require that at least 70% of users who looked at the source return a label. This strict policy is implemented so that only the most confident AL sources are moved to the training set so as to avoid including incorrectly labeled objects.

3.3.4. Cost of Manual Labeling

Standard active learning methods assume that the cost of attaining a label is the same for every data point, and thus aim to minimize the total number of queries performed (or equivalently achieve the lowest error rate for a given number of queries). This assumption is not valid for variable star classification problem, for a variety of reasons. First, higher signal-to-noise light curves with larger number of epochs will be, on average, easier to manually label than sparser, noisier light curves. Second, a star that has been observed and cataloged by multiple surveys (for instance, it is in the SDSS footprint) will have more archival data with which to determine its true class. Third, depending on its coordinates, a star may or may not be readily available for spectral follow-up. To avoid wasting user time on impossible-to-classify objects, these factors must be taken into account when choosing AL samples.

In applying AL to variable star classification, we treat the cost as a multiplicative factor on the querying criteria. That is, the AL function is $S(\mathbf{x}') = S_1(\mathbf{x}')(1 - C(\mathbf{x}'))$, where the cost function, $C(\mathbf{x}')$, is

$$C(\mathbf{x}') = \mathbf{P}(\mathbf{x}' \text{ cannot be manually labeled } |\mathbf{x}' \text{ is queried}), \tag{6}$$

i.e., the cost function is the probability that a user (or set of users) cannot actually determine a label for that source, given that the user was given that object to manually study[10]. High cost means that we will avoid querying that object. Inclusion of a cost function deters us from wasting valuable user time on objects that are too noisy or sparsely sampled to determine their science class. In §6 we describe how we model the cost and derive an empirical cost estimate for each object in the ASAS testing set.

3.3.5. Stopping Criterion

Insofar as the aim of active learning is to improve the performance of a classifier to the greatest extent possible with as little effort as possible, we must determine when to stop manually labeling sources. A reasonable rule of thumb is to stop querying data for active learning when the effort needed to acquire the new labels is larger than the benefit that those labels have on the classifier's performance. However, it is often difficult to compare these gains and losses, especially for problems where there do not exist ground truths with which to judge the classifier performance nor good metrics to measure gains and losses. Alternatively, one can track the intrinsic stability of the classifier (e.g., by measuring its average confidence over the testing set), and stop when a plateau is reached (cf. Vlachos 2008; Olsson & Tomanek 2009). In our implementation of AL, we choose to run iterations until the performance of the classifier levels off (as judged by a few intrinsic and extrinsic metrics, see §6).

4. Experiment: OGLE and *Hipparcos* Variable Stars

In this section, we test the effectiveness of the various methods proposed in §3 in combating sample selection bias for variable star classification. Starting with the set of 1542 well-understood, confidently-labeled variable stars from Debosscher et al. (2007), we randomly draw a sample of 721 training sources according to a selection function, Γ, that varies across the amplitude-period plane as

$$\Gamma(\mathbf{x}) \propto \log(\text{period } \mathbf{x}) \cdot \log(\text{amplitude } \mathbf{x})^{1/4}. \tag{7}$$

[10]Other definitions of the cost are possible, such as the time necessary for a user to manually label a source or the user disagreement rate. As formulated, our "cost" function measures the inutility of the user on each particular source.

This selection function is devised so that the training set under-samples short-period, small-amplitude variable stars. The resultant training and testing sets are plotted in the amplitude-period plane, along with the training set selection function, in Figure 4.

Distributional mismatch between the training and testing sets causes an off-the-shelf Random Forest classifier to perform poorly for short-period small-amplitude sources. The median overall error rate for a Random Forest classifier trained on the training data and applied to classify the testing data is 29.1%. This is 32% larger than the 10-fold cross-validation error rate of 21.8% on the entire set of 1542 sources (see Richards et al. 2011b; the error rate quoted here is slightly lower due to the addition of new features). The average error rate for testing set objects with period smaller than 0.5 days is 36.1%.

To treat the sample selection bias, we use each of the following methods:

- Importance weighting. A single Random Forest is built on the training set, with class-wise[11] importance sampling weights defined as the ratio of the testing set to training set class proportions[12].
- Self-training and co-training. Each algorithm is repeated for 100 iterations, where on each iteration the most confident 3 testing set objects are added to the training set. For co-training, we use both random feature splits (CT) and periodic versus non-periodic features (CT.p).
- Active learning. Using the metrics in Equations 4 (AL1) and 5 (AL2), we perform 10 rounds of active learning, with batch size of 10 objects selected on each round. The classifier is retrained on the available labeled data after each round. Testing set objects are selected for manual labeling either by treating the selection metrics as probability distributions (AL1.d, AL2.d), or by taking the top candidates (AL1.t, AL2.t). We also compare to an AL method that selects objects completely at random (AL.rand).

For each of the active learning approaches, we evaluate the error rate only over those testing set objects that are not queried by the algorithm. This way we do not artificially decrease the error rate by evaluating sources whose labels have been manually obtained. Note that for this experiment, we have assumed that the true labels can be manually obtained with no error.

---

[11]In importance weighting, ratios of feature densities are typically used as the weights. However, in our implementation of Random Forest, weights may only be defined by class.

[12]Since we know the true class of each object, we are able to use this information to derive the weights. In a real problem, the feature or class densities would need to be estimated.

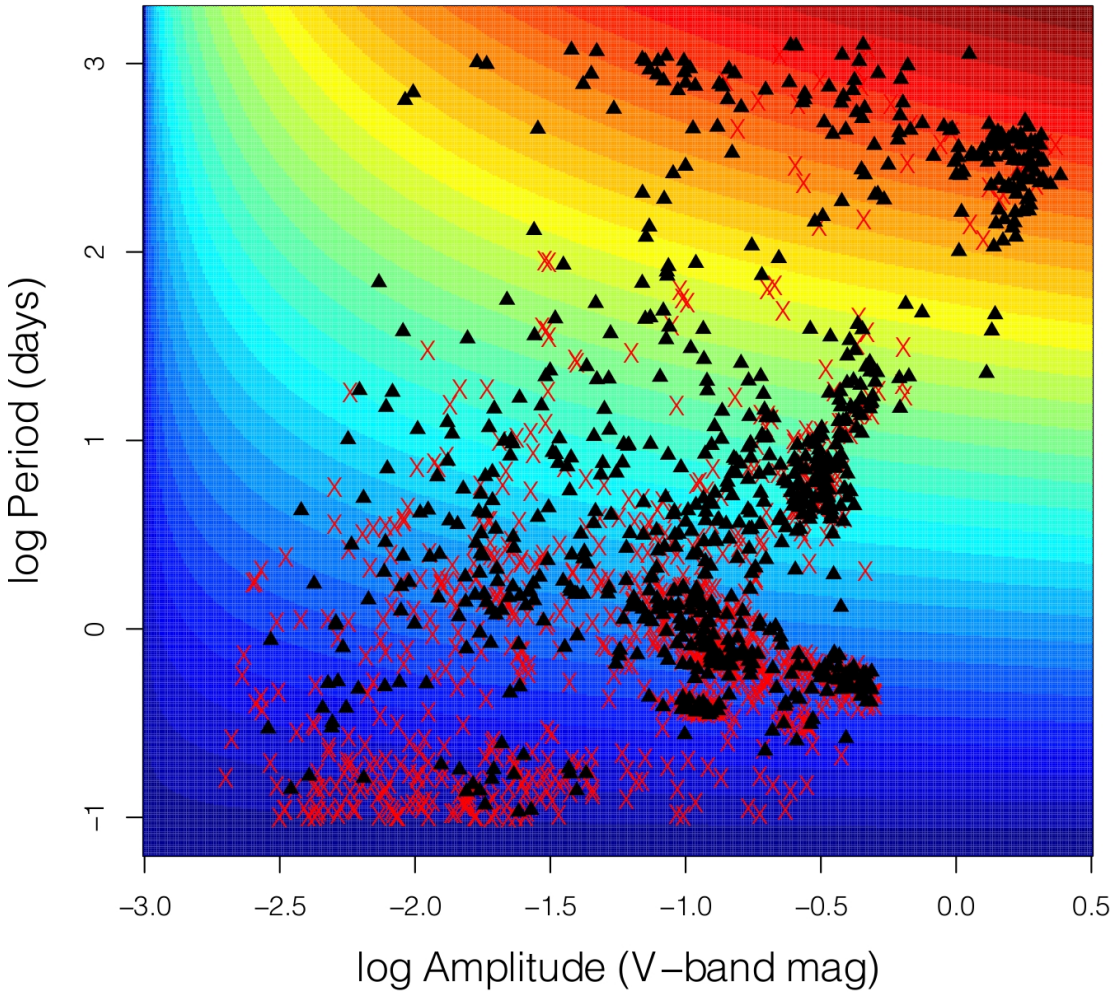

Fig. 4.— Training (black ▲) and testing (red x) data for the simulated example using OGLE & *Hipparcos* data. The 771 training data were randomly sampled from the original 1542 sources according to the sampling distribution plotted in color. Using this sampling scheme, we create sample selection bias by over-sampling long-period, high-amplitude stars and under-sampling the short-period, low-amplitude sources.

Distributions of the classification error rates for each method, obtained over 20 repetitions, are plotted in Figure 5. The largest improvement in error rate is obtained by both AL1.t and AL2.t (25.5% error rate), followed by AL2.d (25.9%). Quoted results for the active learning methods are after querying 100 training set objects (10 AL batches of size 10). AL1.d lags well behind the performance of these other AL querying functions. None of the other methods produces a significant decrease in the error rate of the classifier. Indeed, the ST and CT approaches cause an *increase* in the overall error rate. IW produces a slight decrease in the error rate, by an average of 0.4%, which represents 3 correct classifications. An important observation is that the AL.rand approach of randomly queuing observations for manual labeling does not perform well compared to the more principled AL approaches.

Figure 6 depicts the error rate of the AL approaches as a function of the total number of objects queried. Between the AL1 and AL2 metrics, there is no clear winner, but once large numbers of samples have been observed AL2.d and AL2.t perform better than their AL1 counterparts. We also find in Figure 6 that the AL.d approaches—where objects are drawn with probability proportional to the AL criterion—perform worse than the approaches that always select the top AL candidates. This is unexpected, as selecting only the top methods in batch mode produces samples of objects from the same region in feature space, causing an inefficient use of follow-up resources. However, this observed better performance by the AL.t strategies may be an artifact of using small batch sizes (10 objects); in the application of active learning to ASAS, we typically use batch sizes > 50.

Active learning is able to significantly improve the classification error rate on the set of OGLE & *Hipparcos* testing data because it selectively probes regions of feature space where class labels, if known, would most influence the classifications of a large number of testing data. For the OGLE and *Hipparcos* variable star data, sets of low-amplitude, short-period stars are selected by the AL algorithm, which in turn improve the error rates within the science classes populated by these types of stars, without increasing error rates within the classes that are highly sampled by the training set. We make this more concrete in Table 1, where the classifier error rates within a few select classes are shown. The active learning classifiers show substantial improvement, on average, over the default Random Forest for the classes which are most under-sampled by the training data and show no increase in the error rates for the classes that are most over-represented in the training set.

5. ALLSTARS: Active Learning Light Curve Web Interface

We developed the ALLSTARS (Active Learning Lightcurve classification Service) web based tool as the crowdsourcing user interface to our active learning software. For each

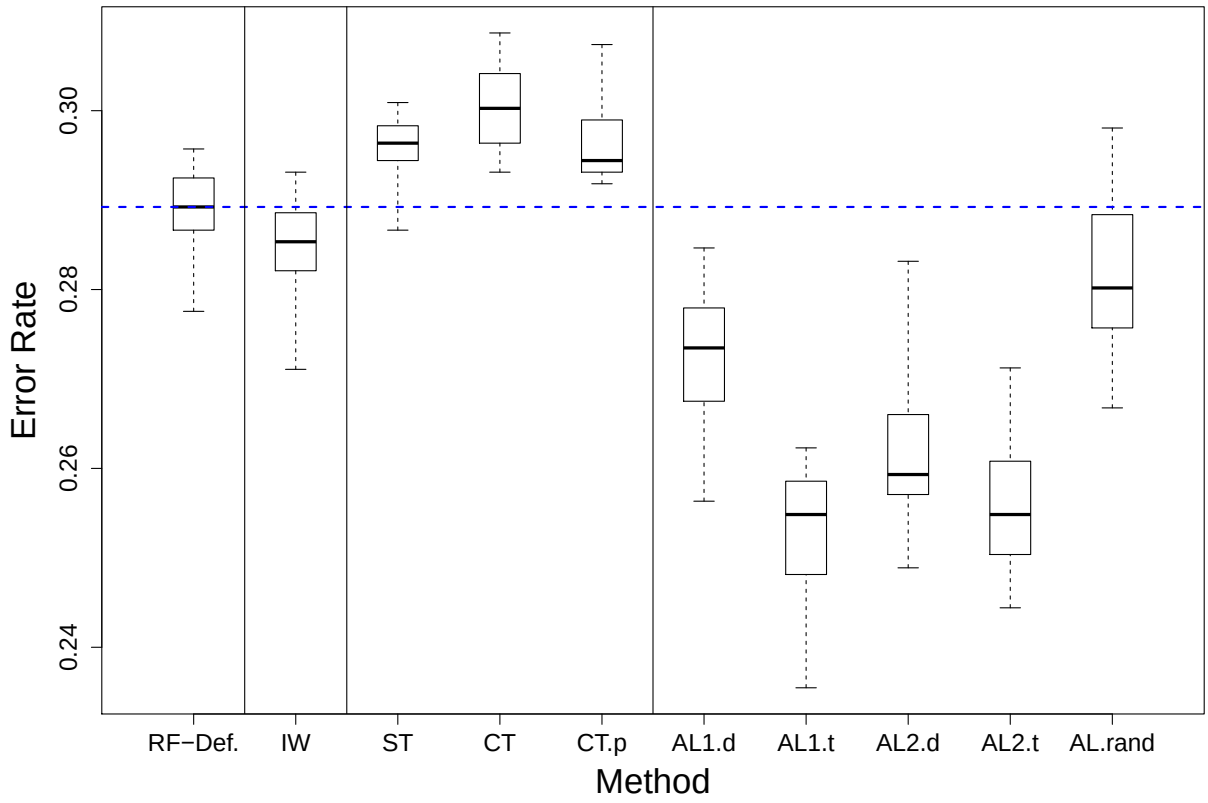

Fig. 5.— Error rates, evaluated over the testing set, of 10 different methods applied to the OGLE & *Hipparcos* simulated data set of 771 training and 771 testing samples. Due to sample selection bias, the default Random Forest (RF-Def.) is ineffective. Importance weighting (IW) improves upon the RF only slightly. The co-training and self-training methods produce an increased error rate. Only the active learning approaches yield any significant gains in the performance of the classifier over the testing set. Note that the AL methods were evaluated over those testing data not in the active learning sample. No large difference is found between the two AL metrics, but both outperform the random selection of AL samples. Note that each boxplot displays the 25th and 75th quantiles as the edges of the boxes, with the center line denoting the median and the whiskers extending to the minimum and maximum.

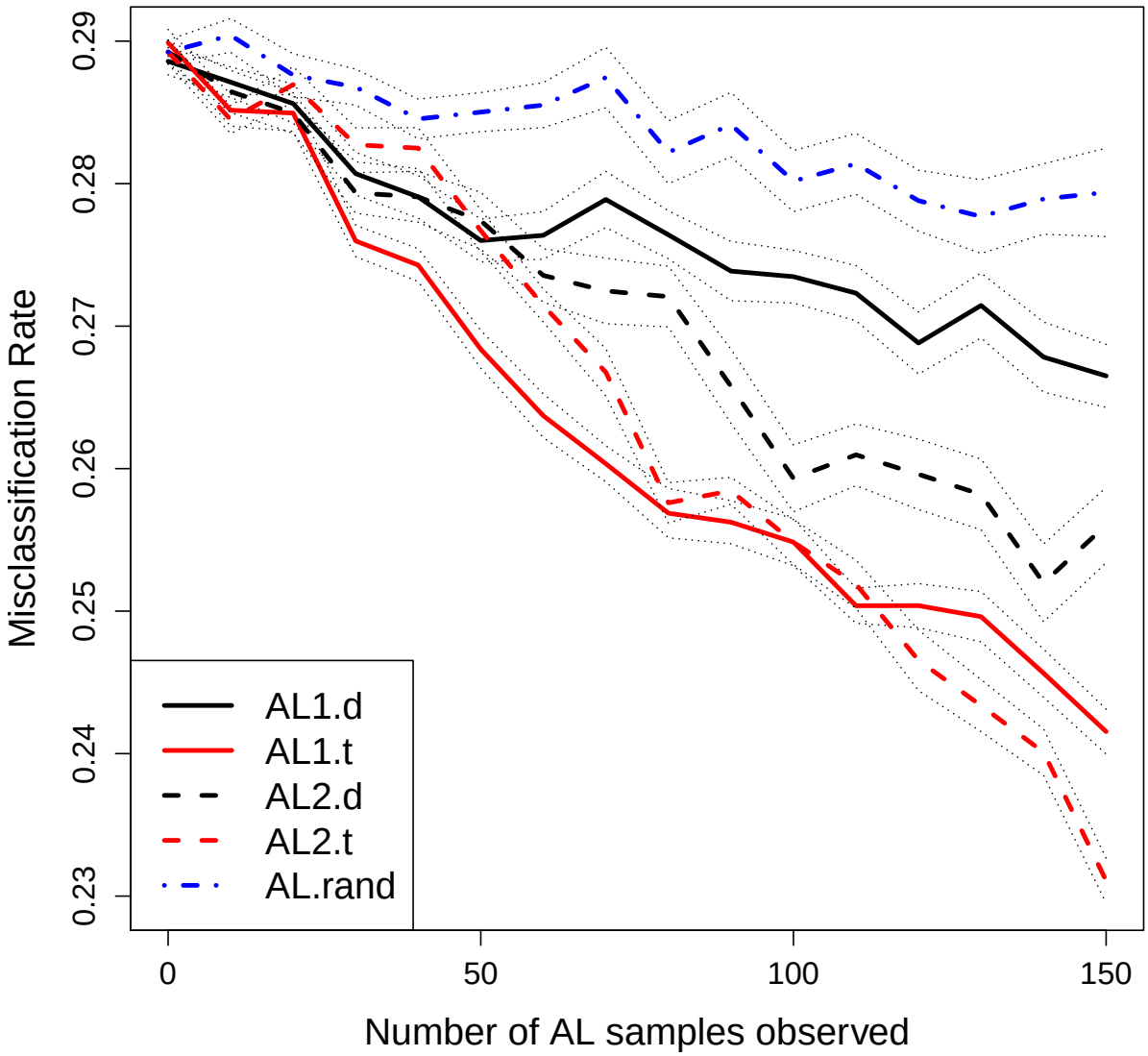

Fig. 6.— Performance of the active learning approaches for the OGLE & *Hipparcos* classification experiment. Both AL1 and AL2 dominate the performance of AL.rand, but there is no clear winner between these two approaches. AL1.t performs best for the first few iterations, but is overtaken by AL2.t after 100 samples are queried. AL2.d performs significantly better than AL1.d after about 50 iterations. For each method, the mean error rate—evaluated over the testing set not included in the AL sample—is plotted along with ±1 standard error bands.

active learning iteration, this website displays to a user the set of AL-queried sources. For each source, users are given access to eight external web resources in addition to several feature space visualizations to facilitate manual classification of that source. A screen shot of the ALLSTARS web interface is in Figure 7. Additionally, for each source a user may make a science classification, a rating of their confidence, a data quality classification, can tag the source as interesting, and also may provide comments and store a manually-determined period. This set of information is used to determine the class of each of the active learning queried sources and to decide which subset of those sources to add to the training set.

ALLSTARS was built using a combination of javascript, PHP, and Python which accesses a MySQL database. Backend feature generators, active learning and classification algorithms were implemented using a combination of Python, C and R. The interactive plots are generated using the Flot jQuery[13] package. External resources made available for classifying each source are:

- NED Extinction Calculator: http://ned.ipac.caltech.edu/forms/calculator.html
- SDSS DR7 Explorer: http://cas.sdss.org/dr7/en/tools/explore/obj.asp
- SDSS DR7 Navigate Tool: http://cas.sdss.org/dr7/en/tools/chart/navi.asp
- SIMBAD Query by coordinates: http://simbad.u-strasbg.fr/simbad/sim-fcoo
- 2MASS Interactive Image (J-band): http://irsa.ipac.caltech.edu/applications/2MASS/IM/interactive.html
- SkyView Original DSS image: http://skyview.gsfc.nasa.gov/cgi-bin/query.pl
- NVO DataScope: http://heasarc.gsfc.nasa.gov/cgi-bin/vo/datascope/init.pl
- DotAstro LightCurve Warehouse: http://dotastro.org/

The initial page for a source includes two color-color plots: $B-J$ vs. $J-K$ and $J-H$ vs. $H-K$, using colors from the SIMBAD source which best matches the location of the given source. The source is also shown on a log-amplitude vs. log-period plot, with sources from the initial *Hipparcos* and OGLE training set displayed in the background. These sources are discriminated using 21 different colors which represent most science classes to which the user may classify. An interactive magnitude vs. time light curve plot is also shown, with

[13] Flot is a Javascript plotting library downloadable from http://code.google.com/p/flot/.

options to display it either unfolded, folded on any of the three most significant periods, or folded using a user entered or zoom-box generated period. The chosen period also updates a black circle on the amplitude-period plot. Also available on this initial page are the top three algorithm classifications and their confidences.

ALLSTARS can be used to display any source available in the http://dotAstro.org Lightcurve Warehouse, allowing a registered user to make a science classification, assess data quality, note a manually found period, or add additional comments for that source. This web interface is an extremely useful tool, not only for performing active learning for variable star classification, but also for following up on outliers discovered via unsupervised learning, for finding typical examples of light curves of desired science classes, and to manually search through subsets of the dotAstro data warehouse.

6. Application of Active Learning to classify ASAS Variable Stars

We use the active learning methodology presented in §3.3 to classify all of ACVS (see §2.1) starting with the combined *Hipparcos* and OGLE training set. We employ the S_2 AL query function (Equation 5), treating it as a probability distribution (AL2.d in §4), and selecting 50 AL candidates on each of 9 iterations (except for the first iteration, where 75 AL candidates were chosen). For a cost function, we employ data from our first AL iteration to train a logistic regression model to predict cost as a function of freq_signif,the statistical significance of the estimated first frequency[14].

A total of 11 users classified sources using the ALLSTARS web interface. To help train new users, the beginning of each iteration was populated with 14-18 high-confidence sources[15]. A total of 615 sources were observed by users (this represents 1.2% of the ACVS catalog). The average user classified 137 sources, with a range from 21–474. User responses were combined using the crowdsourcing methodology in §3.3.3. This led to the inclusion of 415 ASAS sources (67% of all sources that were studied manually) into the training set. In Figure 8 we plot the AL queried data from one iteration in the amplitude-period plane, highlighting those which were selected for inclusion in the training set.

[14]This will bias us away from selecting aperiodic sources, such as T Tauri. However, this is a reasonable approach because (1) there are simply too many aperiodic sources that are impossible to classify manually, and (2) in AL we draw a random sample from the $S_2(\mathbf{x}') * (1 - C(\mathbf{x}'))$ meaning that we are still very likely to select some interesting aperiodic sources with high S_2 score.

[15]As to not throw away useful annotations, these classifications were used along with the AL samples.

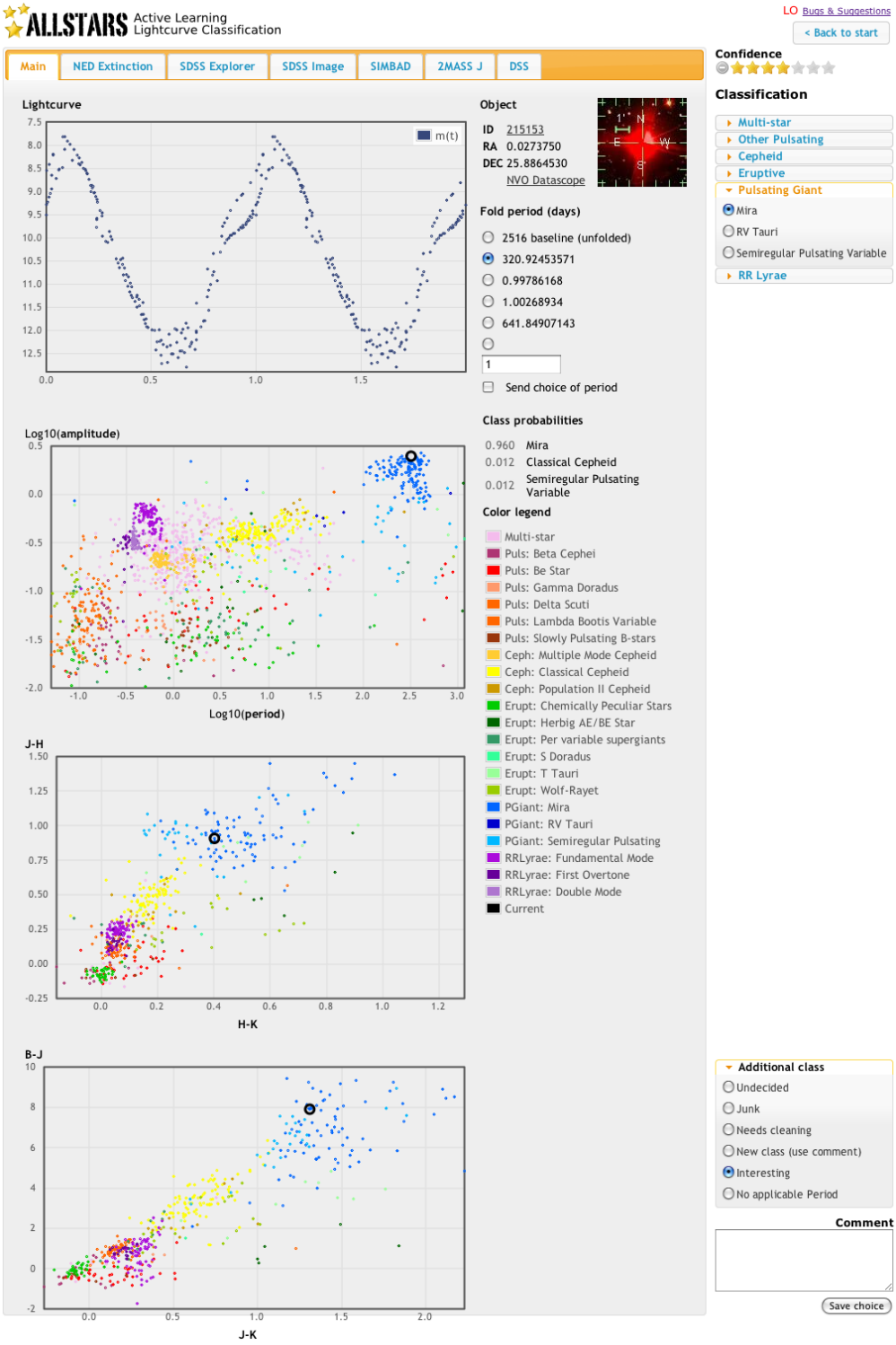

Fig. 7.— Screen shot of the ALLSTARS web interface. Here, a Mira variable from the ASAS survey has been queried by the user. From top to bottom, the user is provided a (folded) ASAS light curve of the source, its location in amplitude-period space, its $J-H$ vs. $H-K$, and its $B-J$ vs. $J-K$ colors. At the top of the page are several tabs which link to external resources. On the left margin the user can make and submit a classification for the source.

As described in §2.1, the default RF only attains a 65.5% agreement with the ACVS catalog. After 9 AL iterations, this jumps to 79.5%, an increase of 21% in agreement rate. The proportion of ACVS sources in which we are confident (defined as $\max_y \widehat{P}_{\rm RF}(y|\mathbf{x}) > 0.5$) climbs from 14.6% to 42.9%. This occurs because the selected ASAS data that are subsequently used as training data fill in sparse regions of training set feature space, thus increasing the chance that ASAS sources are in close proximity to training data and increasing the RF maximum probabilities. As a function of the AL iteration, both the ACVS agreement rate and the proportion of confident classifications achieved by our classifier are plotted in Figure 9. The full evolution of the distribution of $\max_y \widehat{P}_{\rm RF}(y|\mathbf{x})$ is plotted in Figure 10. As the iterations proceed, power is shifted from low to high probabilities.

In Figure 11 we plot a table of the correspondence between our classifications after 9 AL iterations and the ACVS class. Comparing to Figure 3, we see that the AL predictions more closely match the ACVS labels across most science classes. For example, correspondence in the Classical Cepheid class raised from 24% to 61%, RR Lyrae, FM from 79% to 93%, Delta Scuti from 22% to 60%, and Chemically Peculiar from 1% to 72%. We have also identified a number of candidates for more rare classes, such as 117 RV Tauri, 177 Pulsating Be stars, and 43 T Tauri. Additionally, the number of RR Lyrae, DM candidates, which was artificially high for the original RF classifier, has diminished from 9109 to 442. A summary of our ASAS classification active learning, by class, is given in Table 2.

As a consequence of performing active learning on the ASAS data set, we were able to detect the presence of 3 additional science classes of red giant stars. These classes were discovered by one of the AL users upon realizing that many of the queried pulsating red giant stars were low-amplitude with 10-75 day periods. A literature search revealed that these stars naturally break into small-amplitude red giant A and B subclasses (SARG A and B, see Wray et al. 2004). Furthermore, the presence of a red giant subclass of long secondary period (LSP, Soszyński 2007) stars was discovered and added. Via active learning, our classifier identified 3699 SARG A, 8823 SARG B, and 5889 LSP candidates.

Our final experiment is to compare our classification results using active learning with the classification of a Random Forest that is trained on the ACVS labels. The aim of this study is to determine whether our classifier's disagreement with ACVS is due principally to inadequacies in our classifier or mistakes and inconsistencies in the ACVS classifications. Using a 5-fold cross-validation on the ACVS labels, a RF classifier finds a 90% agreement rate with ACVS (compared to 79.5% using AL). A substantial proportion of our disagreement with ACVS results from the use of a finer taxonomy (where, e.g., we can correctly identify some of ACVS Mira candidates as Semi-Regular PVs). Within the classes in which the AL classifier has its poorest agreement with ACVS, the ACVS RF also does not do well: for

Pop. II Cepheids, the ACVS RF finds only 37% agreement (compared to 0%), for Multi. Mode Cepheids it finds 45% agreement (29%), and in Beta Cepheid it finds 0% agreement (0%). This evidence points to the conclusion that the disagreement of our AL classifier to ACVS within these classes is due more to lack of self-consistency of those classes in ACVS (due either to mistakes in ACVS or absence of crucial features) than to any shortcomings in the active learning methodology.

7. Conclusions

We have described the problem of sample selection bias (a.k.a. covariate shift) in supervised learning on astronomical data sets. Though supervised learning has shown great promise in automatically analyzing large astrophysical databases, care must be taken to account for the biases that occur due to distributional differences between the training and testing sets. Here, we have argued that sample selection bias is a common problem in astronomy, primarily because the subset of well-studied astronomical objects typically forms a biased sample of intrinsically brighter and nearby sources. In this paper, we showed the detrimental influence of sample selection bias on the problem of supervised classification of variable stars.

To alleviate the effects of sample selection bias, we proposed a few different methods. We find, on a toy problem using *Hipparcos* and OGLE light curves, that active learning performs significantly better than other methods such as importance weighting, co-training, and self-training. Furthermore, we argue that AL is a suitable method for many astronomical problems, where follow-up resources are usually available (albeit with limited availability). Active learning simply gives a principled way to determine which sources, if followed up on, would help the supervised algorithm the most. We show that in classifying variable stars from the ASAS survey, AL produces hugely significant improvements in performance within only a handful of iterations. Our `ALLSTARS` web interface was critical in this work, as was the participation of knowledgeable ("trained expert") users and sophisticated crowdsourcing methods.

Though we have introduced a couple of AL querying functions, many different options are available. In particular, we argue that the S_2 criterion is appropriate for our classification problem because it targets objects whose inclusion in the training set would induce the largest overall change in the classification predictions over the testing set. However, for each problem, a different AL function will be appropriate. The pertinent querying function depends on the problem at hand, the type of response being modeled, and the kind of supervised algorithm employed, and typically several different choices are available.

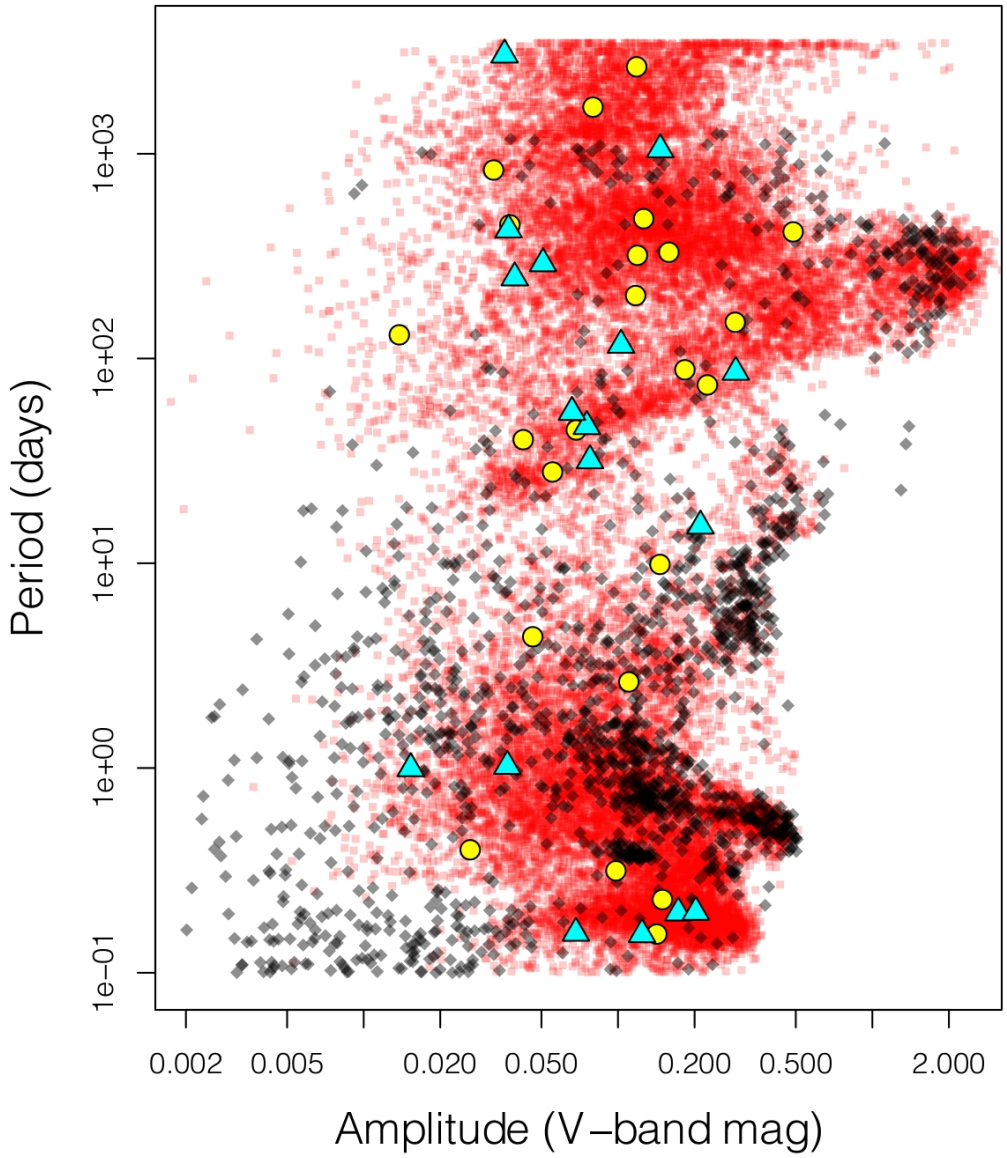

Fig. 8.— Active learning samples on a single iteration of the algorithm. Yellow circles signify points that at least 65% of users were able to classify. These points are included on subsequent iterations of the algorithm. Cyan triangles signify variable stars that were queried, but for which fewer than 65% of users were able to classify. Black diamonds and red squares are the original training and testing data, as in Figure 1.

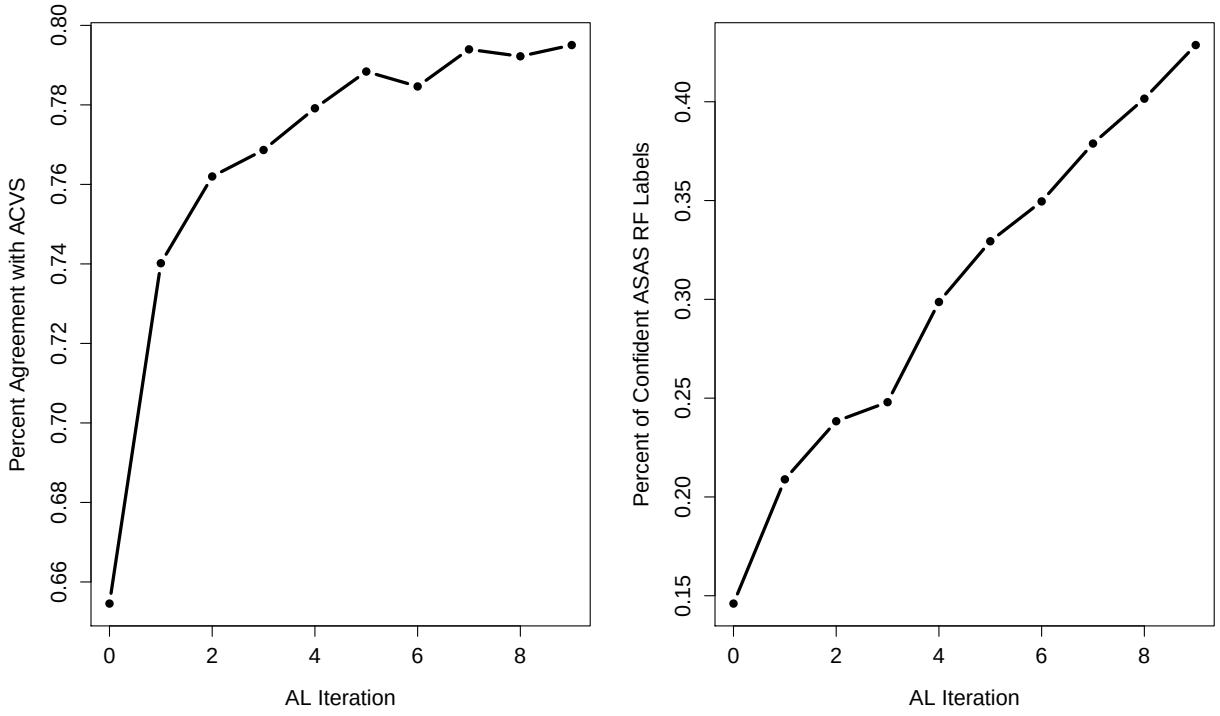

Fig. 9.— Left: Percent agreement of the Random Forest classifier with the ACVS labels, as a function of AL iteration. Right: Percent of ASAS data with confident RF classification (posterior probability > 0.5), as a function of AL iteration. In the percent agreement with ACVS metric, performance increases dramatically in the first couple of iterations and then slowly levels off. In the percent of confident RF labels, the performance increases steadily.

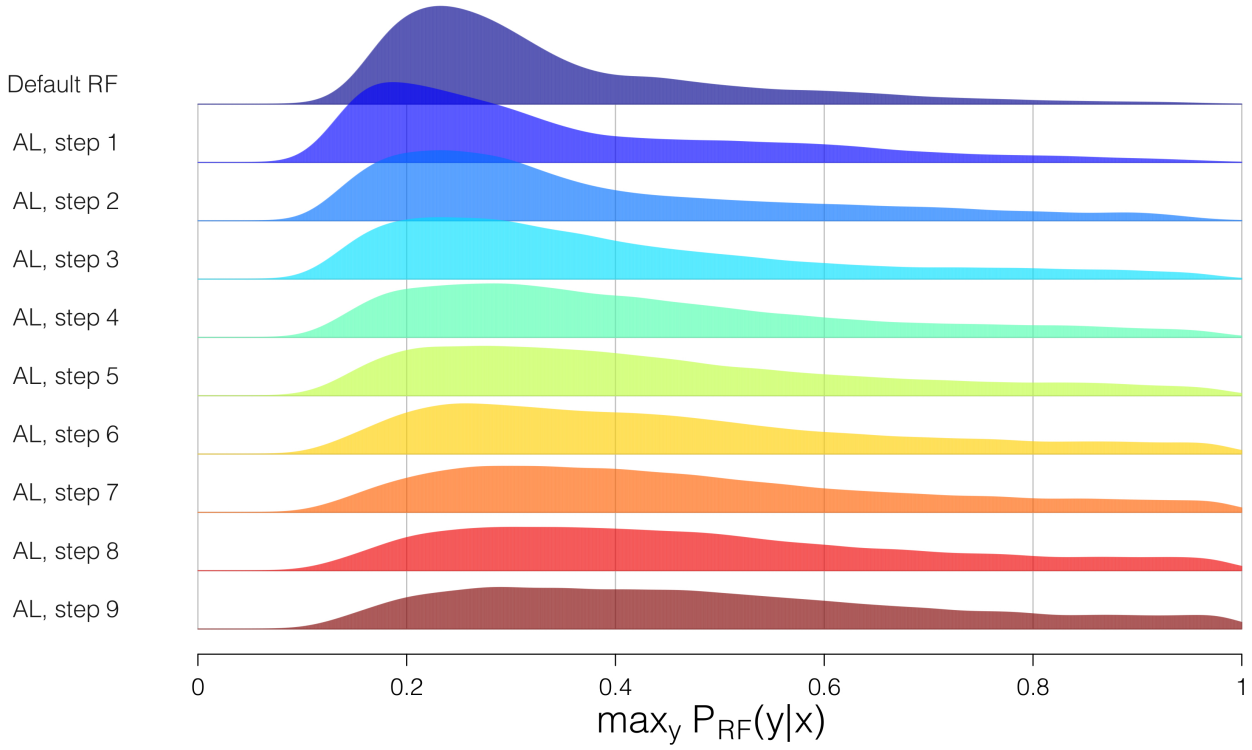

Fig. 10.— Distribution of the Random Forest $\max_y \widehat{P}_{\rm RF}(y|\mathbf{x})$ values for the ASAS data, as a function of AL iteration. For the default RF classifier, most values are smaller than 0.4, meaning that the classifier is confident on very few sources. As the AL iterations proceed, much of the mass of the distribution gradually shifts toward larger values. The distribution slowly becomes multimodal: for a slim majority of sources, the algorithm has high confidence, while for a substantial subset of the data the algorithm remains unsure of the classification.

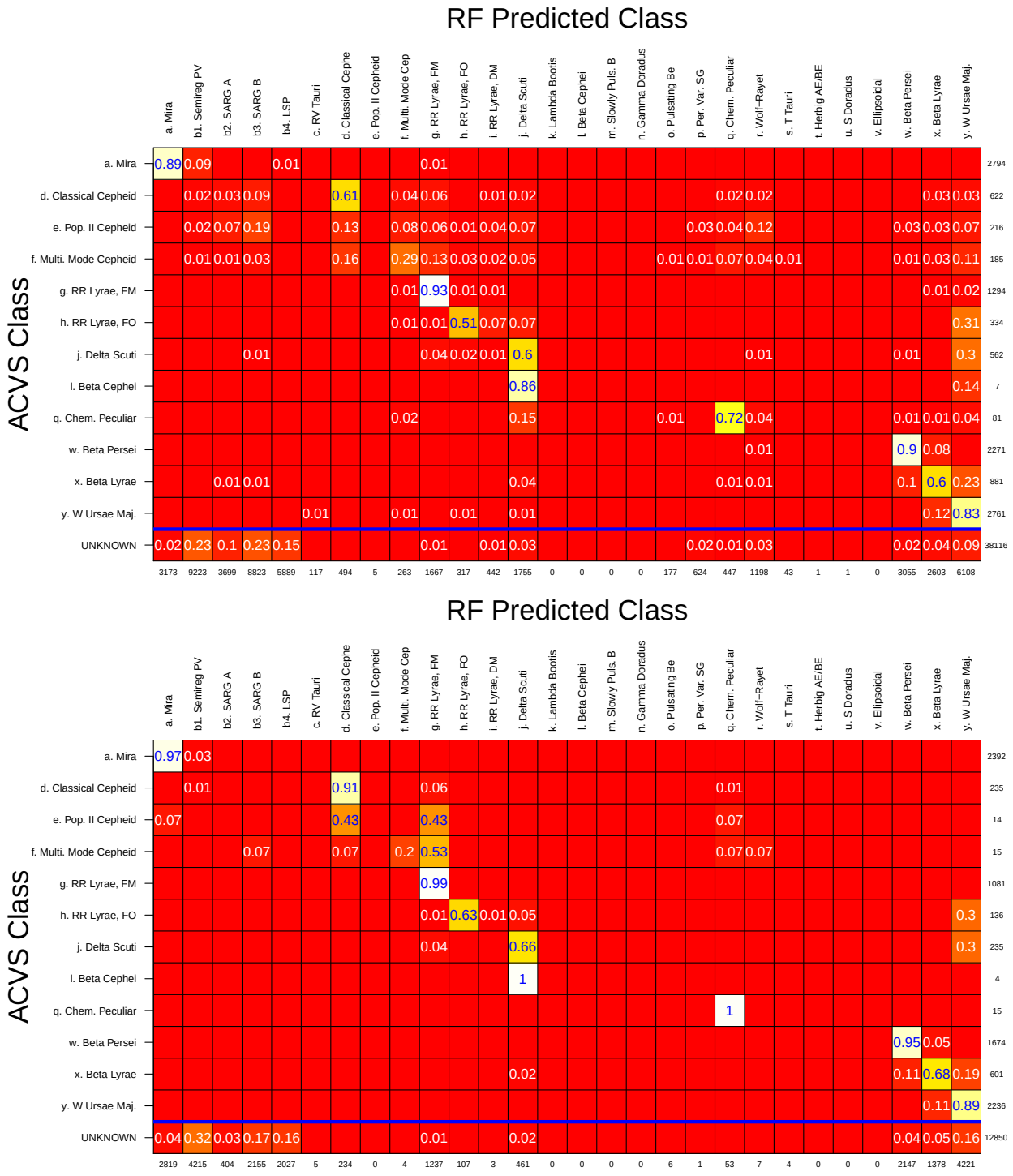

Fig. 11.— Top: Classifications of the active learning RF classifier after 9 iterations of AL. Compared to Figure 3, there is a closer correspondence to the ACVS class labels (y axis). Notably, the RRL, DM artifact has largely disappeared. Bottom: Same for only sources with classification probability > 0.5. Here, the agreement is even higher. The main confusion is in classifying ACVS RR Lyrae, FO and Delta Scuti as W Ursae Maj.

One common cause of sample selection bias in variable star classification is that data from older surveys—whose sources have typically been observed over many epochs—are commonly used to classify data from ongoing surveys, whose sources contain many fewer epochs of observation. In addition to AL, other viable approaches to this particular problem are those of *noisification*, where the training set light curves are artificially modified to resemble those of the testing set, and *denoisification*, where each testing light curve is matched to a (clean) training light curve. These techniques are currently being studied by Long et al. (2011).

Our discussion of sample selection bias has revolved around the use of non-parametric tools (and in particular Random Forests). For the types of complicated classification and regression problems in astrophysics, flexible non-parametric methods are usually necessary. However, in many applications, parametric models are appropriate. In this parametric setting, there are several methods of overcoming sample selection bias, including Bayesian experimental design (Chaloner & Verdinelli 1995).

We conclude by emphasizing the importance of treating sample selection bias for future petabyte-scale surveys such as Gaia and LSST. These upcoming surveys will collect data at such massive rates that rare, unexpected, and yet-undiscovered sources will be prevalent in their data streams. Furthermore, due to superior optics and cameras, they will probe different populations of sources than observed by any previous mission. For these reasons, any conceivable training set constructed prior to the start of these surveys will have significant sample selection bias. Through active learning, we now have a principled way to queue sources for targeted follow-up in order to augment training sets to optimize the performance of machine-learned algorithms and to maximize the science that these missions produce.

The authors acknowledge the generous support of a CDI grant (#0941742) from the National Science Foundation. This work was performed in the CDI-sponsored Center for Time Domain Informatics (`http://cftd.info`). N.R.B. is supported through the Einstein Fellowship Program (NASA Cooperative Agreement: NNG06DO90A). We acknowledge the help of Christopher Klein, Adam Morgan, and Brad Cenko in helping to manually classify variable stars with `ALLSTARS`. We also thank Laurent Eyer for helpful conversations.

A. Derivation of Active Learning Random Forest Metric

In this Appendix, we derive Equation 5 as an AL selection criterion function. Our starting point is to select instances that maximize the total amount of change in the RF predicted probabilities of the testing data $\mathbf{x} \in \mathcal{U}$. Assuming we have a labeled training set

$\mathcal{L}$, the total amount of change in the testing RF probabilities due to the addition of $\mathbf{x}'$ to $\mathcal{L}$ is

$$S_2(\mathbf{x}') = \sum_{\mathbf{x}\in\mathcal{U}} ||\widehat{P}_{\mathrm{RF},\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})||_1 \tag{A1}$$

where we use the notation $\widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})$ to denote the Random Forest probability that the label for instance $\mathbf{x}$ is y, where the RF is trained on the set $\mathcal{L}$. To simplify notation, we rewrite $S_2(\mathbf{x}') = \sum_{\mathbf{x}\in\mathcal{U}} \Delta(\mathbf{x}', \mathbf{x})$, where

$$\begin{aligned} \Delta(\mathbf{x}', \mathbf{x}) &= ||\widehat{P}_{\mathrm{RF},\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})||_1 &\text{(A2)} \\ &= \sum_{y=1}^{C} |\widehat{P}_{\mathrm{RF},\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})| &\text{(A3)} \end{aligned}$$

where C is the total number of classes. Equation A3 follows from the definition of ℓ_1 norm.

From Equation 2, $\widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x}) = \frac{1}{B}\sum_b \theta_{b,\mathcal{L}}(y|\mathbf{x})$, where $\theta_{b,\mathcal{L}}$ is the bth decision tree in the Random Forest built on training set $\mathcal{L}$. Now, assuming that the addition of $\mathbf{x}'$ to $\mathcal{L}$ does not change the structure of any of the B decision trees[16], we can compute the change in the decision tree estimate in terminal node $T_b(\mathbf{x}')$ of tree b. Let $Y(\mathbf{x}')$ denote the true label of source $\mathbf{x}'$. In adding $\mathbf{x}'$ to $\mathcal{L}$, decision tree b changes to

$$\theta_{b,\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) = \begin{cases} \frac{n_b(\mathbf{x}')\theta_{b,\mathcal{L}}(y|\mathbf{x}) + I(Y(\mathbf{x}')=y)}{n_b(\mathbf{x}')+1} & \text{if } \mathbf{x} \in T_b(\mathbf{x}') \\ \theta_{b,\mathcal{L}}(y|\mathbf{x}) & \text{if } \mathbf{x} \notin T_b(\mathbf{x}') \end{cases} \tag{A4}$$

where $n_b(\mathbf{x}')$ is the number points in $\mathcal{L}$ that fall in $T_b(\mathbf{x}')$ and $I(\cdot)$ is a boolean indicator function. The way to understand Equation A4 is that the empirical probability estimates in the terminal node $T_b(\mathbf{x}')$ update to include $Y(\mathbf{x}')$, while the rest of the terminal nodes remain unchanged.

Therefore, if $\mathbf{x} \in T_b(\mathbf{x}')$, then the amount of change in the probability estimate is

$$\begin{aligned} \theta_{b,\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \theta_{b,\mathcal{L}}(y|\mathbf{x}) &= \frac{n_b(\mathbf{x}')\theta_{b,\mathcal{L}}(y|\mathbf{x}) + I(Y(\mathbf{x}')=y)}{n_b(\mathbf{x}')+1} - \theta_{b,\mathcal{L}}(y|\mathbf{x}) &\text{(A5)} \\ &= \frac{I(Y(\mathbf{x}')=y) - \theta_{b,\mathcal{L}}(y|\mathbf{x})}{n_b(\mathbf{x}')+1} &\text{(A6)} \end{aligned}$$

while in all other terminal nodes of b, the change is 0.

[16] In reality, the structure of the trees may change, but analyzing the effect on the RF of adding $\mathbf{x}'$ is intractable if the trees are allowed to change substantially.

Using the result in Equation A6 for tree b, we can compute the total amount of change, $\Delta(\mathbf{x}', \mathbf{x})$, across the entire RF by averaging the response over the B trees:

$$\Delta(\mathbf{x}', \mathbf{x}) = \sum_{y=1}^{C} \left| \frac{1}{B} \sum_{b:\mathbf{x} \in T_b(\mathbf{x}')} \frac{I(Y(\mathbf{x}') = y) - \theta_{b,\mathcal{L}}(y|\mathbf{x})}{n_b(\mathbf{x}') + 1} \right| \tag{A7}$$

where $n_b(\mathbf{x}')$ and $\theta_{b,\mathcal{L}}(y|\mathbf{x})$ are quantities computed for each of the B trees. However, these entities are costly to store for large B and are not available in most RF implementations. To compute Equation A7 directly from the standard RF output (e.g., proximity matrices and predicted probabilities), we need two approximations: (1) $n_b(\mathbf{x}') = \sum_{\mathbf{z} \in \mathcal{L}} \rho(\mathbf{x}', \mathbf{z})$, i.e., replace the number of objects in $T_b(\mathbf{x}')$ by the average number over the B trees, and (2) $\theta_{b,\mathcal{L}}(y|\mathbf{x}) = \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x})$, i.e., approximate the probability vector of each tree by the RF probability. Using these approximations we have that

$$\Delta(\mathbf{x}', \mathbf{x}) \approx \sum_{y=1}^{C} \left| \frac{1}{B} \sum_{b:\mathbf{x} \in T_b(\mathbf{x}')} \frac{I(Y(\mathbf{x}') = y) - \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x})}{\sum_{\mathbf{z} \in \mathcal{L}} \rho(\mathbf{x}', \mathbf{z}) + 1} \right| \tag{A8}$$

$$= \frac{1}{\sum_{\mathbf{z} \in \mathcal{L}} \rho(\mathbf{x}', \mathbf{z}) + 1} \sum_{y=1}^{C} \left| \frac{1}{B} \sum_{b=1}^{B} I(\mathbf{x} \in T_b(\mathbf{x}')) \left(I(Y(\mathbf{x}') = y) - \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x}) \right) \right| \tag{A9}$$

$$= \frac{1}{\sum_{\mathbf{z} \in \mathcal{L}} \rho(\mathbf{x}', \mathbf{z}) + 1} \sum_{y=1}^{C} \left| I(Y(\mathbf{x}') = y) - \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x}) \right| \frac{1}{B} \sum_{b=1}^{B} I(\mathbf{x} \in T_b(\mathbf{x}')) \tag{A10}$$

However, we cannot directly compute this equation because do not know *a priori* what the value of $Y(\mathbf{x}')$ is. Luckily, we can find a lower bound on the term in Equation A10 that includes $Y(\mathbf{x}')$, and use this to produce a *conservative* estimate of $\Delta(\mathbf{x}', \mathbf{x})$. Our lower bound is

$$\begin{aligned} \sum_{y=1}^{C} |I(Y(\mathbf{x}') = y) - \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x})| &= (1 - \widehat{P}_{\text{RF},\mathcal{L}}(Y(\mathbf{x}')|\mathbf{x})) + \sum_{y \neq Y(\mathbf{x}')} \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x}) \\ &\geq 1 - \widehat{P}_{\text{RF},\mathcal{L}}(Y(\mathbf{x}')|\mathbf{x}) \\ &\geq 1 - \max_y \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x}) \end{aligned}$$

Therefore, the smallest possible change in the RF probabilities is given by

$$\Delta(\mathbf{x}', \mathbf{x}) = \frac{1 - \max_y \widehat{P}_{\text{RF},\mathcal{L}}(y|\mathbf{x})}{\sum_{\mathbf{z} \in \mathcal{L}} \rho(\mathbf{x}', \mathbf{z}) + 1} \frac{1}{B} \sum_{b=1}^{B} I(\mathbf{x} \in T_b(\mathbf{x}')) \tag{A11}$$

which is a metric that can be computed.

Now substituting the result of Equation A11 into Equation A1, we have that

$$S_2(\mathbf{x}') = \sum_{\mathbf{x}\in\mathcal{U}} \frac{1-\max_y \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})}{\sum_{\mathbf{z}\in\mathcal{L}}\rho(\mathbf{x}',\mathbf{z})+1}\frac{1}{B}\sum_{b=1}^{B} I(\mathbf{x}\in T_b(\mathbf{x}')) \quad \text{(A12)}$$

$$= \sum_{\mathbf{x}\in\mathcal{U}} \frac{1-\max_y \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})}{\sum_{\mathbf{z}\in\mathcal{L}}\rho(\mathbf{x}',\mathbf{z})+1}\rho(\mathbf{x}',\mathbf{x}) \quad \text{(A13)}$$

which is the AL criterion, S_2, presented in Equation 5.

REFERENCES

Auvergne, M., et al. 2009, A&A, 506, 411

Ball, N. M., Loveday, J., Fukugita, M., Nakamura, O., Okamura, S., Brinkmann, J., & Brunner, R. J. 2004, MNRAS, 348, 1038

Bloom, J. S., & Richards, J. W. 2011, arXiv/1104.3142

Blum, A., & Mitchell, T. 1998, in Proceedings of the eleventh annual conference on Computational learning theory, ACM, 92–100

Bonfield, D. G., Sun, Y., Davey, N., Jarvis, M. J., Abdalla, F. B., Banerji, M., & Adams, R. G. 2010, MNRAS, 405, 987

Breiman, L. 2001, Machine learning, 45, 5

Brinker, K. 2003, in In Proceedings of the 20th International Conference on Machine Learning (AAAI Press), 59–66

Butler, N. R., & Bloom, J. S. 2011, AJ, 141, 93

Carliles, S., Budavári, T., Heinis, S., Priebe, C., & Szalay, A. S. 2010, ApJ, 712, 511

Chaloner, K., & Verdinelli, I. 1995, Statistical Science, 10, 273

Cohn, D. 1996, Neural Networks, 9, 1071

Collister, A. A., & Lahav, O. 2004, PASP, 116, 345

D'Abrusco, R., Staiano, A., Longo, G., Brescia, M., Paolillo, M., De Filippis, E., & Tagliaferri, R. 2007, ApJ, 663, 752

Debosscher, J., Sarro, L. M., Aerts, C., Cuypers, J., Vandenbussche, B., Garrido, R., & Solano, E. 2007, A&A, 475, 1159

Debosscher, J., et al. 2009, A&A, 506, 519

Donmez, P., Carbonell, J., & Schneider, J. 2009, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 259–268

Dubath, P., et al. 2011, arXiv/1101.2406

Gao, D., Zhang, Y.-X., & Zhao, Y.-H. 2008, MNRAS, 386, 1417

Goldman, S., & Zhou, Y. 2000, in Proceedings of the 17th International Conference on Machine Learning, Citeseer

Heckman, J. 1979, Econometrica: Journal of the econometric society, 153

Huang, J., Smola, A., Gretton, A., Borgwardt, K., & Scholkopf, B. 2007, Advances in neural information processing systems, 19, 601

Huertas-Company, M., Rouan, D., Tasca, L., Soucail, G., & Le Fèvre, O. 2008, A&A, 478, 971

Kessler, R., et al. 2010, PASP, 122, 1415

Lewis, D., & Gale, W. 1994, in Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Springer-Verlag New York, Inc., 3–12

Lintott, C. J., et al. 2008, MNRAS, 389, 1179

Liu, Y. 2004, Journal of chemical information and computer sciences, 44, 1936

Long, J., El Karoui, N., Rice, J., Richards, J. W., & Bloom, J. S. 2011, in Preparation

LSST Science Collaborations et al. 2009, arXiv/0912.0201

Matthews, D. J., & Newman, J. A. 2010, ApJ, 721, 456

Newling, J., et al. 2011, MNRAS, 545

Nigam, K., & Ghani, R. 2000, in Proceedings of the ninth international conference on Information and knowledge management, ACM, 86–93

Olsson, F., & Tomanek, K. 2009, in Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, 138–146

Perryman, M., et al. 1997, Astronomy and Astrophysics, 323, L49

Perryman, M. A. C., et al. 2001, A&A, 369, 339

Pojmanski, G. 1997, Acta Astron., 47, 467

—. 2000, Acta Astron., 50, 177

Pojmański, G. 2001, in Astronomical Society of the Pacific Conference Series, Vol. 246, IAU Colloq. 183: Small Telescope Astronomy on Global Scales, ed. B. Paczynski, W.-P. Chen, & C. Lemme, 53

—. 2002, Acta Astron., 52, 397

Pojmanski, G., Pilecki, B., & Szczygiel, D. 2005, Acta Astronomica, 55, 275

Quadri, R. F., & Williams, R. J. 2010, ApJ, 725, 794

Richards, G. T., et al. 2009, AJ, 137, 3884

Richards, J. W., Homrighausen, D., Freeman, P. E., Schafer, C. M., & Poznanski, D. 2011a, arXiv/1103.6034

Richards, J. W., et al. 2011b, ApJ, 733, 10

Roy, N., & McCallum, A. 2001, in Machine Learning-International Workshop, Citeseer, 441–448

Schulz, A. E. 2010, ApJ, 724, 1305

Settles, B. 2010, Active Learning Literature Survey, Tech. rep., CS Tech. Rep. 1648, University of Wisconsin-Madison

Shimodaira, H. 2000, Journal of Statistical Planning and Inference, 90, 227

Smith, K. W., Bailer-Jones, C. A. L., Klement, R. J., & Xue, X. X. 2010, A&A, 522, A88+

Soszyński, I. 2007, ApJ, 660, 1486

Soszyński, I., et al. 2011, Acta Astron., 61, 1

Sugiyama, M., Krauledat, M., & Müller, K. 2007, The Journal of Machine Learning Research, 8, 985

Sugiyama, M., & Müller, K. 2005, Statistics & Decisions, 23, 249

Sypniewski, A. J., & Gerdes, D. W. 2011, in Bulletin of the American Astronomical Society, Vol. 43, American Astronomical Society Meeting, 150.04

Tong, S., & Chang, E. 2001, in Proceedings of the ninth ACM international conference on Multimedia, ACM, 107–118

Tong, S., & Koller, D. 2002, The Journal of Machine Learning Research, 2, 45

Tsalmantza, P., et al. 2007, A&A, 470, 761

Tur, G., Hakkani-Tur, D., & Schapire, R. 2005, Speech Communication, 45, 171

Udalski, A., Soszynski, I., Szymanski, M., Kubiak, M., Pietrzynski, G., Wozniak, P., & Zebrun, K. 1999a, Acta Astron., 49, 1

—. 1999b, Acta Astron., 49, 223

—. 1999c, Acta Astron., 49, 437

Vlachos, A. 2008, Computer Speech & Language, 22, 295

Wadadekar, Y. 2005, PASP, 117, 79

Wozniak, P. R., Udalski, A., Szymanski, M., Kubiak, M., Pietrzynski, G., Soszynski, I., & Zebrun, K. 2002, Acta Astron., 52, 129

Wray, J. J., Eyer, L., & Paczyński, B. 2004, MNRAS, 349, 1059

Yan, R., Yang, J., & Hauptmann, A. 2003, in Ninth IEEE International Conference on Computer Vision (Press), 516–523

Zadrozny, B. 2004, in Proceedings of the twenty-first international conference on Machine learning, ACM, 114

Table 1. Error rates, in %, over all testing data, and for those testing data within selected science classes in the OGLE & *Hipparcos* experiment. The first set of classes are those most under-represented in the training data. The second set are those most over-represented in the training data. Several methods for sample selection bias reduction are compared.

Science Class	$N_{\rm Train}$	$N_{\rm Test}$	RF[a]	IW	ST	CT	CT.p	AL1.d[b]	AL1.t[b]	AL2.d[b]	AL2.t[b]	AL.rand[b]
All	771	771	28.9	28.5	29.6	30.0	29.4	27.3	25.5	25.9	25.5	28.0
Delta Scuti	25	89	15.7	15.7	15.7	15.7	14.6	15.4	14.0	15.6	21.3	12.3
Beta Cephei	9	30	95.0	91.7	96.7	96.7	96.7	90.7	87.5	88.9	84.0	90.7
W Ursa Maj.	16	43	40.7	36.0	51.2	60.5	61.6	27.0	27.3	27.1	19.2	30.1
Mira	121	23	8.7	8.7	8.7	8.7	4.3	9.1	8.7	8.7	8.7	9.8
Semi-Reg. PV	33	9	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	33.3	35.4
Class. Cepheid	122	68	2.9	2.9	1.5	1.5	1.5	3.1	1.5	1.6	1.5	2.8

[a]Default Random Forest.

[b]Errors evaluated over all objects not in the active learning sample.

Table 2. Results, by class, of performing active learning to classify ASAS variable stars.

Science Class	$N_{\rm Train}$	$N_{\rm ALAdd}$[a]	$N_{\rm RF}$[b]	$N_{\rm AL}$[c]
a. Mira	144	20	3587	3173
b1. Semireg PV	42	59	5799	9223
b2. SARG A	0	15	0	3699
b3. SARG B	0	29	0	8823
b4. LSP	0	54	0	5889
c. RV Tauri	6	5	0	117
d. Classical Cepheid	191	16	324	494
e. Pop. II Cepheid	23	0	98	5
f. Multi. Mode Cepheid	94	4	162	263
g. RR Lyrae, FM	124	26	1714	1667
h. RR Lyrae, FO	25	14	51	317
i. RR Lyrae, DM	57	3	9109	442
j. Delta Scuti	114	19	822	1755
k. Lambda Bootis	13	0	0	0
l. Beta Cephei	39	0	0	0
m. Slowly Puls. B	29	0	0	0
n. Gamma Doradus	28	0	0	0
o. Pulsating Be	45	4	10	177
p. Per. Var. SG	55	1	1663	624
q. Chem. Peculiar	51	14	27	447
r. Wolf-Rayet	40	0	6683	1198
s. T Tauri	14	4	753	43
t. Herbig AE/BE	15	0	4	1
u. S Doradus	7	0	0	1
v. Ellipsoidal	13	0	0	0
w. Beta Persei	169	25	2110	3055
x. Beta Lyrae	145	37	11962	2603
y. W Ursae Maj.	59	66	5246	6108

[a]ASAS sources added to the training set after 8 AL iterations.

[b]Number of ASAS sources classified by the default Random Forest.

[c]Number of ASAS sources classified by the RF after 8 AL iterations.