The Astrophysical Journal, 744:192 (19pp), 2012 January 10 doi:10.1088/0004-637X/744/2/192

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]

[1] Astronomy Department, University of California, Berkeley, CA 94720-7450, USA; jwrichar@stat.berkeley.edu

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

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

Received 2011 June 14; accepted 2011 October 6; published 2011 December 22

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 (1) standard assumptions for machine-learned model selection procedures break down and (2) dense regions of testing space might be completely devoid of training data. We explore possible remedies to sample selection bias, including importance weighting, co-training, 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 Optical Gravitational Lensing Experiment, 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 AL to classify variable stars in the All Sky Automated Survey, finding dramatic improvement in our agreement with the ASAS Catalog of Variable Stars, 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.

Key words: methods: data analysis – methods: statistical – stars: variables: general – techniques: photometric

Online-only material: color figures

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 a 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.[4] 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[5] 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

[4] By feature we mean any quantity that can be computed directly as a function of the raw data, while response variable refers to the target parameter to be predicted for each new source. For example, in photometric redshift estimation the features are photometric colors while the response is redshift.

[5] Training data are the subset of data with known response variable that is employed to fit a supervised learning model.

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 ASAS Catalog of Variable Stars (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 variable sources. Upcoming surveys, whose automated prediction algorithms will be trained on data from older surveys and/or idealized models, will suffer from these same maladies if sample selection bias is not treated properly.

To overcome sample selection bias, we examine three methods: importance weighting (IW), co-training (CT), and active learning (AL). On both the ASAS variable star classification problem and a simulated variable star data set, we find that 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 AL 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 AL approach for candidate object selection, especially when data sizes become prohibitively large for people to manually analyze each source.

The structure of the paper is as follows. In Section 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 Section 3, we introduce a few methods that can be used to mitigate the effects of sample selection bias. We describe AL in detail, focusing on its implementation with Random Forest (RF) classification. Next, we test those methods in Section 4, showing that AL attains the best results in a simulated variable star classification experiment. In Section 5 we describe our online AL 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 AL to classify ASAS variable stars in Section 6, showing drastic improvement over the off-the-shelf classifier. Finally, we end with some concluding remarks in Section 7.

2. SAMPLE SELECTION BIAS IN ASTRONOMICAL SURVEYS

A fundamental assumption for supervised machine-learning methods is that the training and testing sets[6] 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 and 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 *COnvection ROtation and planetary Transits* (*CoRoT*; 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 objects that are undersampled 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, data from sources observed by different surveys usually follow distinct distributions with significant offsets. See, for example, Figure 1, where there is an obvious absence of the combined *Hipparcos* and OGLE training data (black diamonds) in the high-frequency, high-amplitude regime where the density of the testing set of ASAS variables (red squares) 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 systematic differences in the distribution of source types.

In other areas of astrophysics and cosmology it is a 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 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). The systematic difference between training and testing sets can cause catastrophic prediction errors when the trained model is applied to new data. This problem arises for two reasons. First, under sample selection bias, standard generalization

[6] Throughout the paper, we call training data those objects with known response variables 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.

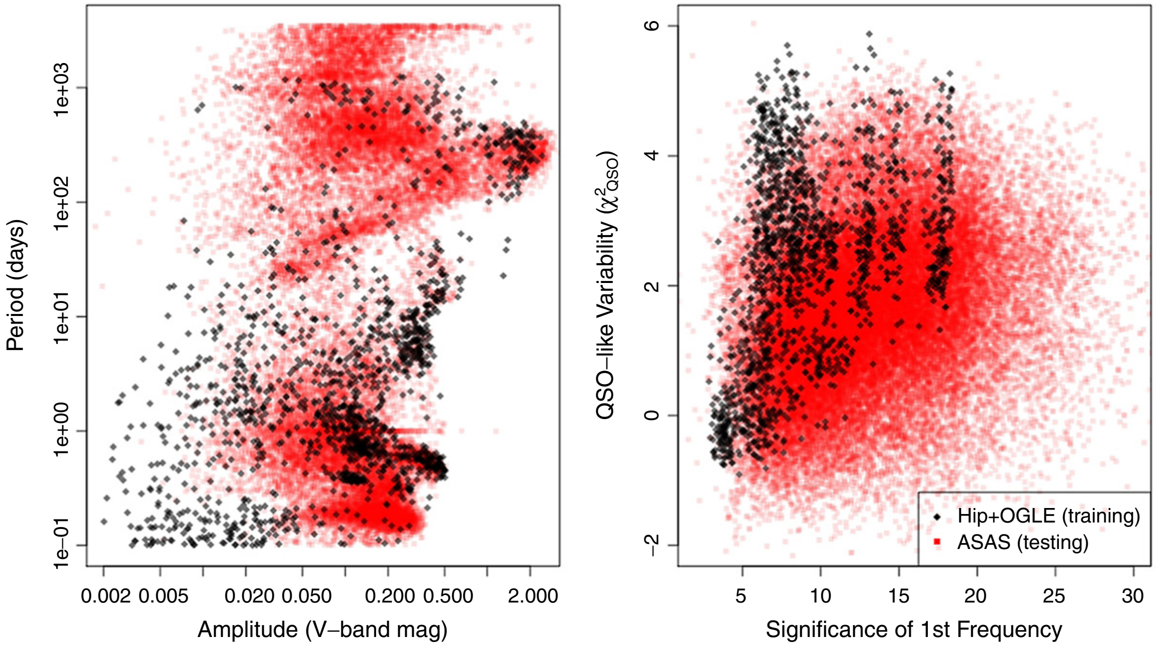


Figure 1. Presence of sample selection bias for ASAS variable star (red □) classification using a training set of well-understood data from *Hipparcos* and OGLE (black ◇) is exhibited by the large discrepancy between the feature distributions of the two data sets. Left: large distributional mismatch exists in the period–amplitude plane. ASAS testing data have high density in short-period, high-amplitude and long-period, moderate-amplitude regions, where there are little training data. Only those ASAS data whose statistical significance of the fist frequency are larger than the median are plotted. 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).

(A color version of this figure is available in the online journal.)

error estimation procedures, such as cross-validation,[7] 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, the chosen model is likely to be suboptimal for prediction on the testing data. In Section 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 Figure 1—causing catastrophically bad extrapolation of the model into those regions. In this case, any classifier trained only on the training data will produce poor class predictions in 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: *CT* (Section 3.2 and *self-training*), where testing instances with most certain class prediction are iteratively added to the training set, and *AL* (Section 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 ASAS. Particularly, we use an automated machine-learning algorithm to classify sources in the ACVS (Pojmanski 2002). ACVS verson 1.1[8] 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 the presence of sample selection bias is obvious: the distributions of the two data sets are clearly different, and several regions of feature space are densely populated with ASAS data while being devoid of training data.

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, 1999c), 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 four harmonics), $J - H$ and $H - K$ colors and IR fluxes to predict the classes of the strictly periodic sources. Many ACVS objects either have multiple labels or are annotated as having low-confidence classifications. For less regular periodic sources, location in the $J - H$ versus $H - K$ plane is tested; if the object falls within an area of late-type irregular or semi-regular stars, it is assigned the label miscellaneous (MISC), else it is inspected by eye. We find that 38,117 ACVS stars, representing 76% of

[7] Cross-validation is a standard technique for model selection in which a subset of the data are used to train the model while the left-out data are used to evaluate performance.

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

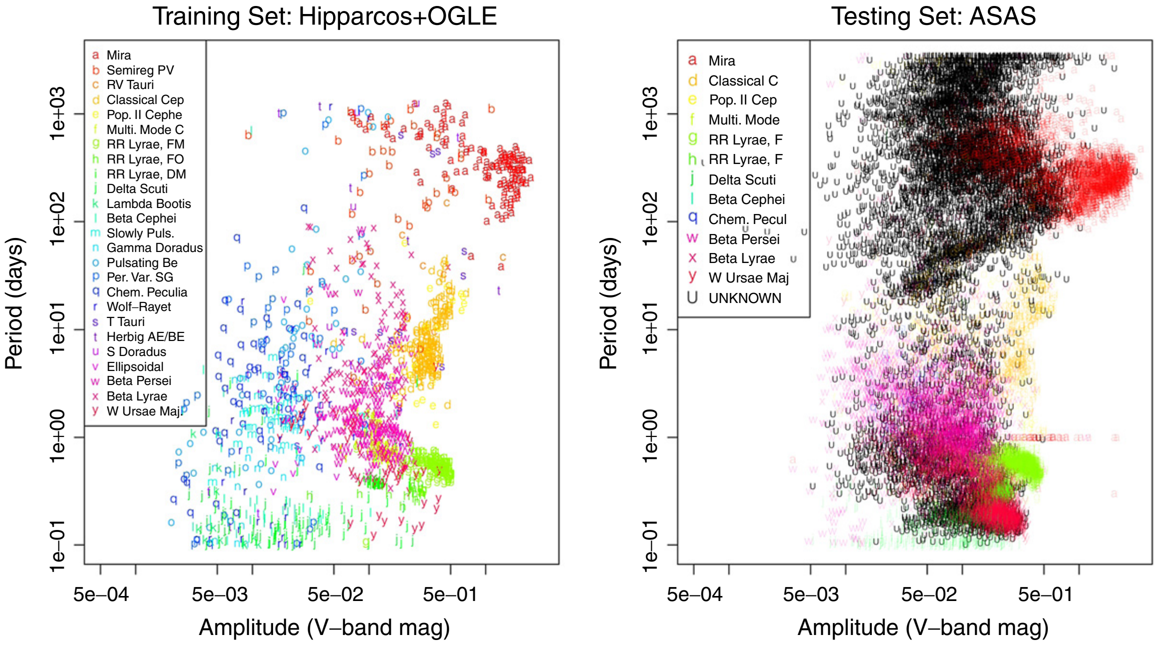


Figure 2. Left: period–amplitude relationship for the 1542 training set sources from the *Hipparcos* and OGLE surveys. Symbols and colors denote the true science class of each object. Right: same for a random sample of size 10,000 from the 50,124 ASAS testing objects, where symbols and colors denote 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 significant distributional difference between the training and testing sets, which can cause poor performance of a machine-learned classifier.

(A color version of this figure is available in the online journal.)

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.[9]

As our base model, we use an RF classifier (Breiman 2001). RF 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. (2011b) as well as five additional light curve features from Dubath et al. (2011). The RF 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 RF 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.[10]

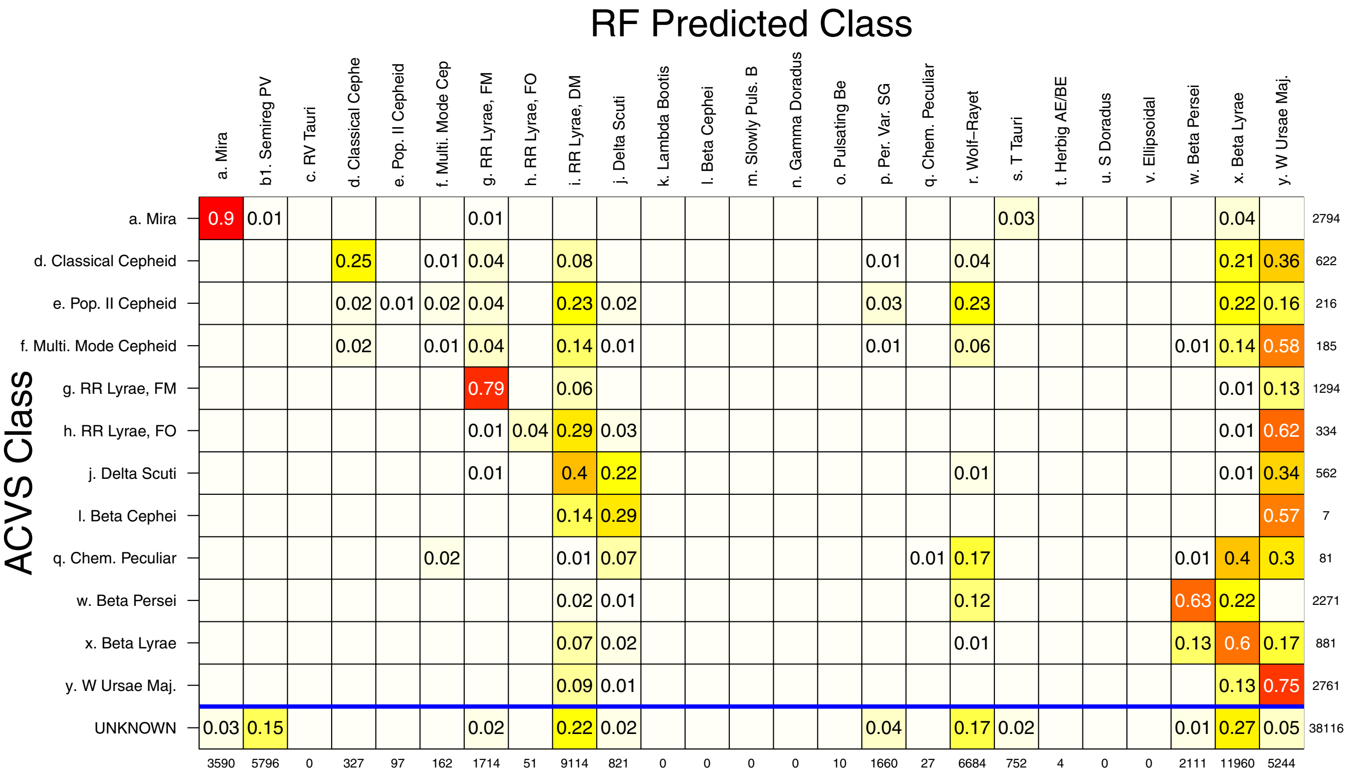
Training an RF classifier on the *Hipparcos* and OGLE data as in Richards et al. (2011b) and applying that classification model to predict the class label of each object in ACVS, we obtain a 65.5% correspondence with the ACVS labels for the 24% of objects that have a confident ACVS label. A table showing the correspondence of our predicted RF classification labels with those of ACVS is plotted in Figure 3. The RF algorithm successfully finds 90% of the Mira and 79% of the 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 RF class taxonomy is finer than that used by ACVS, including twice as many classes; as such, the RF has the ability to identify objects of rarer classes, such as T Tauri and Gamma Doradus stars.

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 appreciably 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 RF 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 RF produce a posterior class probability of $\geqslant 0.5$, meaning that the classifier is confident on the class predictions for less than 1/8th of the entire ASAS ACVS catalog.

Furthermore, in Figure 3 we find that many ASAS sources (9114 of 50,124, or 18.2%) are identified by the RF 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, through visual inspection, only 91 RR Lyrae, DM candidates in the entire OGLE-III catalog, out of 16,836 total RR Lyrae candidates

[9] 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.

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



RF Predicted Class

ACVS Class

	a. Mira	b1. Semireg PV	c. RV Tauri	d. Classical Cephe	e. Pop. II Cepheid	f. Multi. Mode Cep	g. RR Lyrae, FM	h. RR Lyrae, FO	i. RR Lyrae, DM	j. Delta Scuti	k. Lambda Bootis	l. Beta Cephei	m. Slowly Puls. B	n. Gamma Doradus	o. Pulsating Be	p. Per. Var. SG	q. Chem. Peculiar	r. Wolf–Rayet	s. T Tauri	t. Herbig AE/BE	u. S Doradus	v. Ellipsoidal	w. Beta Persei	x. Beta Lyrae	y. W Ursae Maj.	
a. Mira	0.9	0.01					0.01												0.03					0.04		2794
d. Classical Cepheid				0.25		0.01	0.04		0.08							0.01		0.04						0.21	0.36	622
e. Pop. II Cepheid				0.02	0.01	0.02	0.04		0.23	0.02						0.03		0.23						0.22	0.16	216
f. Multi. Mode Cepheid				0.02		0.01	0.04		0.14	0.01						0.01		0.06					0.01	0.14	0.58	185
g. RR Lyrae, FM							0.79		0.06															0.01	0.13	1294
h. RR Lyrae, FO							0.01	0.04	0.29	0.03														0.01	0.62	334
j. Delta Scuti							0.01		0.4	0.22								0.01						0.01	0.34	562
l. Beta Cephei									0.14	0.29															0.57	7
q. Chem. Peculiar						0.02			0.01	0.07							0.01	0.17					0.01	0.4	0.3	81
w. Beta Persei									0.02	0.01								0.12					0.63	0.22		2271
x. Beta Lyrae									0.07	0.02								0.01					0.13	0.6	0.17	881
y. W Ursae Maj.									0.09	0.01														0.13	0.75	2761
UNKNOWN	0.03	0.15					0.02		0.22	0.02						0.04		0.17	0.02				0.01	0.27	0.05	38116
	3590	5796	0	327	97	162	1714	51	9114	821	0	0	0	0	10	1660	27	6684	752	4	0	0	2111	11960	5244	

Figure 3. Off-the-shelf Random Forest classifications of the ASAS data set, using a training set of the 1542 *Hipparcos* and 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.

(A color version of this figure is available in the online journal.)

(0.5%). This classification artifact occurs because RR Lyrae, DM stars 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. 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 in variable star classification. In this section, we will introduce three different principled approaches of treating sample selection bias and argue that AL is the most appropriate of these methods for dealing with biases in astronomical data set. In Section 4, these methods will be compared using variable star data from the OGLE and *Hipparcos* missions.

3.1. Importance Weighting

Under sample selection bias, standard error estimation procedures (such as cross-validation) are biased, resulting in poor model selection. The basic idea is that one wishes to choose the statistical model (e.g., classifier) that minimizes the expected prediction error on the testing data. When the distribution of training and testing data is different, one needs to explicitly account for this difference, else the testing error estimate will likely be wrong. Minimizing an incorrect testing error estimate will cause us to choose a suboptimal model. One can achieve an unbiased estimate of the testing error via IW (see Sugiyama & Müller 2005; Huang et al. 2007, and Sugiyama et al. 2007), whereby training examples are weighted by an empirical estimate of the ratio of testing-to-training set feature densities when computing the statistical risk of a model.[11] Specifically, when training a statistical model, weighting the training data by

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

modifies standard risk estimation procedures to compute the statistical risk over the testing set (see Huang et al. 2007 for further mathematical details). Here, $\mathbf{x}_i$ is the feature vector and y_i is the response for training object i.

To achieve the last equality 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 the training and testing sets. In practice, this last equality will probably not hold for the types of astrophysical data sets that we are interested in: though the mapping from features to response values may be the same for data from different surveys, 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

[11] Statistical risk is the expected error, measured by a loss function, over a particular probability distribution. By default, risk is computed over $\mathbf{P}_{\text{Train}}$; IW modifies this to compute the risk with respect to $\mathbf{P}_{\text{Test}}$.

feature-response densities, $\mathbf{P}(\mathbf{x}, y)$.[12] 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, Equation (1) can be estimated via distribution matching (Huang et al. 2007) or by fitting a probabilistic classifier to the classification problem of training versus testing set and employing the output probability estimates (Zadrozny 2004).

Using the weights defined in Equation (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 undersampled by the training data, with respect to the testing set density. This enforces a higher penalty for making errors in regions of feature space that are underrepresented 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 that are densely populated by testing data (and conversely ignore modeling those regions devoid of testing data). For the ASAS example, IW 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 IW is useful in some problems, it has been shown to be asymptotically sub-optimal when the statistical model is correctly specified[13] (Shimodaira 2000) and with flexible non-parametric models such as RF we observe very little change in performance using IW (see Section 4). An additional, more debilitating drawback is that IW requires the support of the testing distribution[14] be a *subset* of the support of the training distribution,[15] which, in the types of supervised learning problems common in astrophysics, is rarely the case.

3.2. Co-training

In astronomical problems, we typically have much more unlabeled than labeled data. This is due to both the painstaking 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 employ 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 *CT* and *self-training* slowly augment the training set by iteratively adding the most confidently classified testing set cases from the previous iteration.

CT 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 testing instances to the training set of the *other* classifier. This process continues either for N iterations or until all testing 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., $p(y|\mathbf{x}) = h_1(y|\mathbf{x}_1)h_2(y|\mathbf{x}_2)$. CT has shown impressive performance in situations where very few training examples are used to classify many testing cases. Blum & Mitchell (1998) use CT 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 CT 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 Web 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. Here, candidate lists obtained with the first version of the classifier were used to select very probable class members among the *CoRoT* data for subsequent inclusion in the training set. This augmentation procedure led to inclusion of an extra 114 sources into the training set.

Both CT 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, CT 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 Section 4 we apply both self-training and CT to variable star classification, finding that these methods perform poorly in terms of overall error rate, especially for classes that are undersampled by the training data.

3.3. Active Learning

A special 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, this can be achieved via targeted spectroscopic study, visualization of (folded) light curves, or querying of other databases and catalogs. Consider astronomical source classification for future missions: while it is impractical to manually follow-up 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 *AL* (and in particular, pool-based AL, Lewis & Gale 1994). Under pool-based AL for classification, an algorithm iteratively selects, out of the entire set of unlabeled

[12] 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 $\mathbf{P}_{\mathrm{Test}}(\mathbf{x}_i|y_i) = \mathbf{P}_{\mathrm{Train}}(\mathbf{x}_i|y_i)$ in most practical situations; however, if this were to hold then the importance weights would simply reduce to the ratio of response priors.

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

[14] Defined as the subset of feature space with non-zero density: $\mathcal{S} = \{\mathbf{x} : \mathbf{P}(\mathbf{x}) > 0\}$.

[15] 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.

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. AL 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). For a thorough review, see Settles (2010).

The basic procedure for pool-based AL is the following: begin with a training set $\mathcal{L}$ and testing set $\mathcal{U}$. On each AL iteration, 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 query function, see Section 3.3.1. These queried AL 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. This procedure 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. However, the drawback to that approach is that it needs a relatively large initial training sample and requires that for all non-void regions of $\mathbf{P}_{\text{Test}}$, $\mathbf{P}_{\text{Train}}$ also be non-zero. On the other hand, the AL approach is to *expand* the training set in a way that makes it most closely resemble the testing set, thereby curtailing sample selection bias.

3.3.1. Active Learning Query Function

Several strategies have been proposed to determine which testing data about which AL 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.

We have an explicit goal of minimizing the classification error rate over the entire set of testing data, so 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.[16] However, because the true labels of the training data are not known a priori, one must also iterate over the possible labels of the training data, computing a rough estimate of the expected decrease in testing error rate by approximation of 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,[17] 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 IW and the second is motivated by selecting the sources whose inclusion in the training set would produce the largest total change in the predicted class probabilities for the testing sources. To explain how we implement these heuristics, we first revisit the RF 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 RF estimate of probability of class y is simply the empirical proportion,

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

of the B trees that predict class y. Additionally, the RF 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, which is 1 when the statement is true and 0 if it is false. 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 undersampled by the training data, as measured by

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

which is the ratio of the average proximity measure for $\mathbf{x}'$ in the testing ($\mathcal{U}$) to the training ($\mathcal{L}$) set. 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. Thus, the ratio in Equation (4) is large only for testing data that reside in regions of feature space with relatively high testing set density compared to training set density.

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

[16] For many machine-learning algorithms, fast incremental updating algorithms exist, making this approach feasible.

[17] 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.

metric that attempts to choose the $\mathbf{x}'$ that maximizes the total change, over the testing set, of the RF probability vectors, as measured using the ℓ_1 norm.[18] An approximate solution to this problem is to choose the testing data points that maximize

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

where the RF probability, $\widehat{P}_{\mathrm{RF}}(y|\mathbf{x})$, is defined in Equation (2), and $\max_y \widehat{P}_{\mathrm{RF}}(y|\mathbf{x})$ refers to the maximum of the class probabilities for feature vector $\mathbf{x}$. In the Appendix, we work out the details of deriving Equation (5) from the heuristic of selecting testing points whose inclusion in the training set maximally affects the total change of the RF predicted probabilities over $\mathcal{U}$.

The key elements to Equations (4) and (5) are (1) the testing set average proximity, represented by $\sum_{\mathbf{x}\in\mathcal{U}} \rho(\mathbf{x}', \mathbf{x})$, is in the numerator, and (2) the training set average proximity, represented by $\sum_{\mathbf{z}\in\mathcal{L}} \rho(\mathbf{x}', \mathbf{z})$, is in the denominator. This means that we 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 RF prediction uncertainty, represented by $1 - \max_y \widehat{P}_{\mathrm{RF}}(y|\mathbf{x})$, in the numerator. This means that S_2 gives higher weight to those testing points that are difficult to classify, causing the algorithm to focus more attention along class boundaries or other poorly identified regions of feature space, which should lead to better performance.

3.3.2. Batch-mode Active Learning

In typical AL applications, queries are chosen sequentially. 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 astronomical observing run, multiple objects are queued for follow-up observation. In classifying variable stars from the ASAS survey (Section 6), 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 Equations (4) and (5)). Heuristic methods have been devised that create diversity within batches of AL samples (e.g., Brinker 2003 for support vector machines (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 Section 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 AL 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 light curves. 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 Section 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 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 be difficult. 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 AL 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 initialize 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 Section 6, we use the `IEThresh` algorithm with $\epsilon = 0.85$ to crowdsource labels for the ASAS data set. 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 AL 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 the 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 Sloan Digital Sky Survey, 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 spectroscopic 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 criterion. That is, the AL function is $S(\mathbf{x}') = S_1(\mathbf{x}')(1 - C(\mathbf{x}'))$, where the cost

[18] The ℓ_1 norm is defined as $||\mathbf{x}||_1 = \sum_j |x_j|$.

function, $C(\mathbf{x}')$, is

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

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.[19] High cost on a source means that we 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 Section 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 AL 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 AL 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 ground truths do not exist 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 Section 6).

4. EXPERIMENT: OGLE AND *HIPPARCOS* VARIABLE STARS

In this section, we test the effectiveness of the various methods proposed in Section 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

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

This selection function is devised so that the training set undersamples 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.[20]

Distributional mismatch between the training and testing sets causes an off-the-shelf RF classifier to perform poorly for short-period small-amplitude sources. The median overall error rate for an RF 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.

1. *Importance weighting*. A single RF is built on the training set, with class-wise[21] importance sampling weights defined as the ratio of the testing set to training set class proportions.[22]
2. *Self-training and CT*. Each algorithm is repeated for 100 iterations, where on each iteration the most confident three testing set objects are added to the training set. For CT, we use both random feature splits (CT) and a split between periodic and non-periodic features (CT.p).
3. *Active learning*. Using the metrics in Equations (4) (AL1) and (5) (AL2), we perform 10 rounds of AL, 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 AL 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.

Distributions of the classification error rates for each method, obtained over 20 random draws of training data from Γ, 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 AL 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. For comparison, the AL.rand approach of randomly querying observations for manual labeling does not perform well compared to any of the principled AL approaches. None of the other methods produces a statistically 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 three correct classifications.

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

[19] 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 lack of capability of the user in manually labeling each particular source.

[20] All of the code and data used to generate the results and figures in Section 4 are available for download at http://lyra.berkeley.edu/~jwrichar/arXiv1106.2832_sec4.tar.gz.

[21] In importance weighting, ratios of feature densities are typically used as the weights. However, in the R randomForest code that we use, weights may only be defined by class.

[22] 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.

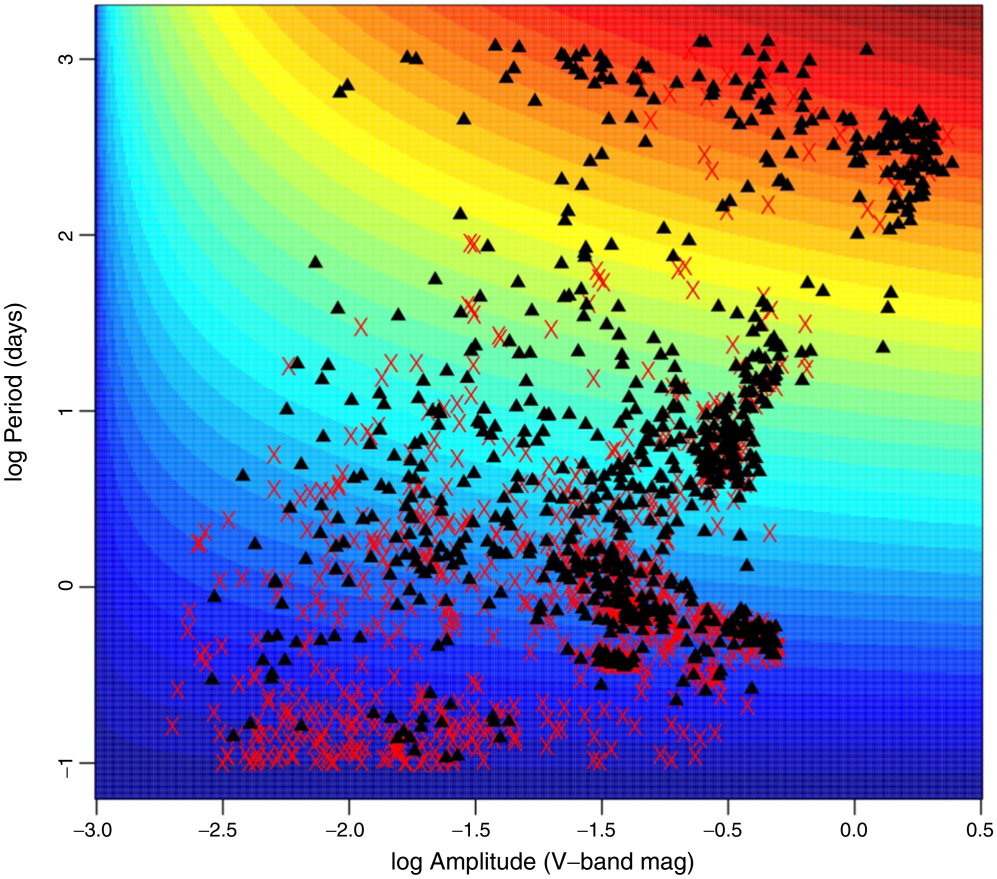


Figure 4. Training (black ▲) and testing (red X) data for the simulated example using OGLE and *Hipparcos* data. The 771 training data were randomly sampled from the original 1542 sources according to the sampling distribution, Γ, plotted in color. Using this sampling scheme, we create sample selection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

(A color version of this figure is available in the online journal.)

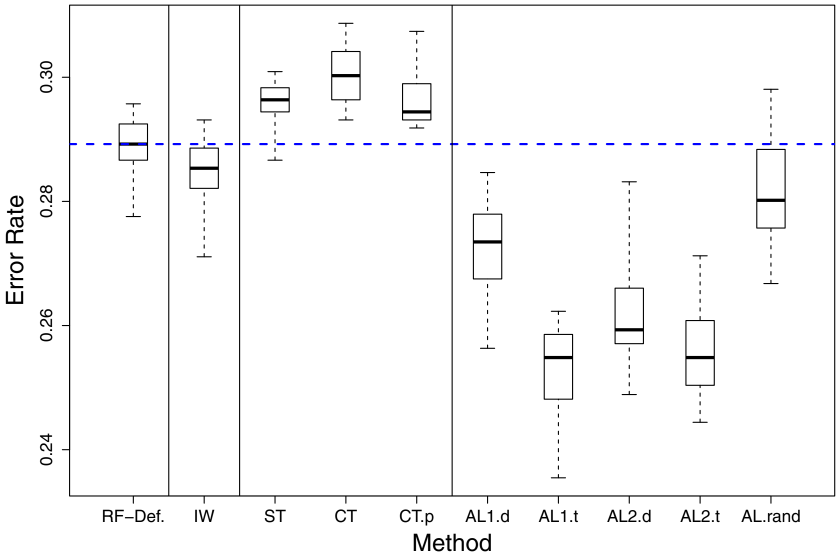


Figure 5. Error rates, evaluated over the testing set, of 10 different methods applied to the OGLE and *Hipparcos* simulated data set of 771 training and 771 testing samples. Due to sample selection bias, the default Random Forest (RF-Def.) is ineffective. Importance weighting (IW) improves upon the 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 error distributions are somewhat wider than for other methods because the AL methods are each evaluated over only the 671 testing data points that were 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. 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.

(A color version of this figure is available in the online journal.)

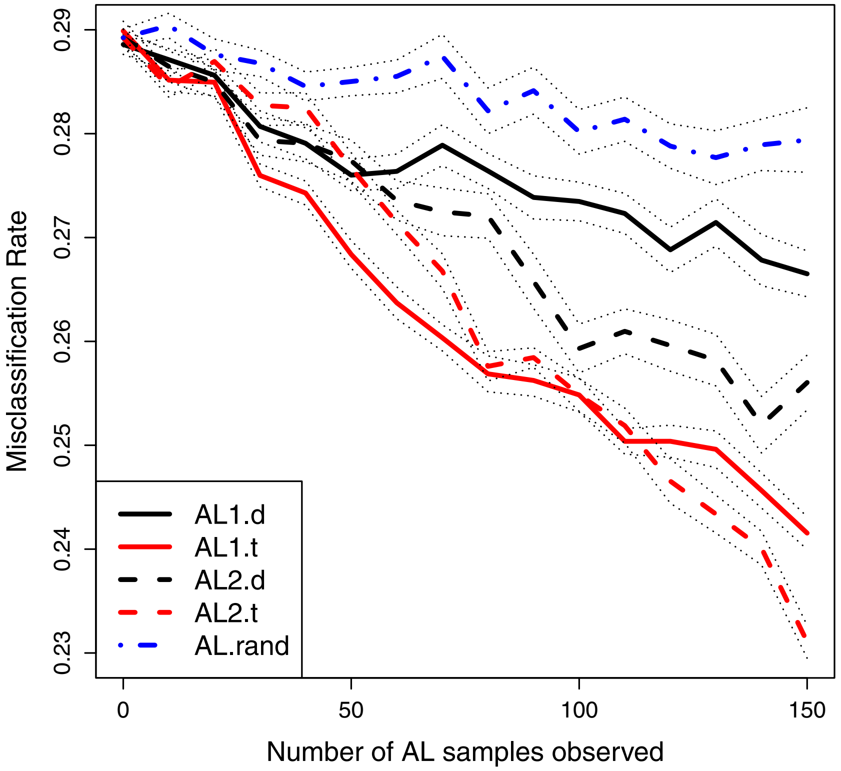


Figure 6. Performance of the active learning approaches for the OGLE and *Hipparcos* classification experiment. Both AL1 and AL2 dominate the performance of AL.rand, but there is no clear winner between these two approaches. AL1.t performs best for the first few iterations, but is overtaken by AL2.t 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.

(A color version of this figure is available in the online journal.)

Table 1

Error Rates, in %, Over All Testing Data, and for those Testing Data within Selected Science Classes in the OGLE and *Hipparcos* Experiment

Science Class	N_{Train}	N_{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

Notes. The first set of classes are those most underrepresented in the training data. The second set are those most overrepresented in the training data. Several methods for sample selection bias reduction are compared.

[a] Default Random Forest.

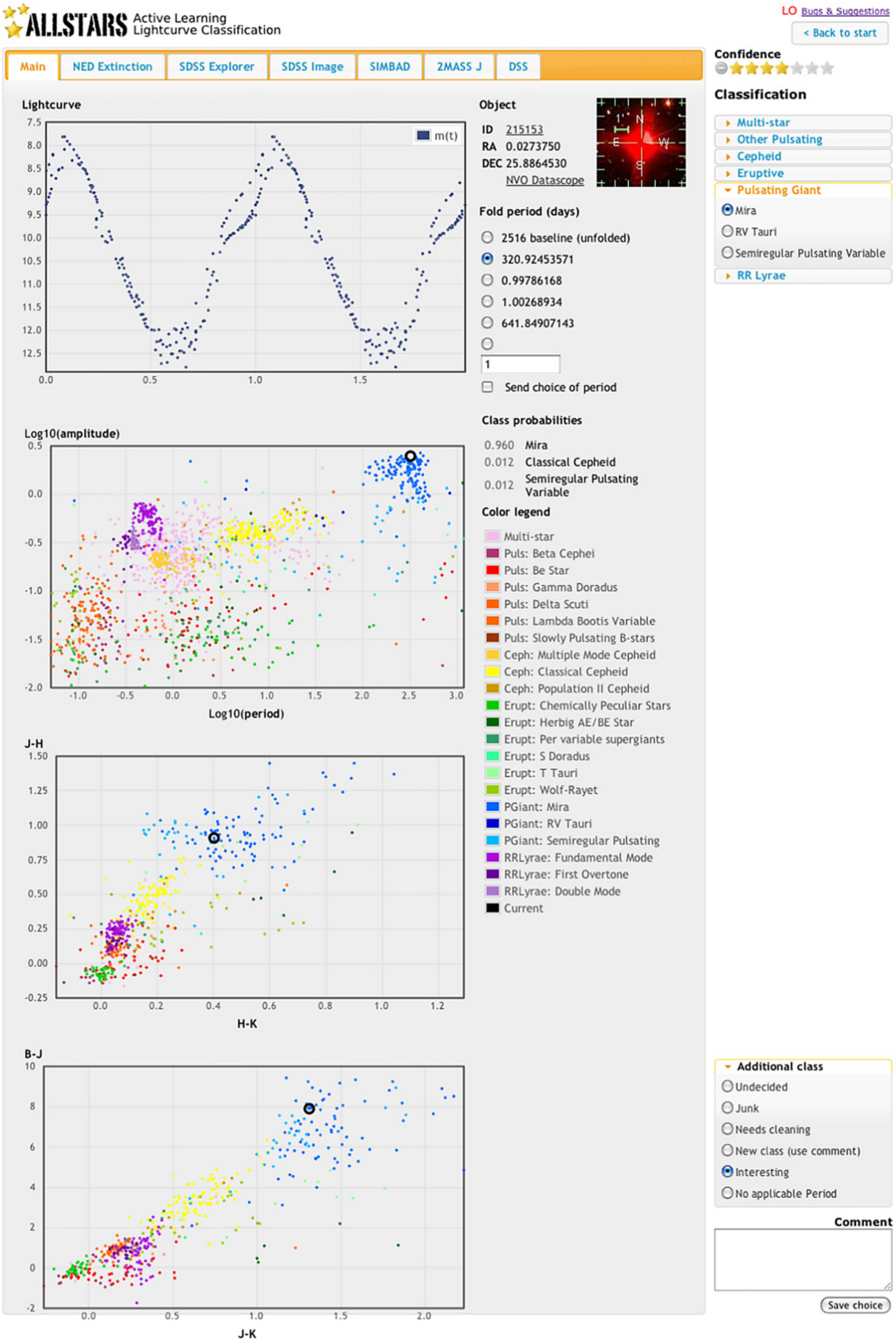
[b] Errors evaluated over all objects not in the active learning sample.

AL.t strategies may be an artifact of using small batch sizes (10 objects); in the application of AL to ASAS, we typically use batch sizes >50.

AL is able to significantly improve the classification error rate on the set of OGLE and *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 AL classifiers show substantial improvement, on average, over the default RF for the classes which are most undersampled by the training data with no increase in the error rates within the classes that are most overrepresented in the training set.

5. ALLSTARS: ACTIVE LEARNING LIGHT CURVE WEB INTERFACE

We developed the ALLSTARS (Active Learning Light curve classification Service) Web-based tool as the crowdsourcing user interface to our active learning software. For each AL iteration, this Web site 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,



Object

ID 215153
RA 0.0273750
DEC 25.8864530
NVO Datascope

Fold period (days)

○ 2516 baseline (unfolded)
● 320.92453571
○ 0.99786168
○ 1.00268934
○ 641.84907143
○ 1

□ Send choice of period

Class probabilities

0.960	Mira
0.012	Classical Cepheid
0.012	Semiregular Pulsating Variable

Color legend

- 🟪 Multi-star
- 🟥 Puls: Beta Cephei
- 🟥 Puls: Be Star
- 🟧 Puls: Gamma Doradus
- 🟧 Puls: Delta Scuti
- 🟧 Puls: Lambda Bootis Variable
- 🟫 Puls: Slowly Pulsating B-stars
- 🟨 Ceph: Multiple Mode Cepheid
- 🟨 Ceph: Classical Cepheid
- 🟨 Ceph: Population II Cepheid
- 🟩 Erupt: Chemically Peculiar Stars
- 🟩 Erupt: Herbig AE/BE Star
- 🟩 Erupt: Per variable supergiants
- 🟩 Erupt: S Doradus
- 🟩 Erupt: T Tauri
- 🟩 Erupt: Wolf-Rayet
- 🟦 PGiant: Mira
- 🟦 PGiant: RV Tauri
- 🟦 PGiant: Semiregular Pulsating
- 🟪 RRLyrae: Fundamental Mode
- 🟪 RRLyrae: First Overtone
- 🟪 RRLyrae: Double Mode
- ⬛ Current

Confidence

★★★★☆☆☆

Classification

- ▸ Multi-star
- ▸ Other Pulsating
- ▸ Cepheid
- ▸ Eruptive
- ▾ Pulsating Giant
  - ● Mira
  - ○ RV Tauri
  - ○ Semiregular Pulsating Variable
- ▸ RR Lyrae

▾ Additional class

- ○ Undecided
- ○ Junk
- ○ Needs cleaning
- ○ New class (use comment)
- ● Interesting
- ○ No applicable Period

Comment

Figure 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.

(A color version of this figure is available in the online journal.)

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 AL 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. Back-end feature generators, AL, and classification algorithms were implemented using a combination of Python, C, and R. The interactive plots are generated using the Flot jQuery[23] package.

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

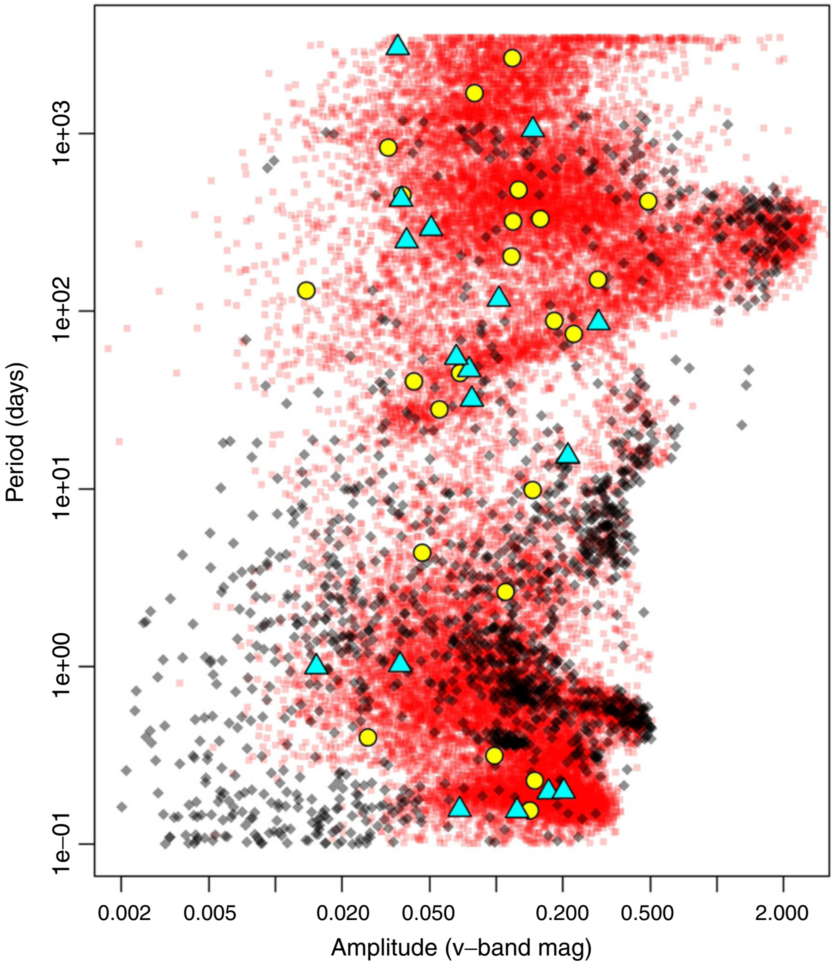


Figure 8. Active learning samples on a single iteration of the algorithm. Yellow circles signify points that at least 65% of users were able to classify. These points are included on subsequent iterations of the algorithm. Cyan triangles signify variable stars that were queried, but for which fewer than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

(A color version of this figure is available in the online journal.)

External resources made available for classifying each source are as follows.

1. NED Extinction Calculator: http://ned.ipac.caltech.edu/forms/calculator.html.
2. SDSS DR7 Explorer: http://cas.sdss.org/dr7/en/tools/explore/obj.asp.
3. SDSS DR7 Navigate Tool: http://cas.sdss.org/dr7/en/tools/chart/navi.asp.
4. SIMBAD Query by coordinates: http://simbad.u-strasbg.fr/simbad/sim-fcoo.
5. 2MASS Interactive Image (*J* band): http://irsa.ipac.caltech.edu/applications/2MASS/IM/interactive.html.
6. SkyView Original DSS image: http://skyview.gsfc.nasa.gov/cgi-bin/query.pl.
7. NVO DataScope: http://heasarc.gsfc.nasa.gov/cgi-bin/vo/datascope/init.pl.
8. DotAstro LightCurve Warehouse: http://dotastro.org/.

The initial page for a source includes two color–color plots: $B-J$ versus $J-K$ and $J-H$ versus $H-K$, using colors from the SIMBAD source which best matches the location of the given source. The source is also shown on a log-amplitude versus 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 versus time light curve plot is also shown, with 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 AL 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 AL methodology presented in Section 3.3 to classify all of ACVS (see Section 2.1) starting with the combined *Hipparcos* and OGLE training set. We employ the S_2 AL query

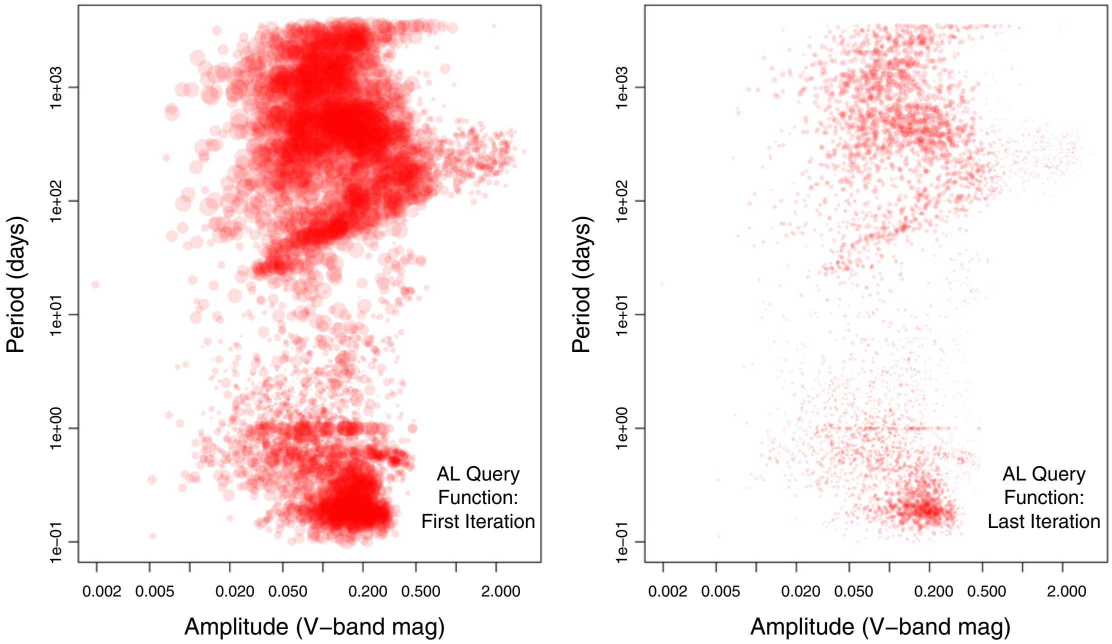


Figure 9. Value of the active learning query function, S_2, before the first iteration (left) and the ninth iteration (right) of active learning. For each variable star, the radius of the data point is proportional to S_2. After including actively learned training samples from eight AL iterations, the average S_2 value is markedly decreased. This occurs because regions of feature space that were originally undersampled by the training set are filled in by AL data, making the training and testing distributions less discrepant.

(A color version of this figure is available in the online journal.)

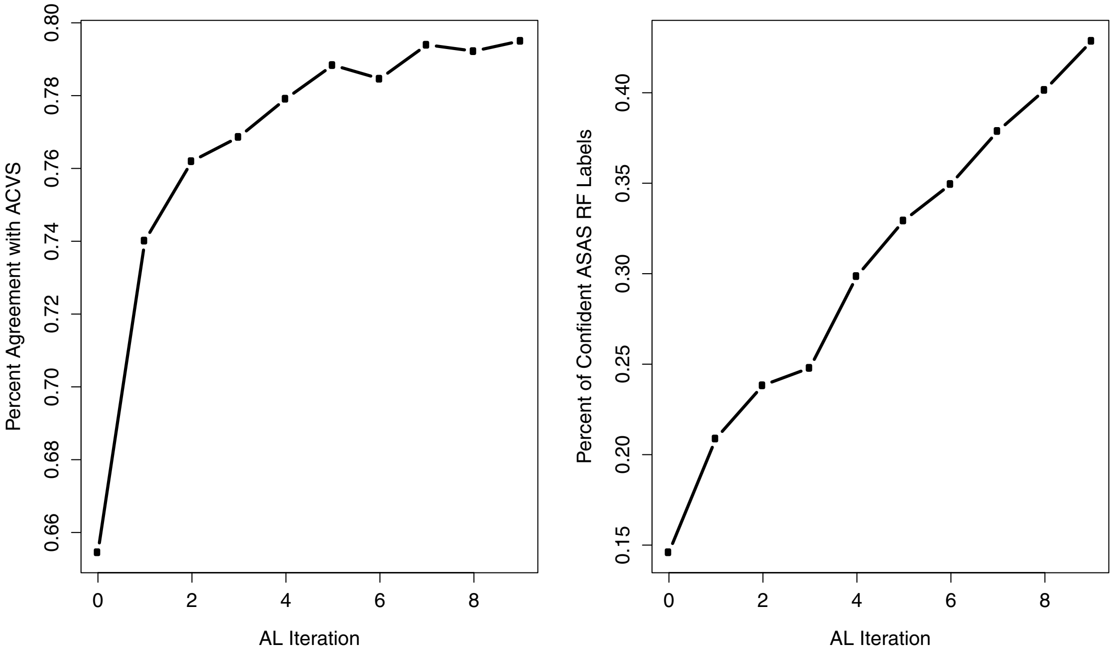


Figure 10. Left: percent agreement of the Random Forest classifier with the ACVS labels, as a function of AL iteration. Right: percent of ASAS data with confident RF classification (posterior probability >0.5), as a function of AL iteration. In the percent agreement with ACVS metric, performance increases dramatically in the first couple of iterations and then slowly levels off. In the percent of confident RF labels, the performance increases steadily.

function (Equation (5)), treating it as a probability distribution (AL2.d in Section 4), and selecting 50 AL candidates on each of nine 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.[24]

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.[25] A total of 615 sources were observed by users (this represents 1.2% of the

[24] 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.

[25] So as to not throw away useful annotations, these classifications were used along with the AL samples.

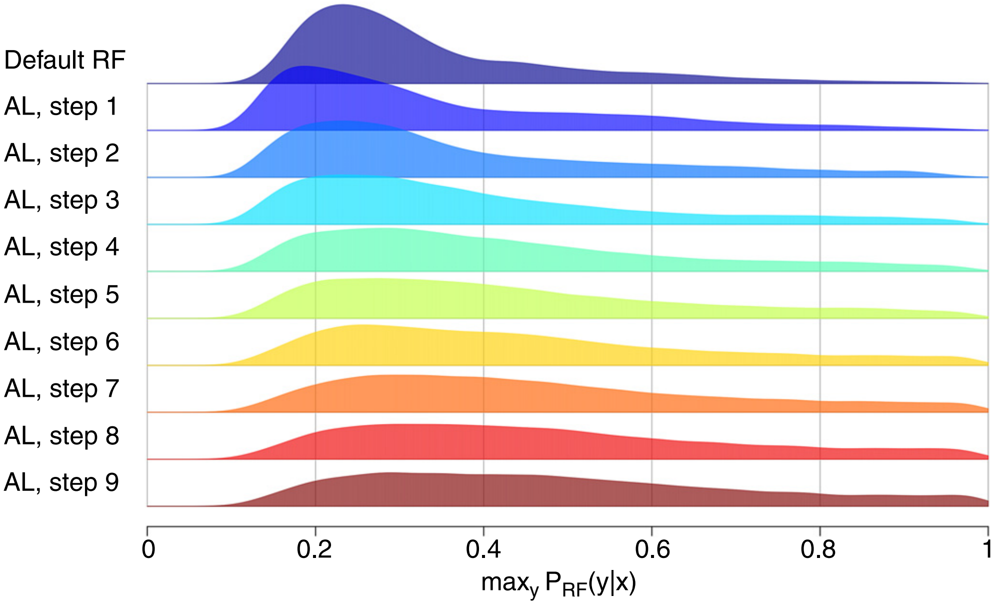


Figure 11. Distribution of the Random Forest $\max_y \widehat{P}_{\rm RF}(y|\mathbf{x})$ values for the ASAS data, as a function of AL iteration. For the default RF classifier, most values are smaller than 0.4, meaning that the classifier is confident on very few sources. As the AL iterations proceed, 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.

(A color version of this figure is available in the online journal.)

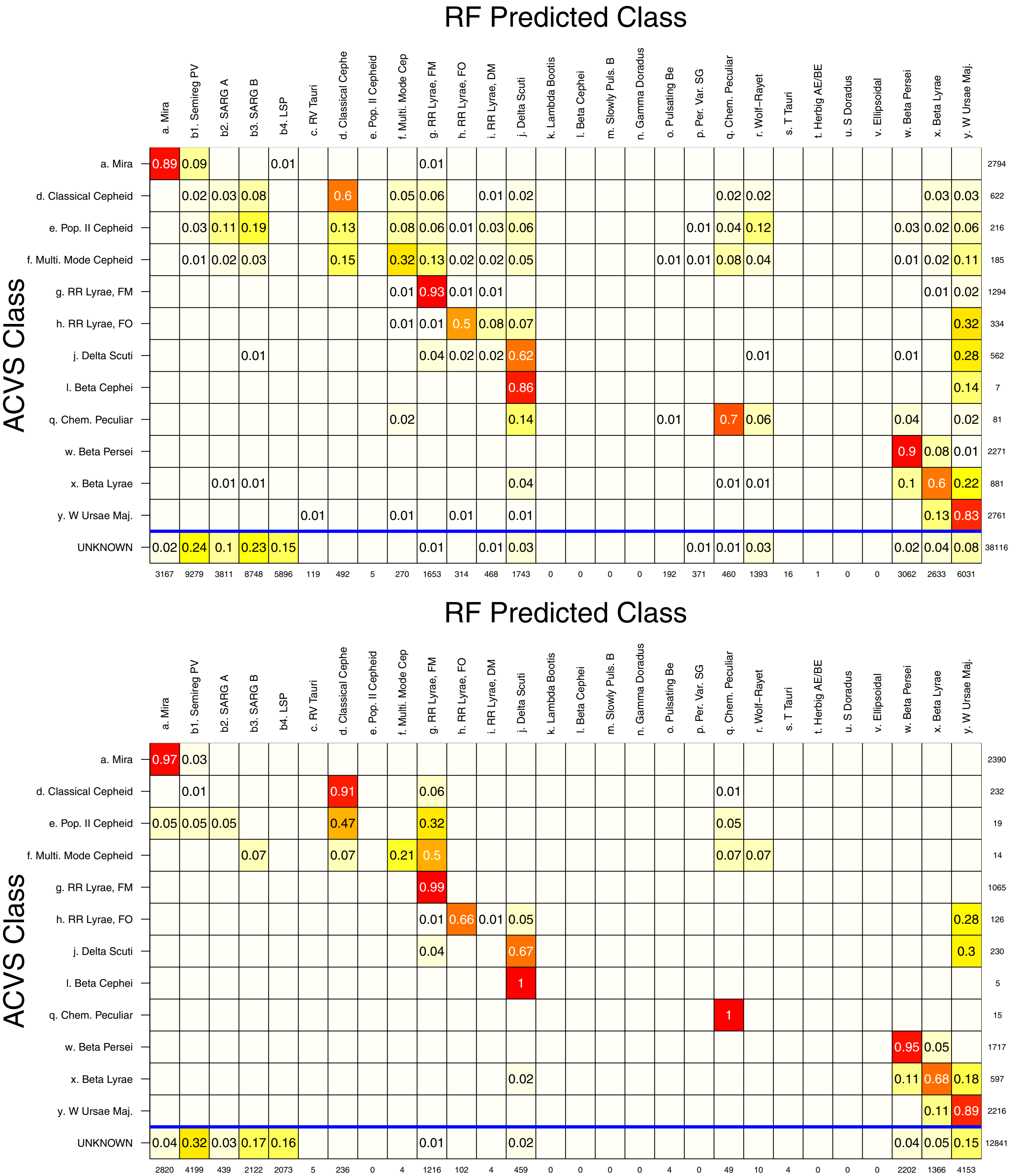
ACVS catalog). The average user classified 137 sources, with a range from 21 to 474. User responses were combined using the crowdsourcing methodology in Section 3.3.3. This led to the inclusion of 415 ASAS sources (67% of all sources that were studied manually, representing 0.8% of the ACVS catalog) into the training set. In Figure 8 we plot the AL queried data from one iteration in the amplitude–period plane, highlighting those objects which were selected for inclusion in the training set. Figure 9 depicts, in period–amplitude space, how the S_2 criterion changes from the first to the last AL iteration: inclusion of the AL samples markedly decreases the average S_2 value and almost completely diminishes the S_2 value for short-period, high-amplitude variables.

As described in Section 2.1, the default RF only attains a 65.5% agreement with the ACVS catalog. After nine AL iterations, this jumps to 79.5%, an increase of 21% in agreement rate. The proportion of ACVS sources in which we are confident (which we define as those objects having RF probability $\geqslant$0.5 for a single class) 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 confidence. As a function of the AL iteration, the ACVS agreement rate and the proportion of confident classifications achieved by our classifier are plotted in Figure 10. The full evolution of the distribution of $\max_y \widehat{P}_{\rm RF}(y|\mathbf{x})$ is plotted in Figure 11. As the iterations proceed, power is shifted from low to high probabilities.

In Figure 12, we plot a table of the correspondence between our classifications after nine AL iterations and the ACVS class. Compared to Figure 3, the AL predictions more closely match the ACVS labels across most science classes. For example, correspondence in the Classical Cepheid class rose from 24% to 60%, RR Lyrae, FM from 79% to 93%, Delta Scuti from 22% to 62%, and Chemically Peculiar from 1% to 70%. We have also identified a number of candidates for more rare classes, such as 119 RV Tauri, 192 Pulsating Be stars, and 16 T Tauri. Additionally, the number of RR Lyrae, DM candidates, which was artificially high for the original RF classifier, has diminished from 9114 to 468. A summary of our ASAS classification AL, by class, is given in Table 2.

As a consequence of performing AL on the ASAS data set, we were able to detect the presence of three 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 AL, our classifier identified 3811 SARG A, 8748 SARG B, and 5896 LSP candidates.

Our final experiment compares our classification results using AL with an RF classifier trained directly on the ACVS labels. The aim of this study is to determine whether our classifier's 20.5% disagreement with ACVS is due principally to inadequacies in our classifier or because of mistakes and inconsistencies in the ACVS labels. Using a five-fold cross-validation on the ACVS labels, an RF classifier finds a 90% agreement rate with ACVS. Hence, half of our discrepancy with ACVS can be explained by inconsistencies in the ACVS labels. Further, our use of a finer taxonomy (where, e.g., we can correctly identify some ACVS Miras as being Semi-Regular PVs) causes more discrepancies between the AL and ACVS classifiers. Additionally, 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



	a. Mira	b1. Semireg PV	b2. SARG A	b3. SARG B	b4. LSP	c. RV Tauri	d. Classical Cephe	e. Pop. II Cepheid	f. Multi. Mode Cep	g. RR Lyrae, FM	h. RR Lyrae, FO	i. RR Lyrae, DM	j. Delta Scuti	k. Lambda Bootis	l. Beta Cephei	m. Slowly Puls. B	n. Gamma Doradus	o. Pulsating Be	p. Per. Var. SG	q. Chem. Peculiar	r. Wolf–Rayet	s. T Tauri	t. Herbig AE/BE	u. S Doradus	v. Ellipsoidal	w. Beta Persei	x. Beta Lyrae	y. W Ursae Maj.	
a. Mira	0.89	0.09			0.01					0.01																			2794
d. Classical Cepheid		0.02	0.03	0.08			0.6		0.05	0.06		0.01	0.02							0.02	0.02						0.03	0.03	622
e. Pop. II Cepheid		0.03	0.11	0.19			0.13		0.08	0.06	0.01	0.03	0.06						0.01	0.04	0.12					0.03	0.02	0.06	216
f. Multi. Mode Cepheid		0.01	0.02	0.03			0.15		0.32	0.13	0.02	0.02	0.05					0.01	0.01	0.08	0.04					0.01	0.02	0.11	185
g. RR Lyrae, FM									0.01	0.93	0.01	0.01															0.01	0.02	1294
h. RR Lyrae, FO									0.01	0.01	0.5	0.08	0.07															0.32	334
j. Delta Scuti				0.01						0.04	0.02	0.02	0.62								0.01					0.01		0.28	562
l. Beta Cephei													0.86															0.14	7
q. Chem. Peculiar									0.02				0.14					0.01		0.7	0.06					0.04		0.02	81
w. Beta Persei																										0.9	0.08	0.01	2271
x. Beta Lyrae			0.01	0.01									0.04							0.01	0.01					0.1	0.6	0.22	881
y. W Ursae Maj.						0.01			0.01		0.01		0.01														0.13	0.83	2761
UNKNOWN	0.02	0.24	0.1	0.23	0.15					0.01		0.01	0.03						0.01	0.01	0.03					0.02	0.04	0.08	38116
	3167	9279	3811	8748	5896	119	492	5	270	1653	314	468	1743	0	0	0	0	192	371	460	1393	16	1	0	0	3062	2633	6031	

	a. Mira	b1. Semireg PV	b2. SARG A	b3. SARG B	b4. LSP	c. RV Tauri	d. Classical Cephe	e. Pop. II Cepheid	f. Multi. Mode Cep	g. RR Lyrae, FM	h. RR Lyrae, FO	i. RR Lyrae, DM	j. Delta Scuti	k. Lambda Bootis	l. Beta Cephei	m. Slowly Puls. B	n. Gamma Doradus	o. Pulsating Be	p. Per. Var. SG	q. Chem. Peculiar	r. Wolf–Rayet	s. T Tauri	t. Herbig AE/BE	u. S Doradus	v. Ellipsoidal	w. Beta Persei	x. Beta Lyrae	y. W Ursae Maj.	
a. Mira	0.97	0.03																											2390
d. Classical Cepheid		0.01					0.91			0.06										0.01									232
e. Pop. II Cepheid	0.05	0.05	0.05				0.47			0.32										0.05									19
f. Multi. Mode Cepheid				0.07			0.07		0.21	0.5										0.07	0.07								14
g. RR Lyrae, FM										0.99																			1065
h. RR Lyrae, FO										0.01	0.66	0.01	0.05															0.28	126
j. Delta Scuti										0.04			0.67															0.3	230
l. Beta Cephei													1																5
q. Chem. Peculiar																				1									15
w. Beta Persei																										0.95	0.05		1717
x. Beta Lyrae													0.02													0.11	0.68	0.18	597
y. W Ursae Maj.																											0.11	0.89	2216
UNKNOWN	0.04	0.32	0.03	0.17	0.16					0.01			0.02													0.04	0.05	0.15	12841
	2820	4199	439	2122	2073	5	236	0	4	1216	102	4	459	0	0	0	0	4	0	49	10	4	0	0	0	2202	1366	4153	

Figure 12. Top: classifications of the active learning RF classifier after nine 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.

(A color version of this figure is available in the online journal.)

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 is due more to lack of self-consistency in ACVS (due either to mistakes in ACVS or absence of crucial features in our classifier) than to any shortcomings in the AL methodology.

Table 2
Results, by Class, of Performing Active Learning to Classify ASAS Variable Stars

Science Class	N_{Train}	N_{ALAdd}[a]	N_{RF}[b]	N_{AL}[c]
a. Mira	144	20	3590	3167
b1. Semi-Reg. PV	42	59	5796	9279
b2. SARG A	0	15	0	3811
b3. SARG B	0	29	0	8748
b4. LSP	0	54	0	5896
c. RV Tauri	6	5	0	119
d. Classical Cepheid	191	16	327	492
e. Pop. II Cepheid	23	0	97	5
f. Multi. Mode Cepheid	94	4	162	270
g. RR Lyrae, FM	124	26	1714	1653
h. RR Lyrae, FO	25	14	51	314
i. RR Lyrae, DM	57	3	9114	468
j. Delta Scuti	114	19	821	1743
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	192
p. Per. Var. SG	55	0	1660	371
q. Chem. Peculiar	51	14	27	460
r. Wolf–Rayet	40	0	6684	1393
s. T Tauri	14	4	752	16
t. Herbig AE/BE	15	0	4	1
u. S Doradus	7	0	0	0
v. Ellipsoidal	13	0	0	0
w. Beta Persei	169	25	2111	3062
x. Beta Lyrae	145	37	11960	2633
y. W Ursae Maj.	59	66	5244	6031

Notes.
[a] ASAS sources added to the training set after eight AL iterations.
[b] Number of ASAS sources classified by the default Random Forest.
[c] Number of ASAS sources classified by the RF after eight AL iterations.

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 AL performs significantly better than other methods such as IW, CT, 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). AL 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.

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. Additionally, the differing cadences between surveys can be debilitating when attempting to utilize data from older surveys to classify within new surveys. 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 J. P. Long et al. (2011, in preparation).

Our discussion of sample selection bias has revolved around the use of non-parametric tools (and in particular RFs). 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 AL, 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.

APPENDIX

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 RF 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

$$\Delta(\mathbf{x}', \mathbf{x}) = ||\widehat{P}_{\mathrm{RF},\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})||_1 \tag{A2}$$

$$= \sum_{y=1}^{C} |\widehat{P}_{\mathrm{RF},\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})|, \tag{A3}$$

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 RF 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,[26] 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

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

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

$$\theta_{b,\mathcal{L}\cup\mathbf{x}'}(y|\mathbf{x}) - \theta_{b,\mathcal{L}}(y|\mathbf{x}) = \frac{n_b(\mathbf{x}')\theta_{b,\mathcal{L}}(y|\mathbf{x}) + I(Y(\mathbf{x}') = y)}{n_b(\mathbf{x}') + 1} - \theta_{b,\mathcal{L}}(y|\mathbf{x}) \tag{A5}$$

$$= \frac{I(Y(\mathbf{x}') = y) - \theta_{b,\mathcal{L}}(y|\mathbf{x})}{n_b(\mathbf{x}') + 1}, \tag{A6}$$

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

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) $\theta_{b,\mathcal{L}}(y|\mathbf{x}) = \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x})$, i.e., approximate the probability vector of each tree by the RF probability. Using these approximations we have that

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

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

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

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

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

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

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

which is a metric that can be computed.

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

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

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

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

Finally, for completeness we note that other approximations can be derived. For example, if the RF class probability estimate

[26] 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.

is modified from Equation (2) to be

$$\widehat{P}_{\mathrm{RF}}(y|\mathbf{x}) = \frac{\sum_{b=1}^{B} n_b(\mathbf{x})\theta_b(y|\mathbf{x})}{\sum_{b=1}^{B} n_b(\mathbf{x})}, \tag{A14}$$

where $n_b(\mathbf{x})$ is the number of training set objects sharing a terminal node with $\mathbf{x}$ in tree b, then one is led to the selection metric

$$S_3(\mathbf{x}') = \sum_{\mathbf{x}\in\mathcal{U}} \frac{B(1 - \max_y \widehat{P}_{\mathrm{RF},\mathcal{L}}(y|\mathbf{x}))}{B\rho(\mathbf{x}',\mathbf{x}) + N(\mathbf{x})}\rho(\mathbf{x}',\mathbf{x}), \tag{A15}$$

where $N(\mathbf{x}) = \sum_b n_b(\mathbf{x})$.

REFERENCES

Auvergne, M., Bodin, P., Boisnard, L., et al. 2009, A&A, 506, 411

Ball, N. M., Loveday, J., Fukugita, M., et al. 2004, MNRAS, 348, 1038

Bloom, J. S., & Richards, J. W. 2011, in Advances in Machine Learning and Data Mining for Astronomy, ed. M. J. Way et al. (Boca Raton, FL: CRC Press)

Blum, A., & Mitchell, T. 1998, in Proc. Eleventh Annual Conf. on Computational Learning Theory (New York: ACM), 92

Bonfield, D. G., Sun, Y., Davey, N., et al. 2010, MNRAS, 405, 987

Breiman, L. 2001, Mach. Learn., 45, 5

Brinker, K. 2003, in Proc. 20th Int. Conf. on Machine Learning (Palo Alto, CA: AAAI Press), 59

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, Stat. Sci., 10, 273

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

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

D'Abrusco, R., Staiano, A., Longo, G., et al. 2007, ApJ, 663, 752

Debosscher, J., Sarro, L. M., Aerts, C., et al. 2007, A&A, 475, 1159

Debosscher, J., Sarro, L. M., López, M., et al. 2009, A&A, 506, 519

Donmez, P., Carbonell, J., & Schneider, J. 2009, in Proc. 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (New York: ACM), 259

Dubath, P., Rimoldini, L., Süveges, M., et al. 2011, MNRAS, 414, 2602

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

Goldman, S., & Zhou, Y. 2000, in Proc. 17th Int. Conf. on Machine Learning, (ICML 2000) (San Mateo, CA: Morgan Kaufmann), 327

Heckman, J. 1979, Econometrica: J. Econom. Soc., 47, 153

Huang, J., Smola, A., Gretton, A., Borgwardt, K., & Scholkopf, B. 2007, Adv. Neural Inf. Process. Syst., 19, 601

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

Kessler, R., Bassett, B., Belov, P., et al. 2010, PASP, 122, 1415

Lewis, D., & Gale, W. 1994, in Proc. 17th Annu. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, ed. W. B. Croft & C. J. van Rijsbergen (New York: Springer), 3

Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, MNRAS, 389, 1179

Liu, Y. 2004, J. Chem. Inf. Comput. Sci., 44, 1936

LSST Science Collaborations, Abell, P. A., Allison, J., et al. 2009, arXiv:0912.0201

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

Newling, J., Varughese, M., Bassett, B., et al. 2011, MNRAS, 414, 1987

Nigam, K., & Ghani, R. 2000, in Proc. Ninth Int. Conf. on Information and Knowledge Management (New York: ACM), 86

Olsson, F., & Tomanek, K. 2009, in Proc. Thirteenth Conf. on Computational Natural Language Learning, Association for Computational Linguistics, 138

Perryman, M. A. C., de Boer, K. S., Gilmore, G., et al. 2001, A&A, 369, 339

Perryman, M. A. C., Lindegren, L., Kovalevsky, J., et al. 1997, A&A, 323, L49

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

Pojmanski, G. 2000, Acta Astron., 50, 177

Pojmański, G. 2001, in ASP Conf. Ser. 246, IAU Colloq. 183, Small Telescope Astronomy on Global Scales, ed. B. Paczynski, W.-P. Chen, & C. Lemme (San Francisco, CA: ASP), 53

Pojmanski, G. 2002, Acta Astron., 52, 397

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

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

Richards, G. T., Deo, R. P., Lacy, M., et al. 2009, AJ, 137, 3884

Richards, J. W., Homrighausen, D., Freeman, P. E., Schafer, C. M., & Poznanski, D. 2011a, MNRAS, accepted

Richards, J. W., Starr, D. L., Butler, N. R., et al. 2011b, ApJ, 733, 10

Roy, N., & McCallum, A. 2001, in Proc. 18th International Conf. on Machine Learning (San Mateo, CA: Morgan Kaufmann), 441

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

Settles, B. 2010, Active Learning Literature Survey, Technical Report, CS Technical Report 1648, Univ. Wisconsin–Madison

Shimodaira, H. 2000, J. Stat. Plan. 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., Dziembowski, W. A., Udalski, A., et al. 2011, Acta Astron., 61, 1

Sugiyama, M., Krauledat, M., & Müller, K. 2007, J. Mach. Learn. Res., 8, 985

Sugiyama, M., & Müller, K. 2005, Stat. Decis., 23, 249

Sypniewski, A. J., & Gerdes, D. W. 2011, BAAS, 43, 150.04

Tong, S., & Chang, E. 2001, in Proc. Ninth ACM Int. Conf. on Multimedia (New York: ACM), 107

Tong, S., & Koller, D. 2002, J. Mach. Learn. Res., 2, 45

Tsalmantza, P., Kontizas, M., Bailer-Jones, C. A. L., et al. 2007, A&A, 470, 761

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

Udalski, A., Soszynski, I., Szymanski, M., et al. 1999a, Acta Astron., 49, 1

Udalski, A., Soszynski, I., Szymanski, M., et al. 1999b, Acta Astron., 49, 223

Udalski, A., Soszynski, I., Szymanski, M., et al. 1999c, Acta Astron., 49, 437

Vlachos, A. 2008, Comput. Speech Lang., 22, 295

Wadadekar, Y. 2005, PASP, 117, 79

Wozniak, P. R., Udalski, A., Szymanski, M., et al. 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 Int. Conf. on Computer Vision, 516

Zadrozny, B. 2004, in Proc. Twenty-first Int. Conf. on Machine Learning (New York: ACM), 114