Learning from Crowd Labels to find Black Holes

Matthew Alger

A thesis submitted in partial fulfillment of the degree of
Bachelor of Science (Advanced) (Honours) at
The Research School of Computer Science
The Australian National University

October 2016

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

Matthew Alger
27 October 2016

Acknowledgements

I would like to thank my supervisor, Dr Cheng Soon Ong, for his invaluable guidance, advice, and commentary throughout the Honours year. I would also like to thank Dr Julie Banfield for her help with both the thesis and project itself, and for answering all of my many astronomy questions. Finally, I would like to thank Jacob Buete, Damon Binder, and Rachel Alger, who gave up their time to read and comment upon the draft of this thesis.

Abstract

In this thesis we present a new supervised learning approach to the astronomical problem of radio cross-identification. We focus in particular on training our methods using crowdsourced data from the Radio Galaxy Zoo, a citizen science website that allows volunteers to cross-identify radio objects.

Cross-identification is the problem of matching of objects detected at one wavelength with objects detected at another. This is a particularly difficult problem for radio, as a large number of radio objects are active galactic nuclei, supermassive black holes at the centre of galaxies emitting huge radio jets. These jets can sprawl across the sky in complex ways as they interact with their environment and in general have no clear relationship with their host galaxy.

In this thesis we have cast the radio cross-identification problem into a machine learning context, framing it as a object localisation problem that can be solved using binary classification. Using this framework, we then classified galaxies in the Chandra Deep Field South according to whether they contain an active galactic nucleus. We used a combination of image features and astronomical features to represent each galaxy. Image features were extracted from images of the radio sky using a convolutional neural network.

We trained the classifier using label data from Radio Galaxy Zoo. We found that a classifier trained on these non-expert labels performs similarly to a classifier trained on expert labels, attaining balanced accuracies of $(87.17 \pm 0.90)\%$ and $(88.74 \pm 0.77)\%$ respectively.

We investigated multiple ways of handling noise and redundancy in the crowdsourced labels, and applied the classification model from the Raykar et al. (2010) paper *Learning From Crowds* to the Radio Galaxy Zoo labels. Comparing this to a simple classification model trained on the majority vote, we found that the majority vote approach obtained the highest classification accuracy.

Finally, we investigated applications of active learning to the radio cross-identification problem, with a focus on the Radio Galaxy Zoo project. We applied query-by-committee to the radio cross-identification problem, finding that query-by-committee outperforms random selection (but not if the random selection is class-balanced). We then suggested an experiment to investigate applications of active learning to Radio Galaxy Zoo, and highlighted problems with current active learning literature when applied to citizen science.

Contents

Chapter 1

Introduction

Radio images of distant supermassive black holes offer unique insight into the inner workings of galaxies and their environment. Newer, larger, and better radio surveys will allow astronomers to learn more than ever before about these objects. The galaxies in which black holes are found do not appear in radio surveys (instead appearing in infrared surveys), but to fully investigate them, astronomers need information about both the black hole and its host galaxy. This leads to a difficult problem: How do we match our radio observations of supermassive black holes to infrared observations of their host galaxies?

An object emits light at different wavelengths depending on its physical properties. As such, looking at the sky in different wavelengths gives us very different pictures. For example, Figure 1.1 shows Centaurus A imaged in optical, infrared, X-ray, and radio light. The disc of the galaxy is only visible in optical and infrared, and the large jets emitted from the centre of the galaxy are only visible in X-ray and radio. This shows that to fully understand an astronomical object, we have to look at it in many different wavelengths.

While it is possible to look at one object in many wavelengths at once, we can only do so for one object at a time. However, most astronomical data comes from *surveys*, where a telescope images large areas of the sky at once in specific wavelengths. These surveys detect many objects, and individually observing all of them is impractical.

Figure 1.1: Centaurus A in different wavelengths. From left to right: optical, infrared, X-ray, and radio. *Image: ESO/WFI/M.Rejkuba et al. (Optical); NASA (Infrared); NASA/CXC/U.Birmingham/M.Burke et al. (X-ray); NSF/VLA/Univ.Hertfordshire/M.Hardcastle (Radio)*

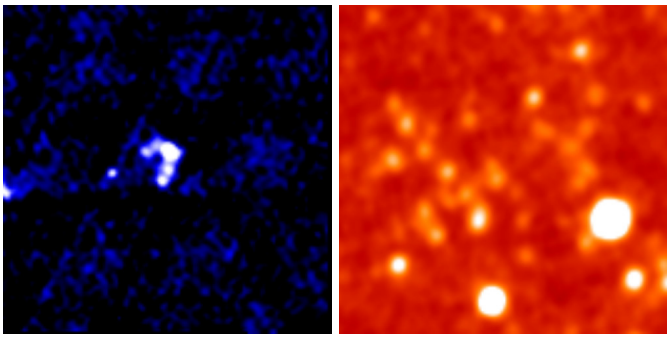


Figure 1.2: The same patch of sky (8h 15m 9.0s +27° 3′ 37″) imaged in radio (left) and infrared (right). *Image: NRAO VLA (Radio); WISE (Infrared)*

This leads us to the problem of *cross-identification*: Given an object detected at one wavelength, what is the corresponding object at another wavelength?

At first, this might sound easy. We could simply take note of the position of the object in an image at one wavelength, and then match it to the same position in an image at the other wavelength. This, however, assumes that objects appear at the same position in multiple wavelengths. While this is true for point-like objects, it is not true in general. Objects that are not point-like may stretch across the sky, and only parts of the object may emit each wavelength. We can again look to Centaurus A (Figure 1.1) as an example: The brightest part of Centaurus A in infrared is far removed from the brightest parts in radio. When objects are very distant, cross-identification becomes even harder. Figure 1.2 shows the same patch of sky in both radio and infrared. The infrared image shows no obvious counterpart for the object visible in the radio image.

The cross-identification problem is compounded by the sheer size of the universe. The scale of modern astronomical surveys is enormous: Astronomers have catalogued millions of radio objects [1] and hundreds of millions of infrared objects [2]. This makes manual cross-identification impossible. Automated cross-identification algorithms exist [3, 4], but are expected to fail for upcoming radio surveys [1].

The Evolutionary Map of the Universe (EMU) [5] is one such survey. EMU will use the new Australian SKA Pathfinder (ASKAP) telescope to image 75% of the sky in radio wavelengths, and is expected to find over 70 million radio galaxies — around 30 times the number of radio galaxies we know about today [1]! These objects will need to be cross-identified with their infrared counterparts, but it is estimated that 10% of these radio galaxies will be too complicated for current automated cross-identification algorithms [1, 5].

Radio Galaxy Zoo (RGZ) [1] is a citizen science project that attempts to crowdsource the cross-identification problem. Volunteers are presented with an image of the sky in both radio and infrared, and are asked to identify radio objects and associate them with the corresponding infrared object. The cross-identification interface is available online, so anyone can help cross-identify radio objects[1].

[1] https://radio.galaxyzoo.org/

RGZ is based on the highly successful Galaxy Zoo [6, 7]. The Zooniverse platform[2] created by Galaxy Zoo provides a way for non-experts to help researchers label data across a wide range of scientific fields, and has resulted in well over 110 publications[3].

To date, with the help of thousands of volunteers, RGZ has managed to cross-identify over 100 000 radio galaxies. While this does not compare in scale to EMU, the hope is that these cross-identifications can be used to train next-generation machine learning algorithms.

1.1 Contributions

We present in this thesis a supervised learning approach to the problem of radio cross-identification, using label data sourced from the Radio Galaxy Zoo. Our approach is to frame the radio cross-identification problem as object localisation, and then as binary classification, allowing standard machine learning techniques to be applied to the problem. This is the first application of machine learning to automated radio cross-identification that we are aware of.

Our methods do not make use of any astronomical models. We believe that this will help in the development of future algorithms for cross-identification of objects detected in EMU, where astronomical models may fail with the discovery of new classes of radio object. Instead of astronomical models, we use features extracted automatically from radio images using a convolutional neural network (Section 4.4.2). To our knowledge, this is the first instance where features have been automatically extracted from radio images.

We have made use of the Radio Galaxy Zoo cross-identification database [1]. This is a very recent data set, and our work is one of the first uses of this data. In particular, this work is the first application of machine learning to labels from Radio Galaxy Zoo.

In the process of developing our cross-identification methods, we have investigated the predictive power of infrared flux ratios, which are believed to correlate with whether a galaxy contains an AGN (Section 4.4.3) [1]. Further, we have shown that the features we have extracted from radio images are better predictors.

We have implemented two prominent crowd learning algorithms by Raykar et al. [8] and Yan et al. [9], and compared their performance against a simple crowd learning algorithm (logistic regression with majority vote, Sections 3.5.2 and 3.5.3). Our implementation is MIT-licensed, and is the only open-source implementation that we are aware of.

In Chapter 5 we have highlighted some problems with applying existing active learning and crowd learning literature to citizen science.

Finally, we have implemented an open-source Python library for crowd-based learning in astronomy. This library, crowdastro, contains all code used in this thesis, and can be used to easily reproduce our experiments. In particular, crowdastro

[2] https://zooniverse.org/

[3] See https://www.zooniverse.org/about/publications for a full list.

contains a pipeline for easily training and running our cross-identification methods. The library is described in Appendix A and is available on GitHub[4].

1.2 Outline

Chapter 2 introduces astronomical concepts such as astronomical surveys, radio active galactic nuclei, and radio cross-identification. These concepts are important to understand both the purpose of this thesis, and to understand the problem we are trying to solve. We also introduce four astronomical surveys — EMU, ATLAS, WISE, and SWIRE — that produced the data we used in our experiments.

Chapter 3 introduces machine learning concepts such as classification and image feature extraction. These concepts are the building blocks for a machine-learned algorithm for automated radio cross-identification. We also perform some experiments to test how selected crowd learning algorithms behave in different contexts.

Chapter 4 brings together Chapters 2 and 3 to develop a machine learning approach to the radio cross-identification task. We formalise the problem, highlight the problems that must be solved for development, trial different methods of approaching the task, and present results on task performance using our classifier.

Chapter 5 discusses active learning and its application to crowdsourced projects like the Radio Galaxy Zoo. We perform some simple experiments and suggest future pathways for research in this area.

Appendix A describes crowdastro, a Python package developed as part of this project. This package was used to perform all experiments described in this thesis.

1.3 Data Products and Packages Used in this Thesis

The work presented in this thesis would not have been possible without the various data products used for experiments and development.

In our experiments in Chapters 4 and 5, we make use of data products from the Wide-field Infrared Survey Explorer (WISE), the Spitzer Space Telescope, and the Australia Telescope Compact Array.

WISE is a joint project of the University of California, Los Angeles, and the Jet Propulsion Laboratory/California Institute of Technology, funded by the National Aeronautics and Space Administration.

The Spitzer Space Telescope is operated by the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA. The Spitzer Wide-area Infrared Extragalactic survey was supported by NASA through the Spitzer Legacy Program under contract 1407 with the Jet Propulsion Laboratory.

The Australia Telescope Compact Array is part of the Australia Telescope, which is funded by the Commonwealth of Australia for operation as a National Facility managed by CSIRO.

[4] https://github.com/chengsoonong/crowdastro

Our experiments in Chapter 3 use the Breast Cancer Wisconsin dataset. The dataset was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg, and was accessed from the UCI Machine Learning Repository [5].

For running our experiments and developing our machine learning methods, we made use of astropy [10], a community-developed core Python package for astronomy, and scikit-learn [11], an open source machine learning package.

Finally, this thesis had been made possible by the participation of more then 10 000 volunteers in the Radio Galaxy Zoo project. Their contributions are individually acknowledged at `http://rgzauthors.galaxyzoo.org`.

1.4 Glossary of Abbreviations

- AGN: Active galactic nucleus (Section 2.2)
- AL: Active learning (Chapter 5)
- ATLAS: Australia Telescope Large Area Survey (Section 2.4.2)
- CDFS: Chandra Deep Field South (Section 2.3.2)
- Dec: Declination (Section 2.1)
- ELAIS-S1: European Large Area ISO Survey - South 1 (Section 2.3.2)
- EMU: Evolutionary Map of the Universe (Section 2.4.1)
- FIRST: Faint Images of the Radio Sky at Twenty-Centimeters
- LR: Logistic regression (Section 3.1.2)
- MV: Majority vote (Section 3.5.1)
- QBC: Query-by-committee (Section 5.2.2)
- RA: Right ascension (Section 2.1)
- RF: Random forests (Section 3.1.4)
- RGZ: Radio Galaxy Zoo (Section 2.6)
- SWIRE: Spitzer Wide-area Infrared Extragalactic Survey (Section 2.3.2)
- WISE: Wide-field Infrared Survey Explorer (Section 2.3.1)

[5] `https://archive.ics.uci.edu/ml/`

Chapter 2

Galaxies and Active Galactic Nuclei

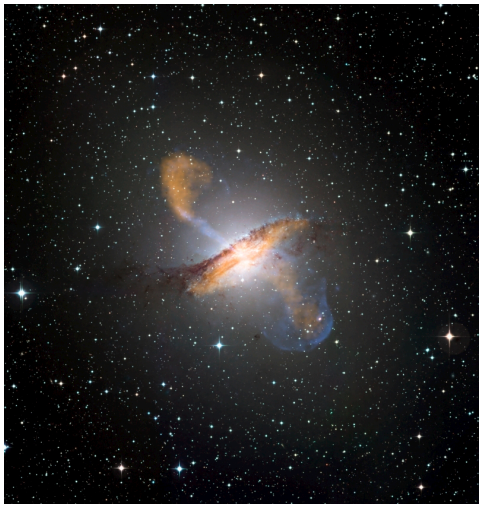


Figure 2.1: Centaurus A, a nearby radio galaxy with an active galactic nucleus. *Image: ESO/WFI (Optical); MPIfR/ESO/APEX/A.Weiss et al. (Submillimetre); NASA/CXC/CfA/R.Kraft et al. (X-ray)*

This chapter develops the problem of cross-identification in an astronomical context. The outcome of the chapter is a solid understanding of cross-identification, as well as knowledge of four key astronomical datasets. These datasets will be used for developing our machine learning approach to cross-identification in Chapter 4.

Section 2.1 will introduce concepts required to understand astronomical surveys. We will look at the equatorial coordinate system, used for describing positions of objects on the sky, and we will describe some basic properties and measurements of light used in surveys.

Section 2.2 will briefly discuss the properties and features of radio active galactic nuclei. Radio active galactic nuclei are radio objects that are extremely common in radio surveys. They are also difficult to cross-identify, and as such pose great difficulty to cross-identification algorithms.

Sections 2.3 and 2.4 will introduce specific infrared and radio surveys, respectively. These surveys form the datasets we will use in Chapter 4, and are also used later in

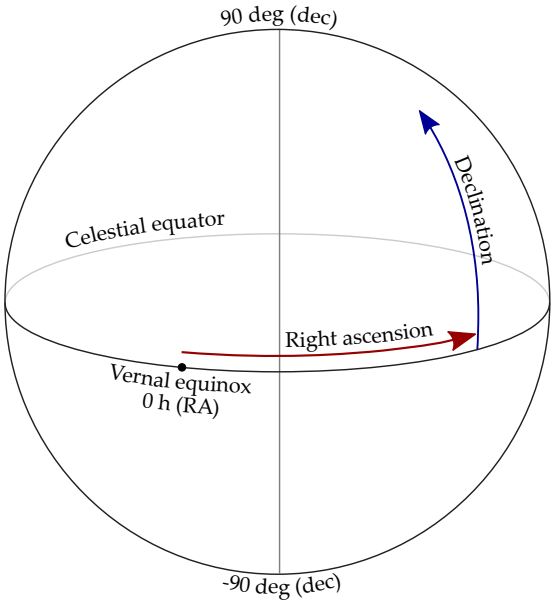


Figure 2.2: The equatorial coordinate system used in astronomy.

this chapter for existing cross-identification catalogues.

Section 2.5 discusses the problem of radio cross-identification, and introduces two catalogues of cross-identified objects that we will refer to for testing our cross-identification methods.

Finally, Section 2.6 describes the Radio Galaxy Zoo, the source of the training data for our machine learning methods.

2.1 Astronomical Observations

2.1.1 Astronomical Coordinates

Astronomy uses the *equatorial coordinate system* to describe the positions of objects on the sky. Each position on the sky is described by two numbers: the *right ascension* (RA) and the *declination* (dec).

The right ascension of an object is the angle eastward from the vernal equinox to the object along the celestial equator. It is measured in hours (h), minutes (min), and seconds (s). There are 60 seconds in 1 minute, 60 minutes in 1 hour, and 1 hour is equal to 15 degrees. Right ascension ranges between 0h and 24h, where both 0h and 24h are located at the vernal equinox. The declination of an object is the angle northward from the celestial equator to the object. It is measured in degrees (°), arcminutes (′), and arcseconds (″). Declination ranges between $-90°$ and $90°$, where $-90°$ is the declination of the south celestial pole and $90°$ is the declination of the north celestial pole. The right ascension and declination are shown in Figure 2.2. It is important to note that while right ascension and declination are both measured in minutes and seconds, these are *different* minutes and seconds. 1 minute (right ascension) is equal to 15

arcminutes (declination).

Objects are usually given a International Astronomical Union (IAU) name based on their coordinates. This name includes the catalogue that identified the object, the right ascension, and the declination. For example, ATLAS3 J002925.7-440256C is from the third release of the ATLAS catalogue, and is located at $00\text{h}\ 29\text{m}\ 25.7\text{s}\ -44^\circ\ 02'\ 56''$.

2.1.2 Wavelength and Frequency

Light is an electromagnetic wave and can thus be characterised by its *wavelength* and *frequency* [12]. The wavelength of light is the distance between two neighbouring peaks; it is measured in metres (m) and usually denoted λ. The frequency of light is the number of waves that have passed a point per second; it is measured in hertz (Hz) and usually denoted ν. Wavelength and frequency are related by the formula

$$c = \lambda\nu,$$

where c is the speed of light.

Humans can see a limited range of wavelengths of light. In this range, the wavelength of light corresponds to its colour: Blue light has shorter wavelength (and higher frequency) than red light. Outside of this range, light still has various different "colours", but we cannot see them. Different ranges of wavelength are assigned different names, such as infrared, x-ray, and radio.

When objects emit light, the wavelength depends on the process by which the light was emitted. For example, thermal radiation is emitted by all objects based entirely on the temperature of the object, with hotter objects emitting light with shorter wavelengths. Another example is synchrotron radiation (Section 2.2), which is emitted in radio wavelengths.

Since telescopes generally only detect brightnesses of objects in the sky, and different wavelengths of light require different mechanisms to detect, telescopes are designed to measure light only at specific ranges of wavelengths. To gain full scientific insight into objects detected in different wavelengths, the objects must be cross-identified with each other.

2.1.3 Flux and Magnitude

The *flux density* of an object is its energy output per unit time per unit area. It is denoted f and is measured in W m^{-2}. The flux density is given by

$$f = \frac{1}{A}\frac{\mathrm{d}E}{\mathrm{d}t}$$

where E is the energy received from the object, t is the time, and A is the apparent area of the object. We cannot usually measure the flux over all frequencies, so flux is observed over a specific frequency range $\Delta\nu$. The *spectral flux density* is then the limit

as the frequency range approaches zero, i.e.

$$f_\nu = \frac{1}{A}\frac{\mathrm{d}^2 E}{\mathrm{d}\nu \mathrm{d}t}.$$

The spectral flux density is measured in janskys (Jy), an astronomical unit equal to 10^{-26} W m^{-2} Hz^{-1} [13].

The *apparent magnitude*, m, of an object is a logarithmic measure of its flux seen from Earth, relative to the star Vega [13]:

$$m = -2.5 \log_{10}\left(\frac{f}{f_{\text{Vega}}}\right). \tag{2.1}$$

f is the flux density of the object and f_{Vega} is the flux density of Vega measured at the same frequency.

The difference between the magnitudes of two objects, m_1 and m_2, represents the logarithm of the ratio of their flux densities:

$$m_2 - m_1 = -2.5 \log_{10}\left(\frac{f_2}{f_1}\right). \tag{2.2}$$

2.2 Radio Active Galactic Nuclei

Figure 2.3: An artist's impression of the accretion disk of an active galactic nucleus. *Image: NASA/JPL-Caltech*

Many galaxies contain a supermassive black hole in their centre [14]. These black holes may accrete matter from the surrounding galaxy into an *accretion disc* (Figure 2.3). The accretion process emits huge amounts of light through different physical processes. These light-emitting black holes are called active galactic nuclei (AGNs). AGNs can be extremely bright, emitting up to 10^{39} J of energy every second — nearly a thousand times more energy than our entire galaxy emits [15]. AGNs are found throughout the universe, with the closest known AGN being Centaurus A (Figure 2.1) at a distance of around 1.2×10^{20} km [16].

Around 10% of AGNs produce *jets* from their accretion disk [17]. Jets are long, thin

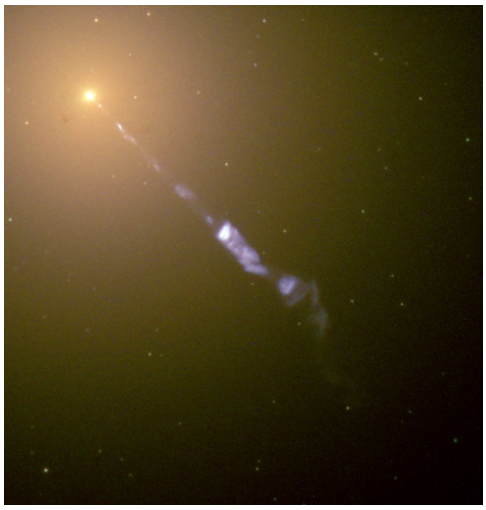


Figure 2.4: M87, a giant elliptical galaxy with a jet. *Image: NASA and The Hubble Heritage Team (STScI/AURA)*

streams of matter such as the one shown in Figure 2.4. These jets can be very large, with "giant" AGNs emitting jets nearly 1 Mpc (3×10^{19} km) in length [18].

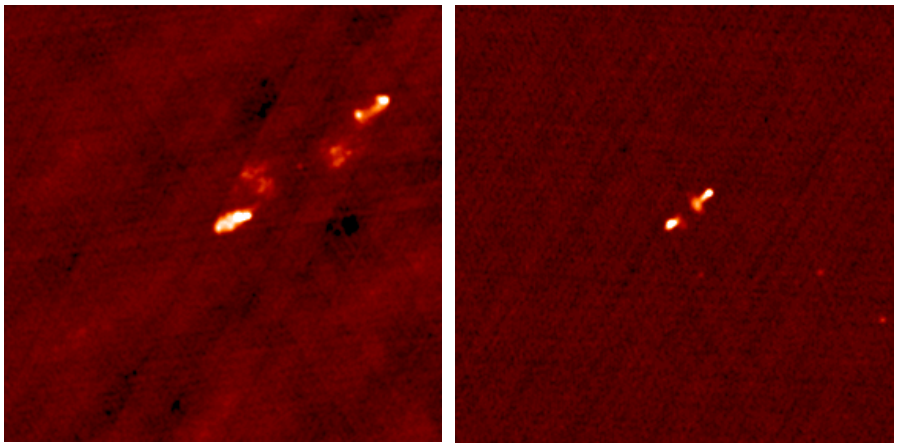
Electrons in jets produce *synchrotron radiation*. This is a form of radiation emitted by charged particles travelling at relativistic speeds as they accelerate in a magnetic field [19]. Synchrotron radiation is emitted in radio wavelengths, and so AGNs emitting synchrotron radiation are called *radio AGNs*. As radio AGNs are the focus of this work, "AGN" will henceforth refer only to radio AGNs unless otherwise specified.

We can observe the jets of radio AGNs with radio telescopes, such as in Figure 2.5. The jets may appear to be separate objects, and there may or may not be a radio source in the middle of the AGN system. If the system is composed of two separate objects such as in Figure 2.5b (one object for each lobe of the jets), then it is called a *radio double*; if a central source is also visible, then it is called a *radio triple*.

Since these jets are not point-like sources of light, and viewing the jets is the only way to observe radio AGNs, common astronomical problems such as cross-identification (Section 2.5) can be very difficult, as the true location of the AGN is unclear.

2.3 Infrared Surveys

A *survey* is an image of the sky made with repeated observations in specific wavelengths, aiming to comprehensively cover some large area of the sky. In this section we look at WISE and SWIRE, two surveys in infrared wavelengths from which we draw data for our experiments.



(a) FIRSTJ093244.5+161050. (b) FIRSTJ130117.5+104121.

Figure 2.5: Two radio AGNs imaged by the NRAO Very Large Array.

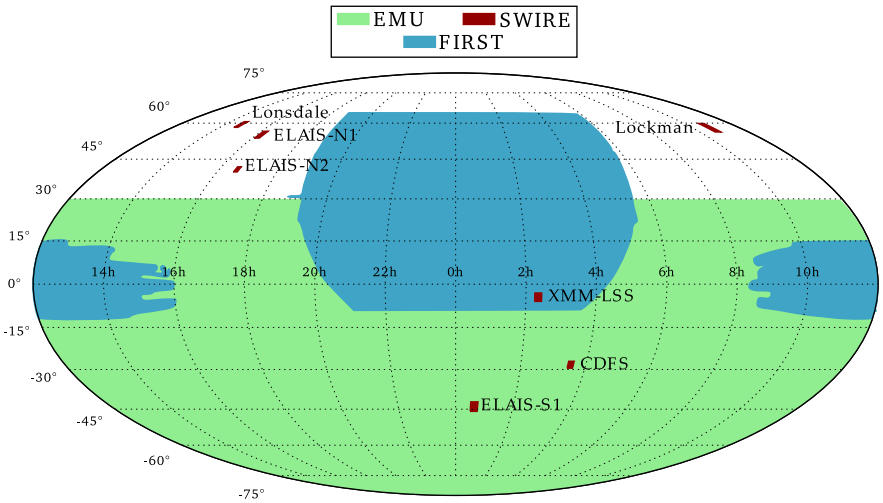


Figure 2.6: A map of the sky, showing the FIRST and EMU radio surveys, and the SWIRE infrared survey. The ATLAS radio survey covers both the CDFS and ELAIS-S1 fields. The WISE infrared survey covers the entire sky.

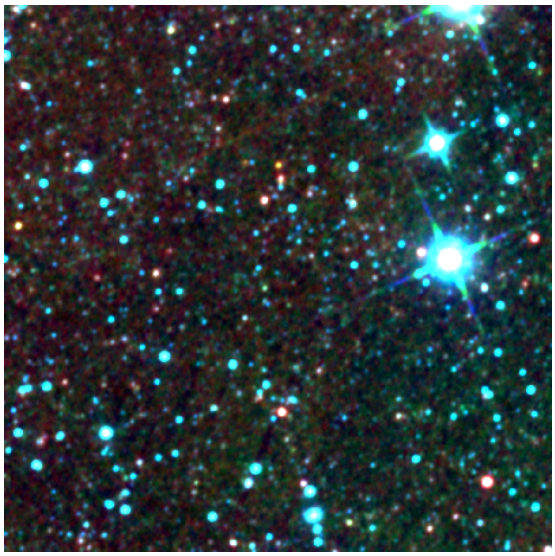


Figure 2.7: Patch of the WISE multi-wavelength composite image centred on 3h30m05.24s -28d34m46.3s.

2.3.1 WISE: Wide-field Infrared Survey Explorer

The Wide-field Infrared Survey Explorer (WISE) is an orbital infrared telescope. In 2009–2010 it was used to survey the entire sky in four wavelengths: 3.4 μm, 4.6 μm, 12 μm, and 22 μm. These wavelengths are referred to as WISE bands $w1$, $w2$, $w3$, and $w4$, respectively. WISE images have resolutions of $6''$–$12''$, with sensitivity between 0.08 and 6 mJy, corresponding to the detection of sources between 16.5 and 7.9 magnitude [20].

The AllWISE catalogue [2] is the most recent WISE catalogue. For each object detected, the catalogue includes the magnitudes in each WISE band, as well as a number of other features we do not make use of in our methods. We will refer to the magnitudes in each band by the names of the bands (i.e. $w1$ – $w4$).

The main goal of WISE was to provide a map of the whole sky in infrared wavelengths for many different reasons: infrared measurements may be used to detect and classify distant galaxies, measurements can be cross-identified to complement other surveys, and so on. There are many other scientific goals of WISE; these are described in detail by Wright et al. [20].

2.3.2 SWIRE: Spitzer Wide-area Infrared Extragalactic Survey

The Spitzer Wide-area Infrared Extragalactic Survey (SWIRE) is a multi-wavelength infrared survey [21]. It observed at four wavelengths: 3.6 μm, 4.5 μm, 5.8 μm, and 8.0 μm.

SWIRE surveyed seven fields. These fields — ELAIS-S1, ELAIS-N1, ELAIS-N2, Lockman, CDFS, XMM-LSS, and Lonsdale — are called the *SWIRE fields*, totalling 63.2 square degrees in area. The fields contain few nearby objects, meaning that observa-

Figure 2.8: Patch of the SWIRE multi-wavelength composite image centred on 3h30m05.24s -28d34m46.3s. This is the same region of the sky as the WISE image in Figure 2.7.

tions in these fields are of very old, distant objects. ELAIS-S1 and CDFS are in the southern sky; all other fields are in the northern sky.

While SWIRE covers far less area of the sky than WISE, it does so at considerably higher resolution and sensitivity: $1.2''$ and 7.3–32.5 μJy, respectively [22, 23].

SWIRE aimed to investigate the evolution of galaxies and AGNs and the relationship between galaxies and AGNs [23].

2.4 Radio Surveys

In this section we look at the EMU and ATLAS radio surveys. EMU does not yet exist, but motivates our work. We will use ATLAS for the experiments in later chapters.

2.4.1 EMU: Evolutionary Map of the Universe

The Evolutionary Map of the Universe (EMU) is an upcoming deep radio survey that aims to provide both high sensitivity and wide coverage of the radio sky [5]. EMU will be sensitive to objects to around 0.015 mJy with an angular resolution of around $10''$, and cover the entire southern sky as far north as $30°$. For comparison, the largest existing radio survey is currently NVSS [24] with sensitivity of 2.5 mJy and a resolution of $45''$, which covers a similar area. With the resolution, sensitivity, and scale of EMU, astronomers hope to investigate galactic evolution, to explore large-scale structure and cosmology of the universe, and to find never-before-seen astronomical objects.

EMU is expected to find huge numbers of radio objects — while the Australia Telescope Large Area Survey [ATLAS] (which has similar resolution and sensitivity to EMU) has detected around 4000 radio objects, EMU is expected to find *70 million*

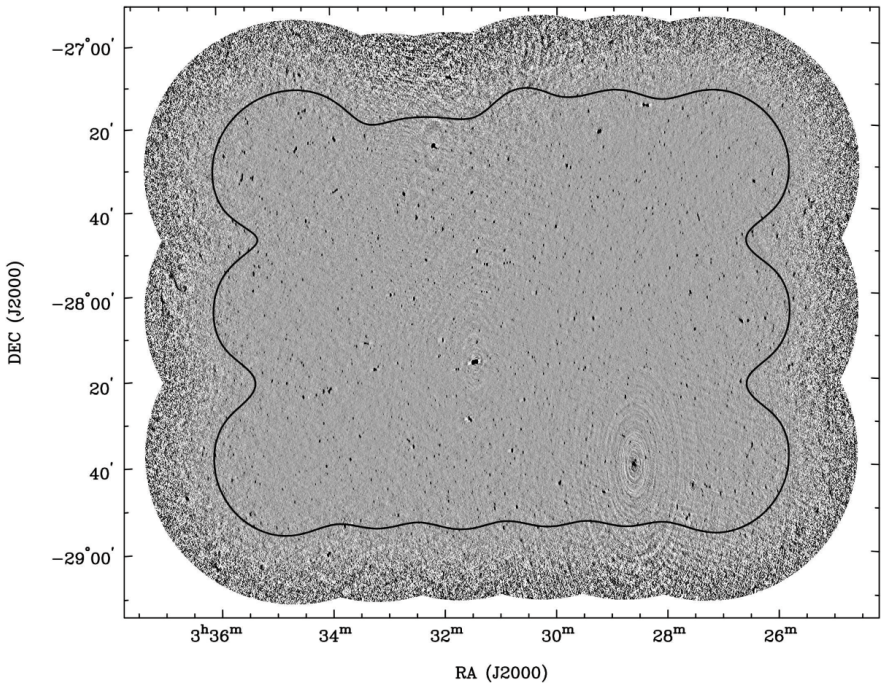


Figure 2.9: ATLAS observations of CDFS. *Reproduced from Franzen et al. [25].*

.

[1]. With such a large number of detected objects, analysis of the data from EMU will be considerably more difficult than analysis of existing surveys. This analysis will be impossible by hand, and will therefore require algorithms to process the data. With so many new objects it is likely that many objects will not fit existing models, meaning that model-based approaches to data processing may be ineffective. Norris et al. [5] estimated that 10% of the newly-found objects will be too complicated for current automated algorithms [1]. It is this problem that motivates the development of new, machine-learned algorithms for processing astronomical data at these large scales. While the EMU survey data is not yet released, development of such algorithms can begin by looking at other datasets with similar sensitivity and resolution, such as ATLAS (Section 2.4.2).

The radio objects found by EMU will eventually be cross-identified with the infrared objects found by WISE (Section 2.3.1). As such, any algorithms that we develop must work with WISE data.

2.4.2 ATLAS: The Australia Telescope Large Area Survey

The Australia Telescope Large Area Survey (ATLAS) is a high sensitivity radio survey which aims to help understand the evolution of early galaxies [26]. The Australia Telescope Compact Array was used to image two small areas of the sky: CDFS and ELAIS-S1. These fields in particular were chosen because they are the two fields imaged in SWIRE (Section 2.3.2) visible from the southern hemisphere. SWIRE produced

high-resolution infrared images of its fields, allowing all objects detected in the ATLAS radio images to be cross-identified with their infrared counterparts.

ATLAS is considered a pilot survey for EMU. EMU and ATLAS image the same wavelengths with similar resolution and sensitivity, so tools and methods developed to process and interpret ATLAS data are expected to work well on the data produced by EMU.

ATLAS provides both a catalogue of detected radio objects and a radio image of the CDFS and ELAIS-S1 fields. The CDFS image covers a total area of 3.7 square degrees and the ELAIS-S1 image covers a total area of 2.7 square degrees. The CDFS image is shown in Figure 2.9. The catalogue is a list of all objects detected in the images with a peak or integrated flux more than 5 times the background noise levels. For each object, the catalogue lists

- an survey identifier,
- an IAU name,
- a position on the sky of the peak flux,
- a peak flux density,
- an integrated flux density,
- an angular size,
- whether the object is extended or compact, and
- a spectral index,

as well as uncertainties associated with each measurement [25].

The ATLAS survey of the CDFS field is the focus of our experiments for three main reasons. It contains around 2400 objects, providing enough training data for machine learning methods, but still remaining a manageable size for our resource-limited tests. As a pilot survey for EMU, we expect methods developed on ATLAS to also work on EMU. Finally, we have three sets of cross-identifications for ATLAS-CDFS (see Sections 2.5 and 2.6), allowing us to check the performance of methods we develop.

2.5 Radio Cross-identification

Observations of astronomical objects in different wavelengths give us information on different physical properties of the objects. We can only make use of this information, however, if we can match observations of the same object in different surveys. This is called *cross-identification*.

The specific cross-identification task we focus on in this thesis is that of cross-identifying a radio object with the associated infrared object. Most often in current radio surveys, a radio object will be a jet from an AGN [5]. In this case, we refer to the infrared counterpart as the *host galaxy* of the AGN. We will assume that all radio objects are AGNs. The main goal of cross-identifying AGNs is to better understand the relationship between AGNs and star-forming activity in their host galaxies [26].

Sometimes, cross-identification is easy: We may have point-like sources of light, allowing us to simply overlay two surveys and identify overlapping objects. In the case of cross-identifying radio AGNs, though, we do not have point-like sources of

light, and cross-identification can be very difficult. AGN jets may be arbitrarily large and complex, and often show up as multiple disconnected radio objects. As such, past attempts to cross-identify radio objects have required human insight [26, 4].

In this section, we look at two different ways astronomers have cross-identified radio objects in the CDFS field of the ATLAS survey. We will make use of the resulting catalogues for testing our cross-identification methods in Chapters 4 and 5.

2.5.1 Norris et al. Catalogue

Along with a catalogue of radio objects found in the CDFS field by the ATLAS survey, Norris et al. [26] produced a catalogue of cross-identifications of these radio objects. The catalogue lists ATLAS objects and their corresponding SWIRE objects.

The cross-identification process was semi-automated. Norris et al. first matched each ATLAS object to the nearest SWIRE object. After this step, around half of the radio objects were matched to a SWIRE object within $1''$, and 79% were matched to a SWIRE object within $3''$. Radio objects greater than $3''$ away from a SWIRE object were then cross-identified manually; radio doubles and radio triples were matched to the SWIRE object nearest their centre.

Additionally, Norris et al. found between 8 and 22 radio objects with no identifiable host galaxy in the infrared. These are called "infrared-faint" radio objects.

Norris et al. estimated the probability of false cross-identifications by displacing all radio objects by $1'$ and repeating the cross-identification process. From the results of this test, they estimated that 9.02% of the cross-identifications are false.

We consider these cross-identifications *expert* cross-identifications, i.e., they are the best possible cross-identifications available for radio objects in the CDFS field. Throughout this thesis, we will use these as an approximation to the unobservable groundtruth cross-identifications.

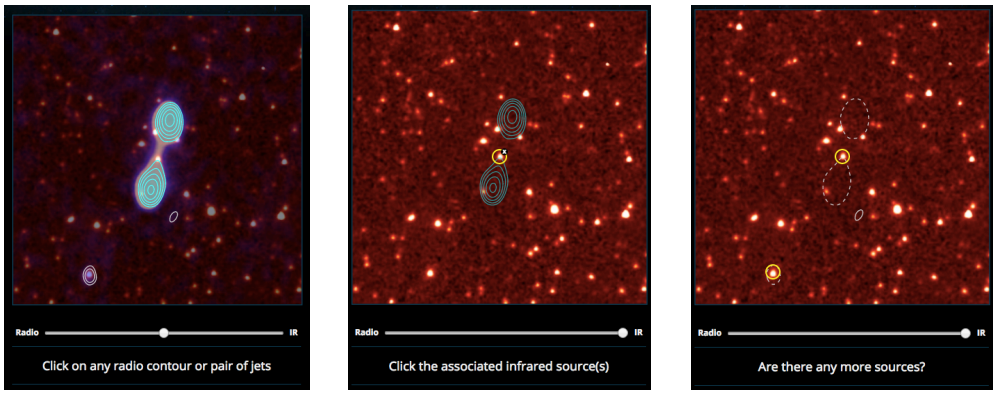
2.5.2 Fan et al. Catalogue

Fan et al. [4] developed an automated cross-identification algorithm that fits astronomical models of AGNs to the radio sky.

The algorithm examines each infrared object in turn. Under the assumption that the infrared object is a host galaxy, it then searches in a $2'$ radius for potential radio components of AGN jets. The radio components with highest likelihood according to a Bayesian model are selected using a greedy algorithm.

This method has some clear limitations — it is model-based, and thus may fail for unexpected radio objects like those we might find in EMU, and it is not able to cross-identify infrared-faint radio objects. Nevertheless, it performs very well when applied to the ATLAS-CDFS field, making 564 cross-identifications identically to Norris et al., missing 31 cross-identifications that Norris et al. reported, and cross-identifying an additional 62 radio objects.

We consider the Fan et al. cross-identifications of ATLAS-CDFS as an alternative source of expert cross-identifications, due to the high (but not total) agreement with



(a) (b) (c)

Figure 2.10: Radio Galaxy Zoo volunteer workflow.
2.10a Volunteers are first asked to identify associated radio objects.
2.10b Volunteers then cross-identify the radio objects with the corresponding host galaxy.
2.10c This is repeated for all radio objects in the image.

the Norris et al. catalogue.

2.6 Radio Galaxy Zoo

The Norris et al. catalogue is highly accurate, but manual expert cross-identification of radio surveys is impractical for large surveys. The CDFS field examined by Norris et al. only contains around 2400 radio objects, which is small even compared to existing surveys (e.g. Faint Images of the Radio Sky at Twenty-Centimeters (FIRST) [27] found 946 000 radio sources). Automated algorithms such as Fan et al. scale better, but most such algorithms are still in their infancy [28] and many are model-based, potentially missing many sources that do not fit the models. The Radio Galaxy Zoo project [1] provides a different approach: Allow many non-expert volunteers to manually cross-identify radio sources with their infrared host galaxies.

Radio Galaxy Zoo[1] is a website where volunteers are presented with a radio image from ATLAS or FIRST and a corresponding infrared image from SWIRE or WISE, and are tasked with matching the radio source with its host galaxy, as well as identifying which radio objects are associated with the same host galaxy (e.g., two radio objects may represent two jets of one AGN). To help reduce noise in these matches, each compact radio object is shown to 5 volunteers, and each complex radio object is shown to 20 volunteers. While both the cross-identification task and the radio object association task are astronomically interesting, we focus only on the cross-identification task for this thesis. An example of the workflow presented to volunteers is shown in Figure 2.10.

Since its launch in December 2013, the Radio Galaxy Zoo project has labelled over 100 000 radio objects. One particularly notable result is the discovery of one of the

[1] https://radio.galaxyzoo.org/

largest known wide-angled tail AGNs by two volunteers, which would be impossible to detect in current automated algorithms [29].

The Radio Galaxy Zoo dataset contains around 177 000 images from FIRST, and 2400 images of the CDFS field from ATLAS. The ATLAS images of the ELAIS-S1 field are not included, and may be used as a testing set for cross-identification algorithms trained on the Radio Galaxy Zoo in the future. Each image is a 2 arcminute-wide square.

The cross-identifications are stored in a MongoDB database as a collection of associated radio objects and the corresponding pixel locations of the volunteers' clicks, as well as the right ascension and declination of the radio object. The radio and infrared image patches associated with each radio object are also included alongside the database. Detailed analysis of these labels will appear in Alger et al. [30].

In Chapter 4, we will train machine learning algorithms to perform the cross-identification task trained on ATLAS-CDFS labels from the Radio Galaxy Zoo.

Chapter 3

Machine Learning on Crowds

In this chapter we will discuss machine learning. While we introduce a number of concepts that will help us develop our cross-identification method, we will not yet apply these concepts to astronomical data, instead deferring this to Chapter 4.

In Section 3.1, we introduce the problem of classification, and discuss some common methods of approach. We will later base our cross-identification methods on these.

In Section 3.2, we look at automated feature extraction from images. Convolutional neural networks will be introduced as feature extractors, which we will eventually apply to radio data from ATLAS.

In Section 3.3, we discuss the benefits and problems of crowdsourcing for obtaining labels for machine learning. Finally, we will look at different ways to handle the problems in Section 3.5.

3.1 Classification

A common machine learning task is *classification*. Given a set $\mathcal{X}$ of *instances* and a set $\mathcal{Y}$ of *labels*, the classification task is to assign a label $y \in \mathcal{Y}$ to each instance $x \in \mathcal{X}$. Effectively, we want to find a map $y : \mathcal{X} \to \mathcal{Y}$.

The "true" label of an instance is called the *groundtruth*, and is denoted z. Classification thus amounts to modelling the conditional distribution $p(z \mid x)$. An example of classification is labelling images of digits, where $\mathcal{X}$ is a set of images and $\mathcal{Y} = \{0, \ldots, 9\}$ [31]. A less obvious example is that of diagnosing breast cancer in patients, where $\mathcal{X}$ is a set of sets of medical observations and $\mathcal{Y}$ is the set $\{\text{malignant}, \text{benign}\}$ [32].

Modelling $p(z \mid x)$ requires some set of *training data* $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. The training data are either used to find parameters for a model or to estimate the model directly. Both processes are called *training* the model. The cardinality of the training data is denoted $N = |\mathcal{D}|$.

Binary classification is classification where each instance must be assigned one of two labels, i.e. $\mathcal{Y} \sim \{0, 1\}$. Instances with a true label of 0 are called *negative instances*, and instances with a true label of 1 are called *positive instances*. For this thesis, we will focus entirely on binary classification and set $\mathcal{Y} = \{0, 1\}$.

Instances are usually represented by a vector of *features*, denoted $\vec{x}$. Choosing features is an important and non-trivial part of modelling a classification task, and can strongly affect the ability of a model to classify instances. The dimensionality of the feature space is denoted D and we assume that $\mathcal{X} \subseteq \mathbb{R}^D$.

We will now look at how to evaluate a classification model, and introduce three common ways of modelling binary classification: logistic regression, neural networks, and random forests.

3.1.1 Evaluating a Classification Model

Given a classification model $y(\vec{x})$ that aims to approximate $p(z \mid \vec{x})$, we want to know how well this model represents the groundtruth. In this thesis we make use of two evaluation metrics: cross-entropy error, and balanced accuracy.

The *cross-entropy error* [33] is a common method for evaluating a binary classification model. The cross-entropy error is given by

$$L(\mathcal{D}) = -\frac{1}{N} \sum_{\vec{x},z \in \mathcal{D}} \left(z \log y(\vec{x}) + (1-z) \log(1 - y(\vec{x})) \right).$$

Higher error indicates a "worse" model.

The *balanced accuracy* is another common method of evaluating classification models. It is the average of the accuracy of classifying positive instances and the accuracy of classifying negative instances, i.e.

$$a = \frac{1}{2} \left(\frac{p_{\text{true}}}{p} + \frac{n_{\text{true}}}{n} \right),$$

where p is the number of instances with $z = 1$, n is the number of instances with $z = 0$, p_{true} is the number of correctly classified instances with $z = 1$, and n_{true} is the number of correctly classified instances with $z = 0$.

Unless otherwise specified, we use the cross-entropy error for training our models, and report the balanced accuracy.

3.1.2 Logistic Regression

Logistic regression is a simple linear model of binary classification [33]. The conditional distribution $p(z \mid \vec{x})$ is modelled as

$$y(\vec{x}) = p(z = 1 \mid \vec{x}) = \sigma(\vec{w}^T \vec{x} + b), \tag{3.1}$$

where $\vec{w}$ and b are parameters to the model and σ is the logistic sigmoid function

$$\sigma(t) = \frac{1}{1 + \exp(-t)}.$$

The elements of $\vec{w}$ are called the *weights*, and control the effect of each feature on the output label probability. b is the *bias*, a constant added to all weighted sums of features independent of the features themselves. b can be incorporated into $\vec{w}$ by adding an extra feature to $\vec{x}$ that is always 1 and increasing the dimension of $\vec{w}$ by 1. We will adopt this convention whenever b is not explicitly included.

Logistic regression is one of the simplest models we can make, and is the classification analogue of linear regression. We can think of it as a linear function with a "soft" threshold, with all output values above the threshold approaching 1 and all other outputs approaching 0. The logistic sigmoid function causes this threshold effect. While we could instead use a step function, a differentiable function like the logistic sigmoid function means that logistic regression will be differentiable.

Since logistic regression is differentiable, *gradient descent* can be used to find optimum values of $\vec{w}$. $\vec{w}$ is modified by the update equation

$$\vec{w}_{t+1} = \vec{w}_t - \lambda \nabla L(\vec{w}_t),$$

where $L : \mathbb{R}^D \to \mathbb{R}$ is the *loss* function, $\lambda \in (0, 1)$ is the *learning rate*, and t is the current step. This update is repeated until convergence.

The loss function represents how well the current model fits the observations. For binary classification, this is usually given by the cross-entropy error with gradient [33]

$$\nabla L(\vec{w}) = -\frac{1}{N} \sum_{\vec{x}, z \in \mathcal{D}} (z - y(\vec{x}))\, \vec{x}.$$

3.1.3 Neural Networks

One key limitation of logistic regression is that it is a linear model. This means that the decision boundary is a hyperplane in the feature space, and thus logistic regression can only classify data with classes that are linearly separable. In general, classes may not be linearly separable, so logistic regression may not be able to accurately model the classification task. One way to improve logistic regression performance on non-linearly separable classes is to introduce a nonlinear *feature map* $\vec{\phi} : \mathbb{R}^D \to \mathbb{R}^F$ to transform the inputs. Logistic regression then becomes

$$y(\vec{x}) = \sigma(\vec{w}^T \vec{\phi}(\vec{x})).$$

By judicious choice of feature transformations, the features may be transformed into a space where the classes are linearly separable. This raises a question: Which feature map should we choose?

We can approximate the feature map by a simple linear function, i.e.

$$\vec{\phi}(\vec{x}) = \vec{\sigma}(A\vec{x}),$$

where $\vec{\sigma}(\vec{v})$ is σ applied elementwise to $\vec{v}$. The weight matrix A can then be learned by gradient methods along with $\vec{w}$. This approximation itself has drawbacks — perhaps a

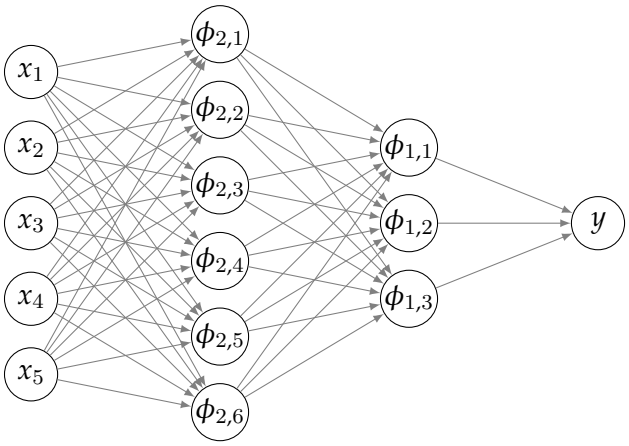


Figure 3.1: A 2-layer neural network represented as a directed acyclic graph.

linear function is not powerful enough to approximate a good choice of feature map — but we can resolve this by applying another nonlinear transformation to $\vec{x}$, arbitrarily many times. This leads to a new model with multiple weights $A_1, \ldots, A_k$:

$$\begin{aligned} y(\vec{x}) &= \sigma(\vec{w}^T \vec{\phi}_1(\vec{x})) \\ \vec{\phi}_1(\vec{x}) &= \vec{\sigma}(A_1 \vec{\phi}_2(\vec{x})) \\ &\vdots \\ \vec{\phi}_K(\vec{x}) &= \vec{\sigma}(A_K \vec{x}). \end{aligned}$$

This is a *K-layer neural network*, so called as it resembles the structure of neurons in the brain. Such a network is able to approximate any function with sufficiently large dimensionality of the matrices A_i [34].

While we have viewed neural networks here as a very natural extension to logistic regression, the class of neural network models is far more varied. Interpreting each element in $\vec{x}, \vec{\phi}_K, \ldots, \vec{\phi}_1$ as a node and the elements of $A_K, \ldots, A_1, \vec{w}$ as weighted edges, we can interpret the above neural network as a directed acyclic graph (Figure 3.1). In this context, each column of the graph is called a *layer*. If the layer is fully connected to its inputs and outputs, then it is called a *dense layer*. We may also introduce different kinds of layers performing arbitrary nonlinear functions; some examples of these will be covered in Section 3.2.4.

3.1.4 Random Forests

A *decision tree* is a classifier that classifies by repeatedly dividing the feature space. Generating the tree proceeds as follows. The tree itself is a binary tree of decisions. The input space is split along one feature axis, giving two subtrees with one less dimension. This is then recursively applied to the subtrees until each leaf of the tree contains less than a given number of examples or until there are no more features to split on. The

Figure 3.2: If we ignore the location of pixels when interpreting an image as a vector, these two images have identical feature vectors, despite the left image clearly containing information not present in the right image.

leaf nodes are then labelled with the most common label of the elements they contain.

A *random forest* is a classifier that uses an ensemble of decision trees to classify data points [35]. A number of decision trees are generated, each with a different subset of features and inputs. This ensures that the decision trees are different. The random forest then classifies by having the decision trees vote on each label, with the most common vote being used as the prediction. The probability of this label being correct can be estimated as the percentage of decision trees in the ensemble that agree with the prediction.

Random forests are nonlinear classifiers which are resistant to problems of feature scale. This makes them a good off-the-shelf classifier to apply to a generic classification problem.

3.2 Feature Extraction from Images

For instances to be input to a classification model, they must be represented by vectors of features. Feature selection is generally a hard problem, but it may be straightforward if instances are simple vectors of measurements (such as in the breast cancer dataset described in Section 3.4.1). If the instances are provided in the form of images, however, feature selection becomes considerably harder.

3.2.1 The Naïve Approach

Images can be thought of as vectors of pixels, where the value of each pixel is treated as an independent feature. If the image is a colour image, then each channel (red, blue, and green) of each pixel can be treated as an independent feature. This is a very simple way to obtain features from images, with two considerable problems.

The first problem is that using pixels directly as independent features fails to accurately capture the image. The pixels are assumed independent when this is not the

(a) Original image.

(b) Max pooling.

(c) Min pooling.

(d) Average pooling.

Figure 3.3: Examples of max, min, and average pooling applied to an image.

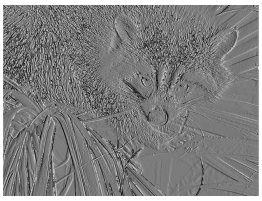
case — neighbouring pixels are likely to be correlated, shapes and structure within the image introduce further correlations, and so on. If we are only looking at certain kinds of images, then the pixel location may also matter — for example, we might have as inputs photographs of landscapes, where the sky tends to be at the top of the image — and this information is also lost. These effects are shown in Figure 3.2. Finally, with independent pixels as features, trained models will be sensitive to small distortions or translations in the input [31]. In Section 3.2.3, we describe a common approach for resolving this problem.

The second problem with this approach is that images tend to be large [31]. Taking each pixel as a feature leads to high dimensional feature spaces. This is not ideal as many algorithms perform poorly in high dimensional spaces. This can be resolved using *pooling*, which we describe in Section 3.2.2.

(a) Original image.

$$\begin{bmatrix} -1 & 1 & 0 \\ -2 & 2 & 0 \\ -1 & 1 & 0 \end{bmatrix}$$

(b) Convolutional filter.



(c) Convolved image.

Figure 3.4: Convolving an image with a 3×3 filter gives a new image. This filter acts as a kind of vertical edge detector.

3.2.2 Pooling

Pooling is a class of operations that reduce the dimensionality of an image, parameterised by a *pool size*[1], p. Nonoverlapping $p \times p$ squares of pixels are condensed into one value, which becomes the corresponding pixel in the output. In this way, the size of the image is reduced from $m \times n$ to $\frac{m}{p} \times \frac{n}{p}$. This results in both a smaller amount of data, and some amount of translation invariance when pooling is used in models like neural networks [36].

We are free to choose whatever aggregation operation we like to condense squares of pixels; common choices are max (called *max pooling*) and mean (called *average pooling*). Some pooling examples are shown in Figure 3.3.

3.2.3 Convolutions

A common way to make use of neighbourhood information in images is by using *convolutions*. A convolution is an operation that transforms an image by applying an $n \times n$ *filter* to each $n \times n$ square of pixels. The filter is a matrix in $\mathbb{C}^{n \times n}$, and it is applied to a square of pixels by treating that square as a matrix, performing an elementwise multiplication between the filter matrix and the pixel matrix, and summing the result. Applying the filter to a square of pixels gives a single pixel value, so by applying the filter across the image, a new image is generated [37].

One ambiguity is how to apply a filter to the edges of an image. There are multiple ways to resolve this. The most common method is to only apply the filter to squares of pixels fully contained within the image, making the output image smaller than the input. Another method is to apply the filter to the edges and assume that pixels outside the image have a constant value (usually 0), making the output image the same size as the input. These methods are commonly known as "valid" and "same" respectively in libraries such as SciPy and Keras.

The effect of a simple convolutional filter on an image is shown in Figure 3.4, and an algorithm for performing a convolution is shown in Algorithm 1.

[1] Another parameter is the *stride*, s, which we ignore here for simplicity, setting $s = p$.

Algorithm 1: One method of performing a convolution. Here, we choose to use the "valid" method of handling edges, resulting in a smaller output than the input. $\odot$ is the elementwise matrix product.

Data:
An image $X \in \mathbb{R}^{m \times n}$
A filter $F \in \mathbb{C}^{d \times d}$
Result: An image $\in \mathbb{R}^{(m-d+d \bmod 2) \times (n-d+d \bmod 2)}$
Initialise output image $Y \in \mathbb{R}^{(m-d+d \bmod 2) \times (n-d+d \bmod 2)}$;
for $y \in \lfloor d/2 \rfloor, \ldots, m - \lfloor d/2 \rfloor$ **do**
  for $x \in \lfloor d/2 \rfloor, \ldots, n - \lfloor d/2 \rfloor$ **do**
    $H \leftarrow d \times d$ square of pixels centred on (x, y);
    $Y_{y,x} \leftarrow (\vec{1}^T (H \odot F) \vec{1}) / (\vec{1}^T F \vec{1})$;
  end
end
return Y

By repeatedly applying convolutions and pooling, we can extract features from images that would not have been present using the naïve approach of Section 3.2.1.

3.2.4 Convolutional Neural Networks

While convolutions are needed to extract features from our images, there are no convolutions that work well in general. Convolutions must be chosen on a domain-dependent basis. This raises the question: Which convolutions should we choose?

One way to answer this question is simply to not choose at all, and then learn the appropriate convolutions as part of a larger classification model. This gives rise to the concept of a *convolutional neural network* (CNN). A CNN is a neural network with layers that convolve their inputs and layers that pool their inputs, called *convolutional layers* and *pooling layers*, respectively. Convolutional layers may contain any number of separate filters. They are generally interspersed with pooling layers. The final layers of the CNN are usually dense layers. Additionally, CNNs introduce some amount of translation invariance into their outputs, since the same filters are applied across the image [31].

CNNs can be used for feature extraction from images. As input, they take images to extract features from, and they are tasked with mapping these to some meaningful value or the intended output of the greater classification problem. The convolutional and pooling layers can then be isolated, forming a network that extracts features from images. An example of a convolutional neural network applied to an image is shown in Figure 3.5.

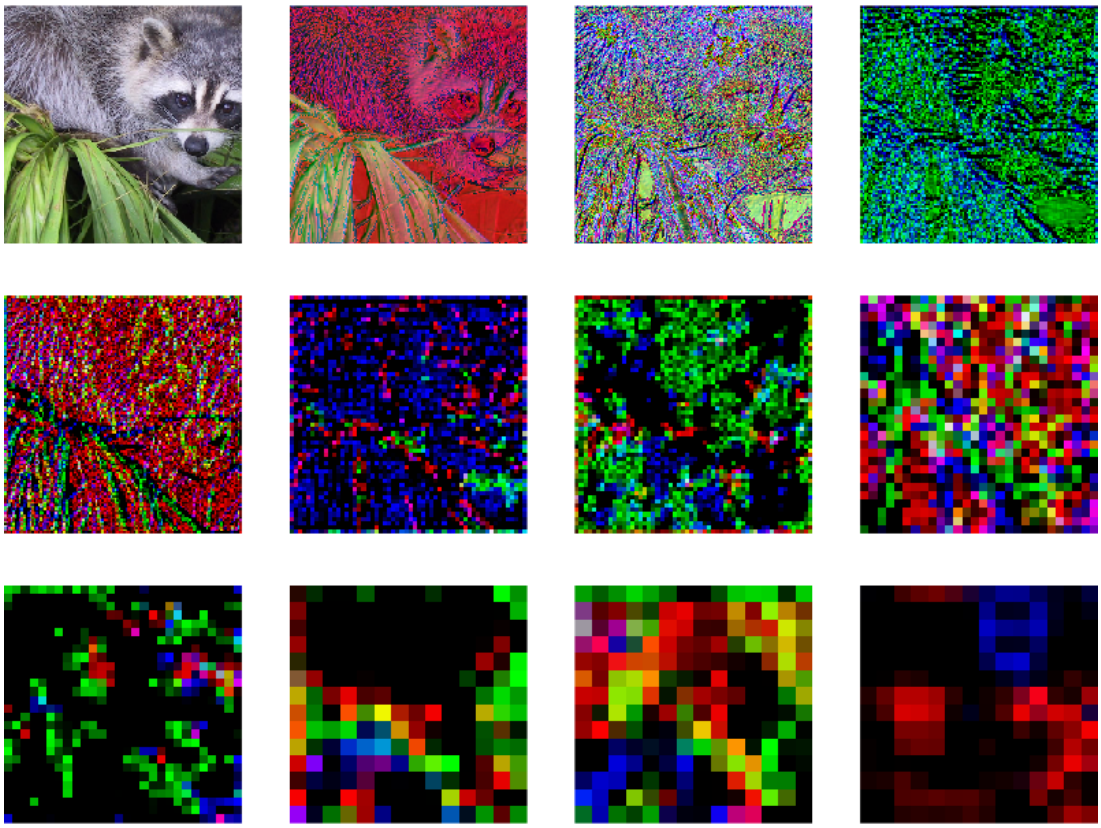


Figure 3.5: A convolutional neural network applied to the top-left image. The other images are the outputs of intermediate convolutional layers of the neural network, from left to right, top to bottom. Colours represent convolutions with different filters. The output, bottom-right, is a higher-level representation of the input image, and its values can be used as inputs for classification. This figure uses Baraldi's implementation of the VGG-16 CNN [39].

3.3 Crowdsourcing Labels

For standard supervised learning tasks, the labels are generally provided by some *expert*. We assume that the expert provides labels that are correct and accurately represent the groundtruth. More recently, however, many projects have *crowdsourced* their labels: Non-expert people (the *crowd*) volunteer or are paid to label data. The crowd may be sourced from websites like Amazon Mechanical Turk[2] where small amounts of money are paid on a per-label basis, or they may volunteer out of interest in the labelling project.

Crowdsourcing allows us to quickly and cheaply label large amounts of data. This comes at the expense of objectivity and reliability — since the crowd are necessarily non-experts, they do not have domain knowledge for the problem, and may not provide high-quality labels [8]. While we can try to reduce noise by requesting multiple labellers label each instance [40, 41], this does not address the fact that the data may be intrinsically hard to label, the labellers may be correlated, and some labellers may even be actively malicious [9]. Some strategies to address these problems are described in Section 3.5.

The form of crowdsourcing we are interested in for this thesis is *citizen science*, as this is the form of crowdsourcing that the Radio Galaxy Zoo (Section 2.6) uses. Citizen science is "scientific work undertaken by members of the general public" [42, 43], and has recently come to refer mainly to websites that allow non-experts to label scientific data. A notable example of citizen science is the Galaxy Zoo project [6], which aims to identify morpologies of galaxies from the Sloan Digital Sky Survey. Galaxy Zoo has gathered tens of millions of labels for nearly 9×10^5 galaxies from over 8×10^4 volunteers [7], and the data collected have been used in 48 publications so far[3]. Many other citizen science projects have been hosted on the Galaxy Zoo's platform, Zooniverse[4], from further astronomical studies like the Radio Galaxy Zoo [1] (Section 2.6), which itself has labelled over 10^5 radio sources from two radio surveys, to ecological studies like Snapshot Serengeti [44], which has classified nearly 11 million camera trap images from Serengeti National Park.

3.4 Simulating a Crowd Labelling Task

The crowd is inherently unpredictable. This makes it difficult to test methods for crowd learning, since there may be biases in existing crowd-labelled datasets, and we want our experiments to be both accurate and reproducible. To mitigate this problem, we instead simulate a crowd labelling task based on a dataset with known groundtruth. In this section, we describe our method for simulating a crowd labelling task, which we will later employ to test methods in Section 3.5. As part of this thesis, we have also implemented this simulation in Python. The implemention is described in Appendix A.2.5.2.

[2] http://mturk.com
[3] https://www.zooniverse.org/about/publications
[4] http://zooniverse.org

3.4.1 The Breast Cancer Wisconsin Dataset

The breast cancer Wisconsin dataset, first used by Wolberg and Mangasarian [32], presents a classification task commonly used for testing classification methods. The dataset can be obtained from the UCI Machine Learning Repository [45] [5]. The classification task is to, given medical observations of a patient, determine if their cancer is benign or malignant. Each instance is a 10-dimensional feature vector, and there are 699 instances. There is some missing data, which we chose to set to 0.

3.4.2 Simulated Labeller Representation

We simulate T labellers. Each labeller is indexed by a number t. The t-th labeller is represented by their *sensitivity* (or *true positive rate*) α_t, and their *specificity* (or *true negative rate*) β_t, i.e.

$$\alpha_t = p(y_t = 1 \mid z = 1)$$
$$\beta_t = p(y_t = 0 \mid z = 0)$$

where y_t is the label assigned by the t-th labeller to an instance with groundtruth z.

[5] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

3.4.3 Obtaining the Simulated Labels

To generate crowd labels from our T simulated labellers for a dataset with N instances, we follow Algorithm 2.

Algorithm 2: Simulating a crowd labelling task.

Data:
A set of groundtruth labels $\{z_1, \ldots, z_N\}$
A set of labellers $\{(\alpha_1, \beta_1), \ldots, (\alpha_T, \beta_T)\}$.
Result: A set of crowd labels $\{y_{1,1}, \ldots, y_{T,N}\}$.
for $i \in 1, \ldots, N$ **do**
  for $t \in 1, \ldots, T$ **do**
    if $z_i = 1$ **then**
      with probability α_t **do**
        $y_{t,i} \leftarrow 1$;
      else
        $y_{t,i} \leftarrow 0$;
      end
    else
      with probability β_t **do**
        $y_{t,i} \leftarrow 0$;
      else
        $y_{t,i} \leftarrow 1$;
      end
    end
  end
end
return $\{y_{1,1}, \ldots, y_{T,N}\}$

3.5 Utilising Crowd Labels

A common method to combat label noise in crowd learning scenarios is to request multiple labels from the crowd for each data point. There is some basis in the literature for such redundancy, with Sheng et al. [41] showing that repeated labelling may improve label and model quality, though Lin et al. [46] show that this is not always the case and results depend on the level and kind of label noise.

Nevertheless, this approach is used in existing citizen science projects such as Galaxy Zoo [6] and Radio Galaxy Zoo [1], both of which request between 5 and 20 crowd labels for each instance.

This raises the question of how best to make use of multiple labels to train classification models. In this section, we look at some methods that directly learn a model from multiple noisy labels, as well as some methods of aggregating the labels to obtain individual labels with (ideally) lower noise.

3.5.1 Majority Vote

One way to reduce label noise is to allow multiple labellers to label the same examples, and then for each example take the *majority vote* to find a "consensus" label. If the label provided by the t-th labeller is denoted y_t and there are T labellers, then the consensus label is given by

$$y = \begin{cases} 1 & \frac{1}{T}\sum_{t=1}^{T} y_t > 0.5 \\ 0 & \frac{1}{T}\sum_{t=1}^{T} y_t < 0.5 \end{cases}$$

with the remaining case decided evenly at random [8].

The idea is that the majority of labels are correct, so with sufficiently large amounts of relabelling we expect the label noise to be reduced. How large "sufficiently large" is is unclear and domain-dependent, and is outside the scope of this thesis, though Sheng et al. [41] and Lin et al. [40] provide some information on how to choose relabelling rates. Of course, this fails in situations where the majority of labellers are not correct, which may be due to intrinsic difficulty or malicious labelling, or where there is no clear majority.

One simple improvement to majority vote is to weight labels by the accuracy of the associated labeller, computing these accuracies based on some known groundtruth for a subset of the data. However, this requires access to the groundtruth for a subset of data, which is not available in general.

3.5.2 Raykar et al. Model

One way to learn from crowdsourced labels when no groundtruth is available is to jointly model both the reliability of each labeller (the *labeller model*) and the groundtruth itself (the *classification model*). Raykar et al. [8] propose such a joint model, which we now describe.

The accuracy of the t-th labeller is modelled by the sensitivity α_t and the specificity β_t, as described in Section 3.4.2. This model assumes that labeller reliability is independent of the example x, but dependent on the groundtruth label z. Note that under this model, the probability that labeller t will assign a given label to a positive example is given by

$$p(y_t \mid z = 1, \alpha_t) = (\alpha_t)^{y_t}(1 - \alpha_t)^{1-y_t}$$

and the probability that they will assign a given label to a negative example is given by

$$p(y_t \mid z = 0, \beta_t) = (\beta_t)^{1-y_t}(1 - \beta_t)^{y_t}.$$

For ease of notation, let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_t)$ and $\vec{\beta} = (\beta_1, \ldots, \beta_t)$.

With this method, the classification model can be any classifier. Raykar et al. choose to use logistic regression, i.e.

$$p(z = 1 \mid \vec{x}, \vec{w}) = \sigma(\vec{w}^T\vec{x}). \tag{3.2}$$

The model parameters $\vec{\theta} = \{\vec{w}, \vec{\alpha}, \vec{\beta}\}$ can be found by maximising the likelihood.

Under the assumption that examples are independently sampled and labellers are independent, the likelihood is given by

$$\begin{aligned} p(\mathcal{D} \mid \vec{\theta}) &= \prod_{i=1}^{N} \prod_{t=1}^{T} p(y_{t,i} \mid \vec{x}_i, \vec{w}, \alpha_t, \beta_t) \\ &= \prod_{i=1}^{N} \prod_{t=1}^{T} \Big[p(y_{t,i} \mid z_i = 1, \alpha_t) p(z_i = 1 \mid \vec{x}_i, \vec{w}) \\ &\qquad + p(y_{t,i} \mid z_i = 0, \beta_t) p(z_i = 0 \mid \vec{x}_i, \vec{w}) \Big]. \end{aligned}$$

Since there are unknown values z, the maximum likelihood problem must be solved using expectation-maximisation. This has closed-form solutions for $\vec{\alpha}$ (Equation 3.3) and $\vec{\beta}$ (Equation 3.4), but gradient methods must be used for finding $\vec{w}$.

μ_i is initialised with majority vote, i.e.

$$\mu_i = \frac{1}{T} \sum_{t=1}^{T} y_{t,i}.$$

The expectation step requires us to compute

$$\begin{aligned} a_i &= \prod_{t=1}^{T} (\alpha_t)^{y_{t,i}} (1 - \alpha_t)^{1 - y_{t,i}} \\ b_i &= \prod_{t=1}^{T} (\beta_t)^{1 - y_{t,i}} (1 - \beta_t)^{y_{t,i}} \\ \mu_i &\propto \frac{a_i \sigma(\vec{w}^T \vec{x}_i)}{a_i \sigma(\vec{w}^T \vec{x}_i) + b_i (1 - \sigma(\vec{w}^T \vec{x}_i))}. \end{aligned}$$

The maximisation step requires us to compute

$$\alpha_t = \frac{\sum_{i=1}^{N} \mu_i y_{t,i}}{\sum_{i=1}^{N} \mu_i} \tag{3.3}$$

$$\beta_t = \frac{\sum_{i=1}^{N} (1 - \mu_i)(1 - y_{t,i})}{\sum_{i=1}^{N} (1 - \mu_i)} \tag{3.4}$$

and find the $\vec{w}$ that maximises the likelihood using gradient methods. In our implementation of this algorithm, we initialised $\vec{w}$ using logistic regression trained on the majority vote, and then used the old $\vec{w}$ to initialise each subsequent maximisation step.

As part of this thesis, we have produced an open source implementation of this algorithm, described in Appendix A.2.5.1. We then performed a series of experiments using this implementation, which are detailed here.

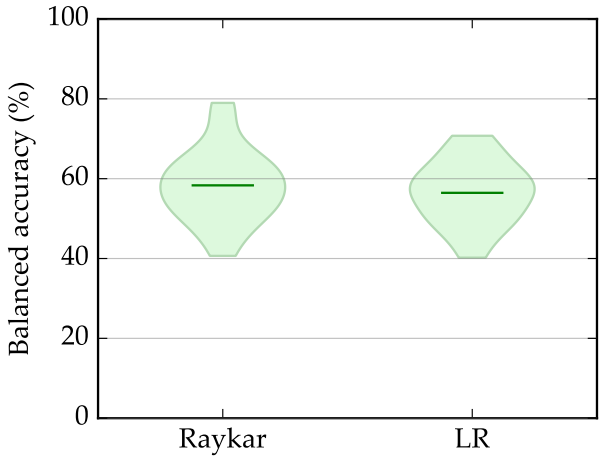


Figure 3.6: Performance of the Raykar et al. classifier against logistic regression (LR) on a simulated crowd labelling of the breast cancer dataset [32]. Spread along the vertical axis represents results over multiple trials; the bar represents the mean.

3.5.2.1 Testing the Raykar Classifier

We first used the implementation to perform a simple, simulated crowd labelling problem. Five simulated labellers were assigned true positive and false positive rates uniformly distributed in the range $[0.25, 0.75]$. Each simulated labeller labelled 50% of the breast cancer dataset (Section 3.4.1). Random samples of 75% of the examples were drawn 20 times and used to train both the Raykar et al. classifier and a logistic regression classifier (using majority vote for the labels). To help find a good set of parameters, the Raykar et al. classifier was retrained 5 times with random initial conditions, and the classifier attaining the highest likelihood was selected. Both logistic regression and Raykar et al. classifiers were tested against the groundtruth of the remaining 25%. The results are plotted in Figure 3.6. The Raykar classifier attained a balanced accuracy of $(58 \pm 8)\%$ and the logistic regression classifier attained a balanced accuracy of $(56 \pm 8)\%$.

From this experiment, we can see that the Raykar et al. algorithm performs comparably to logistic regression, even under high noise. However, training a logistic regression classifier on the majority vote is far simpler and faster, even in an idealised scenario where labeller noise is exactly modelled by the Raykar et al. model. This may be an artefact of the expectation-maximisation algorithm, which is never guaranteed to converge to a global maximum. If so, then increasing the number of random restarts may improve results, at the cost of increased training time.

3.5.2.2 The Effect of Class Imbalance

The second experiment tested the effect of class imbalance on the resulting balanced accuracy of the classifier, as well as on the predicted α and β for each labeller. We

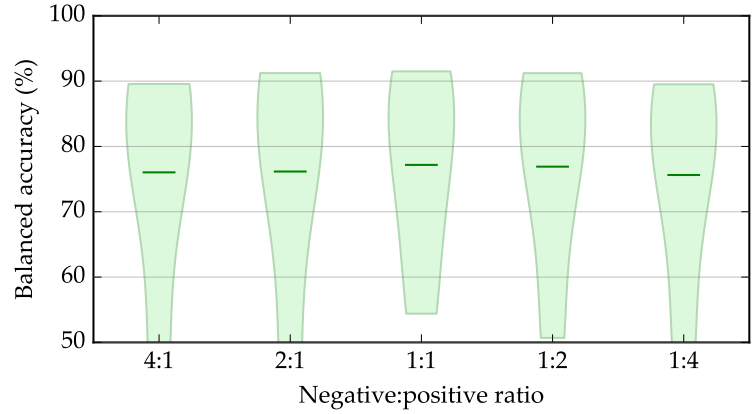


Figure 3.7: Performance of the Raykar et al. classifier on a simulated labelling problem with different class imbalance. Spread along the vertical axis represents results over multiple trials; the bar represents the mean.

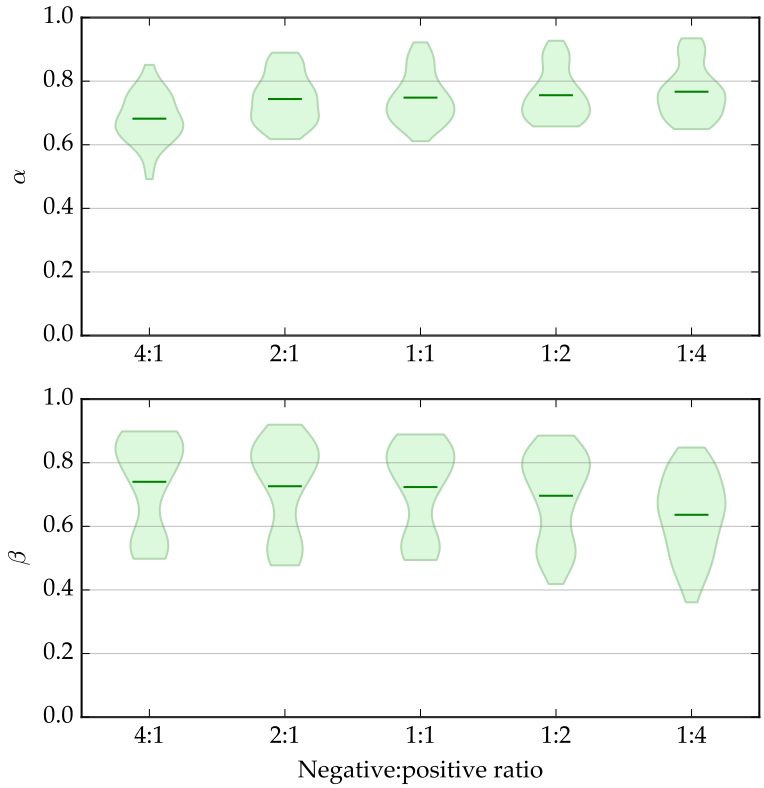


Figure 3.8: Output α and β values of the Raykar et al. classifier on a simulated labelling problem with different class imbalance. Spread along the vertical axis represents results for multiple labellers over multiple trials; the bar represents the mean.

used scikit-learn's `sklearn.datasets.make_classification` function [11] to generate datasets with 5-dimensional features (of which 3 were informative, and 2 were redundant) with a class separation of 1 and 1% of true labels randomly flipped. We generated five datasets, each with 1000 examples, with different ratios of negative to positive examples. The ratios were $4:1$, $2:1$, $1:1$, $1:2$, and $1:4$. We then simulated a crowd labelling task as in Section 3.4, assigning 10 labellers true positive and false positive rates uniformly at random in the interval $[0.5, 0.9]$ and allowing them to "label" the examples. The same labellers were used across all trials and datasets. We ran the Raykar et al. algorithm on each dataset and recorded the balanced accuracy and output α and β values for each trial and dataset. We repeated this experiment 5 times. The resulting balanced accuracies are plotted against the ratios of negative to positive examples in Figure 3.7, and the estimated α and β values are plotted against the ratios in Figure 3.8.

The balanced accuracy is similarly distributed across all trials, with a slight decrease as classes grow more unbalanced. The estimated α and β values are also similar across all trials. This shows that our implementation of the Raykar et al. algorithm is not sensitive to class imbalance.

3.5.2.3 The Effect of Label Noise

The final experiment was to determine the effect of label noise on the performance and output of the algorithm. 10 labellers were simulated with α ranging from 0.2 to 1.0 and β fixed at 1.0. A dataset was generated as for the previous experiment (but with balanced classes), the labellers were tasked with labelling this dataset, and the labels were used to train the algorithm. The balanced accuracy and α values were recorded. This experiment was repeated 5 times, and repeated again with β replacing α. The results are shown in Figure 3.9.

As expected, the balanced accuracy increases as label noise decreases, and the output α and β closely match the true values. This shows that our implementation of the Raykar et al. algorithm is able to recover the noise parameters of our simulated labellers.

3.5.3 Yan et al. Model

A similar approach is described by Yan et al. [9], again jointly learning the groundtruth and labeller models. Unlike Raykar et al., the Yan et al. labeller model is data-dependent. The accuracy of the t-th labeller is modelled by a Bernoulli distribution of $\eta_t(\vec{x})$, parametrised as a logistic regression function $\eta_t(\vec{x}) = \sigma(\vec{\omega}_t^T \vec{x} + \gamma_t)$. The labeller model is thus

$$p(y_t \mid \vec{x}, z, \vec{\omega}_t, \gamma_t) = \eta_t(\vec{x})^{1-|y_t - z|}(1 - \eta_t(\vec{x}))^{|y_t - z|}.$$

For ease of notation, let $\Omega = (\vec{\omega}_1^T, \ldots, \vec{\omega}_T^T)$ and let $\vec{\gamma} = (\gamma_1, \ldots, \gamma_T)$. This model can be obtained from the Raykar et al. model by requiring $\alpha_t = \beta_t = \eta_t(\vec{x})$ and allowing these parameters be data-dependent.

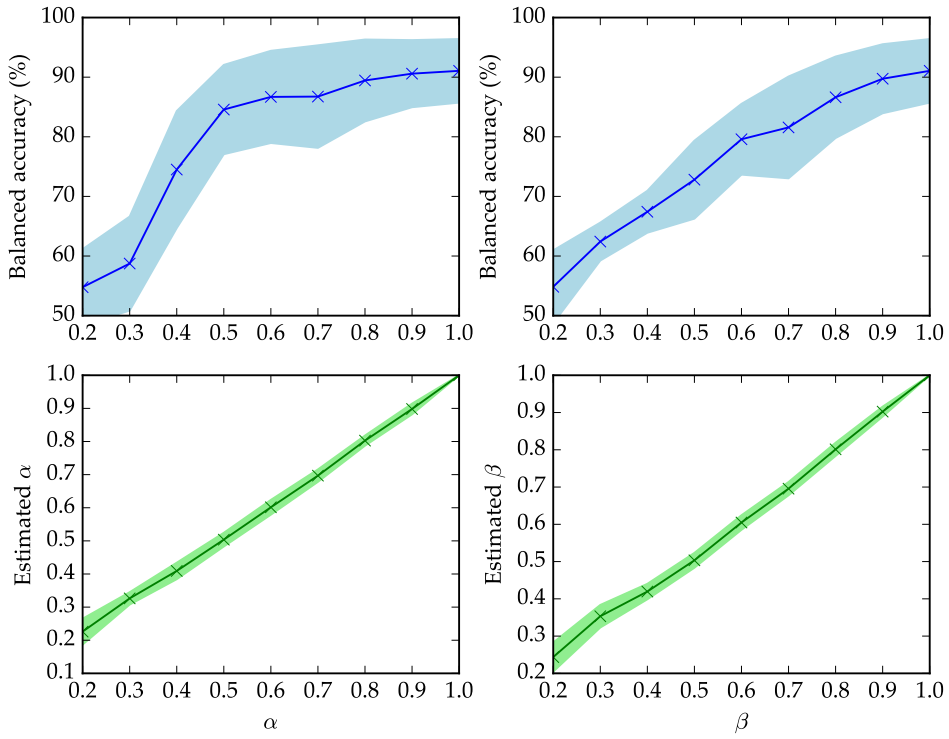


Figure 3.9: Output α, β, and balanced accuracy of the Raykar et al. classifier on a simulated labelling problem with different amounts of label noise. "Estimated α" refers to the value of α output by the algorithm, and α refers to the groundtruth. Filled areas represent the standard deviation of multiple measurements.

As with Raykar et al., the classification model can be any classifier and Yan et al. choose to use logistic regression (Equation 3.2). The parameters $\vec{\theta} = \{\Omega, \vec{\gamma}, \vec{w}\}$ can be found by maximising the likelihood with expectation-maximisation. Under the assumptions that examples are independently sampled and that the labellers are independent, the likelihood is given by

$$\begin{aligned} p(\mathcal{D} \mid \vec{\theta}) &= \prod_{i=1}^{N} \prod_{t=1}^{T} p(y_{t,i} \mid \vec{x}_i, \vec{w}, \Omega, \vec{\gamma}) \\ &= \prod_{i=1}^{N} \prod_{t=1}^{T} \sum_{z=0}^{1} p(y_t \mid \vec{x}_i, z, \vec{\omega}_t, \gamma_t) p(z \mid \vec{x}_i, \vec{w}). \end{aligned}$$

The expectation step requires us to compute

$$\mu_i \propto \prod_{t=1}^{T} p(y_{t,i} \mid \vec{x}_i, z_i = 1, \vec{\omega}_t, \gamma_t) p(z_i = 1 \mid \vec{x}_i, \vec{w}).$$

The maximisation step requires us to maximise

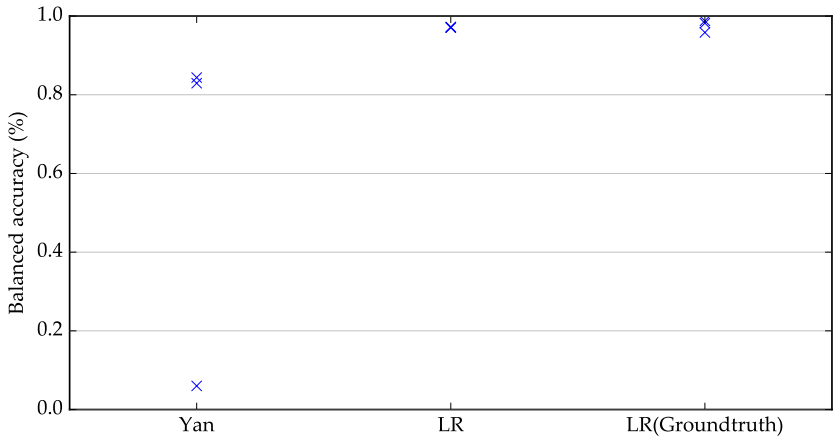
$$\sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{z_i=0}^{1} p(z_i \mid \vec{x}_i, \vec{w}) (\log p(y_{t,i} \mid \vec{x}_i, z, \vec{\omega}_t, \gamma_t) + \log p(z \mid \vec{x}_i, \vec{w}))$$

with respect to $\vec{w}$, Ω, and $\vec{\gamma}$, where $p(z_i \mid \vec{x}_i, \vec{w})$ is fixed to use the previous value of $\vec{w}$.

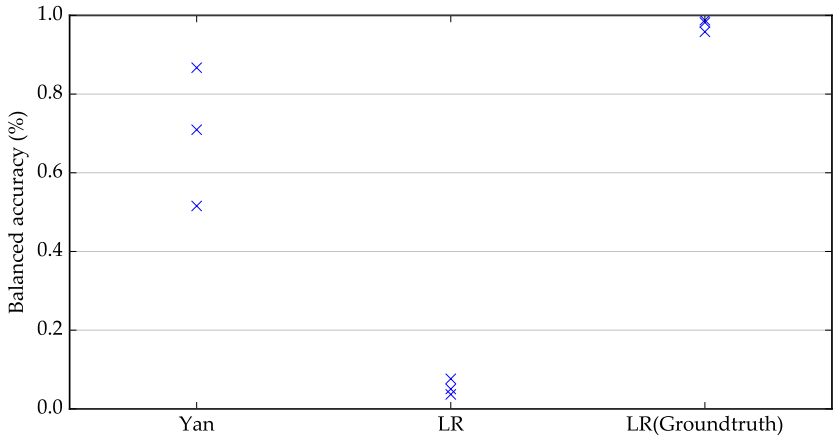
As part of this thesis, we have produced an open source implementation of this algorithm, described in Appendix A.2.5.3. After the algorithm was implemented, we found it to be considerably slower than the Raykar et al. algorithm, so we do not use it later in this thesis. Its inclusion in this thesis is to show that the expectation-maximisation approach can be extended by choice of labeller model.

The Yan et al. algorithm is good for modelling labellers with highly feature-dependent noise. To test the algorithm, we employed it on the breast cancer Wisconsin dataset (Section 3.4.1), but using a different simulation of labellers to that used for the Raykar et al. tests of Section 3.5.2. While the method for generating simulated crowd labels was similar, the labeller model differed to show the ability of the Yan et al. model to handle feature-dependent noise. Instead of using two values α and β to describe each labeller, we clustered the data and assigned each labeller one cluster. The labeller had 100% accuracy for that cluster, and a fixed accuracy υ for all other clusters, giving each labeller a "cluster of expertise". We tested with low noise ($\upsilon = 0.75$) and high noise ($\upsilon = 0.25$). The results are compared with logistic regression trained on the majority vote and logistic regression trained on the groundtruth in Figure 3.10a and Figure 3.10b for low and high noise, respectively.

From these plots, we can tell that with low noise, logistic regression is able to learn a high quality model from the majority vote, while the Yan et al. model performs more poorly. While it is possible for the Yan et al. model to reduce to logistic regression, this is evidently not guaranteed in practice. This may be due to the difficulty of the expectation-maximisation problem, which is never guaranteed to converge to a global



(a) $\upsilon = 75\%$



(b) $\upsilon = 25\%$

Figure 3.10: Yan et al. algorithm compared with logistic regression trained on the majority vote (LR) and logistic regression trained on the groundtruth (LR(Groundtruth)). Labellers have an accuracy of υ when not in their cluster of expertise. Points represent different trials.

maximum in general. When noise is high, however, the Yan et al. algorithm considerably outperforms logistic regression, indicating that in situations with high, feature-dependent noise, the Yan et al. algorithm may be a good choice of classification algorithm. However, this choice of labeller model may not be ideal with large numbers of labellers. There are ND parameters, so the expectation-maximisation algorithm may converge considerably slower and the number of local minima may quickly increase with increasing numbers of labellers.

Chapter 4

Automating Radio Cross-identification

In this chapter, we develop a machine learning approach to the radio cross-identification problem.

We first need to formalise the radio cross-identification problem in a machine learning context. This is the focus of Section 4.1, where we cast the problem as an object localisation problem that can be solved with standard classification methods. Section 4.4 discusses how we can represent galaxies as vectors for classification, and Section 4.5 investigates which classification method should be used for this task.

We will also need some benchmark to test our methods against; our approach to evaluation of methods without groundtruth is discussed in Section 4.2. We describe our experimental design in Section 4.3.

As the Radio Galaxy Zoo label set is sourced from volunteers, we wish to investigate some of the many methods for dealing with crowdsourced labels. This is the focus of Section 4.6, where we look at applying majority vote and the Raykar et al. method to the Radio Galaxy Zoo label set.

In Section 4.7 we train our classifiers on expert labels and on the Radio Galaxy Zoo and compare the results. Finally, in Section 4.8, we conclude and suggest some avenues for future work.

4.1 Formalism

We begin by restating the cross-identification problem (Section 2.5). When we look at the sky with radio telescopes, we see the radio emissions from the jets of AGNs. Given an image of these radio emissions, and possibly other images in other wavelengths, we want to locate the host galaxy containing the associated AGN. In general, there may be multiple hosts associated with one radio object (such as in Figure 4.1), but we make the assumption that there is only one. This greatly simplifies the problem.

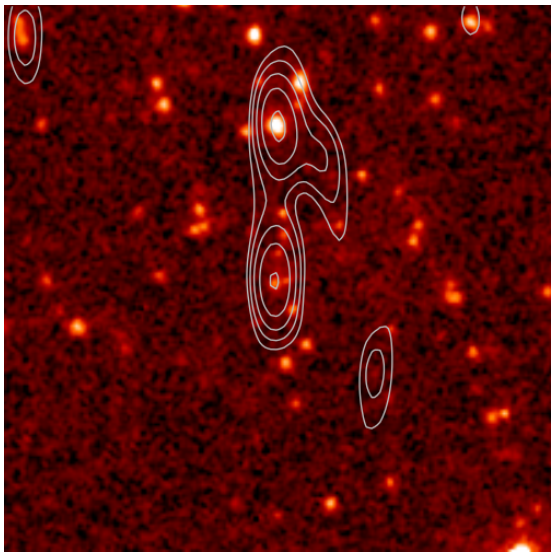


Figure 4.1: A radio object (ARG0003ra1) with two host galaxies. This radio object is actually two radio objects that have been incorrectly detected as one, and there is one host galaxy for each object.

4.1.1 Cross-identification as Object Localisation

We can interpret cross-identification as an object localisation problem. As input, we have an image of the radio sky, and we want to locate a host galaxy in this image. A common way to find an object in an image is by using a sliding window. A fixed-size image patch centred on each pixel is taken as a representation of that pixel. This is then used as input to a classification model which outputs a probability for each pixel, with higher probabilities corresponding to higher likelihood of the object being located at that pixel. The pixel with the highest probability is considered the location of the object.

This may be slow depending on the size of the image. One way to improve upon this approach is to first identify a small number of candidate locations, and then only examine patches around these locations. For the cross-identification problem, we can use galaxy locations as candidate pixels, with galaxies found in infrared surveys such as WISE and SWIRE. Additionally, astronomical measurements such as magnitude are included in these surveys and these may be taken as additional features for each candidate pixel, giving more information to the classifier.

4.1.2 The Galaxy Classification Task

This approach can be formalised as follows. Consider a set $\mathcal{X}$ of candidate host galaxies, and a radio object r that we want to assign a host galaxy. Let $y : \mathcal{X} \to \{0, 1\}$ represent whether a given $x \in \mathcal{X}$ is the host galaxy associated with r. Under the assumption that a radio object has exactly one associated host galaxy, then there exists exactly one $x \in \mathcal{X}$ such that $y(x) = 1$, and for all other $x \in \mathcal{X}$, $y(x) = 0$. The cross-

identification task then amounts to modelling $p(y(x) = 1 \mid x, r)$. Once this distribution is modelled, the host galaxy associated with r is given by

$$\text{host}(r) = \underset{x}{\text{argmax}}\ p(y(x) = 1 \mid x, r). \tag{4.1}$$

Ideally, $\mathcal{X}$ is the set of all galaxies. This is clearly intractable, so as an approximation we use a catalogue of infrared objects near the radio object of interest, taken from an infrared survey. We also make the assumption that the host galaxy is within $1'$ of the radio object — while this doesn't hold in general, systems larger than $1'$ are rare and require human insight to discover [29].

4.2 Evaluating Performance Without Groundtruth

In general, we do not have groundtruth for the cross-identification problem. While the groundtruth exists (a galaxy either has an AGN or does not), it is impossible for us to measure this accurately. Even expert catalogues, such as those provided by Norris et al. (Section 2.5.1), are noisy — [26] estimate that their cross-identifications have a 9.02% false positive rate.

Our primary source of labels for the classification task is the Radio Galaxy Zoo. These labels have been provided by non-experts, and as such are even more noisy. This leads to two problems: A classifier trained on the noisy labels may be inaccurate, and evaluating the performance of a trained classifier against noisy labels will not accurately reflect the true performance of the classifier. We address the former in Section 4.6, and the latter here.

A straightforward way to evaluate classifier performance without groundtruth is to aggregate the Radio Galaxy Zoo crowd labels in some way to approximate the groundtruth, and then use that for evaluation. While there are many ways that we could aggregate the labels (Section 4.6), these aggregates are *not* the groundtruth and may still contain high noise. We want our classifier to be capable of finding the best approximation to the groundtruth, not to the noisy inputs, and evaluation should reflect that. We therefore cannot use aggregated crowd labels for evaluation.

To attempt to mitigate this problem, we combined the Norris et al. catalogue (Section 2.5.1) with the Fan et al. catalogue (Section 2.5.2) to obtain a test set with as little noise as possible. Each galaxy identified as hosting an AGN in the catalogues was matched to the nearest WISE or SWIRE object. Each matched object was then labelled 1. All other galaxies in the infrared survey were labelled 0. This produced two label sets, which we refer to as the Norris et al. labels and the Fan et al. labels respectively. We then took the intersection of these label sets to find a set of galaxies for which the label sets agree. Test sets were drawn only from this set of galaxies. This results in a test set where test instances are "easier" than a test set drawn from the entire data set, but any tests performed against these test sets should be more reliable than would otherwise be the case.

4.3 Experiment Design

To guide our classifier design, we needed to run experiments. This section describes the design and test sets used for the experiments that appear in the following sections.

Note that in the experiments that follow later in this chapter, we have not used SWIRE. AllWISE is the infrared survey that EMU will be cross-identified with, and so we have chosen to focus on it.

To generate the Norris et al. and Fan et al. label sets, we matched each location identified by Norris et al. and Fan et al. to the nearest candidate host in the AllWISE survey. These galaxies were labelled 1. All other candidate hosts were labelled 0.

We generated test sets by first finding all candidate hosts for which the Norris et al. labels and the Fan et al. labels agree. To generate a test set, we followed the following method:

1. Find all ATLAS objects where Norris et al. and Fan et al. agree on the labels of all galaxies within 1′.
2. Draw randomly without replacement 50% of the ATLAS objects.
3. If a candidate host is within 1′ of a drawn ATLAS object, add it to the test set.

This method resulted in a test set with similar class distribution to the true data and no feature overlap between training objects and testing objects. We drew 10 test sets with this method.

To train and test a method, we followed the following method:

1. Choose a test set.
2. Add all candidate hosts not in this test set to the training set.
3. Train the method using the selected training set.
4. Evaluate performance with balanced accuracy against the test set.
5. Repeat for all other test sets.

We report the mean balanced accuracy and standard deviation.

4.4 Feature Selection

To train a machine learning method to solve the galaxy classification task, we must find a representation of galaxies in a feature space $\mathbb{R}^D$. In this section we describe and motivate our choice of galaxy features, extracted from both infrared and radio surveys.

4.4.1 Infrared Features

As described in Section 4.1, we need candidate host galaxies from an infrared catalogue. We have two choices here — the AllWISE catalogue (Section 2.3.1) and the SWIRE catalogue (Section 2.3.2). WISE is lower resolution and less sensitive, but is the survey that will be used to cross-identify EMU (Section 2.4.1) sources when they are available. SWIRE has higher resolution and sensitivity, and thus would give more accurate galaxies, but it does not cover the whole sky. Here, we describe our choice of features for both.

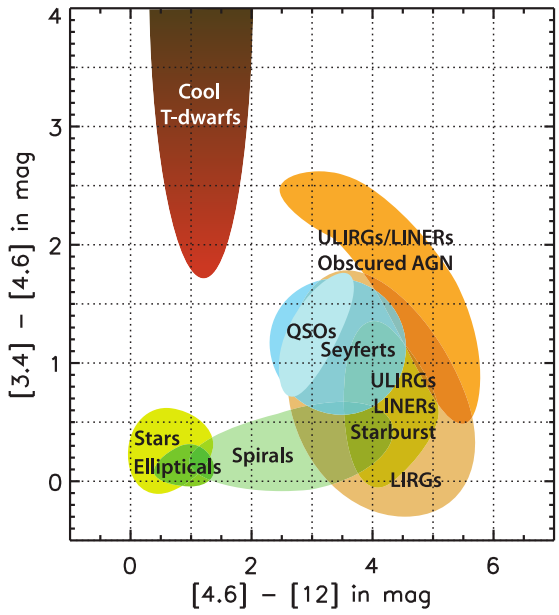


Figure 4.2: WISE colour-colour diagram showing the colours of various astronomical objects. The horizontal axis is $w2 - w3$ and the vertical axis is $w1 - w2$. *Reproduced from Wright et al. [20].*

For WISE, we use all four WISE band magnitudes ($w1, w2, w3$, and $w4$) as features. Since the features are magnitudes, they are on a logarithmic scale, and we found that in practice classification performance improved when the magnitudes were converted to their corresponding flux value. We performed this conversion with the formula

$$f = 10^{-0.4m}. \tag{4.2}$$

This is the inverse of Equation 2.1 with the flux in linear units of f_{Vega} Jy.

The ratios between infrared fluxes are indicators of physical galactic properties such as star formation and dust, and can be used to identify different classes of galaxies [20]. In practice, the $w1 - w2$ and $w2 - w3$ ratios are most commonly used (such as in Figure 4.2), so we chose to use these ratios as features. Once again, we used a linear scale, computing the ratios using equation 2.2 and converting them to linear units with Equation 4.2.

Also available were the unprocessed infrared images captured in the survey. Under the assumption that the large scale infrared structure of a galaxy is unchanged by an AGN, we ignored the images themselves and focused on features obtained from the catalogue. This greatly simplified feature selection with minimal expected impact on the classification performance. Future work may investigate the effect of features extracted from infrared images on classification performance, but this is beyond the scope of this thesis.

SWIRE also contains colour information, but unlike WISE, the information is in the form of fluxes instead of magnitudes. This means that the flux information in

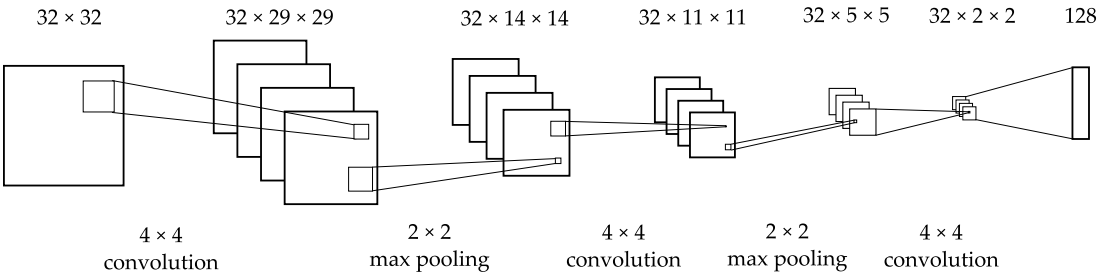


Figure 4.3: Convolutional neural network for radio feature extraction. The leftmost square represents the input image; the rightmost rectangle is the output.

the catalogue can be used as features directly. The ratios can also be used as features: $3.6\ \mu$Jy$/4.5\ \mu$Jy and $4.5\ \mu$Jy$/5.8\ \mu$Jy are the SWIRE equivalents of $w1 - w2$ and $w2 - w3$. Since SWIRE is higher resolution than WISE, the infrared images may contain more useful information and thus be useful for features, but this was not explored in this thesis.

Both WISE and SWIRE catalogues contain the positions of each object. For the final infrared feature, we use the distance from an infrared object to the centre of the closest radio object in ATLAS.

4.4.2 Radio Features

While the infrared catalogues include measurements on individual galaxies, the ATLAS radio catalogue does not. This is because galaxies are not visible in radio, so while galaxies are directly represented in an infrared catalogue they are not in a radio catalogue. Galactic features must thus be extracted from the radio images directly. We use a convolutional neural network for this purpose, as described in Section 3.2.

We note that there is existing research on feature representations of radio galaxies. Fan et al. [4] used an astronomical model of radio galaxies in ATLAS for cross-identification, and Proctor [3] investigated feature selection for radio galaxies in FIRST using hand-designed features. We do not use these methods here for two reasons: First, we wish to investigate automated feature extraction, and second, we wish to avoid dependence on any specific astronomical model. This is because we expect to find completely new objects in EMU that may not fit existing models, and machine learning algorithms like those we develop here will eventually be applied to EMU [1].

4.4.2.1 Building a Model for Feature Extraction

We chose to use a convolutional neural network (CNN) with three convolutional and max pooling layers, shown in Figure 4.3. This resulted in an 128-dimensional feature vector. For training this network, we added a 64-dimensional dense layer mapping to a 1-dimensional output. 10% of ATLAS objects were selected at random and reserved for training the CNN, to ensure later testing data would not overlap the training set. The entire network was then trained for 10 000 epochs to match the consensus Radio

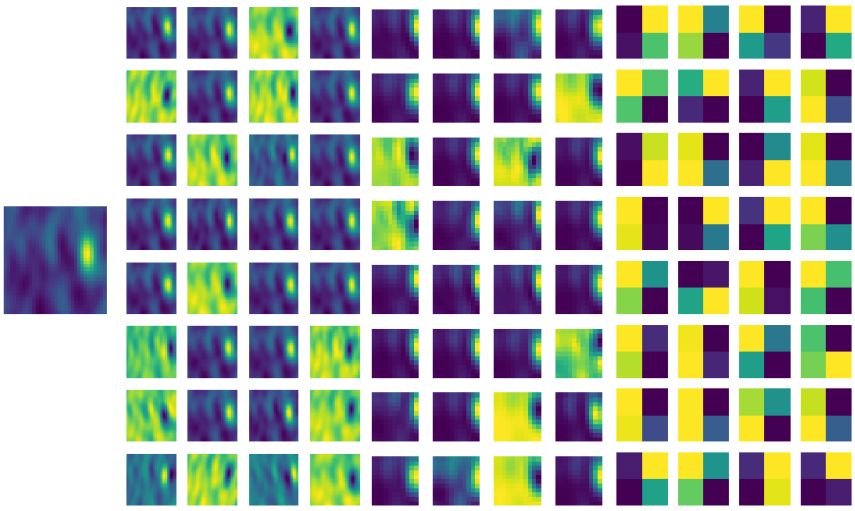


Figure 4.4: The effects of convolutional layers on an input image. The left-most image is the original radio image. The other images are the 32 images output from each of the three convolutional layers.

Galaxy Zoo labels of the galaxies nearest the reserved training set.

A better approach would be to use a convolutional autoencoder, which would allow training on all the radio data and on data with no labels at all, but this was computationally infeasible for this project.

An example of the neural network applied to a radio image is shown in Figure 4.4.

4.4.3 Feature Analysis

To determine the effect of each set of features on classification performance, we performed a feature ablation experiment. We trained a logistic regression classifier on the Norris et al. label set, each time using a different subset of features. The lower the resulting balanced accuracy, the more important the held out features. This was repeated ten times with different 50% subsets of the training set. When holding out the WISE magnitude features, we also held out the CNN features, since we found that the CNN features tended to dominate results. These results are plotted in Figure 4.5 and the means and standard deviations of the balanced accuracies can be found in Table 4.1. It is clear that the features extracted from the radio image are by far the most useful features; removing them causes the balanced accuracy to drop by 17.07 ± 1.88 percentage points. It also seems that some features, such as distance and $w1 - w2$, can act as distractors, though they may be useful features on their own. The inability of our classifier to ignore these features may be due to our use of $L2$ regularisation.

Of particular interest is the effect of the magnitude difference features, as these

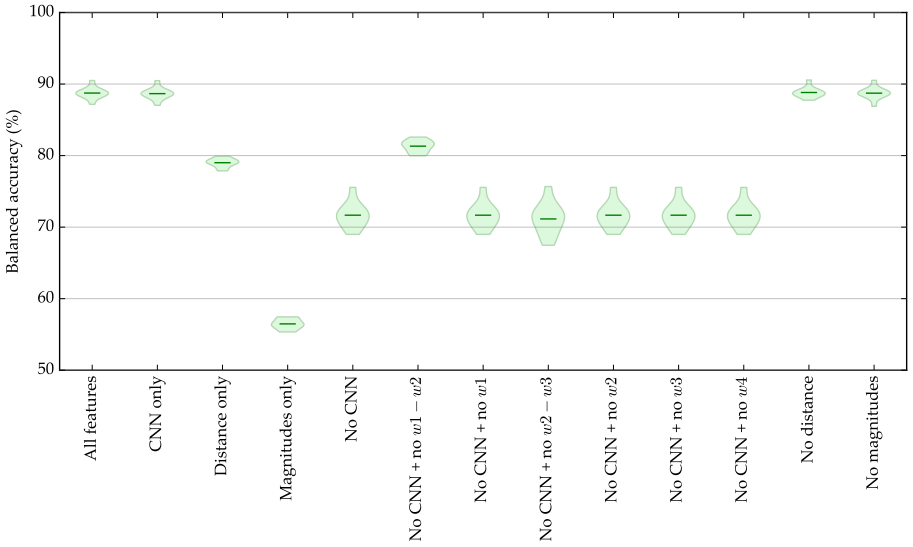


Figure 4.5: Balanced accuracy of logistic regression trained on the Norris et al. label set with different subsets of features. The horizontal axis indicates which features were used for the results in that column. Spread along the vertical axis represents results over multiple test sets; the bar represents the mean.

Features	Mean Balanced Accuracy (%)
No distance	88.82 ± 0.70
All features	88.74 ± 0.78
No magnitudes	88.73 ± 0.83
CNN only	88.66 ± 0.81
No CNN + no $w1 - w2$	81.32 ± 0.81
Distance only	79.01 ± 0.58
No CNN	71.67 ± 1.71
No CNN + no $w1$	71.67 ± 1.71
No CNN + no $w2$	71.67 ± 1.71
No CNN + no $w3$	71.67 ± 1.71
No CNN + no $w4$	71.67 ± 1.70
No CNN + no $w2 - w3$	71.15 ± 2.16
Magnitudes only	56.47 ± 0.70

Table 4.1: Balanced accuracy of logistic regression trained on the Norris et al. label set with different subsets of features. The rows are sorted from highest accuracy to lowest.

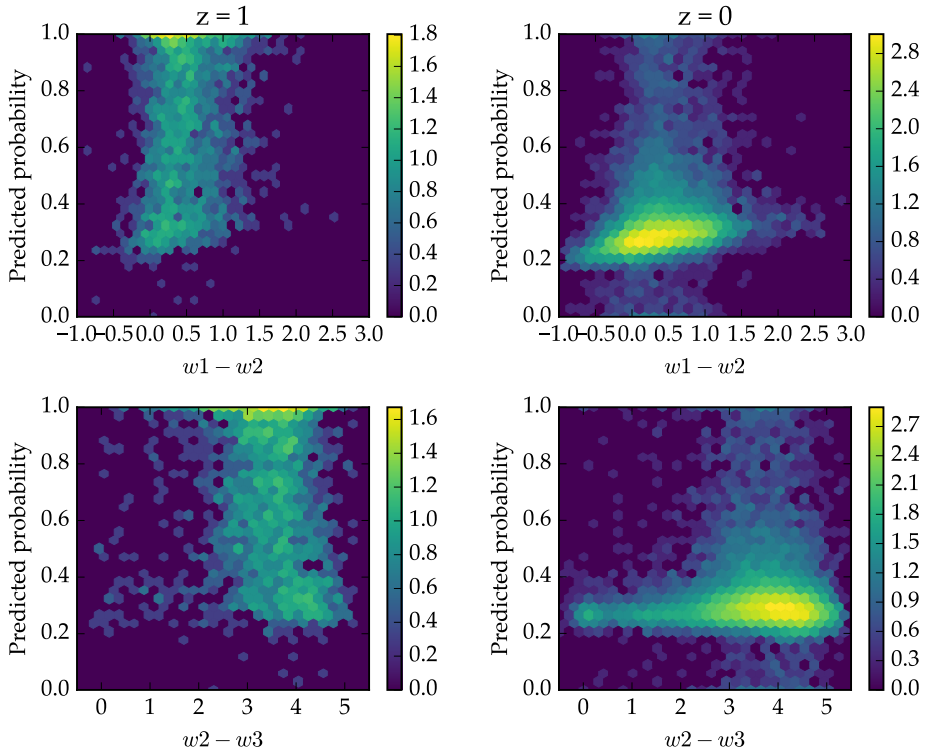


Figure 4.6: WISE magnitude difference features against predicted probability of a galaxy containing an AGN. Colours represent the logarithm of occurrences of instances in each cell.

have significance in astronomy. For this reason, we plot the $w1 - w2$ and $w2 - w3$ features against predicted probability for both positive and negative instances in Figure 4.6. From the plot, we can see that some instances are able to be excluded based on the magnitude differences, though there is significant overlap between positive and negative instances. The magnitude differences are thus weak predictors.

4.5 Choosing a Binary Classifier

To compare the performance of logistic regression with random forests on the galaxy classification task, we performed the following experiment. We first generated five test sets of WISE objects. Then, using all other WISE objects as a training set, we trained a logistic regression classifier with $L2$ regularisation for each test set, separately using labels from both Norris et al. [26] and from Fan et al. [4]. We then computed the balanced accuracy for each classifier. This was repeated for random forests, with each tree using $\sqrt{D}$ features.

We found that logistic regression greatly outperformed random forests. We therefore decided to use logistic regression for later experiments.

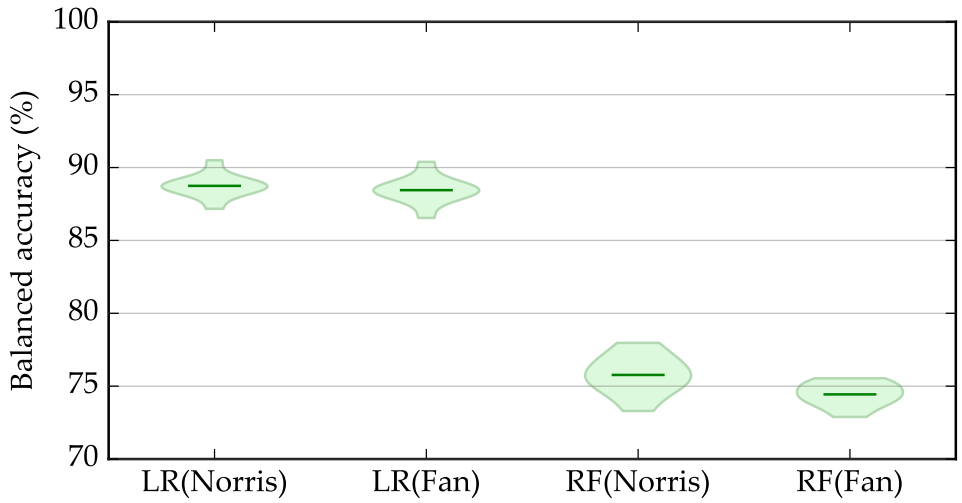


Figure 4.7: Comparison of logistic regression and random forests on the galaxy classification task, trained on the Norris et al. [26] and Fan et al. [4] label sets, and tested against the Norris et al. [26] label set. Spread along the vertical axis represents results over multiple test sets; the bar represents the mean.

4.6 Handling Crowd Labels

While we want to train a classifier on the crowdsourced labels from the Radio Galaxy Zoo, it is unclear how we should aggregate the labels. Some methods for aggregation are described in Section 3.5. Which method will perform best on a dataset is dependent on the dataset itself, so we tested both variants of the Raykar et al. algorithm and majority vote on the galaxy classification task.

One key problem with the use of the Raykar et al. algorithm is that it is very slow with large numbers of labellers. To mitigate this problem, we tested the algorithm on a subset of labellers, chosen by three different measures: the 15 labellers with the most labels, the 15 labellers with the highest balanced accuracy assessed against the Norris et al. labels, and the 15 labellers with the highest balanced accuracy assessed against the majority vote of all labellers. While in practice on datasets such as EMU or FIRST there would be no equivalent to the Norris et al. labels, we can treat this test as a "best-case" method. For each subset of labellers, the Raykar et al. algorithm was tested against logistic regression trained on the majority vote of just these labellers. The results are shown in Figure 4.8 and Table 4.2.

With $T = 10$ labellers, the Raykar et al. algorithm took 83 ± 36 seconds to run. When the number of labellers was increased to $T = 15$, the Raykar et al. algorithm took 152 ± 115 seconds to run. This helps to highlight how slow the algorithm becomes with increasing T.

We also tried using all of the crowd labels directly with logistic regression, without aggregation. In principle, training logistic regression on such a label set may average

Method(Training Set)	Mean Balanced Accuracy (%)
Raykar(Top-15-prolific)	46.34 ± 18.45
LR(Top-15-prolific-MV)	51.98 ± 3.87
Raykar(Top-15-accurate)	86.43 ± 0.10
LR(Top-15-accurate-MV)	86.76 ± 1.01
Raykar(Top-15-est-accurate)	87.25 ± 0.74
LR(Top-15-est-accurate-MV)	87.25 ± 0.90

Table 4.2: Comparison of logistic regression and Raykar et al. on the galaxy classification task. These tests use only the 15 best labellers, with "best" defined in three different ways: Total number of galaxies labelled (Top-15-prolific), balanced accuracy assessed against the Norris et al. labels (Top-15-accurate), and balanced accuracy assessed against the Radio Galaxy Zoo majority vote (Top-15-est-accurate). Logistic regression (LR) is trained on the majority vote of the training labels; the Raykar et al. method (Raykar) is trained on the labels themselves. Uncertainties are standard deviations across multiple test sets.

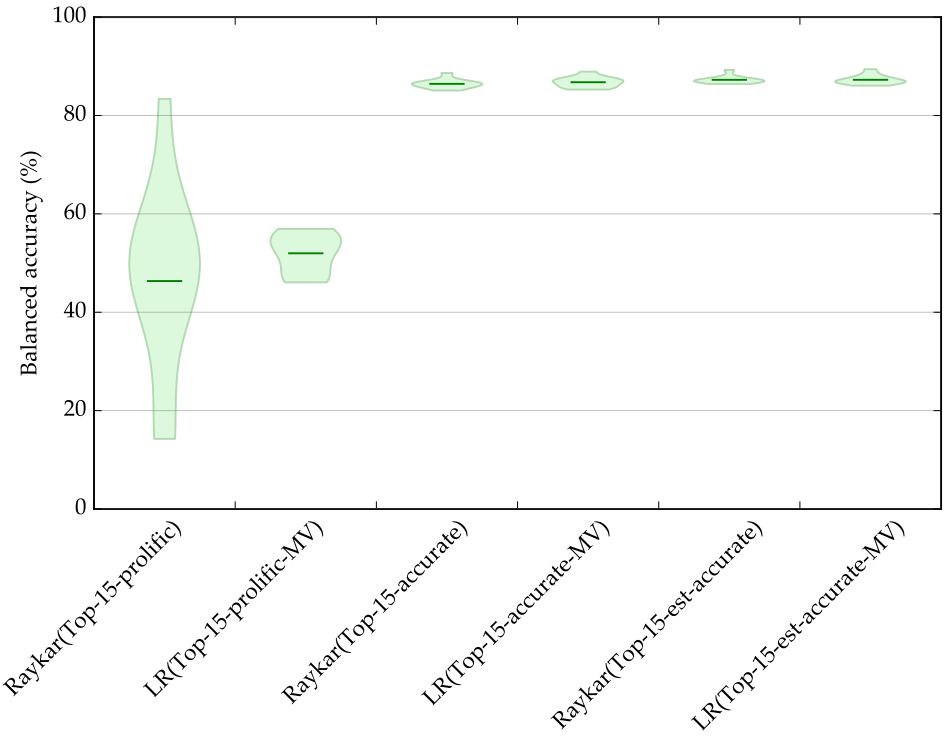


Figure 4.8: Comparison of logistic regression and Raykar et al. on the galaxy classification task. Methods are denoted as in Table 4.2. Spread along the vertical axis represents results across multiple test sets; the bar represents the mean.

Method(Training Set)	Mean Balanced Accuracy (%)
LR(Norris)	88.74 ± 0.77
LR(Fan)	88.44 ± 0.89
LR(RGZ-MV)	87.17 ± 0.90
Raykar(RGZ-Top-50)	85.89 ± 1.16
RGZ-Raw-MV	85.89 ± 0.00

Table 4.3: Performance of logistic regression and the Raykar et al. algorithm on the galaxy classification task, as in Figure 4.9. Uncertainties are standard deviations.

over the conflicting labels and give reasonable results. This was not the case in practice: We found that this method resulted in a balanced accuracy of 50% (i.e. random chance) as the classifier learned to always output 0.

4.7 Galaxy Classification

In this section, we present results using methods developed in this chapter.

4.7.1 Comparison of Methods

We trained logistic regression classifiers on three training sets:

- the Norris et al. labels,
- the Fan et al. labels, and
- the Radio Galaxy Zoo majority vote.

We also trained the Raykar et al. algorithm with the Radio Galaxy Zoo labels. Each volunteer was assigned a unique index and the algorithm was run with the best $T = 50$ labellers, computed against the Radio Galaxy Zoo majority vote. Finally, we compared the predictions of all classifiers to the Radio Galaxy Zoo majority vote. These results are plotted in Figure 4.9 and shown in Table 4.3. The average confusion matrices normalised by class size are shown in Tables 4.4 – 4.8.

The first observation we make is that logistic regression on our feature set does reasonably well. The Norris et al. labels we are testing against are expected to have an error of around 9%, so accuracies upwards of 85% seem reasonable.

The second observation we make is that logistic regression trained on the majority vote outperforms the Raykar et al. classifier which makes use of the labels with no aggregation. This could occur for three reasons: We could have found a local minimum in the expectation-maximisation algorithm, using a subset of labellers instead of all labellers may lower performance, or the Raykar et al. model may simply perform worse at this task than majority vote. Regardless, logistic regression and majority vote are very simple; these results indicate that perhaps we do not need a complicated algorithm for aggregating the Radio Galaxy Zoo labels.

The third observation we make is that logistic regression trained on the Radio Galaxy Zoo labels performs comparably to logistic regression trained on the expert

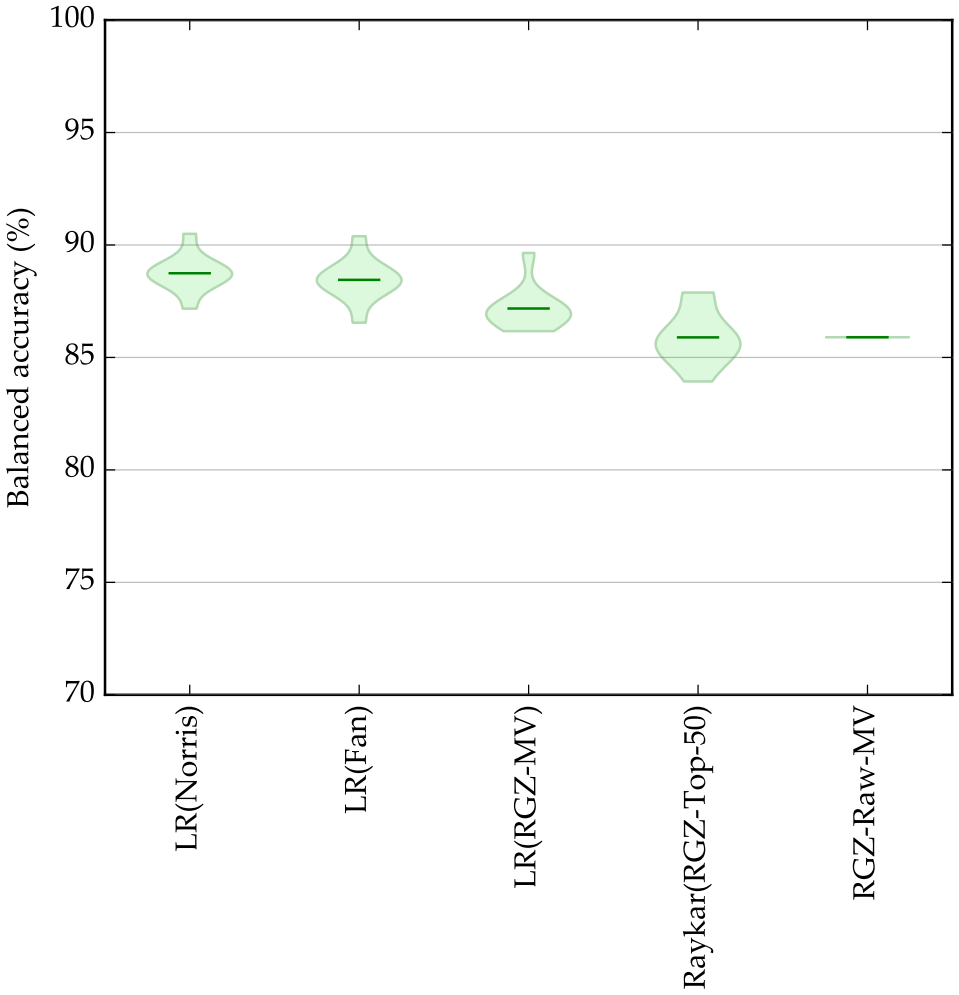


Figure 4.9: Performance of logistic regression and the Raykar et al. algorithm on the galaxy classification task, trained on different sets of labels and tested on the galaxies where Norris et al. and Fan et al. labels agree. LR(Y) indicates logistic regression trained on Y. RGZ-MV is the training set of majority vote labels from the Radio Galaxy Zoo. RGZ-Raw-MV is the majority vote of all crowd labels and is included for comparison. Spread along the vertical axis represents results across multiple test sets; the bar represents the mean.

	$y = 0$ (%)	$y = 1$ (%)
$z = 0$	94.57 ± 0.28	5.43 ± 0.28
$z = 1$	17.08 ± 1.69	82.92 ± 1.69

Table 4.4: Confusion matrix for logistic regression trained on Norris et al..

	$y = 0$ (%)	$y = 1$ (%)
$z = 0$	95.93 ± 0.26	4.07 ± 0.26
$z = 1$	19.03 ± 1.94	80.97 ± 1.94

Table 4.5: Confusion matrix for logistic regression trained on Fan et al..

	$y = 0$ (%)	$y = 1$ (%)
$z = 0$	90.60 ± 0.55	9.40 ± 0.55
$z = 1$	16.25 ± 1.72	83.75 ± 1.72

Table 4.6: Confusion matrix for logistic regression trained on the Radio Galaxy Zoo majority vote.

	$y = 0$ (%)	$y = 1$ (%)
$z = 0$	87.88 ± 1.34	12.12 ± 1.34
$z = 1$	16.09 ± 1.42	83.91 ± 1.42

Table 4.7: Confusion matrix for the Raykar et al. algorithm trained on the Radio Galaxy Zoo labels.

	$y = 0$ (%)	$y = 1$ (%)
$z = 0$	92.47	7.53
$z = 1$	20.67	79.33

Table 4.8: Confusion matrix for the Radio Galaxy majority vote.

labels. This is a very good sign for citizen science, indicating that perhaps expert labels are not necessary, and supervised learning works just as effectively with crowd labels as with expert labels. However, it is hard to tell if this is a real effect; with our features set, logistic regression seems to have a maximum accuracy around 90%, and all classifiers attain accuracies close to this. It is possible that with better features, training on expert labels may give significantly better results than training on crowd labels.

It is interesting to note that the Raykar et al. classifier tends to overpredict positive labels when compared to the other methods. This may be a side-effect of the labeller model, or perhaps the best annotators are more "over-eager" and overpredict positive labels themselves.

The final observation we make is that logistic regression trained on the Radio Galaxy Zoo majority vote outperforms the Radio Galaxy Zoo majority vote. This could be coincidence, or it could indicate some amount of noise reduction from applying logistic regression — for example, our logistic regression model was trained with $L2$ regularisation, which could "smooth over" noise.

4.7.2 Raykar-estimated Labeller Accuracies

The Raykar et al. algorithm estimates a labeller model parametrised by α and β (Section 3.5.2). In this section, we look at the estimated values of α and β for the 50 best Radio Galaxy Zoo volunteers, and compare these to their "true" values.

We computed the true α and true β values for each labeller empirically, with α_t given by the true positive rate of labeller t and β_t given by the true negative rate of labeller t. There were no labellers in the best 50 for whom these rates could not be calculated (i.e. the best 50 labellers had always labelled at least one positive and at least one negative galaxy).

To assess the correlation between the true and predicted values of α and β, we computed the Pearson and Spearman correlation coefficients between the true and predicted values using `scipy.stats` [47].

According to the Pearson correlation coefficient, the true and predicted values of α were weakly correlated, with $r = 0.11$; the true and predicted values of β were not, with $r = -0.01$. According to the Spearman correlation coefficient, the true and predicted values of α were weakly correlated, with $r = 0.32$; the true and predicted values of β were also correlated, with $r = 0.21$.

We note that the Pearson correlation coefficient assumes that the values being tested are normally distributed, which the α and β values are not.

These results may imply that the labeller model does not accurately describe the Radio Galaxy Zoo volunteers. It is possible that a data-dependent model like that proposed by Yan et al. may provide a better model, but due to the difficulty and slow speed of trialling labeller models with many parameters, we do not further investigate this problem in this thesis.

4.8 Conclusion and Future Work

In this chapter, we cast the radio cross-identification problem as a classification problem, and developed a classifier to solve it. We represented galaxies as a combination of astronomical features and features extracted from radio images. We used logistic regression on three different sets of training labels and compared the performance of these to the Raykar et al. crowd learning model, finding that logistic regression outperformed the Raykar et al. model. Additionally, we found that training logistic regression on non-expert labels results in performance comparable to training on experts, showing that efforts to crowd-label scientific data may generalise just as well as expert labels when machine learning is applied.

For the remainder of this section, we will briefly discuss the problems in our method and highlight some avenues for future work.

A clear place that our methods could be improved is in our feature selection. Other studies, such as those by Proctor [3] and Fan et al. [4], have made use of hand-selected features for radio classification. These could possibly be incorporated into the galaxy feature set (though as these hand-selected features are features of radio objects rather than of galaxies, doing so may be non-trivial). We may also be able to make use of information from catalogues at wavelengths other than infrared. In particular, we may be able to make use of information from the ATLAS catalogue, which includes values such as the rotation of extended radio sources.

Our approach to image feature extraction was very simple. We did not fully investigate all of the many different possible architectures for a convolutional neural network, and it is possible that better networks exist for radio data. Improvements could be made in our parameters (filter size, pooling size, etc.) and in our network topology (number of layers, structure of layers, etc.). We could also rescale and rotate training images to ensure rotational and scale invariance in our convolutional features. This was successfully applied to galaxy morphology prediction using Galaxy Zoo data by Dieleman et al. [48], though we note that these results were obtained in optical wavelengths where resolution is higher and images are clearer.

There may be useful information to extract from the infrared images, particularly in higher resolution surveys like SWIRE. As preliminary results suggested there were no strong features in the infrared images, we ignored them for this work, but this was only a passing investigation of infrared features. One possible avenue for investigation is to include infrared *and* radio images as different channels in our convolutional neural network. These networks are capable of supporting multiple colour channels and radio and infrared are effectively colour channels. This would also allow the full use of multi-wavelength data from telescopes like WISE and Spitzer.

An obvious drawback of our approach to object localisation is that infrared-faint radio objects are not able to be classified. There is no clear way around this without dramatically increasing the running time of our method. A possible option is to allow the classifier to output an "undecided" choice when there is no clear infrared counterpart to a radio object. These radio objects could then be investigated by hand, or distributed to a different algorithm.

We note that the Raykar et al. model performed relatively poorly on the galaxy classification task. In Section 4.7.2 we suggested some reasons that this might be the case. Further investigation of these results is needed, in particular whether using a different labeller model would improve results. One such model would be the Yan et al. model, but as we noted in Section 3.5.3 this is very slow and seems to yield mediocre results for large numbers of annotators. Careful analysis of the Radio Galaxy Zoo data set may reveal "clusters" where labellers perform better or worse, and these patterns could be exploited to develop a labeller model that requires less parameters than simply using all features.

Finally, in this work we only investigated the cross-identification problem on a per-object basis. We did not investigate the problem of determining which radio objects are associated with each other, e.g. detecting when two radio objects are simply two jets of the same object. This is potentially a much harder problem than cross-identification, yet it is important for astronomy. One possible approach is to use a classifier like we have developed, use this to identify galaxies that may contain AGNs, and then search around those galaxies for radio objects. This is similar to the approach taken by Fan et al. [4], though their algorithm was applied across the entirety of the CDFS field.

Chapter 5

Active Learning

In this chapter we discuss active learning and look at its application to both the galaxy classification task and to the Radio Galaxy Zoo project. In Section 5.1, we will briefly describe the key concepts of active learning, moving to discuss querying strategies for active learning for binary classification in 5.2. We apply these methods in a simple experiment in Section 5.3, showing that active learning may be useful in an astronomical context.

In Section 5.4 we extend the active learning discussion to a crowdsourcing context, and discuss how crowdsourcing complicates active learning. We propose an experiment for applying active learning to the Radio Galaxy Zoo in Section 5.5, highlighting some difficulties in doing so. Finally, in Section 5.6 we look at the differences between the standard crowdsourcing context and citizen science, pointing out how they differ and how this breaks assumptions of existing methods.

5.1 Introduction

In supervised learning we deal with a set of data points and their associated labels. This dataset may be expensive to obtain, but the main costs may come from collecting labels, rather than from collecting the data points themselves. Examples of such data include text samples [49, 50], and images [51, 6], both of which are now widely and cheaply available through the internet. A more abstract example is scientific hypotheses [52]. Labelling text and images is hard, error-prone, and requires humans; and performing a scientific experiment to test a hypothesis is considerably more expensive than coming up with the hypothesis. It may even be the case that we simply cannot label all the data because there is too much, such as in the Galaxy Zoo [6] and Radio Galaxy Zoo [1] projects.

Active learning (or *query learning* [53, 54, 55]) allows a machine learning algorithm to select specific, unlabelled examples to be labelled by an expert. The algorithm effectively chooses its own training set [53]. The hope is that the algorithm chooses to label only the most useful examples [50], and the expensive process of labelling redundant or useless examples is avoided [56]. Intelligently selecting the training set as in active learning can result in massively reduced labelling costs [49, 52] or even make intractable labelling problems tractable.

While there are many variations of active learning scenarios, we focus on *pool-based* active learning in this thesis. In pool-based active learning, we already have a large pool of unlabelled data points accessible to our algorithms, and our algorithms can choose to present any of these data points to the expert. The pool-based scenario commonly arises when we are able to obtain a lot of unlabelled data at once, such as in astronomical surveys [57, 58, 43].

Active learning has already been successfully applied in astronomy. Pelleg and Moore [57] applied active learning to the Sloan Digital Sky Survey to find anomalies in the survey. Richards et al. [58] applied active learning to classify variable stars from the All Sky Automated Survey. Both papers showed that active learning resulted in a great reduction in the number of labels needed to achieve their respective tasks.

5.2 Query Strategies

A *query strategy* is the approach an active learning algorithm takes to selecting a new data point to label. There are many different query strategies, but here we focus on uncertainty sampling and query-by-committee.

All pool-based query strategies take the same form. We are given some pool of data $\mathcal{X}$ and a set of labelled data $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. We want to select $\tilde{x} \in \mathcal{X}$ such that labelling $\tilde{x}$ maximises our information gain.

5.2.1 Uncertainty Sampling

Uncertainty sampling [49] is perhaps the most common query strategy. Given a classification model $y(\vec{x}) = p(z \mid \vec{x})$ with the ability to output a probability (including probabilistic classifiers like logistic regression, nearest-neighbour classifiers [49], and combinations of probabilistic and non-probabilistic classifiers [59]), the queried point $\tilde{x}$ is the data point for which the model is least certain of the classification. This is not well-defined and an uncertainty sampling algorithm must choose what "least certain" means. There are three common measures of uncertainty — confidence-, entropy-, and margin-based — but in the case of binary classification, they all reduce to one strategy [53]:

$$\tilde{x} = \underset{\vec{x}}{\operatorname{argmax}} \ (0.5 - |y(\vec{x}) - 0.5|) \,.$$

The intuition is that the further a data point is from the decision boundary, the more certain the classifier is of the assigned label, so choosing the closest data point to the decision boundary is equivalent to choosing the most uncertain data point. Another interpretation is that $0.5 - |y(\vec{x}) - 0.5|$ is the expected probability of mislabelling $\vec{x}$ [53].

5.2.2 Query-by-Committee

Query-by-committee (QBC) is an ensemble-based query strategy first proposed by Seung et al. [54]. A committee of classifiers is trained on the known labels, with different

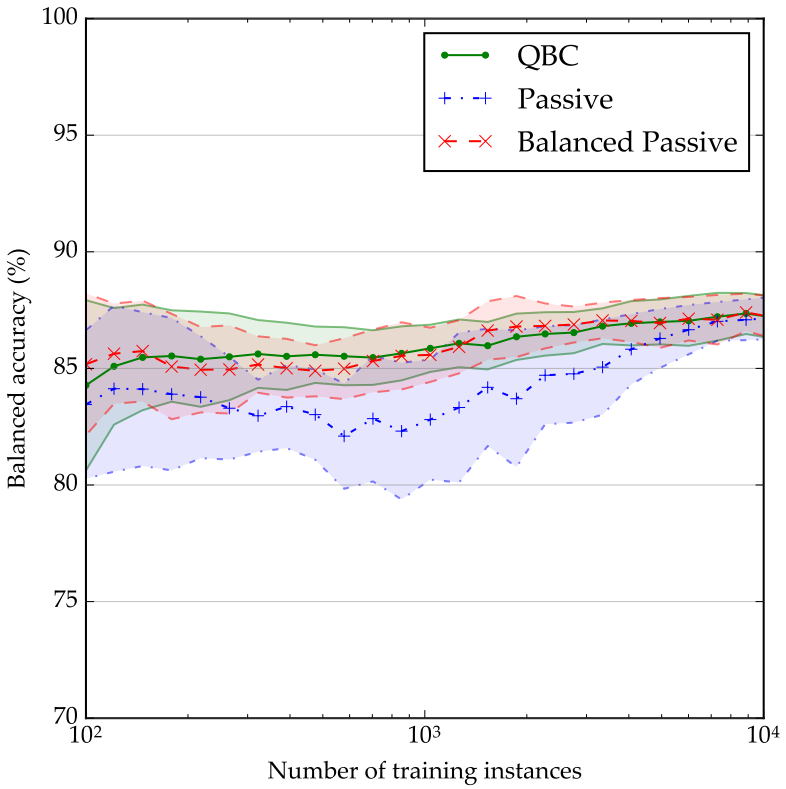


Figure 5.1: Logistic regression trained on the Norris et al. labels with different amounts of training data and two different query strategies. The filled areas represent standard deviation across multiple test sets.

subsets of the labelled data to ensure variety in the committee. The committee then labels the unlabelled pool of data: Each classifier votes on each data point and our prediction is the label with the most votes. The information gain associated with each data point is estimated by the disagreement of the committee on the label, and the data point with the most disagreement is queried.

Disagreement can be measured in multiple ways. The most obvious way is to simply count the number of classifiers that disagree with the majority label [54]. Other methods include computing the entropy of the committee vote [50, 60], and using Kullback-Leibler divergence [50].

5.3 Query-by-Committee on the Galaxy Classification Task

We tested QBC active learning on the galaxy classification task described in Chapter 4, comparing QBC to passive (i.e. random) selection as a query strategy.

We used a committee of 20 logistic regression classifiers for the QBC test. Each was

presented with 75% of the known labels at random, stratified by the labels.

For the passive test, we sampled 100 galaxies at random (stratified by the labels) and trained a logistic regression classifier on these. We then drew a batch of new labels, added these to the existing label set, and retrained the classifier. This was repeated until the classifier had seen the entire training set (10^4 labels). The process for testing QBC was identical, except that instead of drawing new labels at random, the new labels were drawn in order of highest to lowest disagreement of the committee. Disagreement was measured by counting the number of committee labels that disagreed with the majority vote of the labels.

After running the experiment, we observed that QBC outperformed passive selection. We hypothesised that this was because querying at random ignores the fact that there are far more negative examples than positive examples in the galaxy classification task. By this hypothesis, QBC would perform comparably to sampling from the set of positive examples and the set of negative examples at equal rates. To test this, we ran a third test with a random sampler that accounted for class imbalance. We found that this third test performed similarly to QBC.

All three tested querying strategies are plotted in Figure 5.1.

5.4 Active Learning on Crowds

Traditional active learning assumes that we have access to one expert, who always issues correct labels. When labels are sourced from a crowd, these assumptions no longer hold: The crowd are non-experts and can give incorrect labels [61, 62], and there are multiple labellers with different accuracies [62]. We can now ask questions deeper than simply "which label should I request?" — we can, for example, ask "which labeller should I ask?", or "do I need to re-request this label?".

Yan et al. [62] apply the Yan et al. [9] model (Section 3.5.3) to the problem of active learning from crowds. We remind the reader that this model consists of a label model $p(z \mid \vec{x})$ and a data-dependent labeller model $p(y_t \mid \vec{x}, z)$, where $\vec{x}$ is an instance, y_t is the label assigned by labeller t, and z is the groundtruth label. To extend this model into active learning, Yan et al. introduce a query strategy where not only an instance is requested, but also a specific labeller.

First, uncertainty sampling is used with the label model to choose a set of ideal points to query. With logistic regression (Equation 3.2), the decision boundary between positive and negative labels is a hyperplane $\vec{w}^T\vec{x} = 0$; uncertainty sampling would choose to query the points nearest (or on) this hyperplane. The labeller and instance to query are then chosen as solutions to the following optimisation problem:

$$\begin{aligned} &\underset{\vec{x},t}{\text{minimise}}\ \eta_t(\vec{x}) \\ &\text{s.t. } \vec{w}^T\vec{x} = 0. \end{aligned}$$

Intuitively, we query the instance on the decision hyperplane with the least noisy labeller. While this instance may not actually exist in our pool, we simply choose the

instance closest to it (i.e. using Euclidean distance).

This method has similar drawbacks to the Yan et al. passive learning method described in Chapter 3: The number of parameters grows large with large numbers of annotators, and the expectation-maximisation algorithm only converges to a local minimum. In our implementation, training was also quite slow, meaning online active learning may be impractical. It also does not account for the possibility of relabelling instances.

Mozafari et al. [61] suggest an approach similar to uncertainty sampling, with uncertainty computed using a bootstrap method. A full description of this method is beyond the scope of this thesis. Instead, we look at the approach they take to handle noise. Noting that crowds may perform worse on some subsets of the data than other subsets, Mozafari et al. solve an integer linear program to compute the redundancy required for different subsets of the instance pool. First, they partition the data, then estimate the probability of obtaining a correct crowd label for each partition. This estimation is accomplished by querying the crowd on a sample of instances from each partition. The estimated probability is then used to compute the redundancy required. For full details, we refer the reader to the paper [61].

5.5 Active learning for Radio Galaxy Zoo

Radio Galaxy Zoo [1] is a domain where active learning could be very useful. There are many more radio galaxies to classify than there are volunteers, so making the best use we can of their labels is particularly important.

Applying active learning to the task that Radio Galaxy Zoo presents to their volunteers is non-trivial. Volunteers first choose which radio components are associated with the same AGN, and then decide where this AGN is located on an infrared image. Even ignoring the radio components problem, the position is a real-valued vector label, rather than a binary label. In this thesis, we have cast the cross-identification as a binary classification problem, converting these real-valued positions into binary labels by assigning a 1 to the closest galaxy and 0 to all other galaxies. This should still work for active learning, but there is no obvious way to develop a query strategy.

Volunteers are presented with an image of radio object to label, so a query strategy must choose radio objects to present. Methods like uncertainty sampling have no clear application here: How do we aggregate uncertainties in our classifications of neighbouring galaxies into an uncertainty for a radio object? We may be able to perform this aggregation by a number of methods, such as summing, averaging, or maximising the uncertainties. We could even aggregate using something like entropy, looking at the distribution of uncertainties of galaxies in the image. The choice is not obvious and an experiment is required. While we do not perform such an experiment in this thesis, we will suggest one at the end of this section.

Query-by-committee may generalise to Radio Galaxy Zoo. We could label radio objects by considering nearby galaxies and classifying them using our methods from Chapter 4, then selecting the galaxy with the most certain classification as the host

galaxy of the radio object. Multiple classifiers could be used to perform this selection, and the percentage disagreement on the location of the host galaxy would then indicate the uncertainty associated with the radio object. However, this method would likely find radio objects that are intrinsically hard to cross-identify rather than radio objects where labelling would give high information gain. An example of this would be a compact radio object with multiple potential host galaxies very close to its centre. Cross-identifying such an object is usually very easy — the galaxy closest to the centre is most likely to be the host galaxy — but if there are multiple galaxies equidistant from the centre then a committee of classifiers will likely choose equally between them, resulting in high disagreement. In preliminary experiments with using QBC for this task, we found exactly this behaviour.

An ideal experiment for active learning for Radio Galaxy Zoo would compare different generalisations of uncertainty sampling to QBC and passive sampling. In contrast to our approach to the cross-identification problem, instances would be radio objects rather than galaxies. Radio objects would be drawn using the different query strategies. The results of queries would come from an expert catalogue such as the Norris et al. catalogue. The resulting labels would then be used to train a classifier. Training is possible using our methods by assigning all nearby galaxies a positive or negative label based on the result of the query, in a very similar way to how we converted the Norris et al. and Fan et al. cross-identifications into label sets. There is a clear problem of scale — how large a radius around the radio object do we consider "nearby"? — and there is no clear solution to this problem.

After such an experiment was used to determine good query strategies, the approach would need to be extended to work with the crowd. Queries would now be sent to a simulated crowd with realistic (i.e. Radio Galaxy Zoo volunteer-like) noise. This is a much harder problem to solve: One must now consider the problem of relabelling, label noise, and so on. Inspiration for such methods could likely be drawn from Mozafari et al. [61], as their partitioning approach is similar to that chosen by Banfield et al. [1] for the Radio Galaxy Zoo (where the partitions are compact and complex radio objects).

Reviewing these suggested experiments, it is clear that active learning for Radio Galaxy Zoo is a very hard problem to solve! As such, these experiments were beyond the scope of this thesis, but we hope that our classification methods may be applied in an aggregate way to radio objects in future work.

5.6 Active Learning for Citizen Science

Thus far, we have taken the word "crowdsourcing" to mean any scenario where there are multiple labellers. This conflates a number of arguably different scenarios: We may have multiple experts, multiple non-experts with domain knowledge, multiple non-experts without domain knowledge, and so on. The literature often does not disambiguate. Raykar et al. [8] consider the problem of multiple experts who disagree; Yan et al. [9] and Mozafari et al. [61] consider crowdsourcing using a platform such

as Amazon Mechanical Turk, where non-experts are paid to label data on a per-label basis. Nguyen et al. [63] consider a hybrid scenario where we have access to both non-experts *and* experts.

The specific scenario we are interested in is citizen science, where volunteers interested in science contribute to labelling scientific data, typically through web interfaces like Galaxy Zoo [7]. With the rise of internet usage and the amount of available data throughout scientific disciplines, citizen science is steadily increasing in popularity and impact [43]. We believe that citizen science is a crowdsourcing scenario unlike those presented in existing literature, and as such, breaks assumptions often made by active learning methods that operate on crowds.

There is a key difference between citizen science and paid crowdsourcing: In citizen science, we cannot choose our labellers. Using a paid crowdsourcing platform, we have some degree of freedom over who we query; this is (for example) the assumption made in Yan et al. [62] and the crux of their methods. We can choose the best labeller and the best instance to query. In citizen science, volunteers request labels from us.

Another factor is that citizen science tends to involve a very large number of labellers: Galaxy Zoo has involved several hundred thousand volunteers [43]; Radio Galaxy Zoo has over 2000 volunteers involved with just the ATLAS–CDFS observations that we have looked at in this thesis. For contrast, Raykar et al. [8], on whose methods we based our own crowd experiments, used 5 labellers for most of their experiments (with one experiment with 164 labellers); Mozafari et al. [61] and Yan et al. [62] both tested their methods with 3–5 labellers. The large number of labellers involved in practical citizen science raises many challenges. In particular, estimating labeller accuracies is very difficult for large numbers of labellers as the number of model parameters is often tied to the number of labellers.

It may also be difficult to estimate the accuracies of individual volunteers, for a number of reasons. These volunteers may be anonymous, and identifying which labels were assigned by the same individual may be impossible. Indeed, Radio Galaxy Zoo reports that 27% of their labels come from anonymous volunteers.

Finally, we note that volunteers in citizen science vary greatly in how many instances they label. A large number of labels come from volunteers who only label a few instances. This means that trialling volunteers on a "gold standard" of instances with known groundtruth is impractical: If we tested every volunteer on a few known instances, then for many volunteers, the known instances would be all that they label! Yet we cannot treat all volunteers as if they only label a few instances — large contributions are also made by small numbers of volunteers labelling many instances. Marshall et al. [43] highlight this variety with respect to Galaxy Zoo 2. They emphasise the importance of designing projects for both recurring and new volunteers, citing the significant contributions of both groups. This can be visualised by Figure 5.2, which depicts Galaxy Zoo 2 volunteer contributions. We would like to extend their statement: Just as we must design citizen science projects for both recurring and new volunteers, we must design citizen science *algorithms* for both recurring and new volunteers.

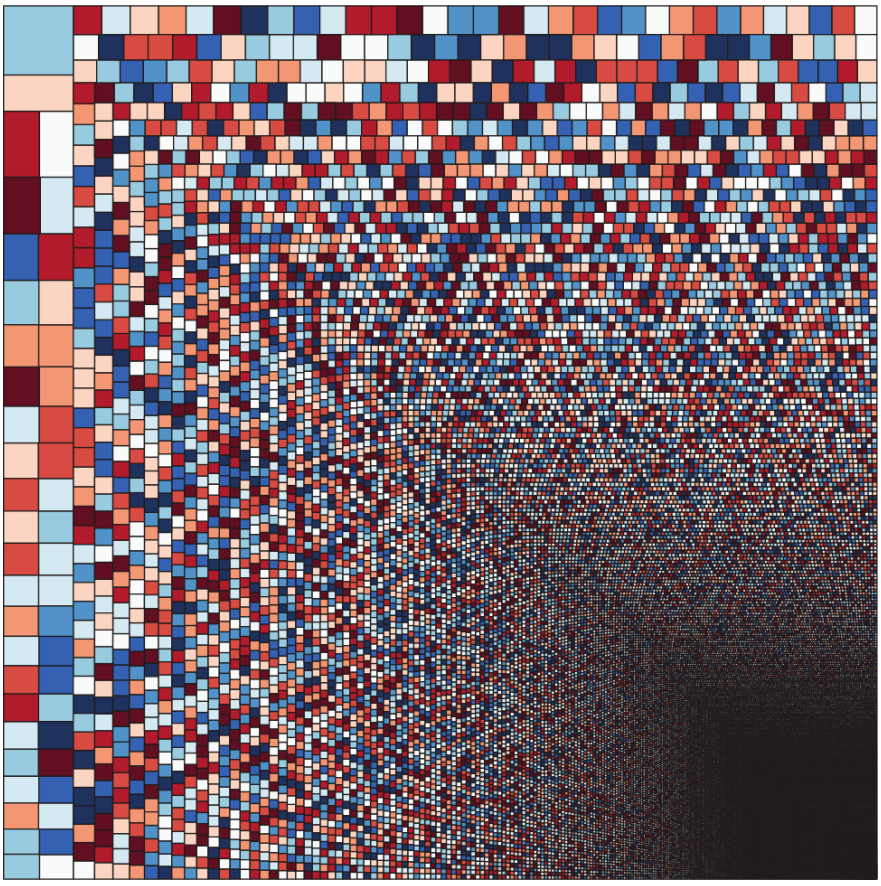


Figure 5.2: Label contributions from volunteers in Galaxy Zoo 2. Each square represents a single user, with the area of the square proportional to the number of instances labelled by that user. Colours are arbitrary. *Image: K. Willett. Reproduced from Marshall et al. [43].*

Chapter 6

Conclusion

Ever larger and ever more detailed radio surveys will bring new challenges to astronomy. In this thesis we have focused on radio cross-identification, a task for which existing algorithms are expected to fail and for which a manual approach is intractable. We have presented a new, astronomical model-free, supervised learning approach to this problem. This approach will hopefully help guide the search for innovative ways to handle the data produced by the Evolutionary Map of the Universe.

Chapters 2 and 3 introduced concepts from astronomy and machine learning respectively. These were then brought together in Chapter 4, where we framed the cross-identification problem first as object localisation and then as binary classification of galaxies. We represented galaxies with the use of a convolutional neural network to extract features from ATLAS radio images, and combined these with features from the WISE telescope. We then tested a variety of classification methods with these features, including logistic regression, random forests, and the Raykar et al. [8] crowd learning algorithm. The results showed that logistic regression with majority vote outperformed the Raykar et al. algorithm, and that a classifier trained on non-expert crowd labels attains comparable accuracy to a classifier trained on expert labels.

Chapter 5 discussed active learning, applying query-by-committee to the cross-identification task in a simple experiment. We then discussed applications of active learning to citizen science.

Our work here is just the start of applying machine learning to radio cross-identification. In Section 4.8 we have suggested a number of avenues for future work. Performance could be improved by incorporating hand-selected features, using information from more wavelengths, and developing better feature extraction methods. Our method could be extended to account for infrared-faint radio objects. Machine learning could be applied to learn to associate related radio objects with each other, and then with their host galaxy. Radio Galaxy Zoo volunteers could be analysed to determine the best possible labeller model for use in crowd learning algorithms. In Section 5.5 we suggested an experiment for applying active learning to Radio Galaxy Zoo and determining an appropriate uncertainty aggregation method for the radio cross-identification task. Finally, in Section 5.6, we highlighted some problems with active learning and crowd learning when applied to citizen science, suggesting that citizen science is a unique form of crowdsourcing that requires additional considerations.

Appendix A

Crowdastro Package

As part of this thesis, we developed an open source Python package called *crowdastro*, containing methods for machine learning on the cross-identification task, and implementations of many of the methods described here. In this appendix, we briefly describe how to obtain this package, and list the submodules available.

A.1 Obtaining and Using Crowdastro

The source code for crowdastro is available on GitHub at `http://github.com/chengsoonong/crowdastro`. Crowdastro can also be installed through pip, by running `pip3 install crowdastro`. The code is MIT licensed.

The crowdastro package can be imported into Python or used with the command-line interface. Documentation for the command-line interface is available on Read the Docs at `https://crowdastro.readthedocs.io`.

A.2 Submodules

In this section, we document the main submodules in crowdastro.

A.2.1 active_learning

`crowdastro.active_learning` contains classes that simulate active learning tasks with different sampling methods. These all implement the same methods, and so can be used as drop-in replacements for each other.

Samplers have access to a pool of unlabelled data and an array of known labels. They also have a `sample_index` method that returns the index of an unlabelled data point to query an expert for the label, and a `sample_indices` method that returns a list of such indices for bulk training. They also have an `add_label` method to add the retrieved label to the array of known labels, and a `add_labels` method which takes a list of such labels for bulk training.

A.2.1.1 qbc_sampler

`crowdastro.active_learning.qbc_sampler` contains a class `QBCSampler` that simulates query-by-committee as described in Section 5.2.2. The committee is composed of a user-defined number of logistic regression classifiers trained on a user-defined percentage of the labelled data. The `QBCSampler` also keeps a single logistic regression called the *reference classifier* which is trained on all known labels; it is this classifier that is used to compute the balanced accuracy to avoid underreporting of accuracy due to sampling.

A.2.1.2 random_sampler

`crowdastro.active_learning.random_sampler` contains classes that simulate an active learning task with passive sampling. There are two such classes:

- `RandomSampler`, which samples completely at random,
- `BalancedSampler`, which samples evenly from binary classes.

A.2.1.3 sampler

`crowdastro.active_learning.random_sampler` contains the `Sampler` base class for other samplers.

A.2.1.4 uncertainty_sampler

`crowdastro.active_learning.uncertainty_sampler` contains a class `ConfidenceUncertaintySampler` that simulates binary uncertainty sampling as described in Section 5.2.1.

A.2.2 classifier

`crowdastro.classifier` contains a `RGZClassifier` object which attempts to locate the host galaxy of a given ATLAS object using the methods developed in this thesis. The nearby galaxy with the highest probability of containing an AGN is selected as the associated host galaxy. The module also contains a `RGZCommittee` object which could be used to estimate information content of ATLAS objects as suggested in Section 5.5.

A.2.3 compile_cnn

`crowdastro.compile_cnn` uses Keras to compile a convolutional neural network model and save the output as JSON.

A.2.4 consensuses

`crowdastro.consensuses` finds the consensus label of each galaxy using crowd labels from Radio Galaxy Zoo. This is referred to as the Radio Galaxy Zoo majority vote throughout this thesis. A major part of this module is the conversion of real-valued labels (i.e. the location that the volunteer chose as the location of the host galaxy) into

binary labels, which is achieved by fitting the Gaussian mixture model with the lowest value of the Bayesian information criterion. More details on the methods implemented in this module will appear in Alger et al. [30].

A.2.5 crowd

`crowdastro.crowd` contains classes for crowd learning. Both classes implement the same methods, and so can be used as drop-in replacements for each other. The module also contains some helper functions for crowd labels and related experiments.

A.2.5.1 raykar

`crowdastro.crowd.raykar` is an implementation of the crowd learning algorithm developed by Raykar et al. [8], described here in Section 3.5.2. The module provides a `RaykarClassifier` object which implements a modified scikit-learn interface.

A.2.5.2 util

`crowdastro.crowd.util` contains useful functions for dealing with crowd labels and performing related experiments shown in this thesis. These functions are:

- `balanced_accuracy`, which computes the balanced accuracy of a classifier against a test set,
- `crowd_label`, which simulates the crowd labelling task as described in Section 3.4,
- `majority_vote`, which computes the majority vote of a set of crowd labels,
- `logistic_regression`, a simple implementation of the logistic regression function (Equation 3.1).

A.2.5.3 yan

`crowdastro.crowd.yan` is an implementation of the crowd learning algorithm developed by Yan et al. [9], described here in Section 3.5.3. The module provides a `YanClassifier` object which implements a modified scikit-learn interface.

A.2.6 experiment

`crowdastro.experiment` contains all the experiments run for this thesis. The scripts in this module can be executed from the command line, e.g.

```
python3 -m crowdastro.experiment.experiment_name
```

A.2.7 generate_annotator_labels

`crowdastro.generate_annotator_labels` generates binary labels for each labeller in Radio Galaxy Zoo by assigning the nearest galaxy to labellers' clicks a label of 1, and assigning all other seen galaxies a label of 0. If a galaxy is never seen by a given labeller,

then the associated label is masked and not used in future calculations. This is referred to as the raw Radio Galaxy Zoo labels in this thesis.

A.2.8 generate_cnn_outputs

`crowdastro.generate_cnn_outputs` runs a trained convolutional neural network on radio patches to obtain features as described in Section 4.4.2.

A.2.9 generate_test_sets

`crowdastro.generate_test_sets` generates training and testing sets for use in experiments. This ensures that training and testing labels are consistent across experiments, and also that no testing data is used to train the convolutional neural network.

A.2.10 generate_training_data

`crowdastro.generate_training_data` generates vector representations of training instances, as well as generating associated labels and metadata.

A.2.11 import_data

`crowdastro.import_data` imports and standardises data used for other modules, including the ATLAS catalogue and images, the WISE catalogue, the SWIRE catalogue, the Norris et al. catalogue, and the Fan et al. catalogue.

A.2.12 plot

`crowdastro.plot` contains functions for generating the plots in this thesis.

A.2.13 rgz_data

`crowdastro.rgz_data` contains low-level functions for handling the Radio Galaxy Zoo database.

A.2.14 train_classifier

`crowdastro.train_classifier` trains and stores a logistic regression classifier on the galaxy classification task.

A.2.15 train_cnn

`crowdastro.train_cnn` trains a convolutional neural network on the galaxy classification task using a subset of training data. The training set is generated in `crowdastro.generate_test_sets` to ensure that it is held out of testing sets.

Bibliography

1. J. K. Banfield, O. I. Wong, K. W. Willett, R. P. Norris, L. Rudnick, S. S. Shabala, B. D. Simmons, C. Snyder, A. Garon, N. Seymour, et al. Radio Galaxy Zoo: host galaxies and radio morphologies derived from visual inspection. *Monthly Notices of the Royal Astronomical Society*, 453(3):2326–2340, 2015.

2. R. M. Cutri, E. L. Wright, T. Conrow, J. W. Fowler, P. R. M. Eisenhardt, C. Grillmair, J. D. Kirkpatrick, F. Masci, H. L. McCallon, S. L. Wheelock, S. Fajardo-Acosta, L. Yan, D. Benford, M. Harbut, T. Jarrett, S. Lake, D. Leisawitz, M. E. Ressler, S. A. Stanford, C. W. Tsai, F. Liu, G. Helou, A. Mainzer, D. Gettings, A. Gonzalez, D. Hoffman, K. A. Marsh, D. Padgett, M. F. Skrutskie, R. P. Beck, M. Papin, and M. Wittman. Explanatory Supplement to the AllWISE Data Release Products. Technical report, November 2013.

3. D. Proctor. Comparing pattern recognition feature sets for sorting triples in the FIRST database. *The Astrophysical Journal Supplement Series*, 165(1):95, 2006.

4. D. Fan, T. Budavári, R. P. Norris, and A. M. Hopkins. Matching radio catalogues with realistic geometry: application to SWIRE and ATLAS. *Monthly Notices of the Royal Astronomical Society*, 451(2):1299–1305, 2015. doi: 10.1093/mnras/stv994. URL http://mnras.oxfordjournals.org/content/451/2/1299.abstract.

5. R. P. Norris, A. M. Hopkins, J. Afonso, S. Brown, J. J. Condon, L. Dunne, I. Feain, R. Hollow, M. Jarvis, M. Johnston-Hollitt, E. Lenc, E. Middelberg, P. Padovani, I. Prandoni, L. Rudnick, N. Seymour, G. Umana, H. Andernach, D. M. Alexander, P. N. Appleton, D. Bacon, J. K. Banfield, W. Becker, M. J. I. Brown, P. Ciliegi, C. Jackson, S. Eales, A. C. Edge, B. M. Gaensler, G. Giovannini, C. A. Hales, P. Hancock, M. T. Huynh, E. Ibar, R. J. Ivison, R. Kennicutt, A. E. Kimball, A. M. Koekemoer, B. S. Koribalski, Á. R. López-Sánchez, M. Y. Mao, T. Murphy, H. Messias, K. A. Pimbblet, A. Raccanelli, K. E. Randall, T. H. Reiprich, I. G. Roseboom, H. Röttgering, D. J. Saikia, R. G. Sharp, O. B. Slee, I. Smail, M. A. Thompson, J. S. Urquhart, J. V. Wall, and G.-B. Zhao. EMU: Evolutionary Map of the Universe. *PASA*, 28:215–248, August 2011. doi: 10.1071/AS11021.

6. C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 389 (3):1179–1189, 2008. doi: 10.1111/j.1365-2966.2008.13689.x. URL http://mnras.oxfordjournals.org/content/389/3/1179.abstract.

7. C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. C. Nichol, M. J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410: 166–178, January 2011. doi: 10.1111/j.1365-2966.2010.17432.x.

8. V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322, 2010.

9. Y. Yan, R. Rosales, G. Fung, M. W. Schmidt, G. H. Valadez, L. Bogoni, L. Moy, and J. G. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *International Conference on Artificial Intelligence and Statistics*, pages 932–939, 2010.

10. Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, W. E. Kerzendorf, A. Conley, N. Crighton, K. Barbary, D. Muna, H. Ferguson, F. Grollier, M. M. Parikh, P. H. Nair, H. M. Unther, C. Deil, J. Woillez, S. Conseil, R. Kramer, J. E. H. Turner, L. Singer, R. Fox, B. A. Weaver, V. Zabalza, Z. I. Edwards, K. Azalee Bostroem, D. J. Burke, A. R. Casey, S. M. Crawford, N. Dencheva, J. Ely, T. Jenness, K. Labrie, P. L. Lim, F. Pierfederici, A. Pontzen, A. Ptak, B. Refsdal, M. Servillat, and O. Streicher. Astropy: A community Python package for astronomy. *AAP*, 558:A33, October 2013. doi: 10.1051/0004-6361/201322068.

11. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

12. D. J. Griffiths. *Introduction to Electrodynamics*. Prentice Hall, 1999. ISBN 9780138053260.

13. P. Francis. Brightness units in astronomy, 2008. URL http://www.mso.anu.edu.au/pfrancis/ObsTech/BrightnessUnits.pdf. [Online; accessed 2016-09-30].

14. D. Richstone, E. Ajhar, R. Bender, G. Bower, A. Dressler, S. Faber, A. Filippenko, K. Gebhardt, R. Green, L. Ho, et al. Supermassive black holes and the evolution of galaxies. *arXiv preprint astro-ph/9810378*, 1998.

15. M. C. Begelman, R. D. Blandford, and M. J. Rees. Theory of extragalactic radio sources. *Reviews of Modern Physics*, pages 255–351, 1984. doi: 10.1103/RevModPhys.56.255.

16. G. L. H. Harris. NGC 5128: The Giant Beneath. *PASA*, 27:475–481, October 2010. doi: 10.1071/AS09063.

17. A. C. Fabian. Active galactic nuclei. *Proceedings of the National Academy of Sciences*, 96(9):4749–4751, 1999.

18. L. Saripalli, R. W. Hunstead, R. Subrahmanyan, and E. Boyce. A complete sample of megaparsec-sized double radio sources from the sydney university molonglo sky survey. *The Astronomical Journal*, 130(3):896, 2005. URL http://stacks.iop.org/1538-3881/130/i=3/a=896.

19. A. A. Sokolov and I. M. Ternov. Synchrotron radiation. *Soviet Physics Journal*, 10 (10):39–47, 1967. ISSN 1573-9228. doi: 10.1007/BF00820300. URL http://dx.doi.org/10.1007/BF00820300.

20. E. L. Wright, P. R. M. Eisenhardt, A. K. Mainzer, M. E. Ressler, R. M. Cutri, T. Jarrett, J. D. Kirkpatrick, D. Padgett, R. S. McMillan, M. Skrutskie, S. A. Stanford, M. Cohen, R. G. Walker, J. C. Mather, D. Leisawitz, T. N. Gautier, III, I. McLean, D. Benford, C. J. Lonsdale, A. Blain, B. Mendez, W. R. Irace, V. Duval, F. Liu, D. Royer, I. Heinrichsen, J. Howard, M. Shannon, M. Kendall, A. L. Walsh, M. Larsen, J. G. Cardon, S. Schick, M. Schwalm, M. Abid, B. Fabinsky, L. Naes, and C.-W. Tsai. The Wide-field Infrared Survey Explorer (WISE): Mission Description and Initial On-orbit Performance. *The Astronomical Journal*, 140: 1868-1881, December 2010. doi: 10.1088/0004-6256/140/6/1868.

21. C. J. Lonsdale, H. E. Smith, M. Rowan-Robinson, J. Surace, D. Shupe, C. Xu, S. Oliver, D. Padgett, F. Fang, T. Conrow, et al. SWIRE: The SIRTF wide-area infrared extragalactic survey. *Publications of the Astronomical Society of the Pacific*, 115(810):897, 2003.

22. S. Laine. Infrared Array Camera (IRAC) pocket guide, 2006. URL http://irsa.ipac.caltech.edu/data/SPITZER/docs/files/spitzer/IRAC_Pocket_Guide_Cold_v5.0.pdf. [Online; accessed 2016-10-19].

23. J. Surace, D. Shupe, F. Fang, C. Lonsdale, E. Gonzalez-Solares, E. Hatziminaoglou11, B. Siana, T. Babbedge, M. Polletta, G. Rodighiero, et al. The SWIRE data release 2: Image atlases and source catalogs for ELAIS-N1, ELAIS-N2, XMM-LSS, and the Lockman hole. *Spitzer Science Centre, California Institute of Technology, Pasadena, CA*, 2005.

24. J. J. Condon, W. D. Cotton, E. W. Greisen, Q. F. Yin, R. A. Perley, G. B. Taylor, and J. J. Broderick. The NRAO VLA Sky Survey. *AJ*, 115:1693–1716, May 1998. doi: 10.1086/300337.

25. T. M. O. Franzen, J. K. Banfield, C. A. Hales, A. Hopkins, R. P. Norris, N. Seymour, K. E. Chow, A. Herzog, M. T. Huynh, E. Lenc, et al. ATLAS–I. third release of 1.4 GHz mosaics and component catalogues. *Monthly Notices of the Royal Astronomical Society*, 453(4):4020–4036, 2015.

26. R. P. Norris, J. Afonso, P. N. Appleton, B. J. Boyle, P. Ciliegi, S. M. Croom, M. T. Huynh, C. A. Jackson, A. M. Koekemoer, C. J. Lonsdale, et al. Deep ATLAS radio

observations of the Chandra Deep Field-South/Spitzer wide-area infrared extragalactic field. *The Astronomical Journal*, 132(6):2409, 2006.

27. R. H. Becker, R. L. White, and D. J. Helfand. The FIRST Survey: Faint Images of the Radio Sky at Twenty Centimeters. *The Astrophysical Journal*, 450:559, September 1995. doi: 10.1086/176166.

28. R. Norris. How citizen scientists discovered a giant cluster of galaxies, 2016. URL https://theconversation.com/how-citizen-scientists-discovered-a-giant-cluster-of-galaxies-59373. [Online; accessed 2016-08-21].

29. J. K. Banfield, H. Andernach, A. D. Kapińska, L. Rudnick, M. J. Hardcastle, G. Cotter, S. Vaughan, T. W. Jones, I. Heywood, J. D. Wing, O. I. Wong, T. Matorny, I. A. Terentev, A. R. López-Sánchez, R. P. Norris, N. Seymour, S. S. Shabala, and K. W. Willett. Radio Galaxy Zoo: discovery of a poor cluster through a giant wide-angle tail radio galaxy. *Monthly Notices of the Royal Astronomical Society*, 2016. doi: 10.1093/mnras/stw1067. URL http://mnras.oxfordjournals.org/content/early/2016/05/05/mnras.stw1067.abstract.

30. M. Alger, J. K. Banfield, and C. S. Ong. Radio Galaxy Zoo: machine learning for cross-identification of radio emission with the host galaxy, 2016. [In preparation; tentative title].

31. Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

32. W. Wolberg and O. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology,. In *Proceedings of the National Academy of Sciences*, pages 9193–9196, Dec 1990.

33. C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

34. G. Gybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

35. M. Kon. Decision trees and random forests, 2016. URL http://math.bu.edu/people/mkon/MA751/L18RandomForests.pdf.

36. D. Scherer, A. Müller, and S. Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, pages 92–101. Springer, 2010.

37. J. Ludwig. Image convolution. *Satellite Digital Image Analysis. Portland State University.*, 2015. URL http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf.

38. L. Baraldi. VGG16 model for Keras, 2015. URL https://gist.github.com/baraldilorenzo/07d7802847aaad0a35d3. [Online; accessed 2016-09-30].

39. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

40. C. H. Lin, Mausam, and D. S. Weld. Re-active learning: Active learning with relabeling. 2016.

41. V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 614–622, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401965. URL http://doi.acm.org/10.1145/1401890.1401965.

42. Oxford English Dictionary. citizen, n. and adj. 2016. URL http://www.oed.com/view/Entry/33513?redirectedFrom=citizen+science. [Online; accessed 2016-10-11].

43. P. J. Marshall, C. J. Lintott, and L. N. Fletcher. Ideas for citizen science in astronomy. *Annual Review of Astronomy and Astrophysics*, 53:247–278, 2015. doi: 10.1146/annurev-astro-081913-035959.

44. A. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific Data*, 2, 2015.

45. M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

46. C. H. Lin, Mausam, and D. S. Weld. To re(label), or not to re(label). In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

47. E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL http://www.scipy.org/. [Online; accessed 2016-09-29].

48. S. Dieleman, K. W. Willett, and J. Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450:1441–1459, June 2015. doi: 10.1093/mnras/stv632.

49. D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers, 1994.

50. A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification, 1998.

51. C. C. Loy, T. Xiang, and S. Gong. *Computer Vision–ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part I*, chapter Stream-Based Active Unusual Event Detection, pages 161–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-19315-6. doi: 10.1007/978-3-642-19315-6_13. URL http://dx.doi.org/10.1007/978-3-642-19315-6_13.

52. R. D. King, K. E. Whelan, F. M. Jones, P. G. K. Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427 (6971):247–252, 2004. ISSN 0028-0836. doi: http://www.nature.com/nature/journal/v427/n6971/suppinfo/nature02236_S1.html. URL http://dx.doi.org/10.1038/nature02236. 10.1038/nature02236.

53. B. Settles. Active learning literature survey. Computer sciences technical report, University of Wisconsin–Madison, 2009.

54. H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee, 1992.

55. D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1986. ISSN 1573-0565. doi: 10.1007/bf00116828. URL http://dx.doi.org/10.1007/BF00116828.

56. S. Argamon-Engelson and I. Dagan. Committee-based sample selection for probabilistic classifiers. 11:335–360, 1999.

57. D. Pelleg and A. W. Moore. Active learning for anomaly and rare-category detection. In *Advances in Neural Information Processing Systems*, pages 1073–1080, 2004.

58. J. W. Richards, D. L. Starr, H. Brink, A. A. Miller, J. S. Bloom, N. R. Butler, J. B. James, J. P. Long, and J. Rice. Active learning to overcome sample selection bias: Application to photometric variable star classification. *The Astrophysical Journal*, 744(2):192, 2012. URL http://stacks.iop.org/0004-637X/744/i=2/a=192.

59. D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the eleventh international conference on machine learning*, pages 148–156, 1994.

60. I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. The Morgan Kaufmann series in machine learning, (San Francisco, CA, USA), 1995.

61. B. Mozafari, P. Sarkar, M. J. Franklin, M. I. Jordan, and S. Madden. Active learning for crowd-sourced databases. *CoRR*, abs/1209.3686, 2012. URL http://arxiv.org/abs/1209.3686.

62. Y. Yan, R. Rosales, G. Fung, and J. G. Dy. Active learning from crowds. *Proceedings of the 28th International Conference on Machine Learning*, pages 1161–1168, 2011.

63. A. T. Nguyen, B. C. Wallace, and M. Lease. Combining crowd and expert labels using decision theoretic active learning. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.