Photometric Classification with Thompson Sampling

Alasdair Nam Thang Tran

A thesis submitted in partial fulfilment of the degree of
Bachelor of Science (Honours) at
The Department of Computer Science
Australian National University

October 2015

© Alasdair Nam Thang Tran

Typeset in Palatino by TEX and LATEX 2ε.

Except where otherwise indicated, this thesis is my own original work.

Alasdair Nam Thang Tran
22 October 2015

To my parents,
Tran Hai Hau and Pham Thi Hong Duc.

Acknowledgements

I would like to thank my supervisors, Dr Cheng Soon Ong and Dr Christian Wolf, for their support and encouragement throughout the year. Thank you for providing me with such an interesting project to work on, for spending countless hours giving me valuable advice, and for always checking up on me to make sure that I am still on track. I would also like to thank Chris Blake, Dominik Klaes, Thomas Erben, Hendrik Hildebrandt, Catherine Heymans, Christian Wolf, and the whole VST ATLAS team for allowing me to use data from their Southern Sky survey in my experiments. Finally I would like to say a special thank you to my friend Lee Wei Yen and my family who have kept me sane throughout my honours year.

Abstract

Recent sky surveys such as the Sloan Digital Sky Survey and the VST ATLAS Survey have given us a large amount of data to work with. Spectroscopic labelling, however, is quite expensive, so we only want to manually label an object if doing so allows us to gain new information. This thesis explores pool-based active learning and the novel application of prominent active learning heuristics to the domain of photometric classification.

We begin by applying standard supervised machine learning techniques to two astronomical datasets. The best-performing classifiers with reliable probability estimates, logistic regression and support vector machines, are then used to conduct the active learning experiment. Our key original contribution is the application of Thompson sampling, a Bayesian solution to the exploration vs exploitation problem, to the selection of six active learning heuristics. To address the problem of class imbalance, we derive an extension of the posterior balanced accuracy to the multi-class setting. This is used to evaluate the performance of our algorithms.

The results are very promising. Even under simplistic assumptions like a normally distributed reward, Thompson sampling manages to automatically identify the optimal heuristic after only 50 examples. In particular, the margin minimisation technique is a clear winner, outperforming random sampling by as much as 9% in the VST ATLAS dataset after 300 examples. Being very computationally efficient, we recommend that astronomers use the margin heuristic with logistic regression to decide which objects to label next in future sky surveys. To help them with this task, we have created an open-source and extendable Python package that allows others to easily apply active learning routines to any dataset and make quick visualisation of photometric data.

Contents

List of Figures

List of Symbols

A	Balanced accuracy
A_i	Recall of class i
$\mathcal{A}_u$	Extinction value in the u-band
a	Inverse hyperbolic sine softening parameter
α	First shape parameter of the Beta distribution
$\mathcal{B}$	Committee of classifiers $\mathcal{B} = \{h_1, h_2, ..., h_B\}$
B	Size of the committee of classifiers
b	Bias term in SVMs
β	Second shape parameter of the Beta distribution
$\text{Beta}(\alpha, \beta)$	Beta distribution
$\text{Bin}(n, p)$	Binomial distribution
C	Penalty parameter in logistic regression and SVMs
d	Number of features
D	Distance between an object and Earth
$D_{\text{KL}}(p\|\|q)$	Kullback–Leibler divergence
∇p	Gradient vector
δ	Incremental increase in the accuracy
$\mathcal{E}$	Random subset of the unlabelled pool $\mathcal{U}$
E	Size of the random subset $\mathcal{E}$
E_{B-V}	Reference interstellar reddening value
$\mathbb{E}[X]$	Expected value of the random variable X
$\eta(\boldsymbol{x})$	Linear predictor in logistic regression
f	Detected flux of an object

f_0	Flux of the object with a conventional magnitude of zero
$F_X(x)$	Cumulative distribution function of the random variable X
$f_X(x)$	Probability density function of the continuous random variable X
G_i	Total number of objects in class i that are correctly predicted
$\boldsymbol{g}$	Vector $(g_1, g_2, ..., g_k)$
g_i	Number of objects in class i in a sample that are correctly predicted
$\Gamma(x)$	Gamma function
γ	Free parameter in the RBF kernel
$H(\mathcal{L}_T; h)$	Entropy of predictions under hypothesis h and with training set $\mathcal{L}_T$
h	Classifier or hypothesis $h : \mathcal{X} \to \mathcal{Y}$
$\mathcal{I}$	Fisher information matrix
$\mathbb{I}$	Indicator function
$I(r)$	Surface brightness profile of a galaxy
$\iota_G(D)$	Gini impurity of the set D
k	Number of classes or labels
$\mathcal{L}$	Pool of labelled data
$\mathcal{L}_S$	Test set
$\mathcal{L}_T$	Training set
λ	Wavelength of light
$\ell(x)$	Loss function
M	Absolute magnitude of an object
m	Apparent magnitude of an object
$\boldsymbol{\mu}$	Vector $(\mu_1, \mu_2, ..., \mu_s)$
μ_i	Mean of the prior distribution of the mean reward of heuristic r_i
n	Size of the training set
N_i	Total number of objects belonging to class i in the universe
n_i	Number of objects belonging to class i in a sample

$\mathcal{N}(\mu, \sigma^2)$	Normal distribution
$\boldsymbol{\nu}$	Vector $(\nu_1, \nu_2, ..., \nu_s)$
ν_i	Mean of the reward of heuristic r_i
O	Outer product of the gradient vector with itself.
$\mathbb{P}(A)$	Probability that event A is true
$p(y\|\boldsymbol{x}; h)$	Probability that $\boldsymbol{x}$ has label y under hypothesis h
$q_D(i)$	Frequency of objects belong to class i in set D
Q_{ij}	Entry in the ith row and jth column in the confusion matrix
$\mathcal{R}$	Set of active learning heuristics
$r(\boldsymbol{x}; h)$	Active learning rule or heuristic
ρ_i	Reward of heuristic r_i
$\mathfrak{R}_p(r)$	Petrosian ratio
r_p	Petrosian radius
$R(w)$	Regularisation term in logistic regression and SVMs
s	Number of heuristics
$\boldsymbol{\sigma}^2$	Vector $(\sigma_1^2, \sigma_2^2, ..., \sigma_s^2)$
σ_i^2	Variance of the prior distribution of the mean reward of heuristic r_i
$\sigma(x)$	Logistic or sigmoid function
$\boldsymbol{\tau}^2$	Vector $(\tau_1^2, \tau_2^2, ..., \tau_s^2)$
τ_i^2	Variance of the reward of heuristic r_i
$\boldsymbol{\theta}$	Coefficient vector in logistic regression
$T(\lambda)$	Transmission function of a filter
$\mathrm{tr}(X)$	Trace of matrix X
$\mathcal{U}$	Pool of unlabelled data
$V(\mathcal{L}_T; h)$	Variance under hypothesis h and with training set L_T
$\boldsymbol{w}$	SVM weight vector
$\mathcal{X}$	Feature space

$\boldsymbol{x}$	Feature vector of an object
$\mathcal{Y}$	Label space
y	Label or class of an object

Acronyms

ANU	Australian National University
CDF	Cumulative Distribution Function
CV	Cross Validation
dec	Declination
KL	Kullback-Leibler
MPBA	Mean of the Posterior Balanced Accuracy
PBA	Posterior Balanced Accuracy
PDF	Probability Density Function
PSF	Point Spread Function
QBB	Query by Bagging
ra	Right Ascension
RBF	Radial Basis Function
SDSS	Sloan Digital Sky Survey
SF11	Schlafly and Finkbeiner [2011]
SFD98	Schlegel, Finkbeiner, and Davis [1998]
SVM	Support Vector Machine
VLT	Very Large Telescope
VST	VLT Survey Telescope
W14	Wolf [2014]
WISE	Wide-field Infrared Survey Explorer

Chapter 1

Introduction

The night sky has always been a source of wonder throughout history. Even before the Age of Science, we used our built-in pattern recognition and tried to assign a purpose to the cosmos with stories of gods and the supernatural. Then came the Ancient Greeks who brought mathematics to astronomy and sought a more rational explanation of celestial bodies. But we could only discover so much with our naked eyes. It was not until the invention of new technology during the Scientific Revolution like the telescope that we began to make significantly more progress. And yet, 400 years after Galileo made the first working telescope, most of the sky still remains unknown to us.

This, however, is starting to change. The Sloan Digital Sky Survey (SDSS) began in 2000 and has since obtained 800 million sets of photometric measurements and 3 million spectra of the Northern Sky. Similar projects are going on to map the Southern Sky including the VST ATLAS and the SkyMapper surveys. We now face another challenge of how to analyse this huge amount of data. Fortunately, we can bring machine learning to astronomy and let the computer do the pattern recognition task. This thesis is an attempt to study a small part of the interaction between these two disciplines.

1.1 Contributions

The thesis centres around the problem of how we can use machine learning to choose objects whose spectroscopic labels would be most informative to the task of photometric classification. Our key novel contribution is the application of Thompson sampling, a Bayesian solution to the exploration-exploitation trade-off, to the selection of six active learning heuristics. The algorithm is described in Section 3.5 and 3.6, and the experimental results are discussed in Chapter 5. As far as we know, this is the first time that such active learning heuristics are applied to the astronomical domain. It is also the first time that the multi-arm bandit problem with Thompson sampling is used in the heuristic selection context. Along the way, we make four other contributions that would support the active learning experiment. These are

- Comparing three sets of dust extinction vectors and seeing which one is the most effective at improving the accuracy (Section 4.3.1).

- Identifying good photometric features using both well-known machine learning techniques like polynomial transformation and domain knowledge such as the use of colours (Section 4.1 and 4.3.2).
- Training three families of classifiers to do photometric classification and using a random forest to predict the class proportion of unlabelled data (Section 4.3.3 and 4.3.4).
- Deriving an extension of the posterior balanced accuracy to the multi-class setting, which will be used to evaluate the performance of our algorithms (Section 3.7).

An important part of this project involves creating an open-source, extendable, and well-documented Python package that allows astronomers to perform active learning routines and make quick visualisations of photometric data. The package, the documentation, and reproducible code of all experiments are available on the project's GitHub repository[1].

1.2 Thesis Outline

The thesis consists of four main chapters, two of which are dedicated to a discussion of ideas and the other two focus on experiments:

- Chapter 2 introduces the reader to the tools that astronomers use to map the sky. This includes important concepts like photometry, magnitudes, dust extinction, and the celestial coordinate system. We also give a brief overview of two datasets, SDSS and VST ATLAS, that are used in our experiments.
- Chapter 3 introduces machine learning and surveys the relevant literature in pool-based active learning. This leads to a discussion of the multi-arm bandit problem and the use of Thompson sampling in the setting where each bandit arm is an active learning heuristic. We end the chapter with a derivation of multi-class posterior balanced accuracy, an important performance measure for data with unbalanced classes.
- Chapter 4 and 5 test the ideas discussed so far in a series of experiments on the SDSS and the VST ATLAS datasets. A detailed protocol and a thorough discussion of the results are given for each experiment. In particular, we offer insights into the areas where Thompson sampling outperforms random sampling and where it underperforms.

For those interested in reproducing the experiments, Appendix A provides information on how to obtain the SDSS dataset, including the necessary SQL queries. Supplementary results are included in Appendix B and C.

[1] https://github.com/alasdairtran/mclearn

Chapter 2

Mapping the Night Sky

When many people think about astronomy, the first things that pop into their mind are breathtaking images of the sky like the famous Hubble Deep Field. But pretty images alone are not enough to do real science. For us to make any meaningful observations about astronomical objects, we need actual quantitative measurements. This chapter begins with an overview of spectroscopy and photometry, the two most important tools in optical astronomy used to study the sky (Section 2.1). We then look at important astronomical concepts like fluxes, magnitudes, colours, and the equatorial coordinate system (Section 2.2, 2.3, and 2.4). This will give us a better understanding of the SDSS and the VST ATLAS datasets, which are introduced in Section 2.5. The chapter ends with a brief discussion on dust extinction, a potential problem in the SDSS dataset that we shall need to take care of (Section 2.6).

2.1 Spectroscopy and Photometry

Spectroscopy was born when Isaac Newton discovered in 1666 that white light can be split into a rainbow by passing it through a glass prism. In modern-day astronomy, we use diffraction grating to disperse light and measure the amount of electromagnetic radiation, or flux, emitted from a celestial object at small wavelength intervals. As the name implies, we end up with a spectrum, like the one shown in Figure 2.1. The shape of the spectrum and its absorption lines allow us to deduce many useful properties about the object such as its temperature and chemical composition.

Unfortunately, it can be very difficult to take high resolution spectra of faint objects since we are spreading light thinly across many wavelengths. Photometry gets around this problem by separating light into fewer groups and thus reducing the wavelength resolution [Romanishin 2002, Chapter 1]. Specifically we have a set of filters, each of which can be put in front of the CCD camera to allow only light from certain wavelengths to pass through. Associated with each filter is a transmission function $T(\lambda)$ that tells us the fraction of light that the filter will transmit at wavelength λ. Figure 2.1 shows $T(\lambda)$ of the five bandpasses (u, g, r, i, and z) that are used in the SDSS.

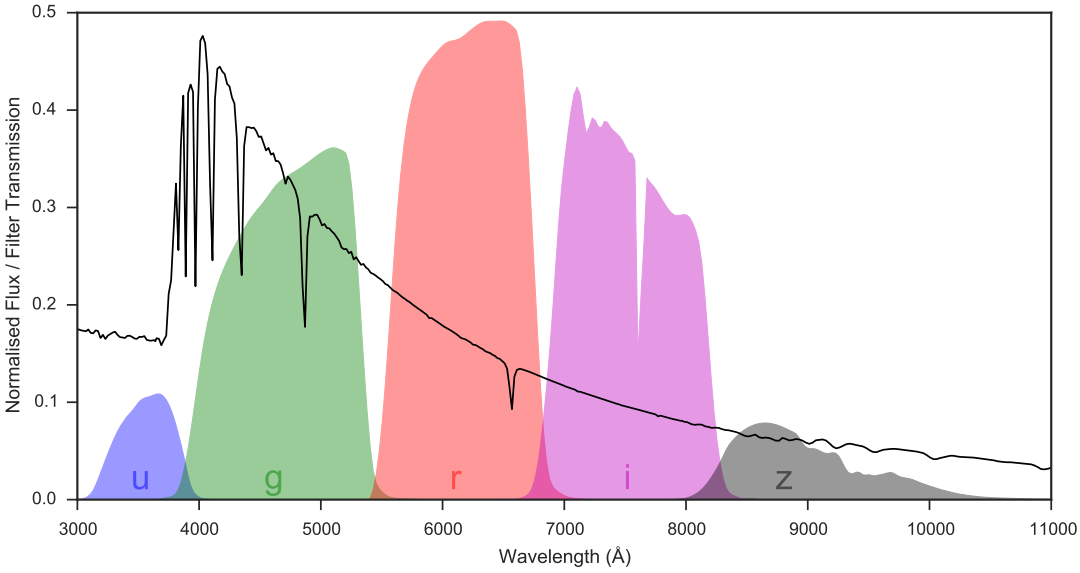


Figure 2.1: The black curve is the spectrum of Vega, the fifth brightest star in the night sky. The spectrum tells us how much radiation Vega emits at each wavelength. We also show five transmission functions, one for each of the five ugriz filters. The transmission function tells us how much light can get through the filter at each wavelength.

2.2 Measuring Fluxes

When light hits the CCD, all we have initially are counts of photons, one for each pixel. The first challenge is to assign the photons to distinct objects. Two models are used in the SDSS, depending on whether we assume the object is an extended source or a point source.

2.2.1 Petrosian Flux

Galaxies are extended-source objects so we need to define an aperture radius, within which all the photons are added together to obtain the flux of the galaxy. Since galaxies have poorly defined edges, a consistent method to pick the aperture radius is required. Blanton et al. [2001] define a quantity called the Petrosian ratio:

$$\mathfrak{R}_p(r) = \frac{\int_{0.8r}^{1.25r} 2\pi s I(s)\, ds}{\pi(1.25^2 - 0.8^2) r^2} \Bigg/ \frac{\int_0^r 2\pi s I(s)\, ds}{\pi r^2}$$

This is the ratio of the mean local surface brightness over an annulus at r to the mean surface brightness within r. The quantity $I(r)$ is the galaxy surface brightness profile and can be estimated from the photon count. With this, we define the Petrosian radius r_p as the radius such that $\mathfrak{R}_p(r_p) = 0.2$. This is one of the features in the SDSS dataset that will prove to be very useful in distinguishing galaxies from point-source objects. The aperture radius is then chosen to be $2r_p$ to ensure that almost all of the light

from a typical galaxy is captured. Finally for each bandpass, we can calculate the corresponding Petrosian flux as

$$f = \int_0^{2r_p} 2\pi s I(s)\, ds$$

2.2.2 Point Spread Function Fitting

Stars, quasars, and white dwarfs are unresolved point sources so they can be modelled by a point spread function (PSF). This approach is particularly useful when we examine a dense region like a globular cluster. In such places, given the amount of overlap, it would be very difficult define an aperture that includes only photons from an object and excludes all others from the neighbours. In PSF fitting, we assume that all objects have the same shape, which allows us to fit a Gaussian model to each of them. We then iteratively vary the position and flux of the objects until the model produces the observed light distribution [Palmer and Davenhall 2001, Chapter 10]. The flux estimated by the converged model is called the PSF flux.

2.3 Magnitudes and Colours

2.3.1 Apparent Magnitudes

The fluxes of the brightest and the dimmest objects in the sky can differ by many orders of magnitudes. This motivates us to take one step further and convert fluxes to inverse hyperbolic sine (or arsinh) apparent magnitudes:

$$m = -\frac{2.5}{\ln(10)}\left[\operatorname{arsinh}\left(\frac{f/f_0}{2b}\right) + \ln(a)\right]$$

where f_0 is the flux of the object with a conventional magnitude of 0 and a is the softening parameter. A nice feature of this magnitude system is that for bright objects with a high signal-to-noise ratio, it behaves like a logarithmic scale, i.e. with every decrease of 1 in the magnitude scale, the object becomes 2.5 times brighter.[1] At the same time, as the flux tends toward zero for fainter objects, the arsinh function (unlike the log function) ensures that the magnitudes are still well-defined [Lupton et al. 1999].

2.3.2 Absolute Magnitudes and Colours

The problem with apparent magnitudes is their dependence on distance. Objects that are further away from us are fainter and hence have higher apparent magnitudes. Of course, being further away does not change anything fundamental about an object like whether it is a star or a galaxy. Thus if we want to study their intrinsic properties, we

[1] The reader might wonder why the scale works in reverse, with a small magnitude corresponding to more brightness. This is the convention created two millennia ago by the Greek astronomer Hipparchus, which, for better or worse, has stuck with us ever since.

need to remove this dependency. One method is to convert to the absolute magnitude, which is defined as the object's apparent magnitude if it were exactly 10 parsecs away from Earth:

$$M = m - 5\log_{10}\left(\frac{D}{10}\right)$$

This conversion requires the knowledge of D, the actual distance of an object in parsecs. In practice, it is often difficult to estimate D, so we resort to an easier method of taking the difference between the amount of light received in two bandpasses. For example, the $u - g$ colour is calculated as

$$\begin{aligned} m_{u-g} &= m_u - m_g \\ &= M_u + 5\log_{10}\left(\frac{D}{10}\right) - M_g + 5\log_{10}\left(\frac{D}{10}\right) \\ &= M_u - M_g \end{aligned}$$

Note that the conversion factor disappears and the colour does not change with distance. In addition, the difference between two magnitudes depends on the average slope of the spectrum. Thus colours measure the general shape of an object's spectrum, which in turn can reveal many useful properties such as a star's temperature.

2.4 Equatorial Coordinate System

Imagine a very large celestial sphere with Earth at its centre. By projecting onto the inside surface of this sphere, we have a way to specify the position of any astronomical object. In this thesis, we use the equatorial coordinate system, where an object's position is specified by two numbers, a right ascension (ra) and a declination (dec). Figure 2.2 shows the Mollweide projection of the celestial sphere. We shall use this map throughout the thesis to visualise some results.

The equatorial coordinate system, like any other systems, needs to have a reference point. Here anything on the celestial sphere that is directly above the Earth's equator will have a declination of 0°. To define a zero point for the right ascension, we use the fact that the centre of the Sun passes through the plane of the Earth's equator twice a year. The first crossing point usually happens on 21 March and is called the vernal equinox. We now define the right ascension of the vernal equinox to be 0° [Spark and Gallagher 2007, Chapter 1].[2]

[2] There is actually a slight complication. Since the Earth's rotation axis is not fixed due to precession, the ra-dec coordinates of an object relative to the origin will actually change slowly over time. Thus we need to also fix a time in which the coordinates are measured. In many surveys like the SDSS, 1 January 2000 12:00 Terrestrial Time is chosen as a reference point.

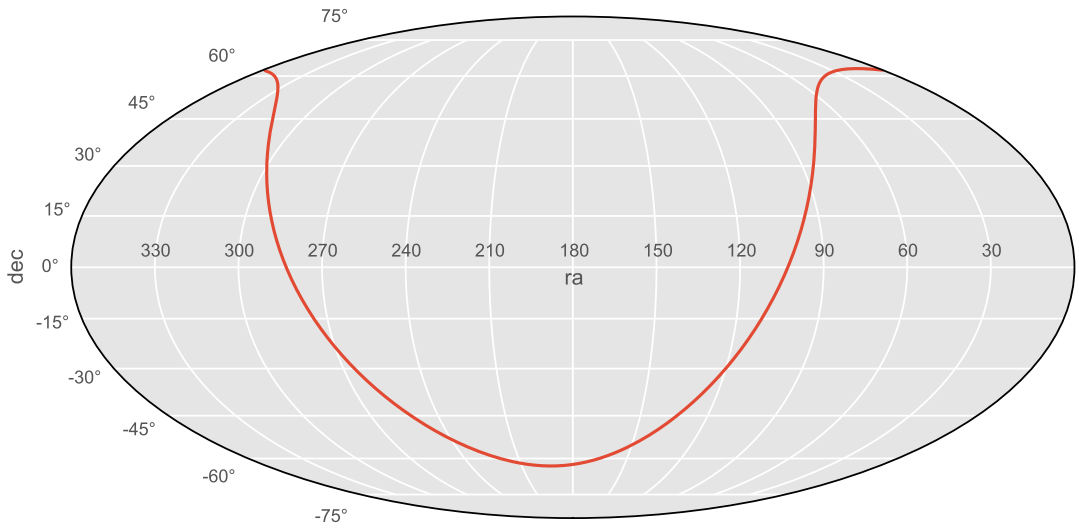


Figure 2.2: This is the Mollweide projection of the celestial sphere under the equatorial coordinate system. The red line indicates the plane of the Milky Way. To avoid cluttering, the coordinate labels will not be shown on later maps.

2.5 Datasets

We now have all the required background to understand the features in the two datasets used in our experiments. Below we give a quick overview of what each dataset contains. For more information on how to obtain them, refer to Appendix A.

2.5.1 SDSS Dataset

The main dataset in our investigation comes from the SDSS, a comprehensive survey of the Northern Sky that began operation in 2000. This dataset consists of 800 million objects, covering 14,055 square degrees, or about a third of the celestial sphere [Alam et al. 2015]. Figure 2.3 shows the coverage of the survey. For each object, we are given 11 features:

- The PSF magnitude in each of the five ugriz bands.
- The Petrosian magnitude in each of the five ugriz bands.
- The Petrosian radius in the r-band.

Only 2.8 million out of the 800 million objects have been spectroscopically classified into three classes: galaxies, quasars, and stars. From Figure 2.4a, we can see that the number of galaxies is twice that of stars and four times that of quasars. This leads to the problem of class imbalance. In Chapter 3 we shall discuss some approaches to minimise the bias toward the dominant class during classification. Also note that both the labelled and the unlabelled sets are not random samples of the sky. For example,

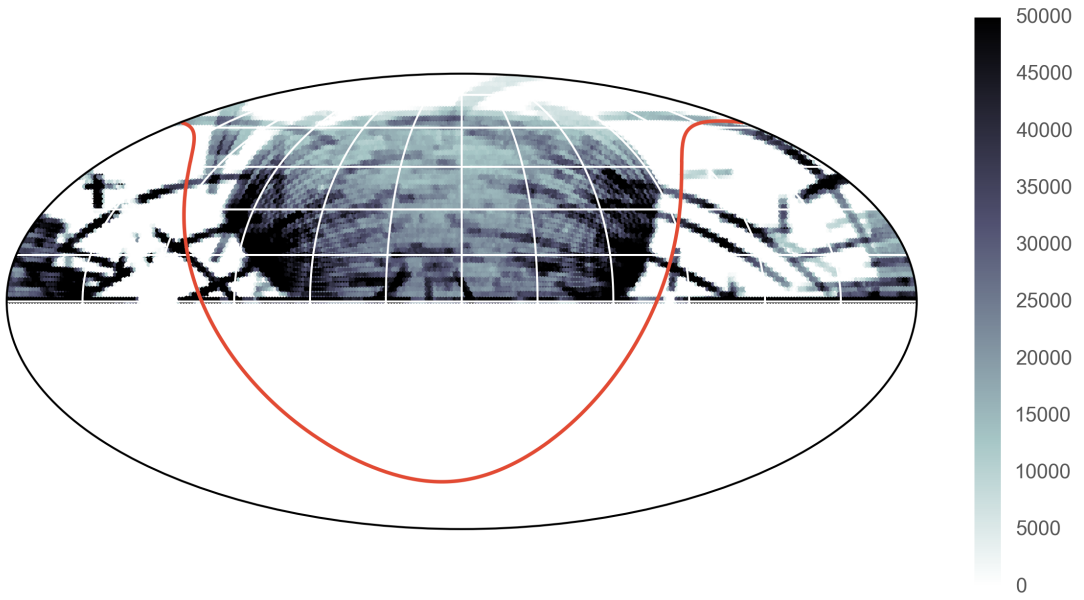
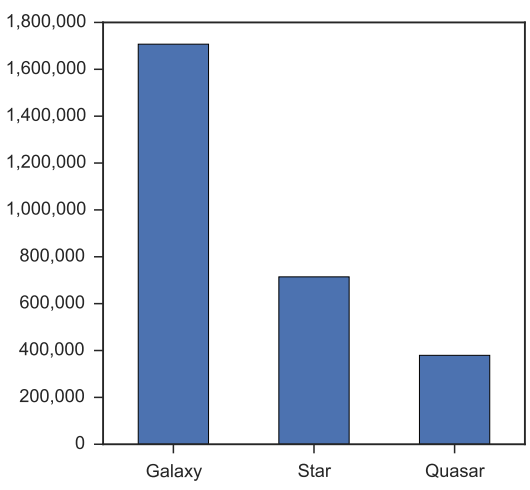
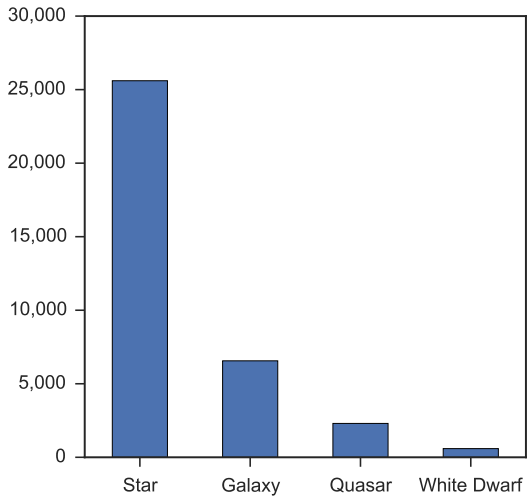


Figure 2.3: The distribution of the 800 million scanned objects in the SDSS: Note that the coverage is not uniform. A darker colour corresponds to more objects being scanned in a particular area. The unit of the colourbar is the object count in a hexagon.



(a) SDSS Dataset



(b) VST ATLAS Dataset

Figure 2.4: Class distribution of the labelled objects in the SDSS and VST ATLAS datasets: Observe that both datasets exhibit the problem of class imbalance. Most objects in the SDSS dataset are galaxies, while stars are the dominant class in the VST ATLAS dataset.

since astronomers are particularly interested in studying quasars, these objects are over-represented in the collection.

2.5.2 VST ATLAS Dataset

A second and much smaller dataset comes from a more recent survey, the VST ATLAS. This project aims to survey 4500 square degrees of the Southern Sky to roughly the same depth as the SDSS [Shanks et al. 2015]. There are about 35,000 objects in total which have been classified by experts into four classes: stars, galaxies, quasars, and white dwarfs. For each object, we are provided with 7 features:

- The calibrated magnitude from the r-band.
- Four colour indices using the ugriz filters: $u - g$, $g - r$, $r - i$, and $i - z$.
- Two colour indices in the infrared: $r - W1$ and $W1 - W2$.

The two infrared channels $W1$ and $W2$ are measured by the Wide-field Infrared Survey Explorer (WISE), one of NASA's space telescopes. These channels will prove useful in distinguishing between classes. As we can see from Figure 2.4b, the problem of class imbalance is even worse in this dataset. For instance, we have 43 times more stars than white dwarfs. Since research is still on-going, the coordinates of the objects are not yet publicly available.

2.6 Dust Extinction

As light travels from its source to us, it can be absorbed and scattered by interstellar dust. The scattering is especially strong at shorter wavelengths. This means that less of the blue light arrives on Earth and the object will appear redder than it actually is.

This would not be a big problem if the reddening process were uniform throughout the celestial sphere, since all the magnitudes and colours will simply shift by a constant term. Indeed, this is the case with the VST ATLAS, since the field where the objects were surveyed exhibits no drift in the reddening. However, the SDSS covers a much larger portion of the sky. As shown in Figure 2.5, the reddening varies by quite a bit throughout the field. The effect is particularly strong in the Milky Way region. Thus we might need to correct the SDSS magnitudes for dust extinction.

Three competing extinction vectors currently exist in the literature. The most popular one for a long time was created by Schlegel, Finkbeiner, and Davis [1998], who used the colours of nearby elliptical galaxies for calibration. Later, Schlafly and Finkbeiner [2011] made some improvement with new data from the SDSS. Recently, Wolf [2014] investigated quasars in the SDSS and found evidence of non-linearity in the reddening map. For convenience, let us call the extinction vectors resulted from the above works SFD98, SF11, and W14, after the authors' initials and year of publication. One interesting question, which we shall explore in Chapter 4, is whether applying any of these vectors to the measurements will have any noticeable effect on the performance of the classifiers.

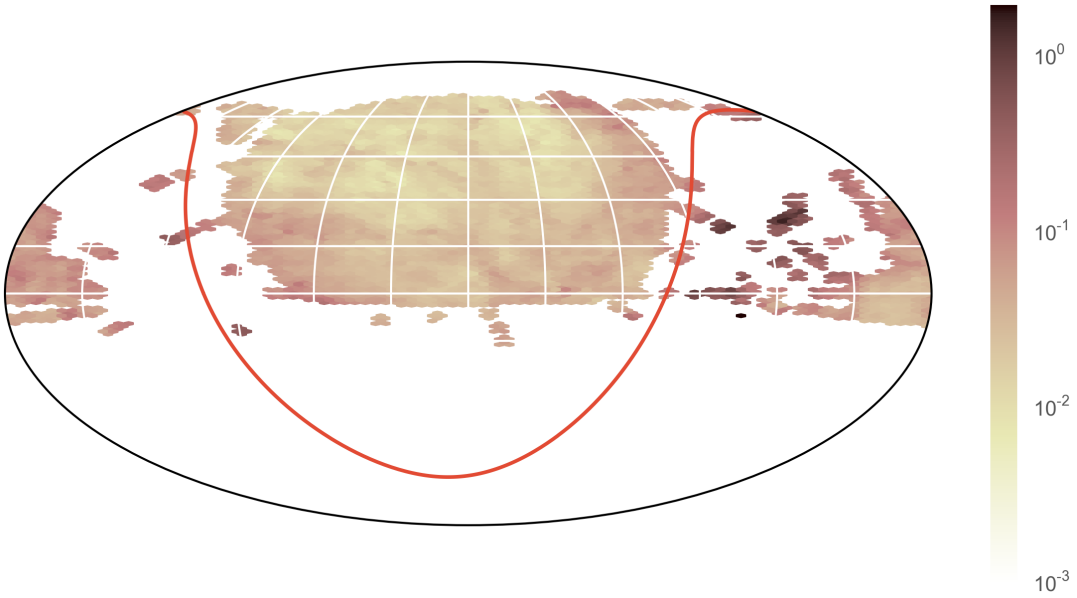


Figure 2.5: Galactic reddening E_{B-V} map in the SDSS: A darker colour indicates a greater amount of reddening.

Chapter 3

Photometric Classification

Let us now bring machine learning to the realm of astronomy. We start with a motivation for why astronomers might find machine learning helpful (Section 3.1) and an overview of three families of classifiers (Section 3.2). We then discuss active learning and six heuristics that can be used to rank unlabelled examples (Section 3.3 and 3.4). Our novel contributions include the application of Thompson sampling to the heuristic selection setting (Section 3.5 and 3.6) and a derivation of the multi-class posterior balanced accuracy (Section 3.7) which can be used to measure the performance of our algorithms.

3.1 Maching Learning in Astronomy

The two most common types of celestial objects are stars and galaxies. There are also some other interesting objects such as quasars and white dwarfs. Quasars are thought to be supermassive black holes surrounded by an accretion disc. They are very luminous and, unlike galaxies, appear as single-source objects. White dwarfs are low to intermediate mass stars that are in their final evolutionary stage. They are very dense and have a faint luminosity.

One way to classify objects into these various groups is to manually inspect their spectra. There have even been attempts to make the process more automatic. For example, Hála [2014] achieved a 95% accuracy rate by training a convolutional neural network on one-dimensional spectra to classify objects into stars, quasars, and galaxies. Even so, it is currently not possible to obtain a spectrum of every object, especially faint ones. This means that only a small number of objects (e.g. 0.35% in the SDSS dataset) can be classified this way. For the rest, we only have photometric measurements.

Fortunately, the field of machine learning came about to solve this kind of problem. In the most basic set-up, we have a collection of objects, each with a vector of photometric measurements $\boldsymbol{x} \in \mathcal{X}$. A subset of them has been spectroscopically classified into some class $y \in \mathcal{Y}$ and they form the labelled set $\mathcal{L} \subset \mathcal{X} \times \mathcal{Y}$. We call $\mathcal{X}$ the feature space and $\mathcal{Y}$ the label space. Let us now partition $\mathcal{L}$ into two disjoint subsets, a training set $\mathcal{L}_T$ and a test set $\mathcal{L}_S$. During the training phase, we feed $\mathcal{L}_T$ to a classifier, and the classifier will then learn a hypothesis $h : \mathcal{X} \to \mathcal{Y}$. The labelling process might

not be perfect, so the goal in machine learning is to capture as much of the underlying trend in the data as possible, while avoiding fitting the random noise. To see how well the hypothesis generalises to unknown data, in the testing phase, we ask the classifier to predict labels of objects in $\mathcal{L}_S$. These predictions are then be compared to the true labels, and an accuracy rate can be calculated.

3.2 Classifiers

Three families of classifiers are used in our experiments. They are random forests, logistic regression, and support vector machines. Below we give a quick overview of how each of them works. We do not implement these classifiers ourselves. Rather we shall use scikit-learn [Pedregosa et al. 2011], the most well-known machine learning package in Python.

3.2.1 Random Forest

To understand the motivation behind random forests, we first need to know how to construct individual decision trees. Building each of these trees is like playing a game of Twenty Questions. We start with the whole training set and at each step, we slice the feature space along the axis of one feature. After many steps, we end up with a set of hyper-dimensional cuboids which form a partition of the feature space. The algorithm stops when each of these cuboids contains data from only one class. There are many criteria that we can use to decide on which feature and where we slice along the axis. In this thesis, we use the Gini impurity which intuitively measures the potential misclassification rate. In particular, let k be the number of classes and $q_D(i)$ be the frequency of objects belonging to class i in set D. Then the Gini impurity of D is the probability that a randomly selected object from D is misclassified if it were labelled according to its frequency in D:

$$\begin{aligned}\iota_G(D) &= \sum_{i=1}^{k} q_D(i)(1 - q_D(i)) \\ &= \sum_{i=1}^{k} q_D(i) - \sum_{i=1}^{k} q_D(i)^2 \\ &= 1 - \sum_{i=1}^{k} q_D(i)^2\end{aligned}$$

When we partition D into subsets $\{D_1, D_2, ..., D_d\}$, the Gini impurity of D is now the sum of the individual Gini impurities, weighted by the size of the subsets:

$$\iota_G(D) = \sum_{i=1}^{d} \frac{|D_i|}{|D|} \iota_G(D_i)$$

Observe that if a subset contains objects from only one class, then its Gini impurity will be zero. This gives us the following splitting criterion: at each step, slice the

feature space along the axis that will result in the greatest drop in the Gini impurity.

One problem with decision trees is that they tend to overfit the data and thus do not generalise well. To solve this, Breiman [2001] proposes that we build many decision trees, thus creating a random forest. The random forest makes its prediction by simply counting up the predictions of all the individual trees and then choosing the most popular choice. By taking an average of the predictions, we avoid the problem of overfitting. Furthermore, for each tree, we only give it a small bootstrap sample and at each split, we only consider a small number of features. This bootstrapping and random subspace selection have been shown empirically to improve the accuracy [Breiman 1996; Ho 1998; Louppe and Geurts 2012]. Another nice feature of random forests is that they are extremely fast and hence scale well with large datasets. Although they do not provide class probability estimates, we can use proportions of the votes as a proxy for the probabilities. However, in practice, we find these probability estimates to be a bit unstable.

3.2.2 Logistic Regression

If we want the hypothesis to model actual class probabilities, then an alternative approach is to use logistic regression. Developed by Cox [1958], the algorithm tries to directly model the probability of being in a class. Let $\boldsymbol{x}$ be the feature vector and $\boldsymbol{\theta}$ be the vector of coefficients. The linear predictor η is defined as

$$\eta(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$$

Since probabilities must lie between 0 and 1, we want our predictor to have the same range. This can be achieved by wrapping η around the logistic function:

$$\begin{aligned} p(y = 1|\boldsymbol{x}; \boldsymbol{\theta}) &= \sigma(\eta(\boldsymbol{x})) \\ &= \frac{1}{1 + e^{-\eta(\boldsymbol{x})}} \end{aligned}$$

We can now interpret $p(y = 1|\boldsymbol{x}; \boldsymbol{\theta})$ as the probability that an object with feature vector $\boldsymbol{x}$ belongs to the positive class. The goal of the algorithm is then to use the training data to estimate the coefficient vector $\boldsymbol{\theta}$. This can be done by finding $\hat{\boldsymbol{\theta}}$ that maximises the log-likelihood function:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^{n} \log p(y_i|\boldsymbol{x}_i; \boldsymbol{\theta}) - \frac{1}{C} R(\boldsymbol{\theta})$$

where n is the size of the training set, $R(\boldsymbol{\theta})$ is the regularisation term, and C is the inverse of the regularisation strength. A low value of C forces the values of the parameters to be small, thus avoiding overfitting. However if C is too low, we might have a hypothesis that is too simple. For the regularisation term, we can either sum

up the absolute values of the coefficients (L1) or the squares of the coefficients (L2):

$$R_{L1}(\boldsymbol{\theta}) = \sum_{i=1}^{m} |\boldsymbol{\theta}_i| \qquad R_{L2}(\boldsymbol{\theta}) = \sum_{i=1}^{m} \boldsymbol{\theta}_i^2$$

where m is the number of features. One advantage of the L1 regularisation is that it leads to sparse solutions, where a lot of coefficients become zero [Tibshirani 1996]. This is useful if we have many features, which for example is the case after we do a polynomial transformation (see Section 3.2.4).

There are a few ways to extend the above model to the multi-class setting. One option is multinomial logistic regression, where we would need to jointly solve a set of $(k-1)$ binary regressions if we have k classes. In practice, when running the multinomial option in scikit-learn, the probability estimates are not very reliable, especially when we have many classes. The cause of this is unknown, but it is more likely due to flaws in the scikit-learn implementation than in the actual theory. A more empirically stable alternative is to use the one-vs-rest strategy, where we run k independent binary regressions. In particular, for class i, we pretend that the dataset contains only objects from class i and class 'not i'. We then train the binary logistic model on this simplified dataset and we do this k times, one for each class. At the end, we end up with k probabilities. These can be interpreted like normal class probabilities after normalisation.

3.2.3 Support Vector Machines

Support vector machines (SVMs), first introduced by Boser, Guyon, and Vapnik [1992], are another popular family of algorithms. They have been used in astronomy, for example by Elting, Bailer-Jones, and Smith [2008], to find non-linear decision boundaries in the colour space of the SDSS dataset. The idea here is to find a decision boundary that can maximise the distance between the boundary and the closest data points, which we call support vectors. This involves finding the weights $\hat{\boldsymbol{w}}$ and the bias $\hat{b}$ that minimise the objective function

$$\underset{\boldsymbol{w},b}{\operatorname{argmin}}\, R(\boldsymbol{w}) + C \sum_{i=1}^{n} \ell(\boldsymbol{x}_i)$$

where $R(\boldsymbol{w})$ can either be L1 or L2 regularisation like in logistic regression, and $\ell(\boldsymbol{x}_i)$ is the loss function. Two common loss functions are the hinge loss

$$\ell(\boldsymbol{x}_i) = \max(0, 1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b))$$

and the square of the hinge loss

$$\ell(\boldsymbol{x}_i) = \max(0, 1 - y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b))^2$$

As usual, the penalty parameter C controls the trade-off between the misclassification of training examples and the simplicity of the decision surface, with a low value of C

resulting in a simple hypothesis. SVMs, in their original formulation, are inherently binary classifiers. However we can still use the one-vs-rest strategy to extend it to the multi-class setting. There has even been an attempt to derive an inherently multi-class SVM [Crammer and Singer 2002].

3.2.4 Learning Complex Hypotheses

Both SVMs and logistic regression are linear classifiers. When dealing with real-world data like those in astronomy, we should not expect to be able to separate classes with a hyper-dimensional plane. If we want them to learn more complex hypotheses, one option is do an explicit polynomial transformation of the original photometric measurements. When we give the classifier the transformed features, it would still find a linear boundary in the transformed space. However, the boundary would most of the time be non-linear in the original space.

A second option is to use the kernel trick which does an implicit map into a high-dimensional feature space. For example, a popular kernel that is often used with SVMs is the radial basis function (RBF), which actually maps the inputs into an infinite-dimensional space. The RBF kernel has the parameter γ which is inversely proportional to the radius of influence of the support vectors. This means that a low value of γ corresponds to a smoother model.

In random forests, we do not have to worry about any transformation. Although in each round, we slice the data along only one axis, there is no limit on how many slices we can take and how small the resulting cuboids can be. This allows us to learn arbitrarily complex hypotheses.

3.3 Overview of Active Learning

We now turn our attention to the construction of the training set. Getting spectroscopic labels is expensive. Until now, astronomers do not have a quantitative method to help them choose objects whose spectroscopic labels would provide them with the most amount of new information. Often, they simply take a random sample of the sky. This, however, might not always be optimal. To see why, imagine that there is a group of objects with very similar photometric measurements. We can obtain spectra from all of them and conclude, for example, that they are all stars. However, a smarter way is to get only one spectrum from this group for labelling and let the classifier generalise to other similar objects. Keeping the size of the training set as small as possible while not sacrificing the classifier accuracy is the goal of active learning.

There are three main types of active learning: membership query synthesis, stream-based selective sampling, and pool-based active learning. In membership query synthesis, we are allowed to request labels for any unlabelled instance in the feature space [Angluin 1988]. Equivalently, we may request the astronomer to find an object with a certain combination of colours and magnitudes. This is not very realistic since such objects might not even exist. In stream-based selective sampling, we sample objects from the source one at a time, and as objects are streaming in, we must decide to either

label or discard each of them [Cohn et al. 1994]. The assumption here is that it is free to obtain unlabelled examples, which again is not applicable to astronomy. Thus we shall not discuss membership query synthesis and stream-based selective sampling further in this thesis.

Instead, we shall focus on pool-based active learning [Lewis and Gale 1994], the most relevant type of active learning for astronomy. In this setting, we keep track of two sets. As usual, we have a labelled set $\mathcal{L}$, which can be further partitioned into a training set $\mathcal{L}_T$ and a test set $\mathcal{L}_S$. There is also an unlabelled set $\mathcal{U} \subset \mathcal{X}$, which contains all the remaining unlabelled examples. The question now is how to select the next example from $\mathcal{U}$ for labelling. In practical terms, where should we next point the telescope to, in order to obtain a spectrum? To answer this question, we need a rule $r(\boldsymbol{x}; h)$ that can assign a score to each object in $\mathcal{U}$, based solely on their photometric features $\boldsymbol{x}$ and the current hypothesis h. This score should reflect the amount of new information we would gain if we were to label the object. Once we have computed $r(\boldsymbol{x}; h)$ for all candidates, we can then pick the example with the highest score and obtain its spectrum. The object's feature vector and its label are then added to the training set and the classifier is retrained to obtain a new h.

Finding an algorithm to compute $r(\boldsymbol{x}; h)$ exactly is still an open problem. For now, the best that we can do is to come up with heuristics that can approximate $r(\boldsymbol{x}; h)$. Another problem is that in practice, the unlabelled pool can be arbitrarily large. For example, there are 800 million unlabelled objects in the SDSS. Thus if we only have a limited computing power, in each round, we might only be able to assign scores to a subset $\mathcal{E} \subseteq \mathcal{U}$ of size E. A formal description of the active learning routine is given in Algorithm 1. As we shall see in Section 3.4, for some active learning heuristics, we need to substitute argmax with argmin in line 4 of Algorithm 1.

Algorithm 1 The general pool-based active learning algorithm

1: **procedure** ACTIVELEARNER($\mathcal{U}$, $\mathcal{L}_T$, h, r, n, E)
2: **while** $|\mathcal{L}_T| < n$ **do**
3: $\mathcal{E} \leftarrow$ random sample of size E from $\mathcal{U}$
4: $\boldsymbol{x}_* \leftarrow \operatorname{argmax}_{\boldsymbol{x} \in \mathcal{E}} r(\boldsymbol{x}; h)$
5: $y_* \leftarrow$ ask the expert to label $\boldsymbol{x}_*$
6: $\mathcal{L}_T \leftarrow \mathcal{L}_T \cup (\boldsymbol{x}_*, y_*)$
7: $\mathcal{U} \leftarrow \mathcal{U} \setminus \boldsymbol{x}_*$
8: $h(\boldsymbol{x}) \leftarrow$ retrain the classifier
9: **end while**
10: **end procedure**

3.4 Active Learning Heuristics

Many methods have been proposed to rank the informativeness of unlabelled objects. The four prominent families of heuristics are uncertainty sampling, version space re-

duction, loss function minimisation, and classifier certainty. All of these heuristics require the class probabilities estimated by the current hypothesis. We now discuss each of them in turn, starting with the least computationally expensive one.

3.4.1 Uncertainty Sampling

Lewis and Gale [1994] introduce the idea of uncertainty sampling, where we select the example whose class membership the classifier is least certain about. These tend to be points that are near the decision boundary of the classifier. One way to quantify the uncertainty is to calculate the entropy [Shannon 1948], which measures the amount of information needed to encode a distribution. Intuitively, the closer class probabilities of an object are to random guessing, the higher its entropy will be. This gives us the heuristic of picking the candidate with the highest entropy:

$$\begin{aligned} \boldsymbol{x}_* &= \operatorname*{argmax}_{x \in \mathcal{E}} r_S(\boldsymbol{x}; h) \\ &= \operatorname*{argmax}_{x \in \mathcal{E}} \left\{ -\sum_{y \in \mathcal{Y}} p(y|\boldsymbol{x}; h) \log \left[p(y|\boldsymbol{x}; h) \right] \right\} \end{aligned}$$

In fact, if we care about how close the class probabilities are to random guessing, there is an even simpler measure. Scheffer et al. [2001] define the margin as the difference between the two highest class probabilities. Since the sum of all probabilities must be 1, the smaller the margin is, the more uncertain we are about the object's class membership. Thus another heuristic is to pick the candidate with the smallest margin:

$$\begin{aligned} \boldsymbol{x}_* &= \operatorname*{argmin}_{x \in \mathcal{E}} r_M(\boldsymbol{x}; h) \\ &= \operatorname*{argmin}_{x \in \mathcal{E}} \left\{ \max_{y \in \mathcal{Y}} p(y|\boldsymbol{x}; h) - \max_{z \in \mathcal{Y} \setminus \{y\}} p(z|\boldsymbol{x}; h) \right\} \end{aligned}$$

3.4.2 Version Space Reduction

Let us define the version space as the set of all possible hypotheses that are consistent with the current training set. Instead of focussing on the uncertainty of individual predictions, we could instead try to constrain the size of the version space, thus allowing the search for the optimal hypothesis to be more precise. To quantify the size the version space, we can train a committee of classifiers, $\mathcal{B} = \{h_1, h_2, ..., h_B\}$, and measure the disagreement among the members about an object's class membership. Each committee member needs to have a hypothesis that is as different from the others as possible but that is still in the version space [Melville and Mooney 2004]. In order to have this diversity, we give each member only a subset of the training examples. Since there might not be enough training data (for example, in our experiments, we have 11 members but only a maximum of 300 labelled points), we need to use bootstrapping and select samples with replacement. Hence this method is often called Query

by Bagging (QBB).

One way to measure the level of disagreement is to calculate the margin using the class probabilities estimated by the committee [Melville and Mooney 2004]. This looks similar to one of the uncertainty sampling heuristic, except now we first average out the probabilities of the members before minimising the margin:

$$\begin{aligned} \boldsymbol{x}_* &= \operatorname*{argmin}_{x \in \mathcal{E}} r_{QM}(\boldsymbol{x}; \mathcal{B}) \\ &= \operatorname*{argmin}_{x \in \mathcal{E}} \left\{ \max_{y \in \mathcal{Y}} p(y|\boldsymbol{x}; \mathcal{B}) - \max_{z \in \mathcal{Y} \setminus \{y\}} p(z|\boldsymbol{x}; \mathcal{B}) \right\} \end{aligned}$$

where

$$p(y|\boldsymbol{x}; \mathcal{B}) = \frac{1}{B} \sum_{b=1}^{B} p(y|\boldsymbol{x}, h_b)$$

In addition to the margin, McCallum and Nigam [1998] offer an alternative disagreement measure which involves picking the candidate with the largest expected Kullback–Leibler (KL) divergence from the average:

$$\begin{aligned} \boldsymbol{x}_* &= \operatorname*{argmax}_{x \in \mathcal{E}} r_{QK}(\boldsymbol{x}; \mathcal{B}) \\ &= \operatorname*{argmax}_{x \in \mathcal{E}} \left\{ \frac{1}{B} \sum_{b=1}^{B} D_{\mathrm{KL}}(p_b \| p_{\mathcal{B}}) \right\} \end{aligned}$$

where

$$D_{\mathrm{KL}}(p_b \| p_{\mathcal{B}}) = \sum_{y \in \mathcal{Y}} p(y|\boldsymbol{x}; h_b) \ln \frac{p(y|\boldsymbol{x}; h_b)}{p(y|\boldsymbol{x}; \mathcal{B})}$$

The KL divergence measures the amount of information lost when $p_{\mathcal{B}}$ is used to approximate p_b. Intuitively, the larger the KL divergence is, the more disagreement there is between $p_{\mathcal{B}}$ and p_b. In the active learning context, $p_{\mathcal{B}}$ is the average prediction probability distribution of the committee, while p_b is the prediction of a particular committee member h_b.

3.4.3 Loss Function Minimisation

The third approach involves minimising a loss function directly, which in turn will minimise the future generalisation error. A commonly used loss function is the squared loss that has the following decomposition:

$$\mathbb{E}[\text{Squared Loss}] = \text{Bias}^2 + \text{Variance} + \text{Noise}$$

Since the noise is intrinsic to the data and represents the expected loss under the optimal hypothesis, there is nothing we can do about it. The squared bias reflects the error due to the model class itself. For example, there will be bias if we use a linear

hypothesis to learn a non-linear function. Thus the bias is fixed under the same classifier. However, under certain assumptions like the consistency of parameter estimates, the variance will vanish as the training set size approaches infinity. This gives us the heuristic of picking the candidate that would cause the greatest drop in the variance if we knew its label. Unfortunately, this is a chicken-and-egg problem since we need to know the labelling information before we can calculate the drop in the variance, which defeats the purpose of the approach. The next best thing we can do is to pick the candidate that will result in the lowest expected variance:

$$\begin{aligned} \boldsymbol{x}_* &= \underset{x \in \mathcal{E}}{\operatorname{argmin}}\, r_V(\boldsymbol{x}; h) \\ &= \underset{x \in \mathcal{E}}{\operatorname{argmin}}\, \mathbb{E}[V(\mathcal{L}_T \cup (\boldsymbol{x}, y))] \\ &= \underset{x \in \mathcal{E}}{\operatorname{argmin}} \left\{ \sum_{y \in \mathcal{Y}} p(y|\boldsymbol{x}; h) V(\mathcal{L}_T \cup (\boldsymbol{x}, y); h) \right\} \end{aligned}$$

where the expectation is over the class probability distribution under the current hypothesis and $V(\mathcal{L}_T \cup (\boldsymbol{x}, y); h)$ is the variance of the model after $(\boldsymbol{x}, y)$ has been added to the label set $\mathcal{L}_T$. Note that this is quite an expensive computation, since for us to assign a score to each candidate, we first need to give it each of the possible labels, add it to the training set to get an updated hypothesis, and calculate the new variance.

In addition, estimating $V(\mathcal{L}_T; h)$ requires a bit of work. In multinomial logistic regression, we can take the first two terms of the Taylor series expansion of the probability

$$p(y|\boldsymbol{x}, \hat{\boldsymbol{\theta}}, h) \approx p(y|\boldsymbol{x}, \boldsymbol{\theta}, h) + \nabla p(y|\boldsymbol{x}, \boldsymbol{\theta}, h)^T (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ are the expected and the current estimates of the model parameters, respectively, and $\nabla p(y|\boldsymbol{x}, \boldsymbol{\theta}, h)$ is called the gradient vector. Let $\mathcal{I}$ be the Fisher information matrix and

$$O = \sum_{\boldsymbol{x} \in \mathcal{U}} \sum_{y \in \mathcal{Y}} \nabla p(y|\boldsymbol{x}, \boldsymbol{\theta}, h)\ \nabla p(y|\boldsymbol{x}, \boldsymbol{\theta}, h)^T$$

Schein and Ungar [2007] show that

$$V(\mathcal{L}_T; h) = \operatorname{tr}(O\mathcal{I}^{-1})$$

where the trace function $\operatorname{tr}(X)$ is the sum of the elements along the main diagonal of the square matrix X. Note that the above expression is specific to multinomial logistic regression. If we use the one-vs-rest strategy with binary logistic regression or another entirely different classifier like SVMs in our experiments, the same approximation might not hold and we should not expect to get good results. We leave the variance estimation of other learning algorithms to future work.

3.4.4 Classifier Certainty

Finally, instead of minimising the variance of $\mathcal{U}$, MacKay [1991] proposes minimising the entropy of the classifier's predictions on $\mathcal{U}$:

$$H(\mathcal{L}_T; h) = -\sum_{\boldsymbol{x}\in\mathcal{U}} \sum_{y\in\mathcal{Y}} p(y|\boldsymbol{x}, h) \log \left[p(y|\boldsymbol{x}, h)\right]$$

Although this sounds similar to one of the uncertainty sampling heuristics, here instead of picking the most uncertain candidate, we pick the candidate that is expected to increase the classifier's prediction certainty by the the greatest amount:

$$\begin{aligned}
\boldsymbol{x}_* &= \operatorname*{argmin}_{x\in\mathcal{E}} r_H(\boldsymbol{x}; h) \\
&= \operatorname*{argmin}_{x\in\mathcal{E}} \mathbb{E}[H(\mathcal{L}_T \cup (\boldsymbol{x}, y))] \\
&= \operatorname*{argmin}_{x\in\mathcal{E}} \left\{ \sum_{y\in\mathcal{Y}} p(y|\boldsymbol{x}; h) H(\mathcal{L}_T \cup (\boldsymbol{x}, y); h) \right\}
\end{aligned}$$

Like the variance minimisation technique, we need to calculate the expectation over the possible classes. Thus to get the score of just one candidate, we need to retrain the classifier k times, where k is the number of labels. In practice, both the variance and the classifier certainty heuristics are too computationally expensive to run.

3.4.5 Summary of Heuristics

Table 3.1: Summary of active learning heuristics used in our experiments

Name	Notation	Objective
Entropy	$r_S(\boldsymbol{x}; h)$	$\operatorname*{argmax}_{x\in\mathcal{E}} \left\{ -\sum_{y\in\mathcal{Y}} p(y|\boldsymbol{x}; h) \log \left[p(y|\boldsymbol{x}; h)\right] \right\}$
Margin	$r_M(\boldsymbol{x}; h)$	$\operatorname*{argmin}_{x\in\mathcal{E}} \left\{ \max_{y\in\mathcal{Y}} p(y|\boldsymbol{x}; h) - \max_{z\in\mathcal{Y}\setminus\{y\}} p(z|\boldsymbol{x}; h) \right\}$
QBB Margin	$r_{QM}(\boldsymbol{x}; h)$	$\operatorname*{argmin}_{x\in\mathcal{E}} \left\{ \max_{y\in\mathcal{Y}} p(y|\boldsymbol{x}; \mathcal{B}) - \max_{z\in\mathcal{Y}\setminus\{y\}} p(z|\boldsymbol{x}; \mathcal{B}) \right\}$
QBB KL	$r_{QK}(\boldsymbol{x}; h)$	$\operatorname*{argmax}_{x\in\mathcal{E}} \left\{ \frac{1}{B} \sum_{b=1}^{B} D_{\mathrm{KL}}(p_b \| p_{\mathcal{B}}) \right\}$
Pool Variance	$r_V(\boldsymbol{x}; h)$	$\operatorname*{argmin}_{x\in\mathcal{E}} \left\{ \sum_{y\in\mathcal{Y}} p(y|\boldsymbol{x}; h) V(\mathcal{L}_T \cup (\boldsymbol{x}, y); h) \right\}$
Pool Entropy	$r_H(\boldsymbol{x}; h)$	$\operatorname*{argmin}_{x\in\mathcal{E}} \left\{ \sum_{y\in\mathcal{Y}} p(y|\boldsymbol{x}; h) H(\mathcal{L}_T \cup (\boldsymbol{x}, y); h) \right\}$

3.5 Multi-arm Bandit

Out of the six heuristics discussed, how do we know which one is the optimal, anyway? There have been some attempts in the literature to do a theoretical analysis of them. However proofs are scarce, and when there is one available, they normally only work under simplifying assumptions. For example, Freund et al. [1997] show that the query by committee algorithm (a slight variant of our QBB heuristics) guarantees an exponential decrease in the prediction error with the training size, but only under certain restrictions such as there is no noise. Thus whether any of these heuristics is guaranteed to beat random sampling is still an open question. We shall not worry too much about the theoretical analysis in this thesis. Instead we shall focus on an empirical analysis in the astronomical domain.

To help us automatically choose the optimal heuristic, we now turn our attention to the multi-armed bandit problem in probability theory. The colourful name originates from the situation where a gambler stands in front of a slot machine with n levers. When pulled, each lever gives out a random reward according to some unknown distribution. The goal of the game is to come up with a strategy that can maximise the gambler's lifetime rewards while minimising the number of pulls.

Our key novel contribution is the application of this theory to the problem of heuristic selection. Suppose we have a set of n heuristics $\mathcal{R} = \{r_1, ..., r_s\}$. Each heuristic has a different ability to identify the candidate whose labelling information is most valuable. An appropriate reward is then the incremental increase in the accuracy rate after the candidate is added to the training set. We assume that the heuristic rewards are independent of each other. This is reasonable since the theories with which we use to derive the heuristics are mostly unrelated.

Let ρ_i be the reward of heuristic $r_i \in \mathcal{R}$. Observe that even with the optimal heuristic, there are still two sources of error. First, there could be error during the labelling process that causes the accuracy rate to decrease. In addition, even without label noise, the classifier trained on finite data might not be the right one, so we still cannot score perfectly due to having a poor h. Conversely, a bad heuristic might be able to pick an informative candidate due to pure luck. Thus there is always a certain level of randomness in the reward received. These errors are probably normally distributed, so

$$(\rho_i \mid \nu_i) \sim \mathcal{N}(\nu_i, \tau_i^2)$$

and the probably density function (PDF) of the reward is

$$f(\rho_i \mid \nu_i) = \frac{1}{\tau_i \sqrt{2\pi}} \exp\left[-\frac{(\rho_i - \nu_i)^2}{2\tau_i^2}\right] \tag{3.1}$$

If we knew both the mean ν_i and the variance τ_i^2 for all heuristics, the problem would become trivially easy since we just need to always use the heuristic that has the highest mean reward. In practice, we do not know ν_i, so let us assume that it follows a normal

distribution

$$\nu_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

Thus the PDF of ν_i is

$$f(\nu_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{(\rho_i - \mu_i)^2}{2\sigma_i^2}\right] \tag{3.2}$$

To make the problem tractable, assume that τ_i^2 is a known constant. The goal now is to find a good algorithm that can estimate μ_i and σ_i^2.

One problem in multi-arm bandits is the trade-off between exploration and exploitation . Suppose we have managed to estimate μ_i and σ_i^2 for all i, then by always selecting the heuristic with the highest possible mean, or the greedy heuristic, we would be exploiting our current knowledge. If however we select one of the other non-greedy heuristics, we would then be exploring with the intention of improving our estimates of μ_i and σ_i^2. There are many instances in which we find our previously held beliefs to be completely wrong. Thus by always exploiting, we could miss out on the optimal heuristic. On the other hand, if we explore too much, it might take a long time to reach the desired accuracy and the strategy ends up being no different from random sampling.

3.6 Thompson Sampling

There are two main methods in the literature that address this exploration vs exploitation problem. The algorithm with a strong theoretical guarantee is Upper Confidence Bound [Auer et al. 2002]. We shall, however, focus on a simpler and much older algorithm called Thompson sampling. First introduced by Thompson [1933], this algorithm solves the trade-off from a Bayesian perspective. It has been shown to achieve results that are comparable and sometimes even better than Upper Confidence Bound [Chapelle and Li 2011].

Under the heuristic selection setting, Thompson sampling works as follows. We start with a prior knowledge of μ_i and σ_i^2 for all i. So long as we do not choose anything stupid, e.g. a zero variance, our choice of prior should not matter too much in the long run. Since initially we do not have any information about the performance of each heuristic, the appropriate prior value for μ_i is 0, i.e. there is no evidence (yet) that any of the heuristics offer an improvement to the accuracy.

In each round, we draw a random sample ν_i' from the distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ for each i and select heuristic r_* that has the highest sampled value of the mean reward:

$$r_* = \underset{i}{\operatorname{argmax}}\, \nu_i'$$

We then use this heuristic to assign scores to the candidates. The object that is deemed to be the most informative is then added to the training set $\mathcal{L}_T$ and the classifier is

retrained. Next we use the updated hypothesis to predict the labels of objects in the test set $\mathcal{L}_S$. Let δ be the reward observed, which is the incremental increase in the accuracy rate on $\mathcal{L}_S$. We now have a new piece of information that we can use to update our prior belief about the mean ν_* and the variance σ_*^2 of r_*. In the Bayesian setting, (3.2) is called the prior and (3.1) is the likelihood. From Bayes' theorem, the posterior distribution is proportional to the product of the prior and the likelihood:

$$
\begin{aligned}
f(\nu_* \mid \rho_* = \delta) &\propto f(\nu_*) f(\delta \mid \nu_*) \\
&\propto \exp\left[-\frac{1}{2\sigma_*^2}(\nu_* - \mu_*)^2\right] \exp\left[-\frac{1}{2\tau_*^2}(\delta - \nu_*)^2\right] \\
&= \exp\left[-\frac{1}{2\sigma_*^2}(\nu_* - \mu_*)^2 - \frac{1}{2\tau_*^2}(\delta - \nu_*)^2\right] \\
&= \exp\left[-\frac{\nu_*^2}{2}\left(\frac{1}{\sigma_*^2} + \frac{1}{\tau_*^2}\right) + \nu_*\left(\frac{\mu_*}{\sigma_*^2} + \frac{\delta}{\tau_*^2}\right) - \left(\frac{\mu_*^2}{2\sigma_*^2} + \frac{\delta^2}{2\tau_*^2}\right)\right] \\
&\propto \exp\left[-\frac{\nu_*^2}{2}\left(\frac{1}{\sigma_*^2} + \frac{1}{\tau_*^2}\right) + \nu_*\left(\frac{\mu_*}{\sigma_*^2} + \frac{\delta}{\tau_*^2}\right)\right] \\
&= \exp\left[-\frac{\nu_*^2}{2}\left(\frac{\sigma_*^2 + \tau_*^2}{\sigma_*^2\tau_*^2}\right) + \nu_*\left(\frac{\mu_*\tau_*^2 + \delta\sigma_*^2}{\sigma_*^2\tau_*^2}\right)\right] \\
&= \exp\left[-\frac{1}{2{\sigma'_*}^2}(\nu_*^2 - 2\nu_*\mu'_*)\right] \\
&\propto \exp\left[-\frac{1}{2{\sigma'_*}^2}(\nu_*^2 - 2\nu_*\mu'_* + {\mu'_*}^2)\right] \\
&= \exp\left[-\frac{1}{2{\sigma'_*}^2}(\nu_* - \mu'_*)^2\right]
\end{aligned}
$$

where we have defined

$$
\mu'_* = \frac{\mu_*\tau_*^2 + \delta\sigma_*^2}{\sigma_*^2 + \tau_*^2} \qquad {\sigma'_*}^2 = \frac{\sigma_*^2\tau_*^2}{\sigma_*^2 + \tau_*^2}
$$

Thus the posterior distribution of the mean reward of r_* remains normal:

$$
(\nu_* \mid \rho_* = \delta) \sim \mathcal{N}(\mu'_*, {\sigma'_*}^2)
$$

For compactness, let:

$$
\begin{aligned}
\boldsymbol{\nu} &= (\nu_1, \nu_2, ..., \nu_s) & \boldsymbol{\mu} &= (\mu_1, \mu_2, ..., \mu_s) \\
\boldsymbol{\sigma^2} &= (\sigma_1^2, \sigma_2^2, ..., \sigma_s^2) & \boldsymbol{\tau^2} &= (\tau_1^2, \tau_2^2, ..., \tau_s^2)
\end{aligned}
$$

Algorithm 2 on page 24 shows the formal specification of the Thompson sampling procedure with normally distributed rewards. If we then combine this with the general pool-based active learning algorithm, we end up with a multi-arm bandit version of active learning where each arm is a heuristic (Algorithm 3). Note again that in line 8 of Algorithm 3, some heuristics have argmin in place of argmax.

Algorithm 2 Thompson sapmling with normally distributed rewards

procedure THOMPSONSAMPLING($\mathcal{R}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\tau}^2$, n)
  for each $t \in \{1, 2, ..., n\}$ **do**
    for each $i \in \{1, 2, ..., |\mathcal{R}|\}$ **do**
      $\nu_i' \leftarrow$ draw a sample from $\mathcal{N}(\mu_i, \sigma_i^2)$
    end for
    $r_* \leftarrow \underset{i}{\operatorname{argmax}}\, \nu_i'$
    Observe reward δ
    $\mu_* \leftarrow \dfrac{\mu_* \tau_*^2 + \delta \sigma_*^2}{\sigma_*^2 + \tau_*^2}$
    $\sigma_*^2 \leftarrow \dfrac{\sigma_*^2 \tau_*^2}{\sigma_*^2 + \tau_*^2}$
  end for
end procedure

Algorithm 3 The multi-arm bandit active learning algorithm

procedure ACTIVEBANDIT($\mathcal{U}$, $\mathcal{L}_T$, h, n, E, $\mathcal{R}$, $\boldsymbol{\mu}$, $\boldsymbol{\sigma}^2$, $\boldsymbol{\tau}^2$)
  while $|\mathcal{L}_T| < n$ **do**
    for each $i \in \{1, 2, ..., |\mathcal{R}|\}$ **do**
      $\nu_i' \leftarrow$ draw a sample from $\mathcal{N}(\mu_i, \sigma_i^2)$
    end for
    $r_* \leftarrow \underset{i}{\operatorname{argmax}}\, \nu_i'$
    $\mathcal{E} \leftarrow$ random sample of size E from $\mathcal{U}$
    $\boldsymbol{x}_* \leftarrow \underset{\boldsymbol{x} \in \mathcal{E}}{\operatorname{argmax}}\, r_*(\boldsymbol{x})$
    $y_* \leftarrow$ ask the expert to label $\boldsymbol{x}_*$
    $\mathcal{L}_T \leftarrow \mathcal{L}_T \cup (\boldsymbol{x}_*, y_*)$
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \boldsymbol{x}_*$
    $h(\boldsymbol{x}) \leftarrow$ retrain the classifier
    $\delta \leftarrow$ incremental increase in the accuracy
    $\mu_* \leftarrow \dfrac{\mu_* \tau_*^2 + \delta \sigma_*^2}{\sigma_*^2 + \tau_*^2}$
    $\sigma_*^2 \leftarrow \dfrac{\sigma_*^2 \tau_*^2}{\sigma_*^2 + \tau_*^2}$
  end while
end procedure

How does Thompson sampling balance between exploration and exploitation? If we only wanted to exploit our current knowledge, then we would naturally select the heuristic that maximises the immediate reward:

$$\begin{aligned} r_* &= \operatorname*{argmax}_i \mathbb{E}[\rho_i] \\ &= \operatorname*{argmax}_i \mathbb{E}\{\mathbb{E}[\rho_i \mid \boldsymbol{\nu}]\} \\ &= \operatorname*{argmax}_i \mathbb{E}\{\mathbb{E}[\rho_i \mid \nu_i]\} \\ &= \operatorname*{argmax}_i \int_{-\infty}^{+\infty} \mathbb{E}[\rho_i \mid \nu_i] f(\nu_i)\, d\nu_i \\ &= \operatorname*{argmax}_i \int_{-\infty}^{+\infty} \nu_i f(\nu_i)\, d\nu_i \\ &= \operatorname*{argmax}_i \mathbb{E}[\nu_i] \\ &= \operatorname*{argmax}_i \mu_i \end{aligned}$$

where we have used the law of total expectation and the fact that both $(\rho_i \mid \nu_i)$ and ν_i are normally distributed. Unsurprisingly, the heuristic that maximises the immediate reward is simply the one with the highest μ_i. However, there is a chance that we could be wrong about our estimate of $\boldsymbol{\mu}$. Thus a better option is to only select a heuristic at the same frequency as the probability that it is optimal. For a given heuristic r_i, this probability is

$$\begin{aligned} \mathbb{P}(\mathbb{E}[\rho_i] = \max_j \mathbb{E}[\rho_j]) &= \mathbb{E}[\mathbb{P}(\mathbb{E}[\rho_i] = \max_j \mathbb{E}[\rho_j] | \boldsymbol{\nu})] \\ &= \int_{\boldsymbol{\nu}} \mathbb{P}(\mathbb{E}[\rho_i] = \max_j \mathbb{E}[\rho_j] \mid \boldsymbol{\nu}) f(\boldsymbol{\nu})\, d\boldsymbol{\nu} \\ &= \int_{\boldsymbol{\nu}} \mathbb{P}(\mathbb{E}[\rho_i \mid \boldsymbol{\nu}] = \max_j \mathbb{E}[\rho_j \mid \boldsymbol{\nu}]) f(\boldsymbol{\nu})\, d\boldsymbol{\nu} \\ &= \int_{\boldsymbol{\nu}} \mathbb{P}(\nu_i = \max_j \nu_j) f(\boldsymbol{\nu})\, d\boldsymbol{\nu} \\ &= \int_{\boldsymbol{\nu}} \mathbb{I}(\nu_i = \max_j \nu_j) f(\boldsymbol{\nu})\, d\boldsymbol{\nu} \end{aligned}$$

where $\mathbb{I}$ is the indicator function. In fact we do not have to evaluate this integral directly. If in each round, we draw a random sample of $\boldsymbol{\nu}$ and act optimally according to the sample values, then over the long run, the frequency that we select each heuristic will approach its probability of being optimal. This is exactly what Thompson sampling does.

Finally, as we shall see in Chapter 5, the reward function dynamically evolves as the training size increases. Intuitively, the accuracy rate can never go beyond 100%, so we would expect the incremental change in the accuracy to become smaller over time. Attempts to address this problem have been made in the literature. For example Gupta, Granmo, and Agrawala [2011] introduce the Dynamic Thompson Sampling

method that can adapt to the evolving parameters faster than Algorithm 2 and 3. However, we shall leave the investigation of this method to future work.

3.7 Posterior Balanced Accuracy

Certain astronomical objects are either rarer or more difficult to detect than others. In the SDSS labelled set, there are 4.5 times as many galaxies as quasars. The problem of class imbalance is even more severe in the VST-ATLAS set, with 43 times more stars than white dwarfs. An easy fix is to undersample the dominant class when creating training and test sets. This, of course, means that the size of these sets are limited by the size of the minority class.

When we do not want to alter the underlying class distributions or when larger training and test sets are desired, we need a performance measure that can correct for the class imbalance. Brodersen et al. [2010] show that the posterior balanced accuracy distribution can overcome the bias in the binary case. We now extend this idea to the multi-class setting.

Suppose we have k classes. For each class i between 1 and k, there are N_i objects in the universe. Given a hypothesis, we can predict the label of every object and compare our prediction to the true label. Let G_i be the number of objects in class i that are correctly predicted. Then we define the recall A_i of class i as

$$A_i = \frac{G_i}{N_i}$$

The problem is that it is not feasible to get the actual values of G_i and N_i since that would require us to obtain the true label of every object. Thus we need a method to estimate these quantities when we only have a sample. Initially we have no information about G_i and N_i, so we can assume that each A_i follows a uniform prior from 0 to 1. This is the same as a Beta distribution with shape parameters $\alpha = \beta = 1$:

$$A_i \sim \text{Beta}(1, 1)$$

The PDF of A_i is then

$$\begin{aligned} f_{A_i}(a) &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, a^{\alpha-1}(1-a)^{\beta-1} \\ &\propto a^{1-1}(1-a)^{1-1} \end{aligned} \tag{3.3}$$

where $\Gamma(\alpha)$ is the gamma function.

After we have trained the classifier, suppose we have a test set containing n_i objects in class i. Running the classifier on this test set is the same as conducting k binomial experiments, where, in the ith experiment, the sample size is n_i and the probability of success is simply A_i. Let g_i be the number of correctly labelled objects belonging to class i in the test set. Then, conditional on the accuracy rate, g_i follows a binomial

distribution:

$$(g_i \mid A_i) \sim \text{Bin}(n_i, A_i)$$

The probability mass function of $(g_i \mid A_i = a)$ is thus

$$\begin{aligned} p_{g_i|A_i}(g_i) &= \binom{n_i}{g_i} a^{g_i}(1-a)^{n_i-g_i} \\ &\propto a^{g_i}(1-a)^{n_i-g_i} \end{aligned} \tag{3.4}$$

In the Bayesian setting, (3.3) is the prior and (3.4) is the likelihood. To get the posterior PDF, we simply multiply the prior with the likelihood:

$$\begin{aligned} f_{A_i|\boldsymbol{g}}(a) &\propto f_{A_i}(a) \times f_{g_i|A_i}(g_i) \\ &\propto a^{1-1}(1-a)^{1-1} \times a^{g_i}(1-a)^{n_i-g_i} \\ &= a^{1+g_i-1}(1-a)^{1+n_i-g_i-1} \end{aligned}$$

Thus, with respect to the binomial likelihood function, the Beta distribution is conjugate to itself. The posterior recall rate A_i also follows a Beta distribution, now with parameters

$$(A_i \mid g_i) \sim \text{Beta}(1+g_i, 1+n_i-g_i)$$

Our goal is to have a balanced accuracy rate, A, that puts an equal weight in each class. One way to achieve this is to take the average of the individual recalls:

$$\begin{aligned} A &= \frac{1}{k}\sum_{i=1}^{k} A_i \\ &= \frac{1}{k} A_T \end{aligned}$$

Here we have defined A_T to be the sum of the individual recalls. We call $(A \mid \boldsymbol{g})$ the posterior balanced accuracy (PBA), where $\boldsymbol{g} = (g_1, ..., g_k)$. Most of the time, we simply want to calculate its expected value:

$$\begin{aligned} \mathbb{E}[A \mid \boldsymbol{g}] &= \frac{1}{k}\,\mathbb{E}[A_T \mid \boldsymbol{g}] \\ &= \frac{1}{k}\int a \cdot f_{A_T|\boldsymbol{g}}(a)\, da \end{aligned}$$

Let us call this the mean posterior balanced accuracy rate (MPBA). Note that there is no closed form solution for the PDF $f_{A_T|\boldsymbol{g}}(a)$. However assuming that A_T is a sum of k independent Beta random variables, $f_{A_T|\boldsymbol{g}}(a)$ can be approximated by numerically convolving k Beta distributions. The independence assumption is reasonable here, since there should be little to no correlation between the individual class accuracy rates. Knowing that a classifier is really good at recognising stars does not tell us much about how well that classifier can recognise galaxies.

Having the knowledge of $f_{A|\boldsymbol{g}}(a)$ will allow us to make violin plots, construct confidence intervals and do hypothesis tests. To get an expression for this, let us first rewrite the cumulative distribution function (CDF) as

$$\begin{aligned} F_{A|\boldsymbol{g}}(a) &= \mathbb{P}(A \leq a \mid \boldsymbol{g}) \\ &= \mathbb{P}\Big(\frac{1}{k} A_T \leq a \,\Big|\, \boldsymbol{g}\Big) \\ &= \mathbb{P}(A_T \leq ka \mid \boldsymbol{g}) \\ &= F_{A_T|\boldsymbol{g}}(ka) \end{aligned} \tag{3.5}$$

Differentiating (3.5) with respect to a, we obtain the PDF of $(A \mid \boldsymbol{g})$:

$$\begin{aligned} f_{A|\boldsymbol{g}}(a) &= \frac{\partial}{\partial a} F_{A|\boldsymbol{g}}(ka) \\ &= \frac{\partial}{\partial a}(ka) \cdot \frac{\partial}{\partial ka} F_{A_T|\boldsymbol{g}}(ka) \\ &= k \cdot f_{A_T|\boldsymbol{g}}(ka) \end{aligned}$$

We shall use the posterior balanced accuracy rate throughout the experiments to report the overall performance of a classifier on a test set.

Chapter 4

Experiment 1: Learning with Random Sampling

Before we can get to the main investigation of Thompson sampling in Chapter 5, we need to first prepare the datasets and study them in the standard setting of supervised machine learning. This, in turn, will help us design the protocol of the active learning experiment. In particular, we start by doing feature selection (Section 4.1) and compare three sets of extinction vectors (Section 4.3.1) with the SDSS dataset. We then optimise the hyperparameters of our classifiers (Section 4.3.2) and obtain learning curves of both the SDSS and VST ATLAS datasets (Section 4.3.3). Finally, the best-performing classifier is used to predict the unknown class proportion of the 800 unlabelled objects in the SDSS (Section 4.3.4).

4.1 Preparation of Data

The VST ATLAS dataset has been specifically prepared for us by Christian Wolf from the ANU Research School of Astronomy & Astrophysics, so there is no need to do any cleaning or feature selection. The SDSS dataset, on the other hand, was extracted from the Sloan SkyServer. Appendix A contains instructions on how to replicate the data, including relevant SQL queries. In particular, we extract two sets of data. For the labelled objects, we only include those that have a clean spectrum and hence a reliable spectroscopic label. We do not impose any limit on how faint an object can be, but we do exclude those with a photometric error greater than 3 mag (i.e. 3 orders of magnitudes in brightness). This is deliberate since it allows us to investigate how well our classifiers can deal with noisy data. Given the constraints, the resulting labelled set contains 2.8 million objects. Finally, the second set of SDSS data contains photometric measurements of all 800 million objects in the database. No constraints were applied to this set.

Originally, each object has 5 PSF and 5 Petrosian magnitudes. As we discussed in Section 2.3.2 however, it is better to take the differences and obtain the colours. By convention, we calculate the $u - g$, $u - r$, $r - i$, and $i - z$ colours. In addition, we keep one apparent magnitude in the set of features. The magnitude in the r-band is chosen since it lets the the greatest fraction of light through the filter. This, hopefully, would

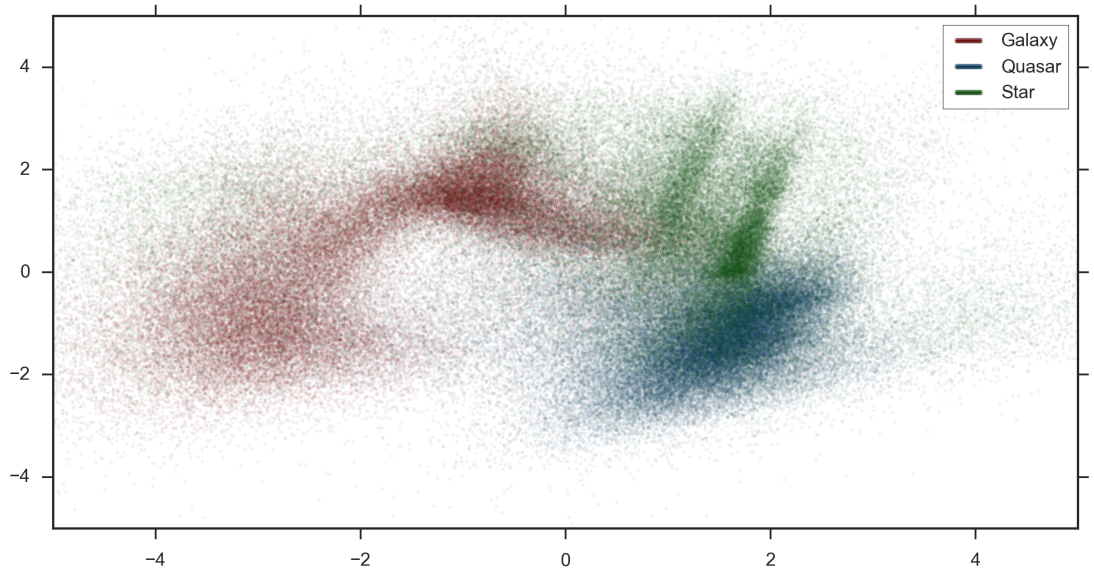


Figure 4.1: We use principal component analysis (PCA) to reduce the 11 features of the SDSS dataset down to two dimensions. This allows us to do a quick visual scan and see how separable the three classes are.

give us a high signal-to-noise ratio. Although it has been not been rigorously verified, this combination of four colours and one magnitude has been found to produce good results by astronomers in the past [Wolf, pers. comm.]. In the end, we end up with 11 features:

- Four PSF and four Petrosian colours ($u-g$, $u-r$, $r-i$, $i-z$).
- The r-band PSF and the r-band Petrosian magnitudes.
- The r-band Petrosian radius.

For a quick visualisation of these features, we make a scatter plot in Figure 4.1 of the first two principal components. Although we can clearly see clusters of stars, quasars, and galaxies, there is also enough overlap between classes, especially between stars and quasars, to make the problem interesting.

Finally, we scale all the features in both datasets to have a zero mean and a unit variance. Although this has little effect in random forests, feature scaling is important for logistic regression and SVMs.

4.2 Experimental Protocol

This experiment consists of three parts. We first compare three sets of reddening correction. We then examine the performance of various classifiers with random sampling. Finally we pick the best classifier and use it to predict the unknown proportions in the unlabelled set.

4.2.1 Reddening Correction

To compare the three extinction vectors (SDF98, SF11, and W14), we first split the labelled pool into a balanced training set of size 600,000 and a balanced test set of size 300,000. We then apply each correction to the measurements and train a random forest of 300 decision trees. A random forest is chosen due to its speed of execution. Finally the balanced posterior accuracy on the test set allows us to decide on the best performing reddening correction set. This set will be applied in all subsequent experiments.

4.2.2 Comparing Classifiers

Next we would like to compare the performance of four classifiers: random forests, logistic regression, linear SVMs, and SVMs with an RBF kernel. In the random forest, we again build 300 trees and the Gini impurity is used to measure the quality of a split. With the other classifiers, there are a few hyperparameters that require tuning before training, including the degree of the polynomial transformation of the original features. To find the optimal values, we conduct a search in the parameter space with logarithmic steps. For each combination, we do a five-fold cross validation with a training and test size of 300 each. We then plot the results on a heat map.

Once the hyperparameters are optimised, we compare the performance of the classifiers by looking at their learning curves. The VST ATLAS data is fairly small, so we can use all of the examples. Since using all of the SDSS labelled examples would take too long on some classifiers like logistic regression, we stop at 300,000 examples. To smooth out the curve, we do a stratified shuffle split with 5 iterations, and then take an average of the results.

Since we would like to use as much data as possible, no attempt is made to balance the classes. Instead we give to the classifiers a weight vector that is proportional to the inverse of the class frequencies. This means that rare objects like white dwarfs are given more weight during training. In addition, we use the posterior balance accuracy rate to remove any bias toward the dominant class.

Finally since we have 800 million unlabelled objects in the SDSS, we pick the best performing classifier and predict the class proportion on the unlabelled data. Information from the confusion matrix is used to correct for the potential misclassification.

4.3 Results and Discussion

4.3.1 Comparison of Reddening Correction Sets

Figure 4.2 on page 32 shows the violin plot of the posterior balanced accuracy on the test set when we apply each of the three extinction vectors to the measurements. Appendix C contains more detailed results. Overall, the results are quite uninteresting since no statistical differences can be found between the three sets. In fact, even if we do not apply any correction, the accuracy rate still remains unchanged.

This could be because in the SDSS did not survey many objects in the Milky Way band, the region in which there is the most dust extinction. Indeed, if we look back at

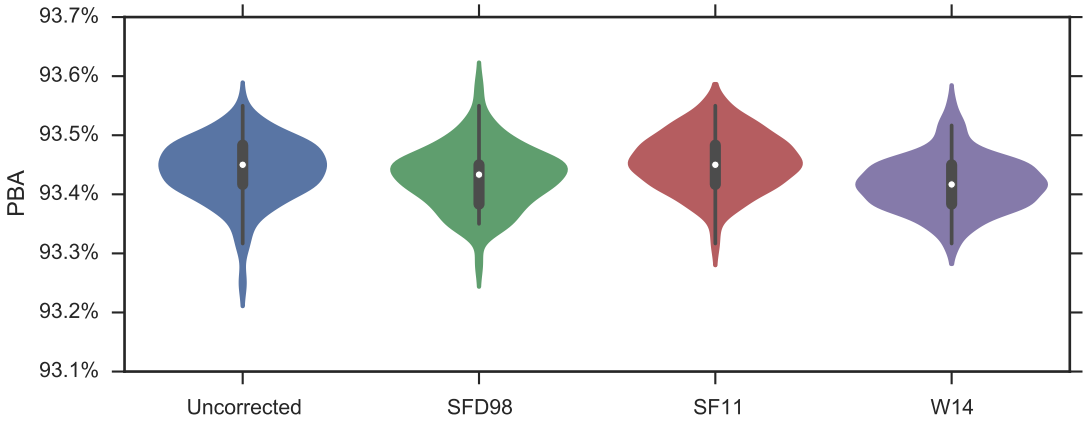


Figure 4.2: Violin plots showing the posterior distribution of the balanced accuracy rates: Inside each violin is a boxplot, where the white dot represents the mean. Given the overlapping, there is no statistical difference between the four extinction vectors.

the reddening map in Figure 2.5, although the extinction amount does vary through the field, such variation is probably fairly small and random forests, after all, are quite robust to noise. In any case, for good measure, we shall correct all photometric measurements for dust extinction using the latest W14 set. Other projects like SkyMapper will survey regions closer to the Milky Way band and thus future work using its data might be able to provide more conclusive evidence of which of the three extinction vectors provides the biggest improvement to the accuracy rate.

4.3.2 Hyperparameter Optimisations

The next three pages contain heat maps of the cross-validation accuracy rate for many different combinations of hyperparameters. For readability, we provide detailed reasoning under each figure of how we choose the optimal values of the parameters. Below is a summary of our decisions:

- **Logistic Regression**: We first do a degree 2 polynomial transformation of both the SDSS and the VST ATLAS features. For the classifier, we use the one-vs-rest strategy and an L1-norm for the penalisation, thus giving us sparse solutions. The inverse regularisation term C is 1 in SDSS and 100 in VST ATLAS.

- **Linear SVM**: For the SDSS data, we do a degree 2 polynomial transformation and our classifier uses the one-vs-rest strategy with the squared hinge loss function and, an L1-norm, and $C = 0.1$. For the VST ATLAS data, we simply use the Crammer-Singer approach with $C = 1,000$.

- **SVM with RBF Kernel**: The optimal hyperparameter values for the SDSS data are $\gamma = 0.01$ and $C = 1,000$. For the VST ATLAS data, they are $\gamma = 0.001$ and $C = 1,000,000$.

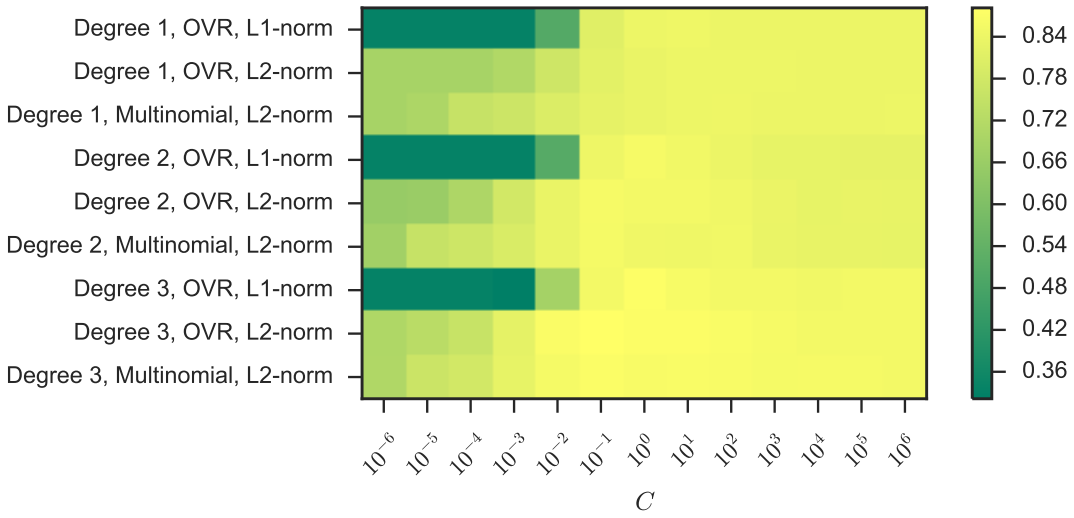


Figure 4.3: Heat map of linear logistic regression's cross-validation accuracy in the SDSS dataset: The best-performing combination involves doing a degree 3 polynomial transformation of the features. In practice, this would make it too slow to run subsequent experiments. Thus we sacrifice a bit of accuracy and pick the optimal parameters that involve only a degree 2 polynomial transformation. In particular, we set the inverse regularisation term C to be 1 and use the one-vs-rest strategy with L1-norm.

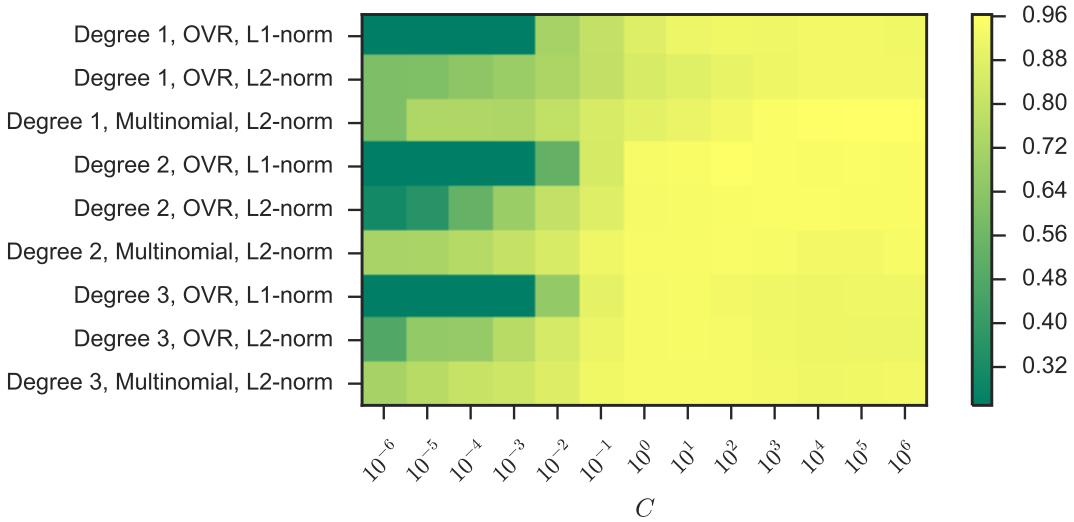


Figure 4.4: Heat map of logistic regression's cross-validation accuracy in the VST ATLAS dataset: The optimal combination involves using multinomial logistic regression. This might seem like a good choice since we get true probability estimates. However it turns out that the scikit-learn implementation of multinomial regression gives unstable probabilities. Thus we resort to the next best combination, where we use one-vs-rest, a degree 2 polynomial transformation of the features, and $C = 100$.

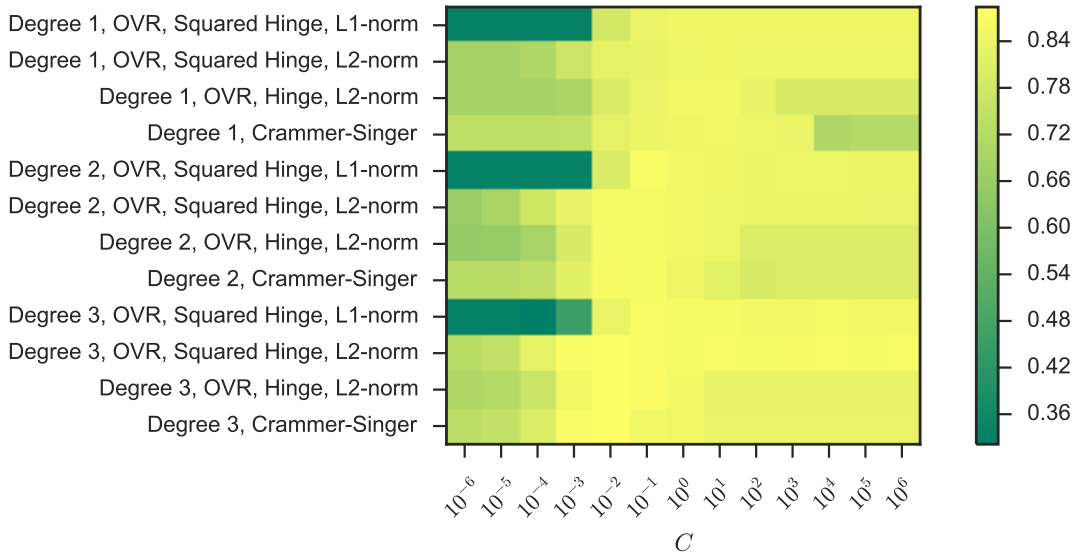


Figure 4.5: Heat map of linear SVM's cross-validation accuracy in the SDSS dataset: Like Figure 4.3, the optimal combination involves a degree 3 polynomial transformation of the features. Due to constraints on processing power, we shall instead pick the next best alternative, which involves a degree 2 transformation, the one-vs-rest strategy, the squared hinge loss function, and the L1-norm for the penalisation.

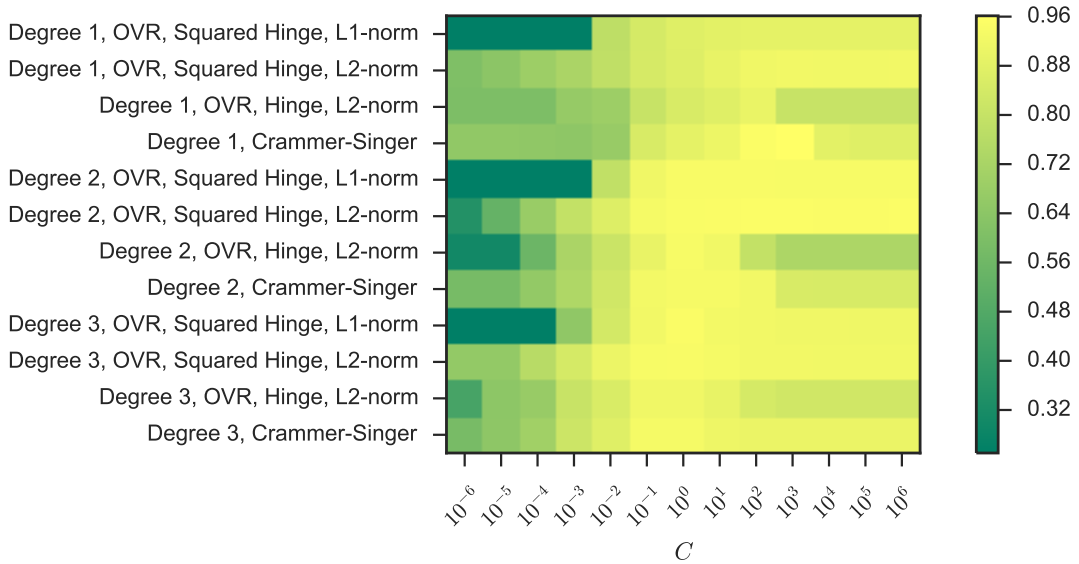


Figure 4.6: Heat map of linear SVM's cross-validation accuracy in the VST ATLAS dataset: Interestingly, the optimal combination does not require any polynomial transformation of the features, and instead of the usual one-vs-rest strategy, we use the Crammer-Singer method with $C = 1000$ gives the best result. In theory, the Crammer-Singer approach can give us true probability estimates, however this has not been implemented in scikit-learn.

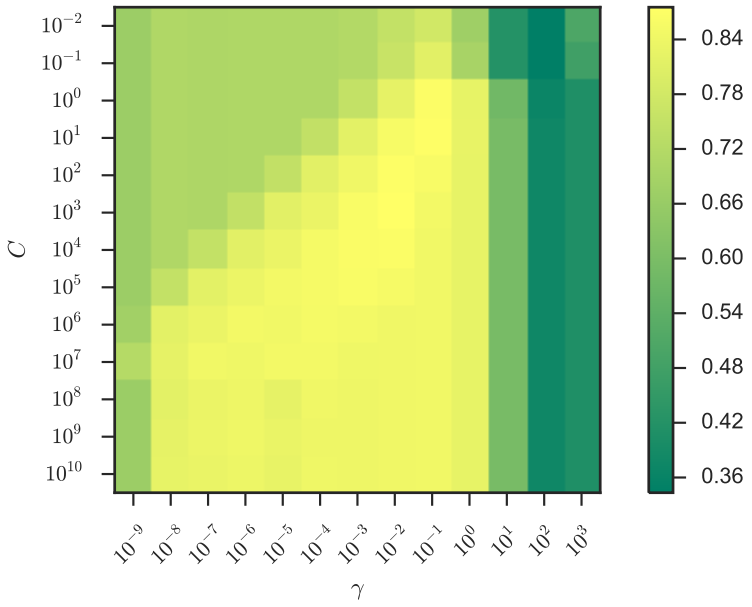


Figure 4.7: Heat map of RBF SVM's cross-validation accuracy in the SDSS dataset: Here the optimal values for the hyperparameters are $\gamma = 0.01$ and $c = 1,000$, giving an accuracy of around 88%.

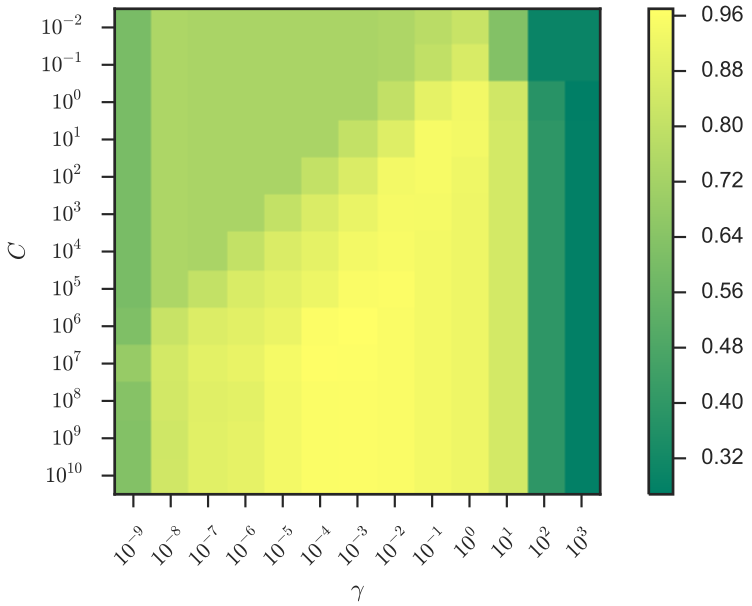
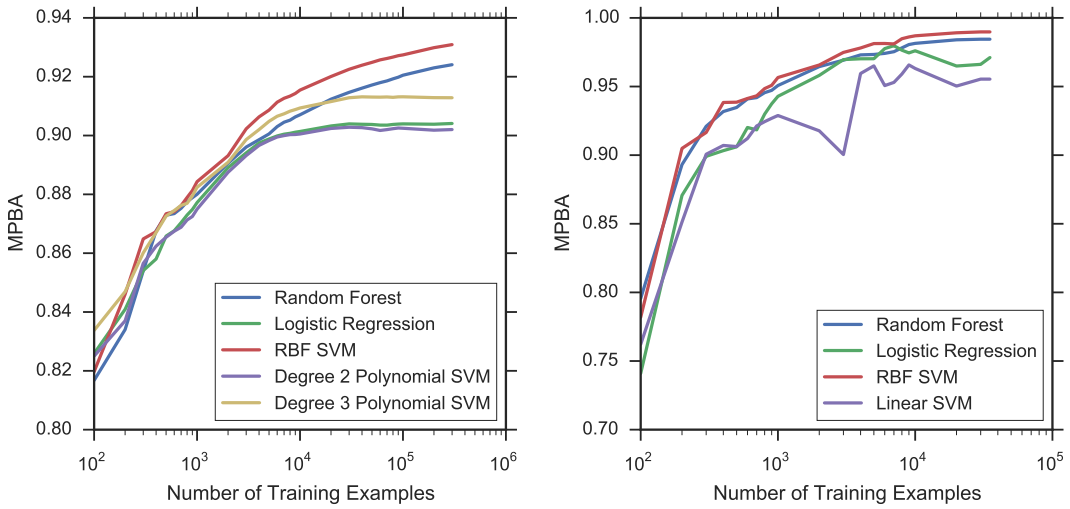


Figure 4.8: Heat map of RBF SVM's cross-validation accuracy in the VST ATLAS dataset: The optimal values are $\gamma = 0.001$ and $C = 1,000,000$, giving us an accuracy of 97%. Observe that with a large value of C, we would need more support vectors during training and hence the the model will be somewhat slower at prediction.



(a) With SDSS data

(b) With VST ATLAS data

Figure 4.9: These are the average learning curves (of 5 trials) with random sampling. Note that we use a log scale for the x-axis since generally, it gets exponentially more difficult to improve the accuracy rate as the accuracy approaches 1.

4.3.3 Learning Curves with Random Sampling

Now that we have tuned the hyperparameters, we are ready to compare the classifiers. Figure 4.9 shows the average learning curves of the two datasets. Overall the two best performers are random forests and SVMs with an RBF kernel. The VST ATLAS data appears to be very clean, with highest achievable balanced accuracy rate of 99%. Logistic regression with a degree 2 polynomial transformation is the slowest algorithm, surprising even slower than RBF SVMs (in theory, it should be the other way around). This could simply be due to the very efficient kernel approximation implemented in scikit-learn.

4.3.4 Class Proportion Estimation

Let us now make some predictions on the unlabelled SDSS data. Although the RBF SVM is the best-performing classifier, we nonetheless choose the random forest due to its fast training time. We retrain the forest, now with a training set of size 837,000 and a test set of size 300,000. Figure 4.10 shows the confusion matrix on the test set. Observe that it is easiest to classify galaxies and the main difficulty is distinguishing between stars and quasars.

There are exactly 794,014,031 objects in the entire database. Out of these, the random forest predicts that

- 357,910,241 (45.1%) are galaxies
- 266,083,661 (33.5%) are stars

		Predicted: Galaxy	Predicted: Star	Predicted: Quasar
Actual	**Galaxy**	97,608 95.7%	500 0.5%	1,892 1.8%
Actual	**Star**	1,633 1.6%	89,489 95.4%	8,878 8.5%
Actual	**Quasar**	2,801 2.7%	3,823 4.1%	93,376 89.7%

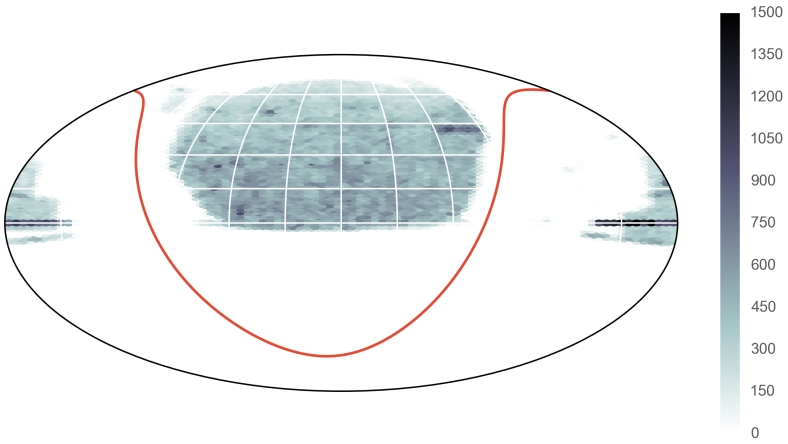
Figure 4.10: The confusion and the normalised confusion matrix of the random forest on the SDSS test set. For example, out of all the objects predicted as quasars, 8.5% of them are actually stars. Out of all the objects predicted as stars, 4.1% of them are actually quasars.

- 170,020,129 (21.4%) are quasars

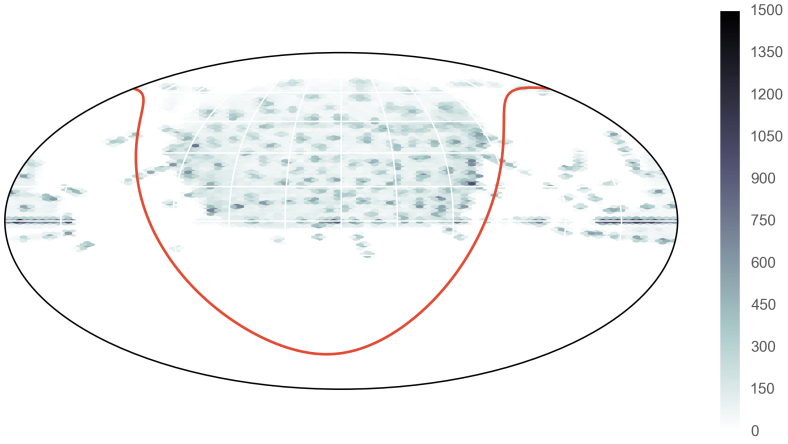
Using information from the normalised confusion matrix, we might be able to correct for the potential misclassification. For example, only 95.7% of objects predicted as galaxies are actually galaxies, while 0.5% of objects predicted as stars are galaxies and 1.8% of objects predicted as quasars are galaxies. Let Q_{ij} the entry in the ith row and jth column of the normalised confusion matrix. Let N_G, N_S, and N_Q be the predicted number of galaxies, stars, and quasars, respectively; and let N'_G, N'_S, and N'_Q be the actual numbers. Then the estimated actual number of galaxies, stars, and quasars in the unlabelled pool are

$$\begin{aligned} N'_G &= N_G Q_{11} + N_S Q_{12} + N_Q Q_{13} = 376,281,067 \\ N'_S &= N_G Q_{21} + N_S Q_{22} + N_Q Q_{23} = 145,832,927 \\ N'_Q &= N_G Q_{31} + N_S Q_{32} + N_Q Q_{33} = 271,900,036 \end{aligned}$$

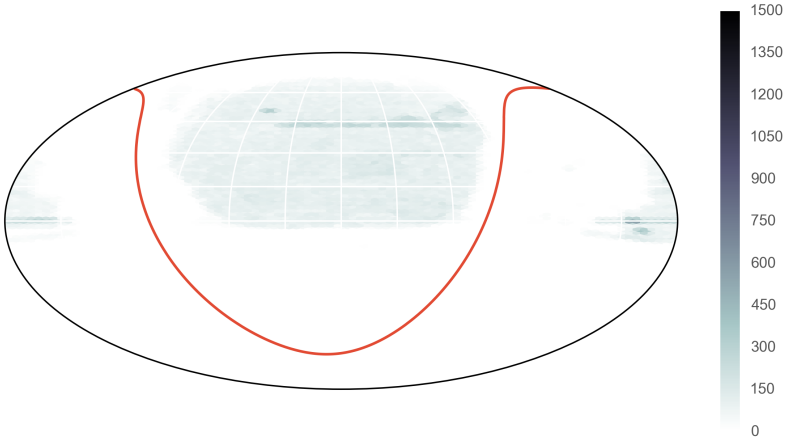
Thus after the correction, around 43.7% of objects are galaxies, 34.5% are stars, and 21.8% are quasars. Note however that even the full SDSS dataset is not a random sample of the sky. More sophisticated methods are needed if we want to recover the true proportions of the random sky.



(a) Distribution of galaxies.

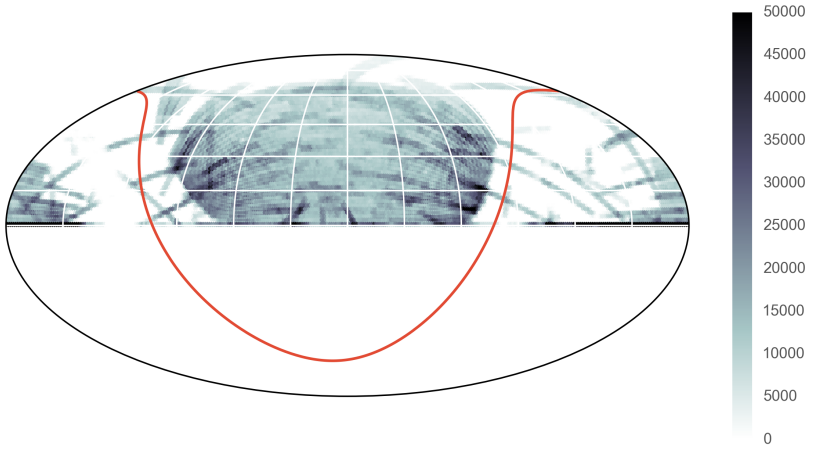


(b) Distribution of stars.

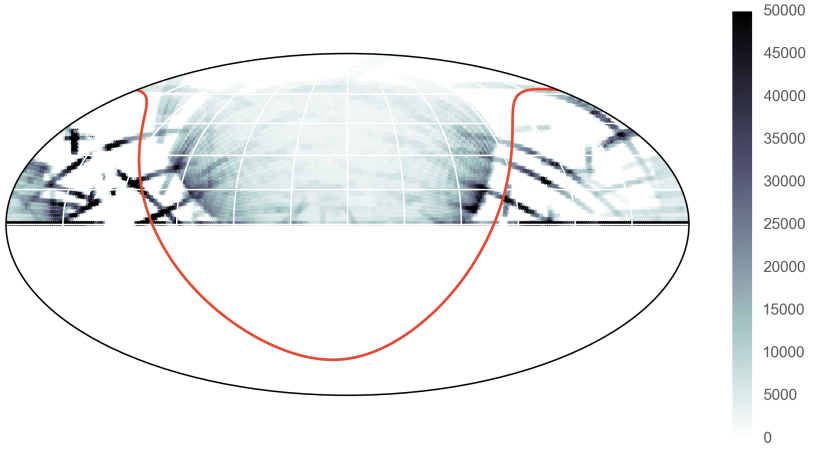


(c) Distribution of quasars.

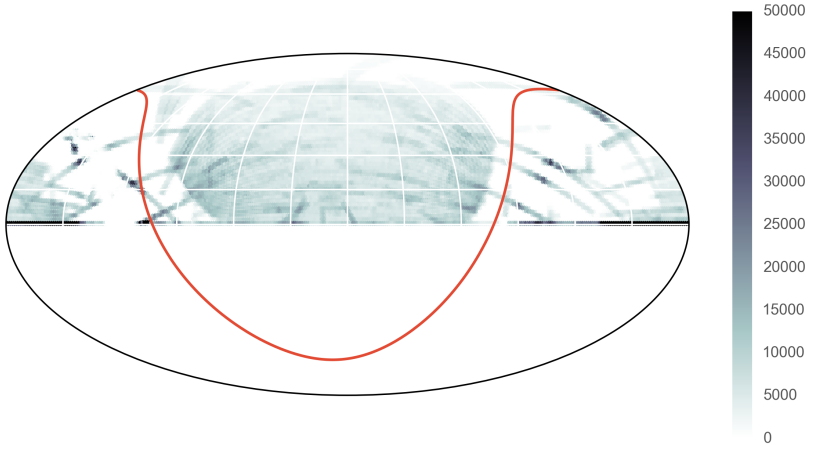
Figure 4.11: The distribution map of the 2.8 million labelled objects in the SDSS: Observe that the galaxies are mostly uniformly distributed in the survey, while the stars are not. We also do not have a lot of examples of quasars.



(a) Distribution of galaxies.

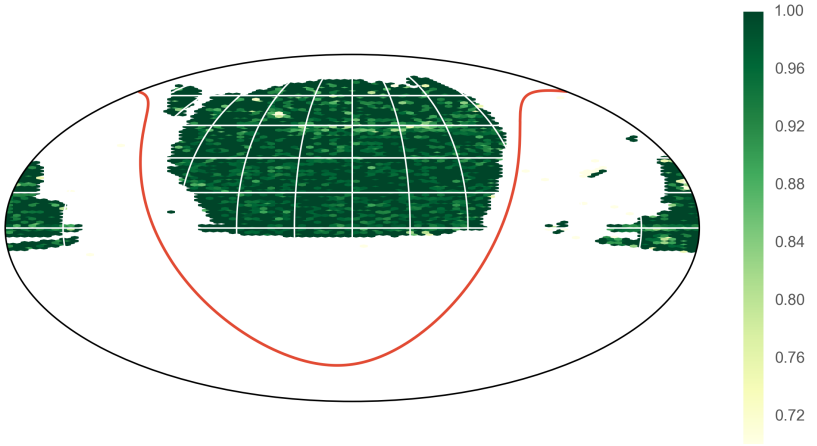


(b) Distribution of stars.

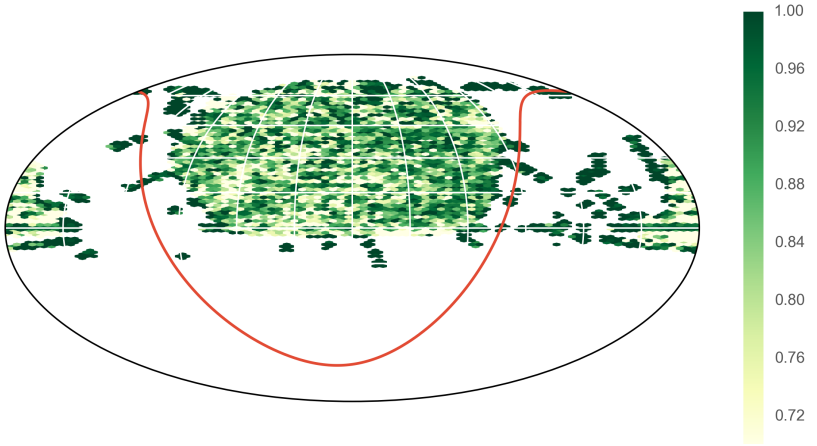


(c) Distribution of quasars.

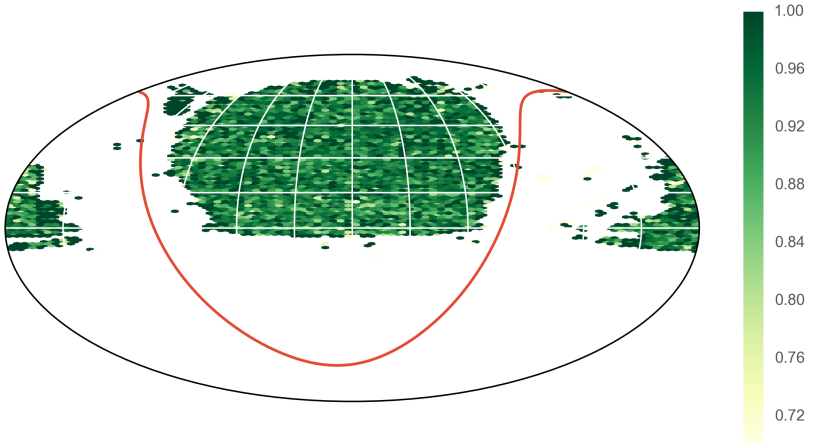
Figure 4.12: Map of predicted labels on the whole SDSS dataset using random forest: This map is, however, only moderately useful, since it is not a random sample of the sky.



(a) Recall of galaxies.



(b) Recall of stars.



(c) Recall of quasars.

Figure 4.13: Recall maps of the random forest: The recall of galaxies is almost perfect, while the recall of stars is fairly average.

Chapter 5

Experiment 2: Learning with Thompson Sampling

Let us now turn our attention to active learning. In this chapter, we investigate the six active learning heuristics described in section 3.4: two uncertainty sampling heuristics (entropy and margin), two version space reduction heuristics (QBB margin and QBB KL), variance minimisation, and classifier certainty. We then use Thompson sampling to see if it can automatically filter out the bad heuristics and select the optimal ones.

5.1 Experimental Protocol

Out of the four classifiers studied in Chapter 4, only logistic regression and RBF SVMs provide reliable probability estimates. Random forests are able to predict class probabilities by looking at the distribution of votes. However, we find in practice that they are not stable. With linear SVMs, the probability functionality has not been implemented in scikit-learn. Thus we are left with only two choices: logistic regression and RBF SVMs. The hyperparameters of these classifiers are the same as those in the previous chapter.

For the unlabelled pool, we examine two cases. In one case, the pool is balanced, with an equal number of objects in each class. This allows us to find out the maximum performance of the heuristics in the best possible scenario. Of course, in real life, the classes are not evenly distributed. Thus in the second case, we keep the original class distribution and see if the performance would deteriorate in any way.

Given two choices of classifiers and two ways to construct the unlabelled pool, we have in total four variations for each dataset and heuristic. In each run, we do a stratified shuffle split of the training and test data with 10 trials. For the SDSS data, the training and test sets each contain 10,000 examples in each split. For the VST ATLAS data, each split uses 70% of all the available data (or 24,539 objects) for training and the remaining (10,516 objects) for testing. So that we can have a meaningful comparison, a random seed is used to make sure that the same splits are used for all the variations and heuristics.

All heuristics require probability estimates. Thus we need to start with a partially trained classifier. Initially, 50 random examples are chosen for training. We find that

any number smaller than this would result in a very poor hypothesis with unreliable probability estimates, and if we pick a number that is too high, the learning curves might be less interesting. A starting sample of size 50 seems to be a good balance.

Some clarification about the terminology is needed. In this chapter, the training set is used as the unlabelled pool. In other words, we initially pretend that we do not have any labels for objects in the training set yet, except for the initial seed of 50 examples. In each round, the active learner is only allowed to assign a score and label objects in this set. The test set is hidden away from the active learner and is used to calculate the posterior balanced accuracy at the end of each round. This can then be used to make the learning curves.

The values of the following parameters are chosen so that the experiments can finish in a reasonable amount of time:

- $E = 300$: In each round, we pick 300 random objects from the pool to be ranked.
- $n = 300$: We only run each active learning experiment until we have 300 labelled objects in the training set.
- $B = 11$: For the two heuristics that require a committee of classifiers, QBB margin and QBB KL, we choose a committee of size 11.
- Finally, the pool variance and pool entropy heuristics require the estimation of the variance and the entropy, respectively, of the entire unlabelled pool. Given that this is quite an expensive computation, we only select a sample of 300 from the unlabelled pool.

In Thompson sampling, we also need to start with some prior knowledge and a likelihood variance:

- $\mu_i = 0$: Since initially, we know nothing about the performance of the heuristics, it is reasonable to set the prior mean of the mean reward of all heuristics to zero.
- $\sigma_i^2 = \tau_i^2 = 0.02$: An arbitrarily small non-zero number is chosen for the prior and the likelihood variances. These values should not be zero, since otherwise we would not be able to update our beliefs.

5.2 Results and Discussion

We split the discussion of the results into two parts. We first focus on how well active learning does with the SDSS dataset (Section 5.2.1). We then examine the results with the VST ATLAS data (Section 5.2.2), which is much cleaner but has a rare class of white dwarfs. For each dataset, the discussion is accompanied by seven pages of visualisations. These are

1. Violin plots of the posterior balanced accuracy distribution after 300 objects have been labelled: These plots give us a quick visual glance of how well the heuristics perform relative to each other.

2. Average learning curves of heuristics that perform worse than random sampling.
3. Average learning curves of heuristics that perform better than random sampling: We separate out the bad heuristics from the good ones mainly to avoid clutter in the plots.
4. Plots showing the total number of selections of the six heuristics in Thompson sampling: These allow us to see how well the Thompson sampling algorithm balances between exploration and exploitation.
5. Plots of the selection frequencies of the six heuristics in Thompson sampling.
6. Plots of cumulative rewards of the six heuristics in Thompson sampling.
7. Plots of the mean rewards of the six heuristics in Thompson sampling.

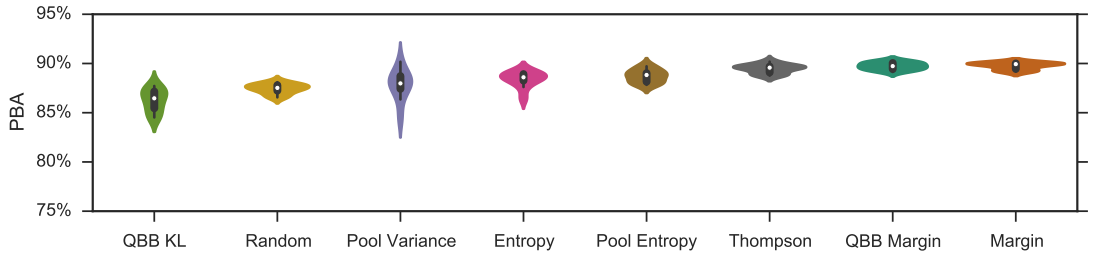
To make the comparison easier, we have associated each heuristic with the same colour throughout all the plots. The are also two more sets of visualisations which we put in Appendix C.

5.2.1 Learning with the SDSS Dataset

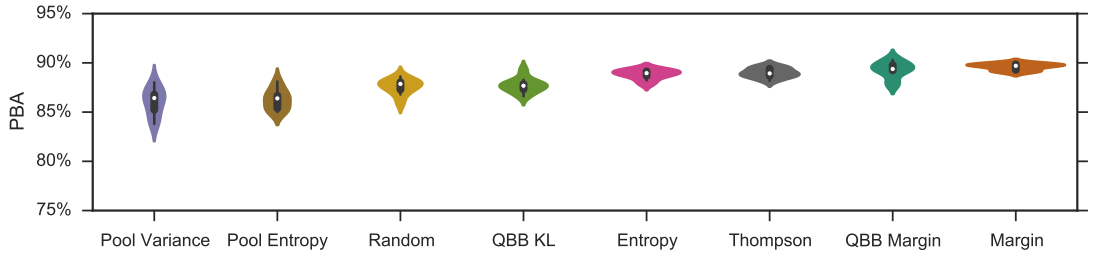
Figures 5.1 to 5.7 show the results of the experiment on the SDSS dataset. Overall, margin and QBB margin have consistently been the top two heuristics. They can outperform random sampling by as much as 2%. QBB margin appears to be no worse the the simple margin approach. This is expected since the bagging technique has been shown to improve the stability of the predictions [Breiman 1996].

The QBB KL heuristic gives a very poor result in logistic regression, while it is comparable with random sampling in RBF SVMs. The pool variance heuristic is derived from the Taylor series approximation of multinomial logistic regression and it seems to work reasonably well with the one-vs-rest strategy. As we would expect, the theory of SVMs are completely different from that of logistic regression, making the pool variance heuristic unsuitable here. It is interesting to see that the pool entropy heuristic also performs quite badly with SVMs, while the uncertainty sampling entropy approach does very well. Finally whether the original data pool is balanced or unbalanced seems to have a very little effect on the outcome.

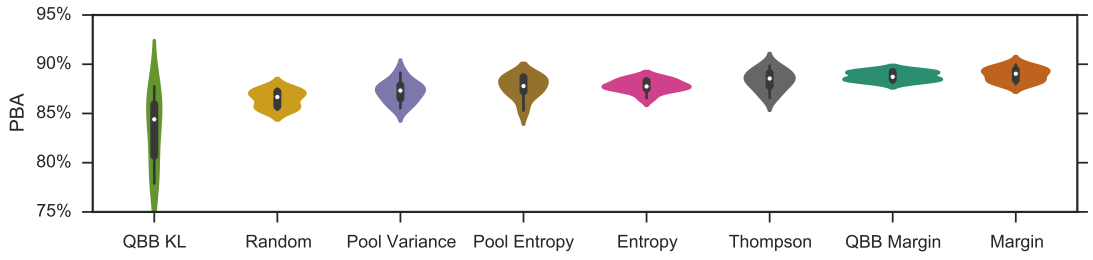
For the Thompson sampling, it seems that even under the very simplifying assumption of a normal prior and a normal likelihood, the algorithm still manages to automatically detect the optimal heuristic. In most cases, the algorithm discovers the margin heuristic to be optimal after only around 50 samples. Although Thompson sampling does not perform as well as using the margin heuristic alone, this is expected since it needs to do some exploration. Still, it gives us the third best learning curve in all cases, after margin and QBB margin. Finally, there is indeed the problem of drifting in the mean of the rewards. It would be interesting to find out how much quicker the algorithm can recognise the optimal heuristic if we were to implement the Dynamic Thompson Sampling method.



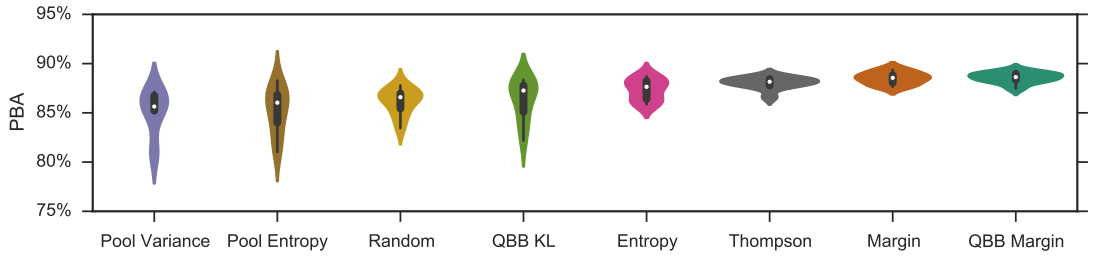
(a) Balanced pool and logistic regression



(b) Balanced pool and RBF SVM

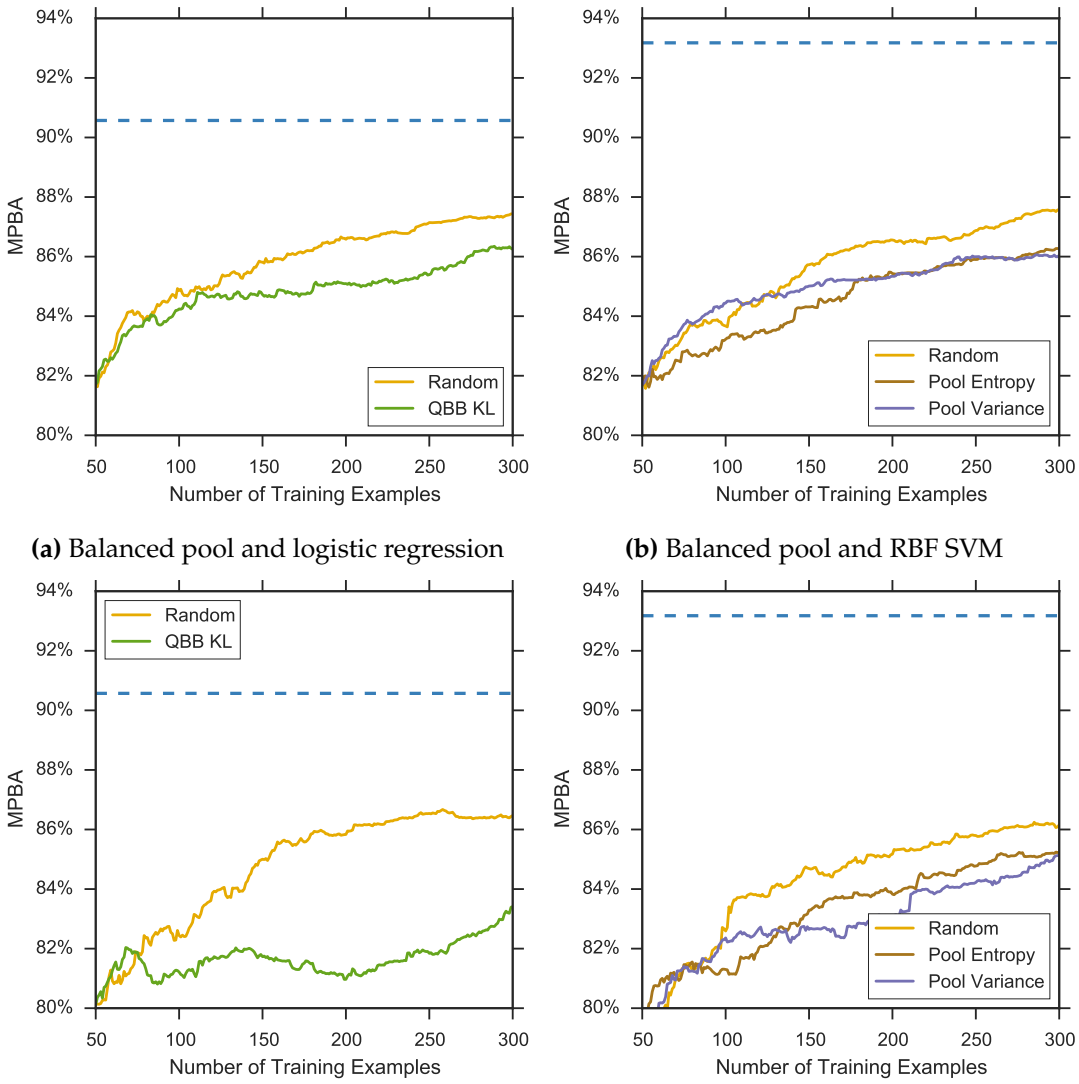


(c) Unbalanced pool and logistic regression



(d) Unbalanced pool and RBF SVM

Figure 5.1: Posterior distributions of the balanced accuracy when the SDSS training set size reaches 300: Overall, with the exception of QBB KL and pool variance, all the other heuristics manage to outperform random sampling. Both margin and QBB margin are consistently in the top two, with Thompson sampling always at third place. This is expected since there is always some exploration in Thompson sampling. Although the pool variance heuristic on average outperforms random sampling slightly, it has a large dispersion and thus unreliable.



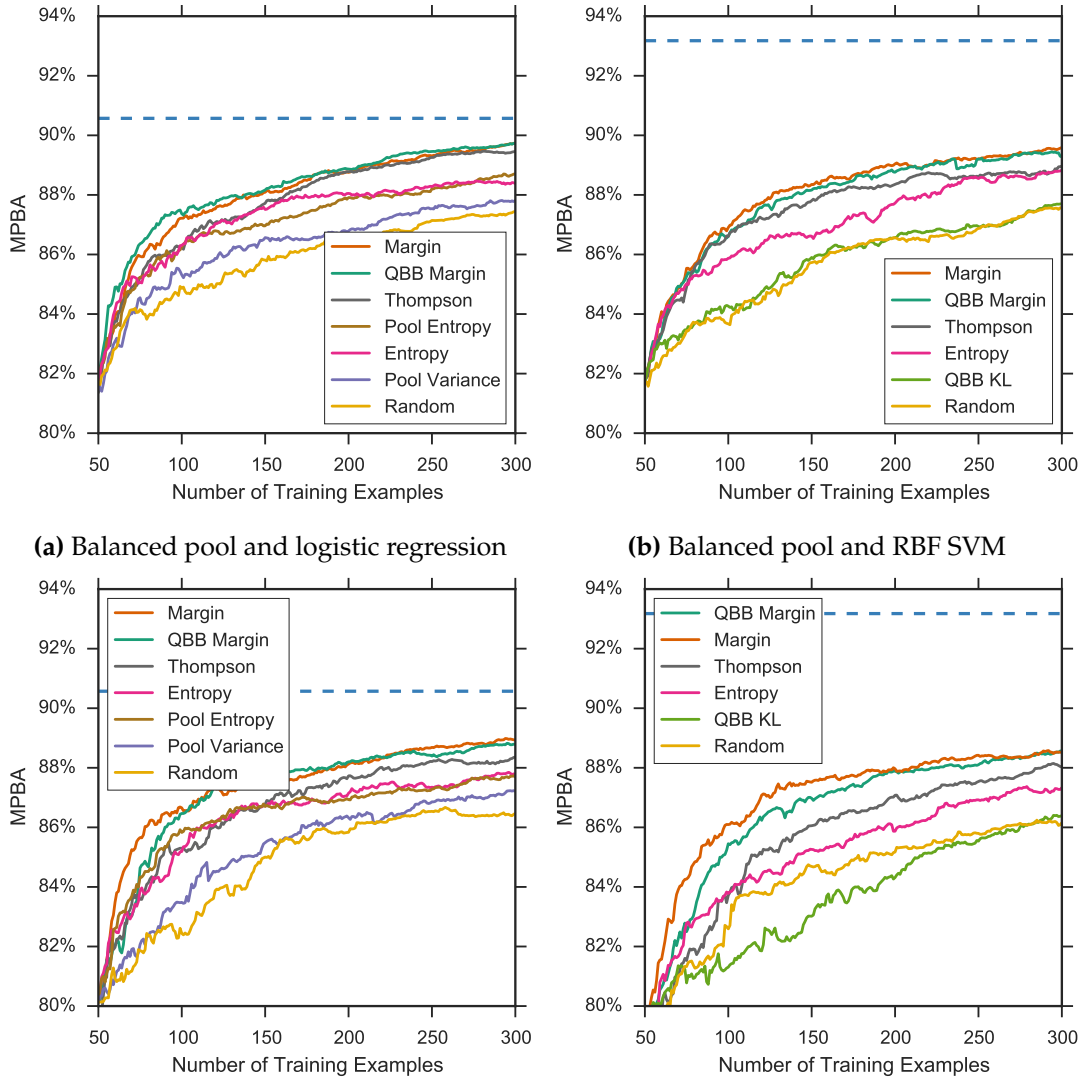
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.2: Learning curves (average of 10 trials) of heuristics that perform worse than random sampling in the SDSS dataset: We can clearly see that with logistic regression, QBB KL performs much worse than random sampling. For RBF SVM, the bad heuristics are pool variance and pool entropy. The underperformance of the pool variance heuristic is expected since the theory does not quite apply to SVMs. The dashed line is the maximum accuracy achievable by the classifier (i.e. the maximum value of the learning curve in Chapter 4)



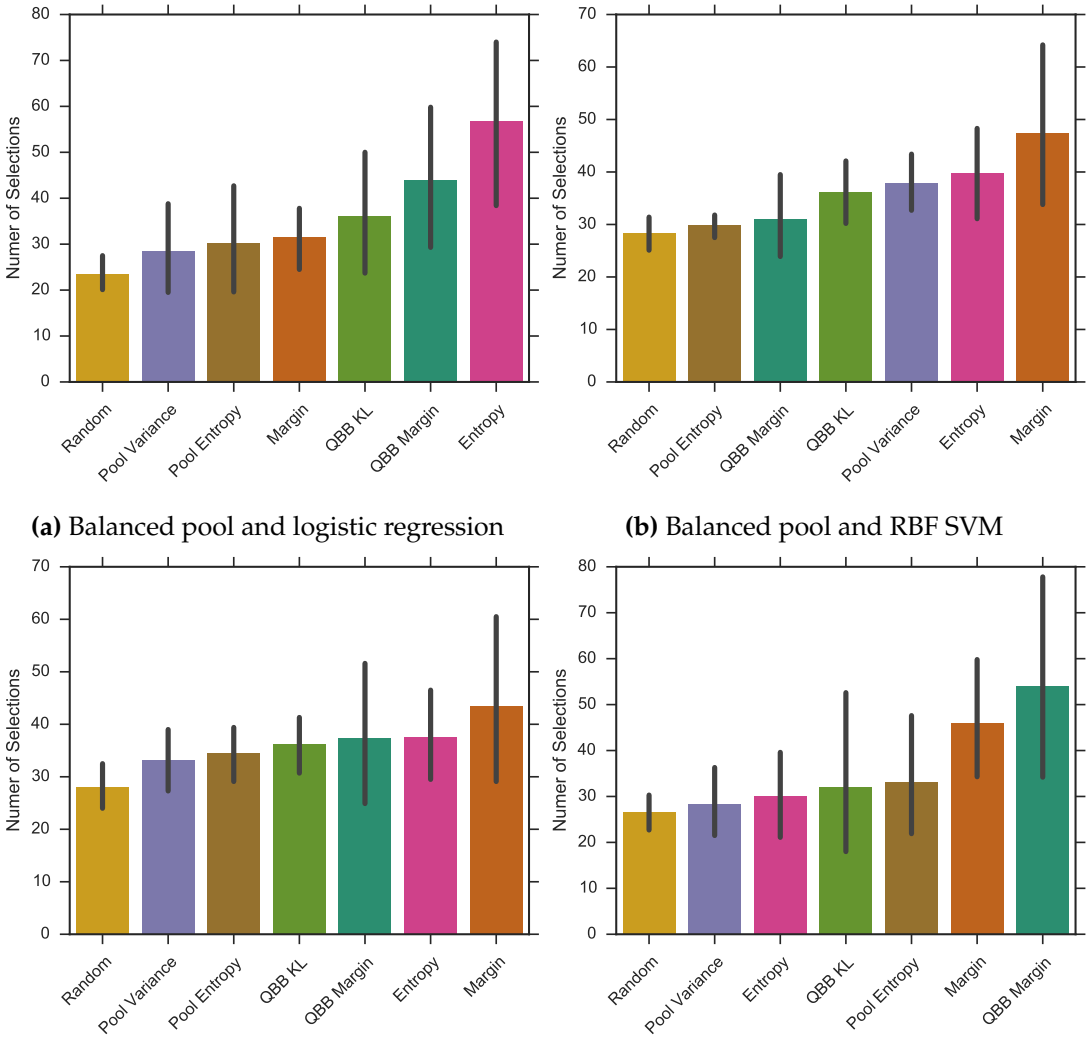
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.3: Learning curves (average of 10 trials) of heuristics that outperform random sampling in the SDSS dataset: Again the dashed line represents the maximum accuracy achievable by the classifier. Although the learning curves of RBF SVM have a lot more potential to go higher, when the sample size is small, its performance is not much different from logistic regression, even with active learning. Another interesting observation is that with the unbalanced pool and RBF SVM, the QBB KL heuristic is worse than random sampling initially but has managed to surpass it just before 300 samples. It would be interesting to see their relative performance as we go beyond 300.



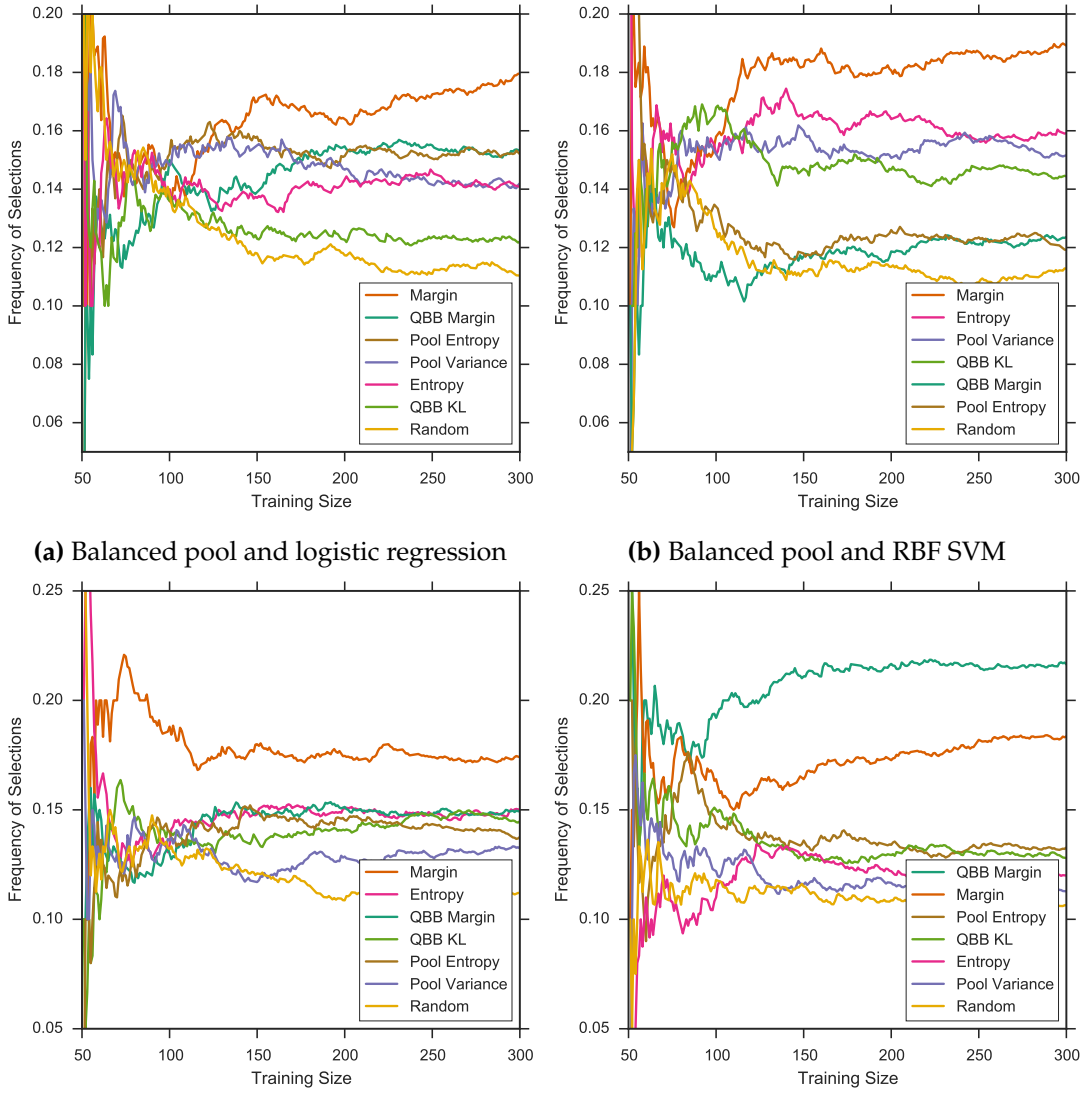
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.4: Total number of selections (average of 10 trials) of the six heuristics in Thompson sampling with the SDSS dataset: It seems that Thompson sampling has managed to select the better heuristics slightly more often. There also a reasonable amount of exploration.



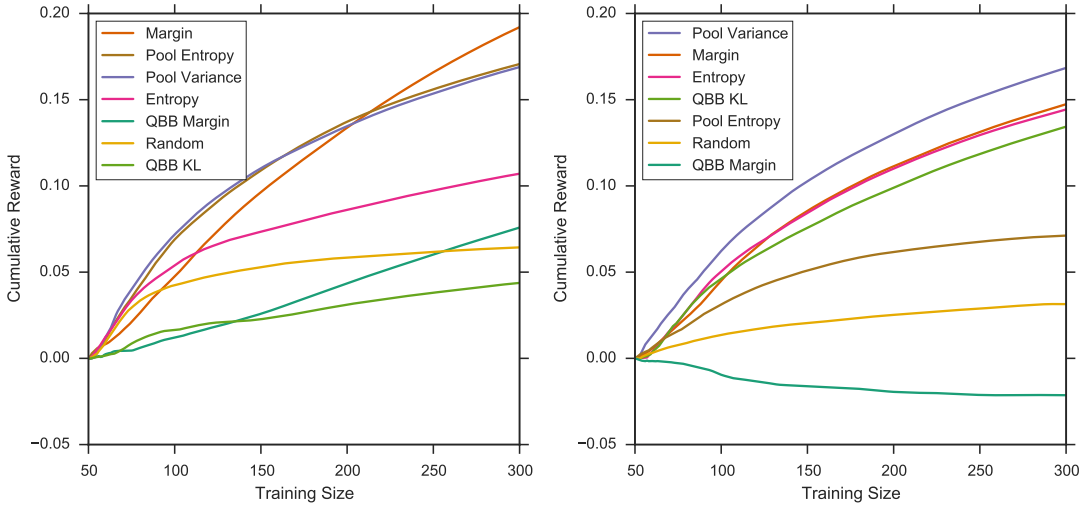
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

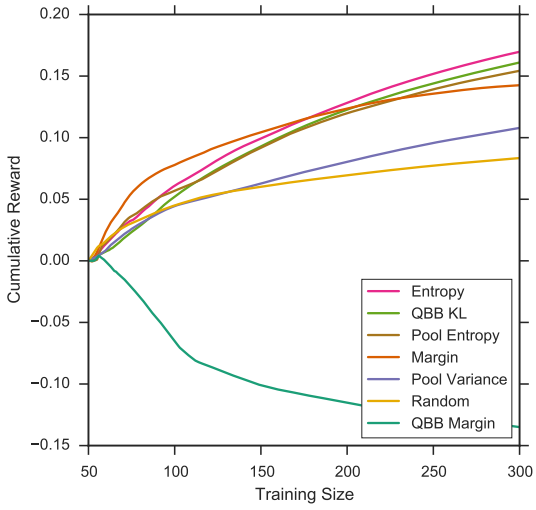
(d) Unbalanced pool and RBF SVM

Figure 5.5: Heuristic selection frequency (average of 10 trials) in Thompson sampling with the SDSS dataset: Here we get a better look of how the selection frequency changes with the sample size. After only around 20 round, the algorithm has managed to identify to optimal heuristics like margin.

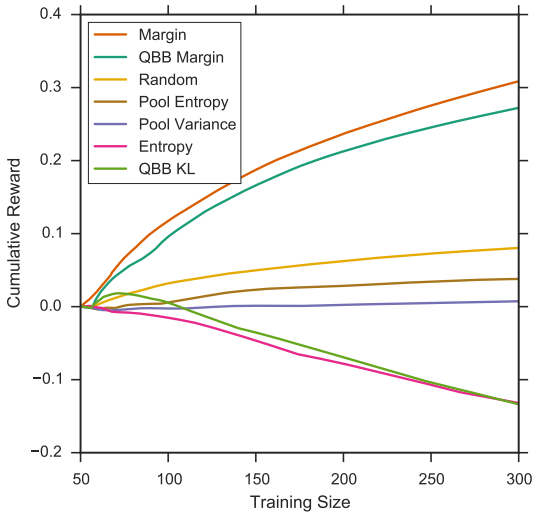


(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

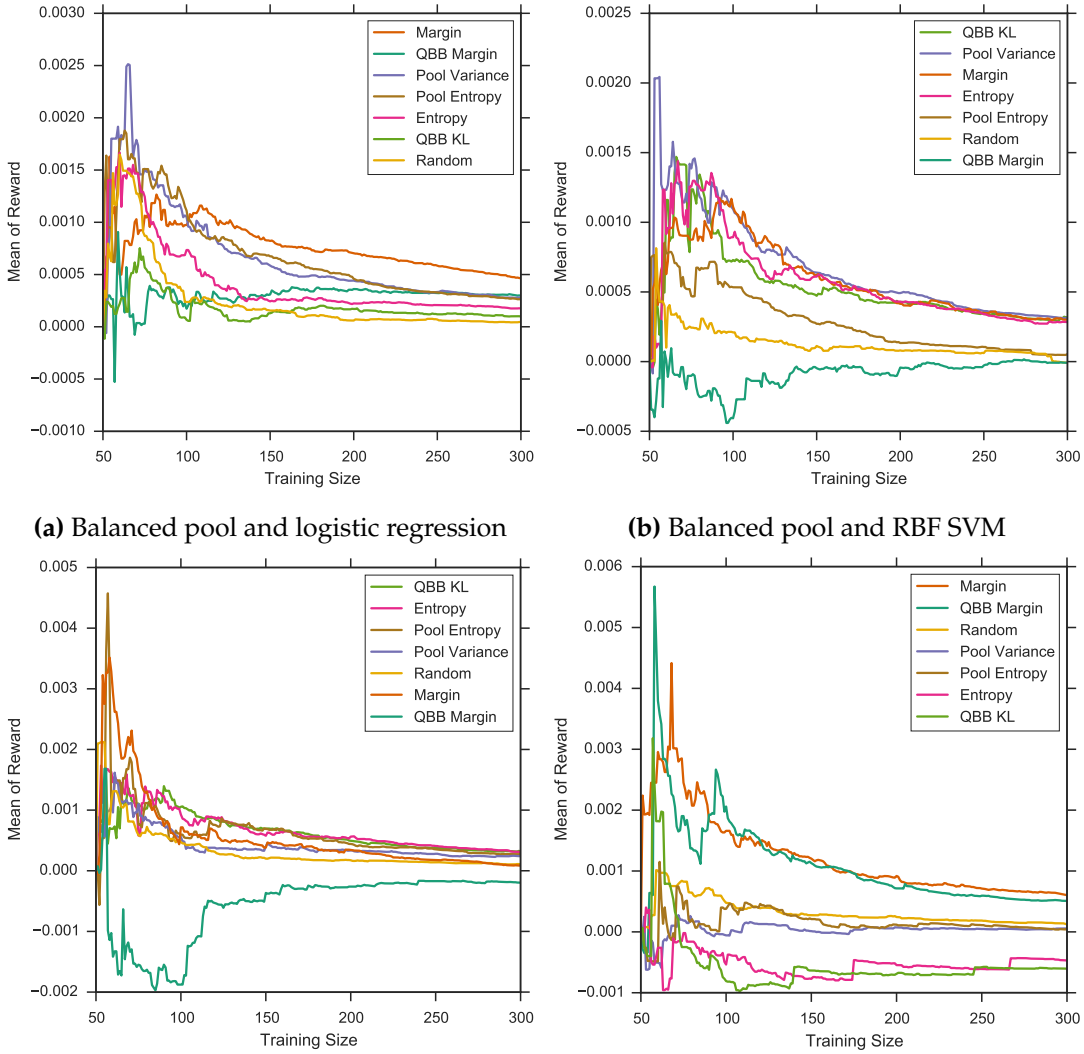


(c) Unbalanced pool and logistic regression



(d) Unbalanced pool and RBF SVM

Figure 5.6: Cumulative reward (average of 10 trials) in Thompson sampling with the SDSS dataset: One anomaly is with the unbalanced pool and logistic regression, where the average cumulative reward of QBB margin is very low. However, it is supposed to be the second best heuristic. This could simply be due to the averaging process, in which the result is skewed due to an unlucky round. In such rounds, QBB margin might give an unusually low reward (resulting in a false belief).



(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

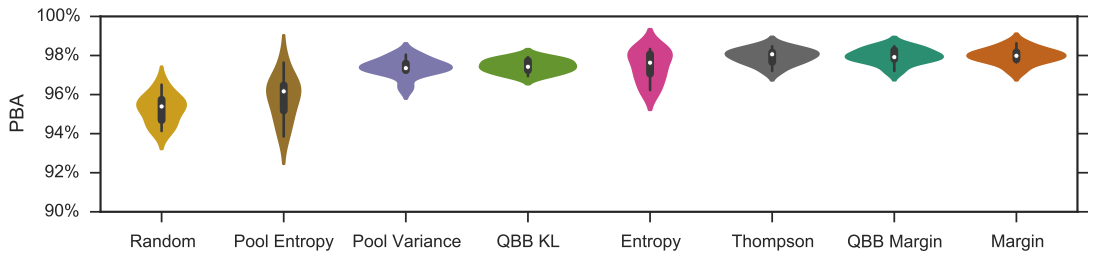
Figure 5.7: Mean reward (average of 10 trials) in Thompson sampling with the SDSS dataset: As expected, the rewards do become smaller over time and drift toward zero.

5.2.2 Learning with the VST ATLAS Dataset

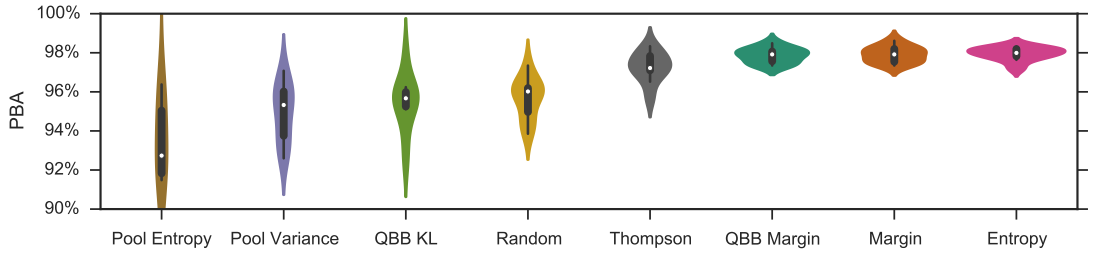
Figures 5.8 to 5.14 show the results of the experiment on the VST ATLAS dataset. The VST ATLAS is a much cleaner dataset, and the active learning seems to work much better here. Margin, QBB margin, and entropy all perform much better than random sampling. At times, especially when the dataset is unbalanced, the difference in the performance can be as great as 9%. This is a great result since in practice, most datasets have unbalanced classes.

Thompson sampling is still doing well. Even with the unbalanced pool and logistic regression, where the algorithm underestimates the usefulness of QBB margin, Thompson sampling still manages to finish third at the end. This is actually a great feature since it means that it is not necessary for Thompson sampling to rank all the heuristics in the exact same order as the order that we get when we study the heuristics individually. A small amount of false belief does not seem to hurt its overall performance.

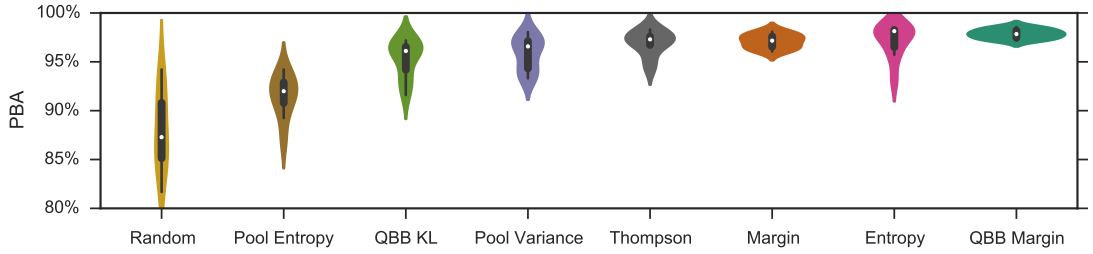
Finally, depending on the setting, the three worst heuristics are pool variance, pool entropy and QBB KL. Often these heuristics are even worse than random sampling. In a way, this is actually good news because they are also the three most computationally expensive heuristics and thus they are not very practical anyway. In particular, even when we run the pool variance and the pool entropy heuristics on the Amazon Elastic Compute Cloud using 36 virtual CPU cores, it still takes a few days to complete. On the other hand, the margin heuristic can be run comfortably on a normal laptop.



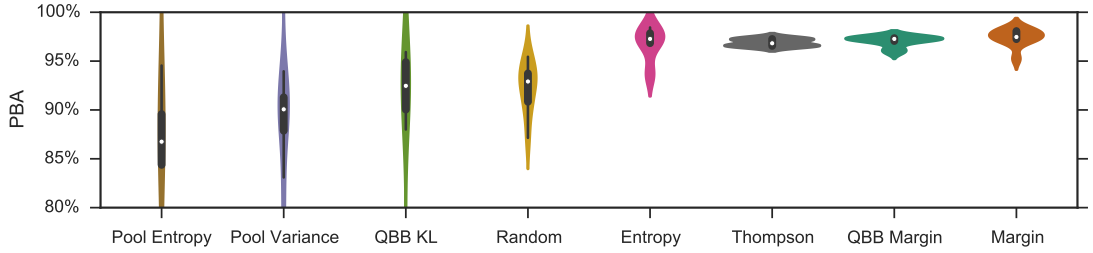
(a) Balanced unlabelled pool and logistic regression



(b) Balanced unlabelled pool and RBF SVM

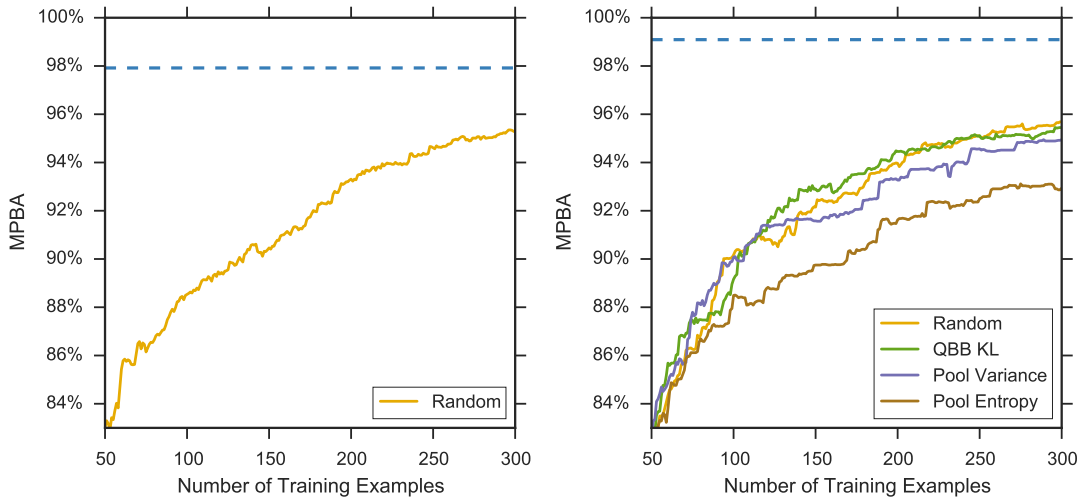


(c) Unbalanced unlabelled pool and logistic regression



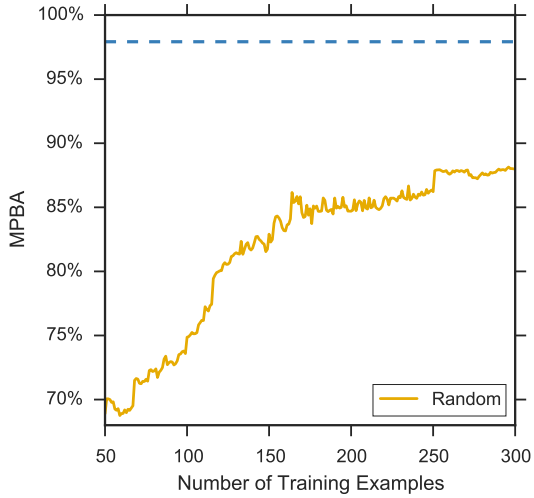
(d) Unbalanced unlabelled pool and RBF SVM

Figure 5.8: Posterior distributions of the balanced accuracy when the VST ATLAS training set size reaches 300: Overall, all heuristics outperform random sampling when we use logistic regression. With RBF SVM, the two pool heuristics and QBB KL do not seem to do as well. In particular, as we can see in Figure 5.8d, when the classes are unbalanced, these three heuristics become very unreliable.

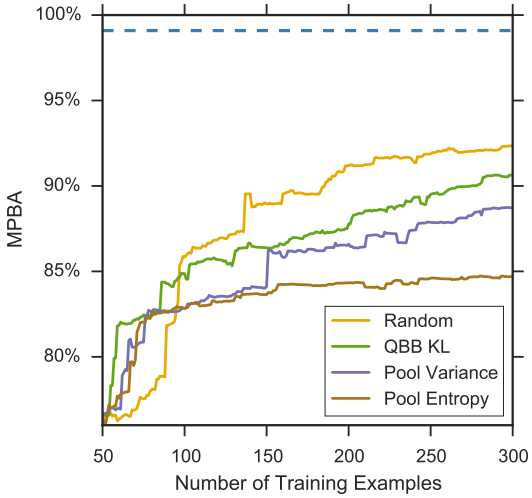


(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

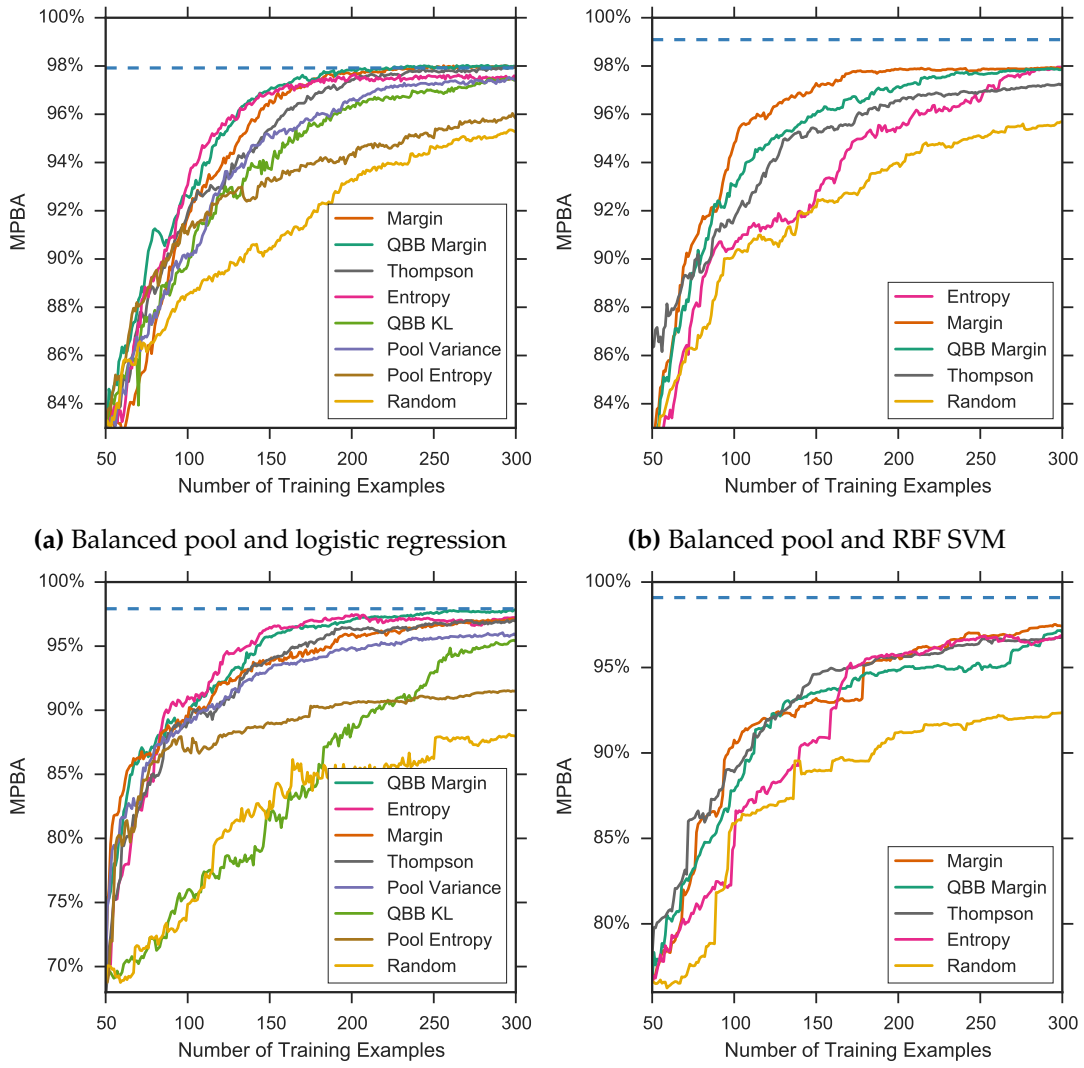


(c) Unbalanced pool and logistic regression



(d) Unbalanced pool and RBF SVM

Figure 5.9: Learning curves (average of 10 trials) of heuristics that perform worse than random sampling in the VST ATLAS dataset: The plots for logistic regression are quite uninteresting since all heuristics beat random sampling. In Figure 5.9d, the learning curve for the pool entropy heuristic appears to flatten out after 100 samples.



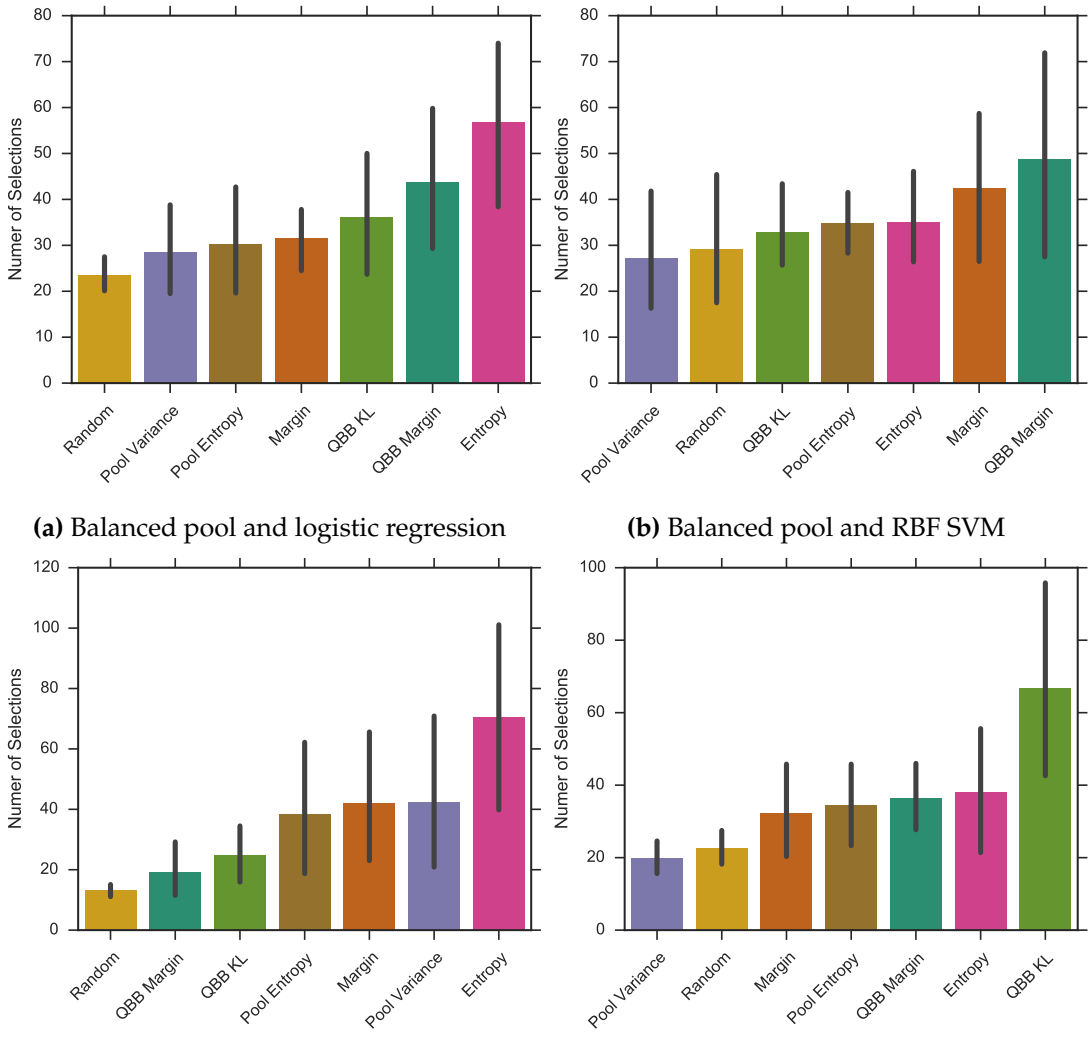
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.10: Learning curves (average of 10 trials) of heuristics that outperform random sampling in the VST ATLAS dataset: It is great to see that with logistic regression, we have basically reached the maximum accuracy (indicated by the dashed line) after only 200 samples.



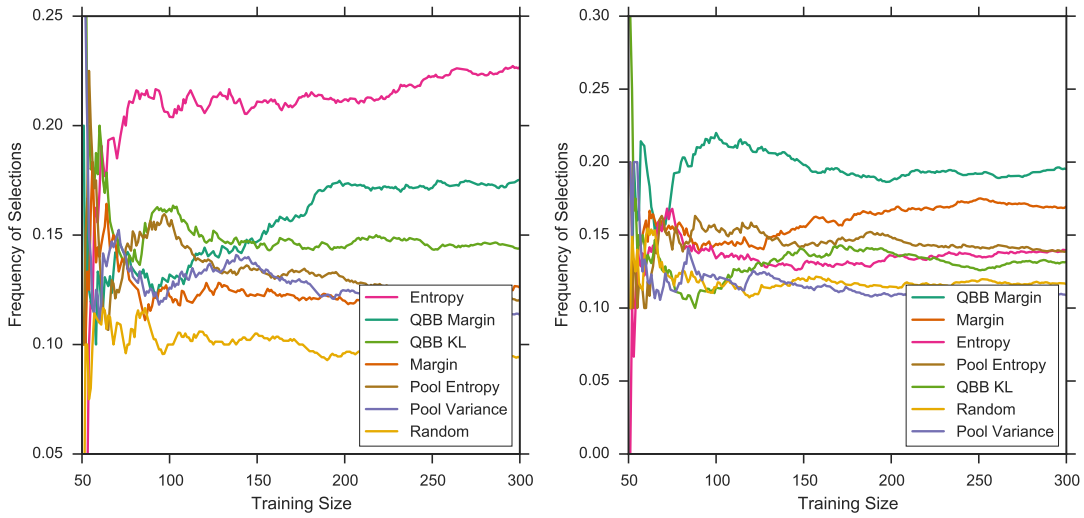
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

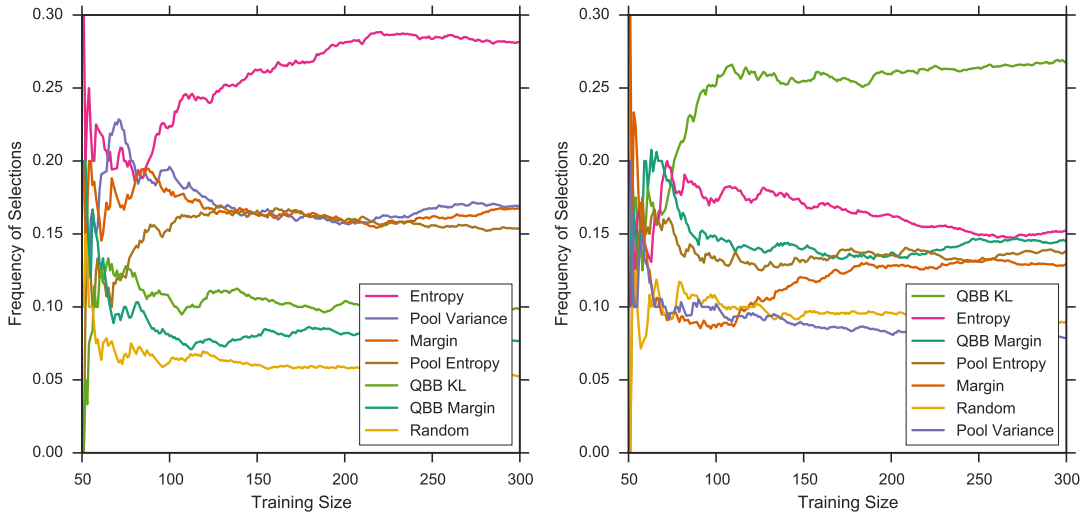
(d) Unbalanced pool and RBF SVM

Figure 5.11: Total number of selections (average of 10 trials) of the six heuristics in Thompson sampling with the VST ATLAS dataset: Like the SDSS, it seems that most of the time, Thompson sampling manages to select the better heuristics more often than average, while maintaining a good exploration. There are also a few anomalies. For example with an unbalanced pool and RBF SVM, QBB KL is picked the most number of times. However when we study it individually, it performs worse than random sampling. And yet, Thompson sampling still finishes third, as shown in Figure 5.10d. Thus a small amount of false belief seems to not be a problem.



(a) Balanced pool and logistic regression

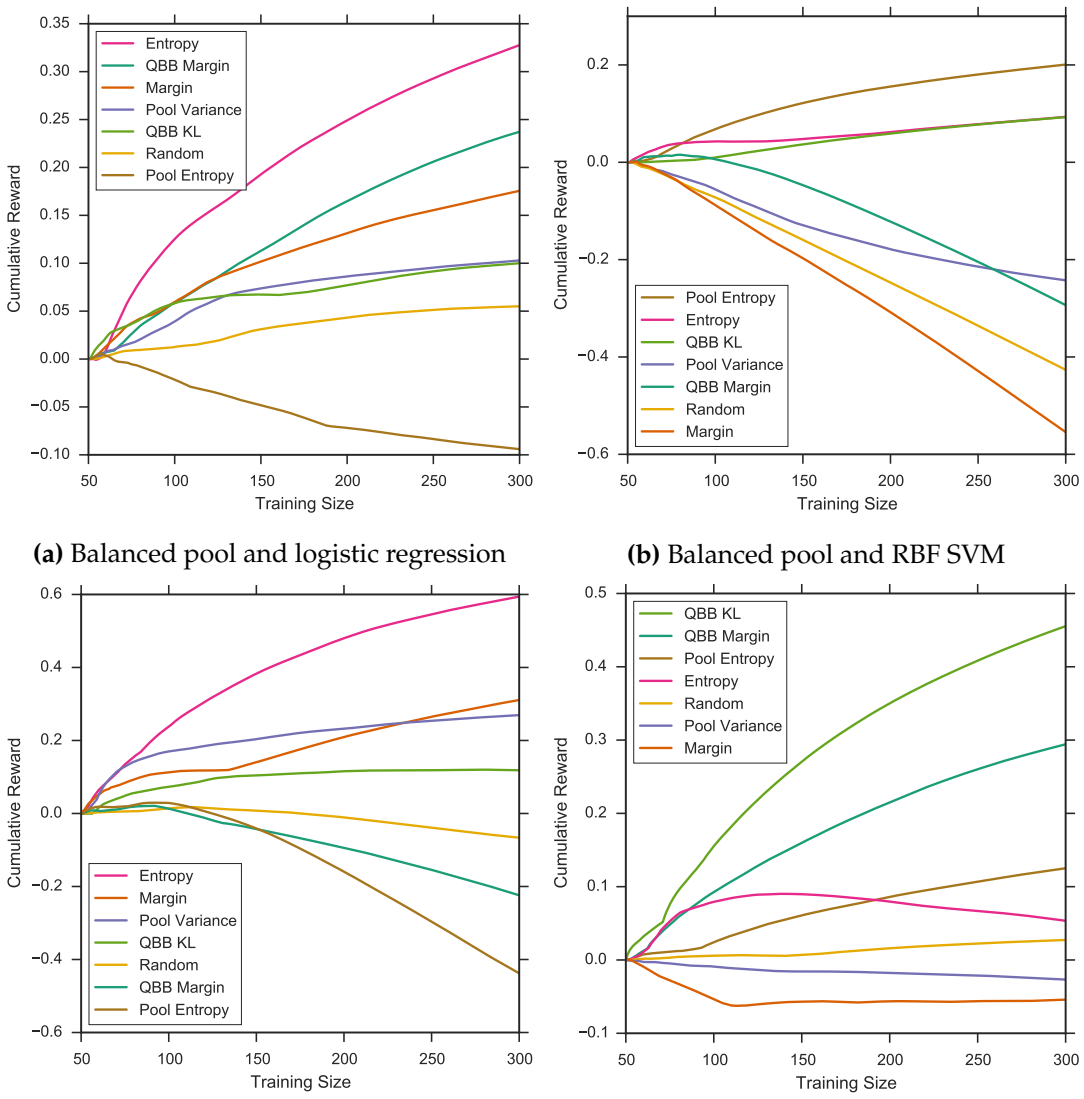
(b) Balanced pool and RBF SVM



(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.12: Heuristic selection frequency (average of 10 trials) in Thompson sampling with the VST ATLAS dataset.



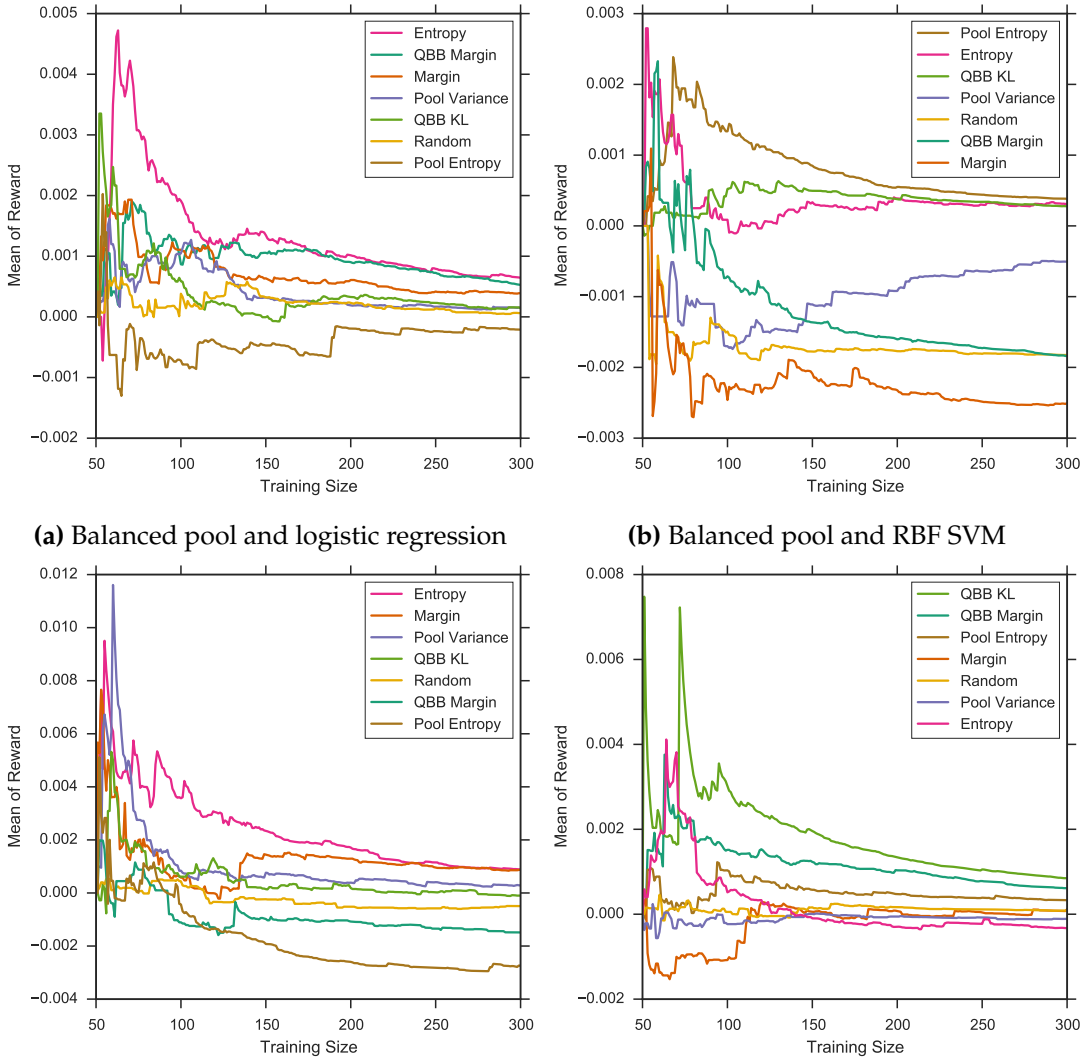
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.13: Cumulative reward (average of 10 trials) in Thompson sampling with the VST ATLAS dataset.



(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure 5.14: Mean reward in Thompson sampling with the VST ATLAS dataset.

Chapter 6

Conclusion

"Simplicity is the final achievement. After one has played a vast quantity of notes and more notes, it is simplicity that emerges as the crowning reward of art."

— Frédéric Chopin, as quoted in *If Not God, Then What?*

This thesis provides an initial set of experiments that explore how active learning can help astronomers with photometric classification. Through our novel contribution of bringing the classic Thompson sampling algorithm to the setting of heuristic selection, we have managed to learn the optimal heuristic automatically very quickly, while still have a reasonable amount of exploration. The cleaner the dataset is and the better the probability estimates are, the more effective active learning will be. For example, we achieve the best improvement with logistic regression and the VST ATLAS dataset. The margin heuristic comes out as a clear winner, even under noisy conditions like in the SDSS. This is great news because it is also the fastest heuristic to run. Simplicity wins at the end of the day.

6.1 Related Works

Our thesis is built mainly upon the work of Schein and Ungar [2007], who investigate the individual active learning heuristics described in this thesis and find that QBB margin provides the most promising results. There are other works in the literature that also do an empirical analysis of active learning in other domains such as text classification [Tong and Koller 2002] and sequence labelling tasks [Settles and Craven 2008]. Most of these works suggest that active learning does indeed offer an improvement over random sampling . However, as far as we know, this thesis is the first in applying active learning to optimal astronomy.

Machine learning methods other than active learning have, of course, been applied to astronomical data. Hála [2014] uses neural networks to learn directly from spectra. Elting et al. [2008] employ a mixture model and clustering to perform class discovery on a portion of the SDSS photometric data. Finally, Bazell et al. [2006] train an SVM with an RBF kernel to predict class proportions in the unlabelled SDSS set, like what we did in Section 4.3.4. However they are more cautious and only predict labels of

objects that are within the inner 90% of the colour space of the training set and have a low measurement error. Given the constraints, they find that 2.3% of the unlabelled objects are quasars. Our prediction, where we do not impose any constraint on the colour space, is 21.8%.

6.2 Future Works

The work in this thesis is only a beginning. In particular, we have only done an empirical investigation in the astronomical domain. It would be an interesting exercise to conduct a theoretical analysis to see under what assumptions heuristics such as margin would guarantee to outperform random sampling. Another possible future direction is to derive a variance estimation method for both SVMs and one-vs-rest logistic regression, since the variance estimate that we used here is only true under the multinomial logistic regression. We can also try to implement the Dynamic Thompson Sampling approach to address the reward drifting problem.

In our investigation, the algorithm only works with one sample at a time. What astronomers would normally do in practice, however, is batch active learning, where in each round, n objects are selected to be labelled simultaneously. This problem is slightly more challenging since we need to deal with the setting where we could have two objects whose class membership we are currently very uncertain about, but the knowledge of class membership of one of them would allow us to predict the other object's class easily. Cluster analysis might be able to help us with this problem of batch active learning.

Appendix A

How to Obtain the Datasets

The following SQL query is used to get the main SDSS dataset containing 2.8 million labelled objects from the Sloan SkyServer:[1]

```
SELECT
    -- right ascension and declination in degrees
    p.ra, p.dec,

    -- class of object, expert opinion (galaxy, star, or quasar)
    CASE s.class WHEN 'GALAXY' THEN 'Galaxy'
                 WHEN 'STAR' THEN 'Star'
                 WHEN 'QSO' THEN 'Quasar'
                 END AS class,

    s.subclass, -- subclass of object

    -- redshift of object from spectrum with error, expert opinion
    s.z AS redshift,
    s.zErr AS redshiftErr,
    s.zWarning,

    -- PSF and Petrosian mags in 5 bands (ugriz) with error
    p.psfMag_u, p.psfMagErr_u,
    p.psfMag_g, p.psfMagErr_g,
    p.psfMag_r, p.psfMagErr_r,
    p.psfMag_i, p.psfMagErr_i,
    p.psfMag_z, p.psfMagErr_z,

    p.petroMag_u, p.petroMagErr_u,
    p.petroMag_g, p.petroMagErr_g,
    p.petroMag_r, p.petroMagErr_r,
    p.petroMag_i, p.petroMagErr_i,
    p.petroMag_z, p.petroMagErr_z,

    -- extinction values
    p.extinction_u, p.extinction_g, p.extinction_r,
    p.extinction_i, p.extinction_z,

    -- size measurement in r-band in arc seconds
```

[1] http://skyserver.sdss.org/CasJobs/

```
    p.petroRad_r, p.petroRadErr_r

FROM PhotoObj AS p
JOIN SpecObj AS s
ON s.bestobjid = p.objid

WHERE
    -- only include objects with complete and reasonably accurate data
    p.psfMagErr_u BETWEEN 0 AND 3
    AND p.psfMagErr_g BETWEEN 0 AND 3
    AND p.psfMagErr_r BETWEEN 0 AND 3
    AND p.psfMagErr_i BETWEEN 0 AND 3
    AND p.psfMagErr_z BETWEEN 0 AND 3
    AND p.petroMagErr_u BETWEEN 0 AND 3
    AND p.petroMagErr_g BETWEEN 0 AND 3
    AND p.petroMagErr_r BETWEEN 0 AND 3
    AND p.petroMagErr_i BETWEEN 0 AND 3
    AND p.petroMagErr_z BETWEEN 0 AND 3
    AND p.petroRadErr_r BETWEEN 0 AND 3
    AND s.zErr BETWEEN 0 AND 0.1
    AND s.zWarning = 0    -- spectrum is ok
```

The following query is used to extract photometric measurements from all 800 million in the database. Since the file is fairly big (around 200 GB in size), a special request might need to be made.

```
SELECT
    p.ra, p.dec,
    CASE s.class WHEN 'GALAXY' THEN 'Galaxy'
                 WHEN 'STAR' THEN 'Star'
                 WHEN 'QSO' THEN 'Quasar'
                 END AS class,
    s.subclass,
    s.z AS redshift,
    s.zErr AS redshiftErr,
    s.zWarning,
    p.psfMag_u, p.psfMagErr_u,
    p.psfMag_g, p.psfMagErr_g,
    p.psfMag_r, p.psfMagErr_r,
    p.psfMag_i, p.psfMagErr_i,
    p.psfMag_z, p.psfMagErr_z,
    p.petroMag_u, p.petroMagErr_u,
    p.petroMag_g, p.petroMagErr_g,
    p.petroMag_r, p.petroMagErr_r,
    p.petroMag_i, p.petroMagErr_i,
    p.petroMag_z, p.petroMagErr_z,
    p.extinction_u, p.extinction_g, p.extinction_r,
    p.extinction_i, p.extinction_z,
    p.petroRad_r, p.petroRadErr_r
FROM PhotoObj AS p
LEFT JOIN SpecObj AS s
ON s.bestobjid = p.objid
```

Dust Extinction Vectors

The SDF98 extinction values are given in the SDSS dataset. To calculate the other two extinction vectors, we start with a reference reddening quantity

$$E_{B-V} = \frac{\mathcal{A}_r}{2.751}$$

where $\mathcal{A}_r$ is the SDF98 extinction value in the r-band.

As a check, we can actually recover the SDF98 extinction vector as follows:

$$\begin{aligned}
\mathcal{A}_u &= 5.155 \cdot E_{B-V} \\
\mathcal{A}_g &= 3.793 \cdot E_{B-V} \\
\mathcal{A}_r &= 2.751 \cdot E_{B-V} \\
\mathcal{A}_i &= 2.086 \cdot E_{B-V} \\
\mathcal{A}_z &= 1.479 \cdot E_{B-V}
\end{aligned}$$

Later, Schlafly and Finkbeiner [2011] applied a different extinction curve, giving us the following correction values:

$$\begin{aligned}
\mathcal{A}_u &= 4.239 \cdot E_{B-V} \\
\mathcal{A}_g &= 3.303 \cdot E_{B-V} \\
\mathcal{A}_r &= 2.285 \cdot E_{B-V} \\
\mathcal{A}_i &= 1.698 \cdot E_{B-V} \\
\mathcal{A}_z &= 1.263 \cdot E_{B-V}
\end{aligned}$$

Recently, Wolf [2014] remapped the E_{B-V} scale to

$$E'_{B-V} = \begin{cases} E_{B-V} & \text{if } E_{B-V} \in [0, 0.04], \\ E_{B-V} + 0.5(E_{B-V} - 0.04) & \text{if } E_{B-V} \in [0, 0.08], \\ E_{B-V} + 0.02 & \text{if } E_{B-V} \in [0.08, +\infty]. \end{cases}$$

which can then be used to calculate a new set of correction values:

$$\begin{aligned}
\mathcal{A}_u &= 4.305 \cdot E'_{B-V} \\
\mathcal{A}_g &= 3.288 \cdot E'_{B-V} \\
\mathcal{A}_r &= 2.261 \cdot E'_{B-V} \\
\mathcal{A}_i &= 1.714 \cdot E'_{B-V} \\
\mathcal{A}_z &= 1.263 \cdot E'_{B-V}
\end{aligned}$$

Each of these correction values need to be subtracted from the corresponding magnitudes to make up for the loss of the scattered light.

Appendix C

Supplementary Results

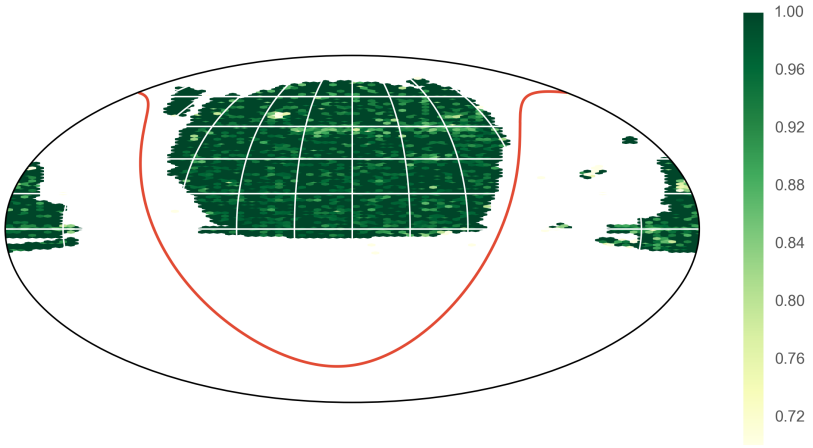
In this Appendix, we present results that are not vital to the main narrative but still somewhat interesting.

C.1 Effects of Dust Extinction on Recall

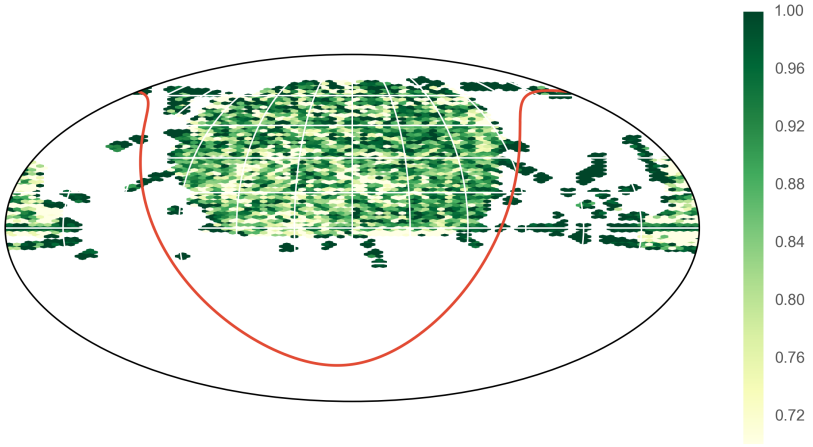
In Chapter 4 we tested the effects of three different extinction vectors on the accuracy rate. Figure C.1 shows how the recall rate is distributed over the celestial sphere. Overall, the recall on galaxies is almost perfect, while the recall on stars is fairly average. On the three pages after that, Figures C.2, C.3, and C.4 show the improvement on recall after each extinction vector is applied. The interesting bit is that there is a patch of stars right next to the Milky Way plane that gets a big improvement in recall after reddening correction. Thus the extinction vector would be very important if there were more objects that are closer to the Milky Way plane (which is the case in the SkyMapper project).

C.2 Variance of the Mean Reward in Thompson Sampling

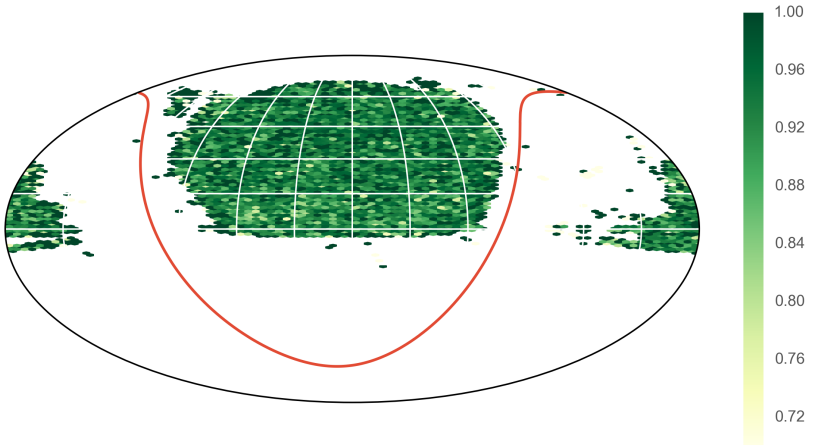
Finally, Figures C.5 and C.6 show how σ^2, the variance of the expected reward, changes with the training set size. As we would expect, the variance decreases exponentially over time, since as we increase the training size, we become more certain of the classifier's accuracy. In addition, the incremental change of the accuracy rate shrinks over time as we approach an accuracy of 100%.



(a) Recall map of galaxies.

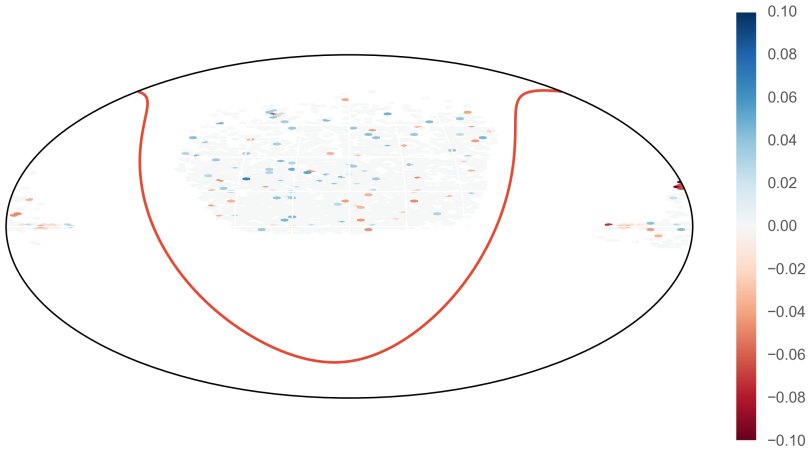


(b) Recall map of stars.

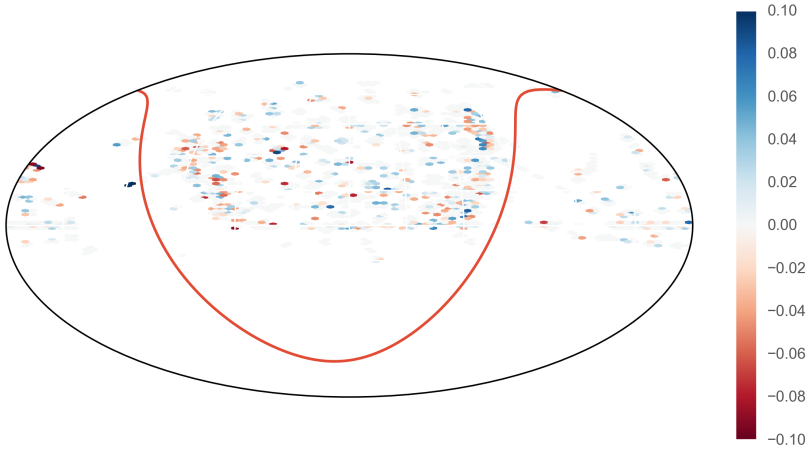


(c) Recall map of quasars.

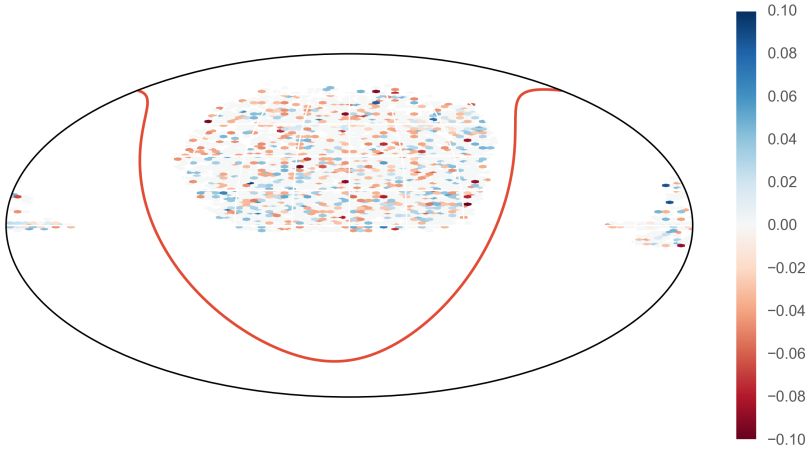
Figure C.1: Recall maps when there is no corrections.



(a) Recall improvement map of galaxies.

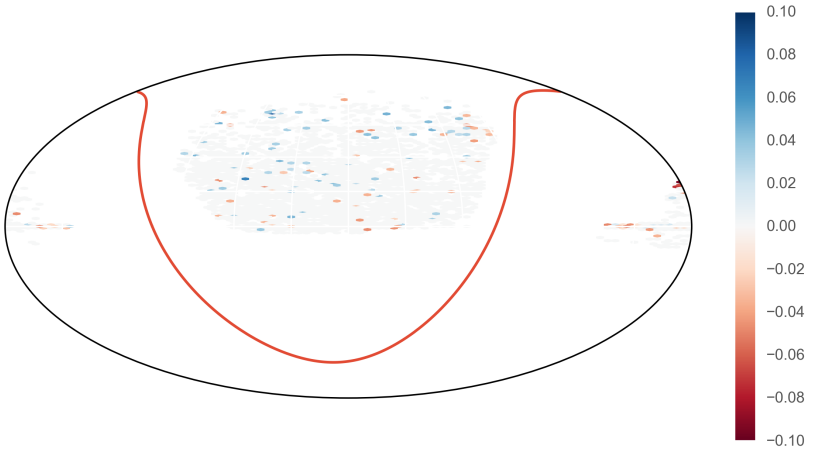


(b) Recall improvement map of stars.

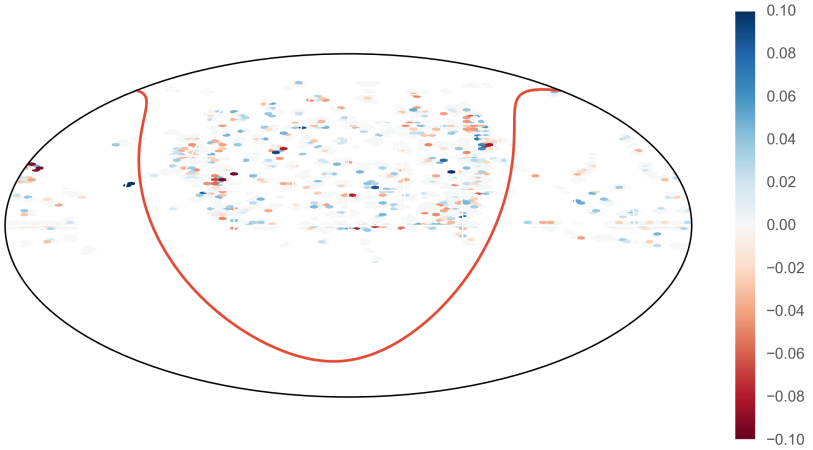


(c) Recall improvement map of quasars.

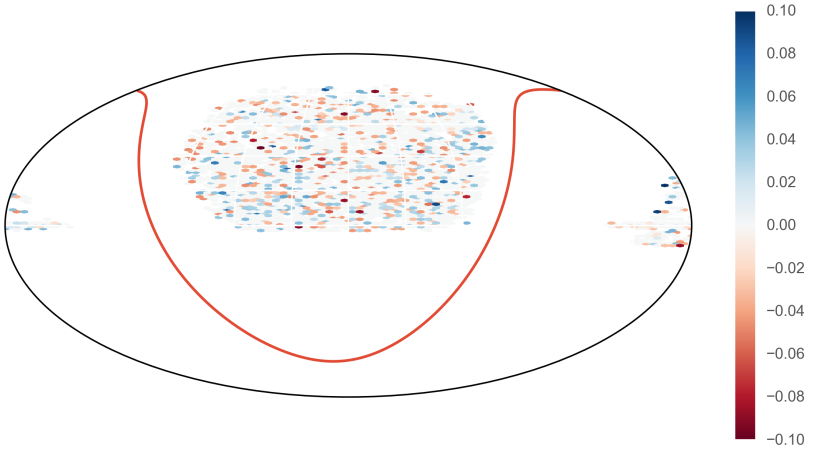
Figure C.2: Recall improvement maps when the SFD98 extinction vector is used.



(a) Recall improvement map of galaxies.

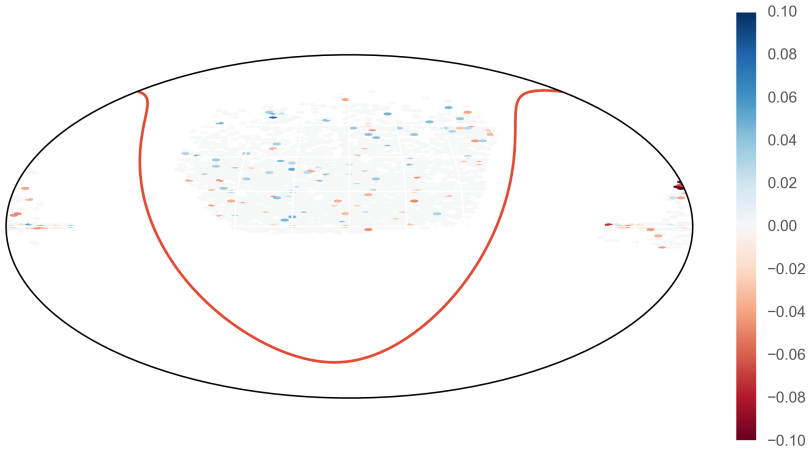


(b) Recall improvement map of stars.

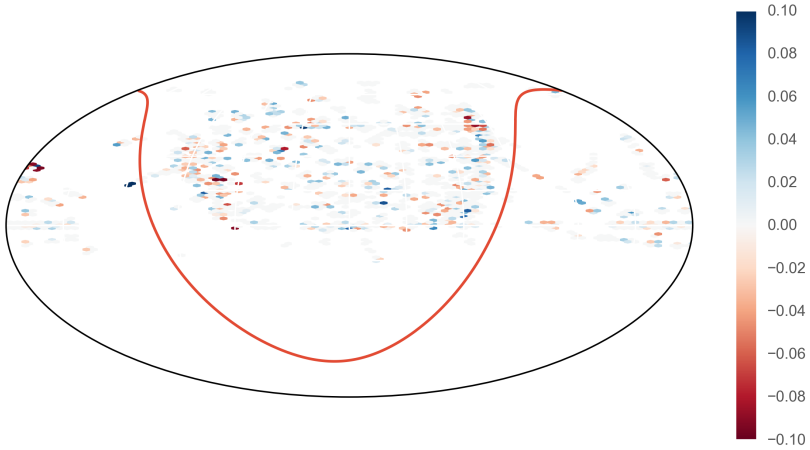


(c) Recall improvement map of quasars.

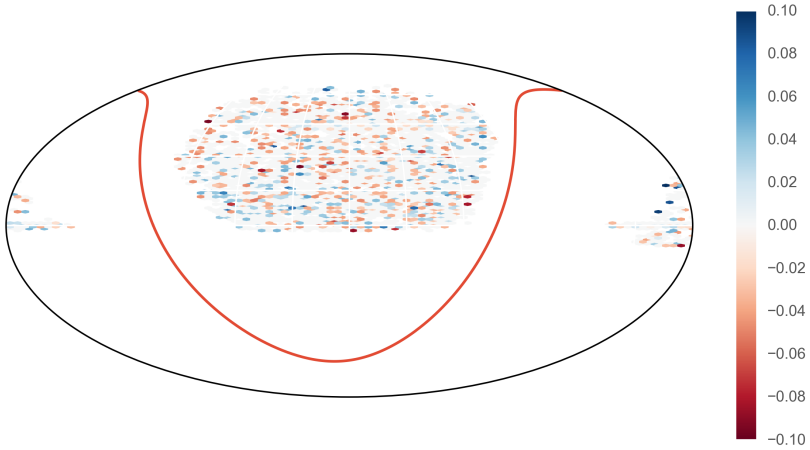
Figure C.3: Recall improvement maps of when the SF11 extinction vector is used.



(a) Recall improvement map of galaxies.

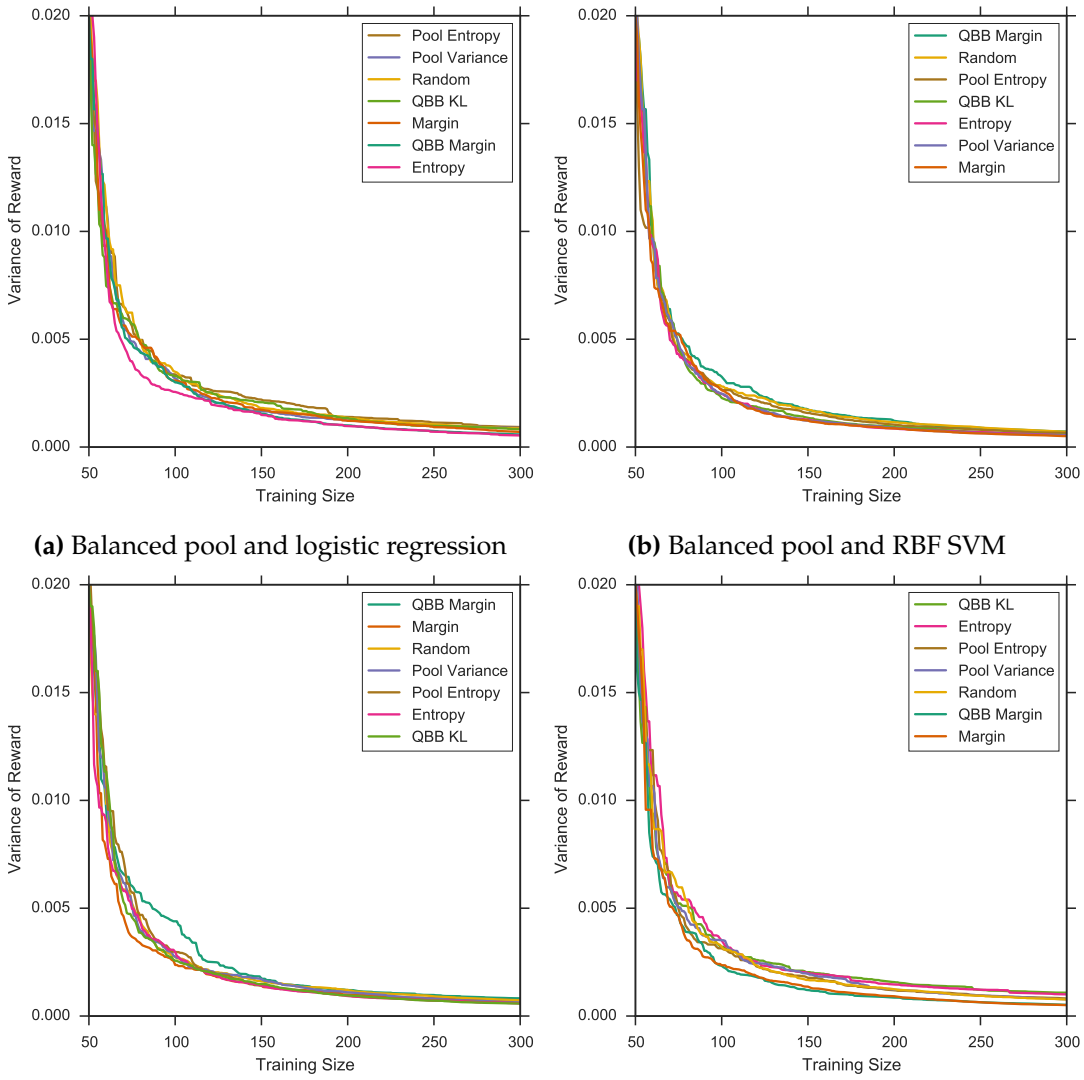


(b) Recall improvement map of stars.



(c) Recall improvement map of quasars.

Figure C.4: Recall improvement maps of when the W14 extinction vector is used.



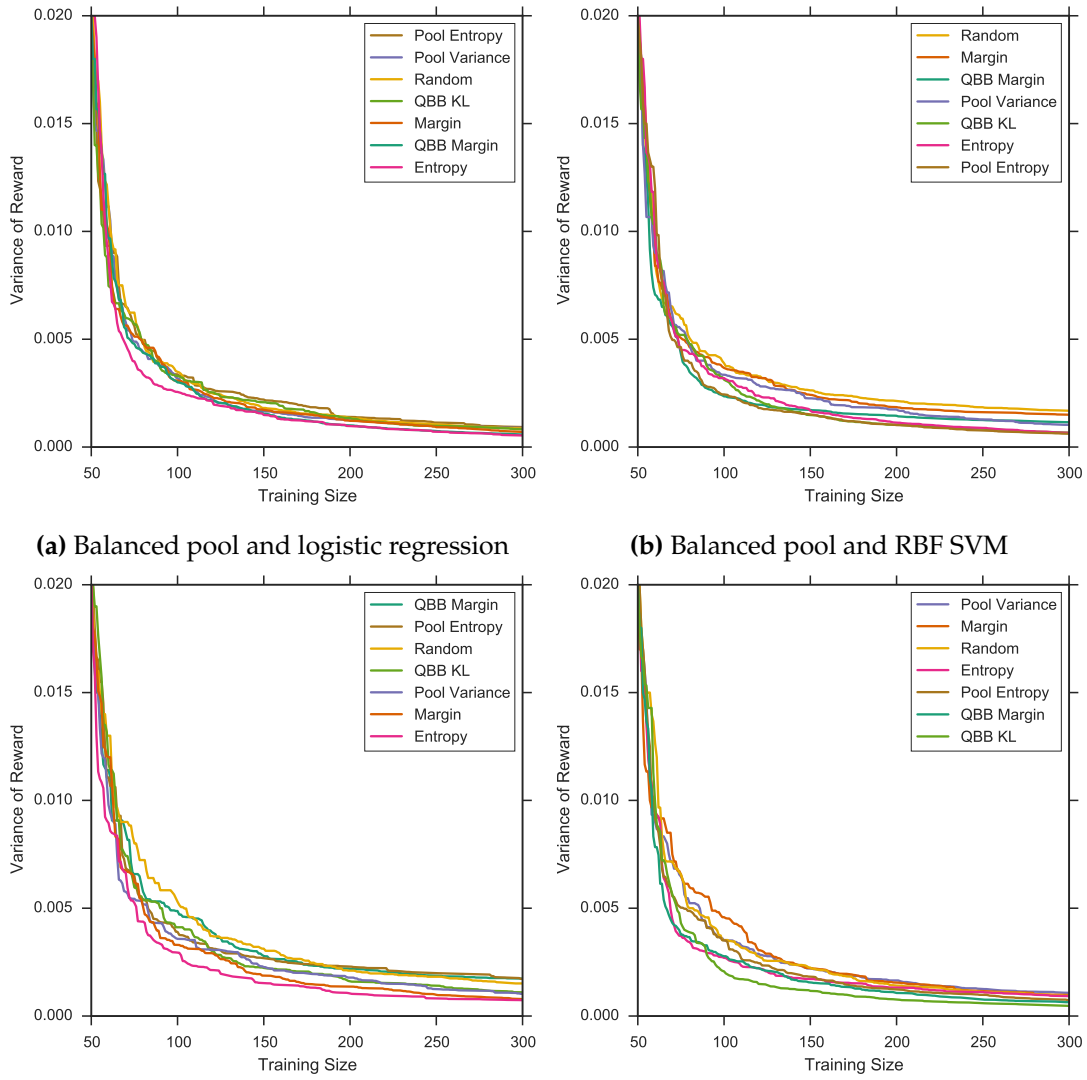
(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure C.5: Variance (average of 10 trials) of the expected reward in Thompson sampling with the SDSS dataset.



(a) Balanced pool and logistic regression

(b) Balanced pool and RBF SVM

(c) Unbalanced pool and logistic regression

(d) Unbalanced pool and RBF SVM

Figure C.6: Variance (average of 10 trials) of the expected reward in Thompson sampling with the VST ATLAS dataset.

Bibliography

ALAM, S., ALBARETI, F. D., ALLENDE PRIETO, C., ANDERS, F., ANDERSON, S. F., ANDERTON, T., ANDREWS, B. H., ARMENGAUD, E., AUBOURG, É., BAILEY, S., AND ET AL. 2015. The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III. *The Astrophysical Journal Supplement Series 219*, 12. (p. 7)

ANGLUIN, D. 1988. Queries and concept learning. *Machine learning 2*, 4, 319–342. (p. 15)

AUER, P., CESA-BIANCHI, N., AND FISCHER, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn. 47*, 2-3 (May), 235–256. (p. 22)

BAZELL, D., MILLER, D. J., AND SUBBARAO, M. 2006. Objective subclass determination of sloan digital sky survey unknown spectral objects. *Astrophys. J. 649*, 678–691. (p. 59)

BLANTON, M. R., DALCANTON, J., EISENSTEIN, D., LOVEDAY, J., STRAUSS, M. A., SUBBARAO, M., WEINBERG, D. H., ANDERSON, J. E., JR., ANNIS, J., BAHCALL, N. A., AND ET AL. 2001. The Luminosity Function of Galaxies in SDSS Commissioning Data. *The Astronomical Journal 121*, 2358–2380. (p. 4)

BOSER, B. E., GUYON, I. M., AND VAPNIK, V. N. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92 (New York, NY, USA, 1992), pp. 144–152. ACM. (p. 14)

BREIMAN, L. 1996. Bagging predictors. *Machine learning 24*, 2, 123–140. (pp. 13, 43)

BREIMAN, L. 2001. Random forests. *Machine learning 45*, 1, 5–32. (p. 13)

BRODERSEN, K. H., ONG, C. S., STEPHAN, K. E., AND BUHMANN, J. M. 2010. The balanced accuracy and its posterior distribution. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10 (Washington, DC, USA, 2010), pp. 3121–3124. IEEE Computer Society. (p. 26)

CHAPELLE, O. AND LI, L. 2011. An empirical evaluation of thompson sampling. In J. SHAWE-TAYLOR, R. ZEMEL, P. BARTLETT, F. PEREIRA, AND K. WEINBERGER Eds., *Advances in Neural Information Processing Systems 24*, pp. 2249–2257. Curran Associates, Inc. (p. 22)

COHN, D., ATLAS, L., AND LADNER, R. 1994. Improving generalization with active learning. *Machine Learning 15*, 2, 201–221. (p. 16)

COX, D. R. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological) 20*, 2, pp. 215–242. (p. 13)

CRAMMER, K. AND SINGER, Y. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research 2*, 265–292. (p. 15)

ELTING, C., BAILER-JONES, C. A. L., AND SMITH, K. W. 2008. Photometric Classification of Stars, Galaxies and Quasars in the Sloan Digital Sky Survey DR6 Using Support Vector Machines. In C. A. L. BAILER-JONES Ed., *American Institute of Physics Conference Series*, Volume 1082 of *American Institute of Physics Conference Series* (Dec. 2008), pp. 9–14. (pp. 14, 59)

FREUND, Y., SEUNG, H. S., SHAMIR, E., AND TISHBY, N. 1997. Selective sampling using the query by committee algorithm. *Machine learning 28*, 2-3, 133–168. (p. 21)

GUPTA, N., GRANMO, O.-C., AND AGRAWALA, A. 2011. Thompson sampling for dynamic multi-armed bandits. In *Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops - Volume 01*, ICMLA '11 (Washington, DC, USA, 2011), pp. 484–489. IEEE Computer Society. (p. 25)

HÁLA, P. 2014. Spectral classification using convolutional neural networks. (pp. 11, 59)

HO, T. K. 1998. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 8 (Aug), 832–844.

LEWIS, D. D. AND GALE, W. A. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (1994), pp. 3–12. Springer-Verlag New York, Inc. (pp. 16, 17)

LOUPPE, G. AND GEURTS, P. 2012. Ensembles on random patches. In *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*, ECML PKDD'12 (Berlin, Heidelberg, 2012), pp. 346–361. Springer-Verlag.

LUPTON, R., GUNN, J. E., AND SZALAY, A. 1999. A modified magnitude system that produces well-behaved magnitudes, colors, and errors even for low signal-to-noise ratio measurements. *Astron. J. 118*, 1406. (p. 5)

MACKAY, D. J. 1991. *Bayesian methods for adaptive models*. PhD thesis, California Institute of Technology. (p. 20)

MCCALLUM, A. AND NIGAM, K. 1998. Employing EM and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98 (San Francisco, CA, USA, 1998), pp. 350–358. Morgan Kaufmann Publishers Inc. (p. 18)

MELVILLE, P. AND MOONEY, R. J. 2004. Diverse ensembles for active learning. In *Proceedings of the twenty-first international conference on Machine learning* (2004), pp. 74. ACM. (pp. 17, 18)

PALMER, J. AND DAVENHALL, A. C. 2001. *The CCD Photometric Calibration Cookbook*. Council for the Central Laboratory of the Research Councils. (p. 5)

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830. (p. 12)

ROMANISHIN, W. 2002. *An Introduction to Astronomical Photometry Using CCDs*. University of Oklahoma. (p. 3)

SCHEFFER, T., DECOMAIN, C., AND WROBEL, S. 2001. Active hidden markov models for information extraction. In F. HOFFMANN, D. HAND, N. ADAMS, D. FISHER, AND G. GUIMARAES Eds., *Advances in Intelligent Data Analysis*, Volume 2189 of *Lecture Notes in Computer Science*, pp. 309–318. Springer Berlin Heidelberg. (p. 17)

SCHEIN, A. I. AND UNGAR, L. H. 2007. Active learning for logistic regression: An evaluation. *Mach. Learn. 68*, 3 (Oct.), 235–265. (pp. 19, 59)

SCHLAFLY, E. F. AND FINKBEINER, D. P. 2011. Measuring Reddening with Sloan Digital Sky Survey Stellar Spectra and Recalibrating SFD. *The Astrophysical Journal 737*, 103. (pp. 9, 63)

SCHLEGEL, D. J., FINKBEINER, D. P., AND DAVIS, M. 1998. Maps of Dust Infrared Emission for Use in Estimation of Reddening and Cosmic Microwave Background Radiation Foregrounds. *The Astrophysical Journal 500*, 525–553. (p. 9)

SETTLES, B. AND CRAVEN, M. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing* (2008), pp. 1070–1079. Association for Computational Linguistics. (p. 59)

SHANKS, T., METCALFE, N., CHEHADE, B., FINDLAY, J. R., IRWIN, M. J., GONZALEZ-SOLARES, E., LEWIS, J. R., YOLDAS, A. K., MANN, R. G., READ, M. A., SUTORIUS, E. T. W., AND VOUTSINAS, S. 2015. The VLT Survey Telescope ATLAS. *Monthly Notices of the Royal Astronomical Society 451*, 4238–4252. (p. 9)

SHANNON, C. E. 1948. A mathematical theory of communication. *Bell System Technical Journal 27*, 3, 379–423. (p. 17)

SPARK, L. S. AND GALLAGHER, J. S. 2007. *Galaxies in the Universe: An Introduction*. Cambridge University Press, Cambridge, UK. (p. 6)

THOMPSON, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika 25*, 3/4, pp. 285–294. (p. 22)

TIBSHIRANI, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288. (p. 14)

TONG, S. AND KOLLER, D. 2002. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research 2*, 45–66. (p. 59)

WOLF, C. 2014. Milky Way dust extinction measured with QSOs. *Monthly Notices of the Royal Astronomical Society 445*, 4252–4258. (pp. 9, 63)

Index