University of Manchester
School of Computer Science
Project Report 2017

Automatic Birdsong Recognition

Author: Victor F. Lampreia Rodrigues

Supervisor: Dr. Andrea Schalk

Abstract

Automatic Birdsong Recognition

Author: Victor F. Lampreia Rodrigues

sup my niggas

Supervisor: Dr. Andrea Schalk

Contents

List of Figures

List of Tables

INTRODUCTION

1.1 Motivation

Automatic birdsong recognition is far from solved outside of controlled environments or with heavy user input. A system which can recognize species from birdsong from a wide variety of species is an exciting application of signal processing, computer vision and machine learning technologies. Such software would be beneficial to the bird enthusiasts and environmental health associations, and others.

1.2 Goals

The primary aim of this project is to research and develop potential methods for automatic birdsong recognition, with little to no user interaction. The system is to recognize up to 50 distinct bird species from field recordings provided by arbitrary sources around the world. The recordings should not have any form of rigorous quality control, aside from the seletion of the least noisy recordings.

limit to songs only

1.3 Potential Approaches

is this section needed?

All approaches in this section deal with either the waveform image of a birdsong recording, a spectrogram of the recording, or both.

1.3.1 DTW / archetypes

previous student used waveform and DTW to match against an archetype waveform.

1.3.2 Image recognition

spectral shapes etc some have used image recognition to match templates representative of specific bird species to identify birds in a recording using it's spectrogram.

1.3.3 Spectral Analysis

The mean spectral frequency and energy information in spectrograms may give some clues as to which species of bird is vocalizing.

1.3.4 Pitch Tracking

forgot what this was called, was used for whalesong recognition. may not work well for birdsong recognition due to relative pitches etc?

1.4 Our Approach

The approach taken for this project is a combination of image recognition and spectral analysis through machine learning. The spectral analysis component was not completed, however section x touches on the subject with speculative notes.

move bulk of subsections to appendices

1.4.1 Process Overview

we take the samples, we process them a bit, then for each sample, we do template matching, then we feed to classifier, then we get results. new samples are matched against the previous known templates and fed through classifier.

1.4.2 Architecture Overview

We developed the artefact as a single software package. All components and features described in this report are part of the program unless stated otherwise.

1.4.3 Implementation Technologies

All code was written in Python 2.7. Libraries used include:

- open-CV 2.0

WORKING DATA

This project bases all mechanisms on one single birdsong representation, the source field recordings. This section describes the data used, how it is collected, and prepared for usage within the program.

2.1 Source Data

2.1.1 Format

The birdsong recordings used during development of this project are encoded using 22kHz to 192kHz 16-bit WAV files.

2.1.2 Xeno-Canto Database

Xeno-Canto (ref) provides a substantial number of field recordings of various bird species, both calls and songs. Recordings are available with varying levels of quality, ranging from extremely clear with minimal perceivable noise, to extremely noisy recordings. Recordings may have single or multiple instances of the same species, and/or different species.

note on copyright

Automatic Sample Retrieval

Manually selecting and downloading recordings is a time-consuming process. A public API does not exist for Xeno-canto, therefore we developed a web scraper specifically for automatic retrieval.

The scraper allows the user to filter on species and recording quality before downloading a sample. Species filtering may be done by exlusion or exlusive inclusion.

When repeated fetching is required, an interval may be set in order to reduce strain on Xeno-canto's servers. Once the interval has been set, the scraper will continuously download samples at the specified rate until it has been interrupted by the user.

The webscraper is written in Python 2.7 using the XXXX package.

Metadata Usage

Xeno-canto provides the following metadata with each recording:

- Date and time

- Recording location
- Species recorded
- Existence of other species

This program only makes use of the prominent species tagged in the recording. The location of the recording could be used to improve the accuracy of the program, as certain species are restricted to certain parts of the world. This greatly affects the probability of a certain species being identified in a recording, as long as location metadata is present.

2.2 Preparation

Recordings are resampled to 22000 khz to reduce the memory footprint and processing power required to operate on each recording. Resampling to 22000 khz was found to have no significant reduction in quality or information retained, despite the high frequency vocalizations in birdsong.

All recordings are processed into spectrograms through a fast fourier transform (FFTS) method provided by the xxx python library. This representation provides a visualization of energy present in each frequency band in function of time. Each frequency is quantized into discrete bands according to the parameters set. Time is quantized into etc. The absolute energy is preserved, we don't lose any information, we gain it. **show spectrogram image under waveform of same section**

The following parameters are used:

- hamming window: ur mom

See Appendix for details on FFTS.

Frequencies above x and below y are removed from the spectrograms as these do not contain signals belonging to any bird species **(cite)**.

2.3 Sample Selection

initial db: x species, y sgrams per species, each with z templates take i, j, k round robin selection for template limiation, random would be better. memory limitations guide selection process some words on merge scheme, main should be in different section/chap

FEATURE ENGINEERING AND SELECTION

3.1 Spectrogram Preprocessing

show sgram with noise

Even with prefiltering performed when extracting recordings from xenocanto, noisy samples are still common. Because a high number of samples is required for our approach, we developed an automatic noise reduction stage.

global parameters for sweeping noise reductions are suboptimal, what may work for one recording might not work for another.

yet this is what we do, works ok, specific values were found through trial and error.

objective is to process spectrograms to find contiguous blobs with similar scope, that is, either phrases or individual vocalizations.

Some noise reduction steps are semi-automatic, such as adaptive thresholding.

a better approach would be to operate on a per-spectrogram basis, to find the optimal parameters for each.

This is a non trivial problem.

conceptually the noise reduction algorithm would perform some parameter search with a heuristic based on the dimensions and quantity of contiguous blobs, with the aim of reducing the number of small blobs which may resemble noise or disjoint parts of a single vocalization or segment.

The target scope should be specifiable, so that either individual vocalizations or complete segments or songs could be extracted.

In some cases it might not be possible to achieve total correct segmentation.

Since different birds have different lengths for specific sounds or parts of song, the spectral dimensions can not be generalized. This means that each species will have different aims for quantity and dimensionality, which must be constructed either by manual input or some feedback mechanism.

a feedback system can then be used to determine which type of segmentation works best by filtering to various sizes and measuring the accuracy obtained after classification.

3.2 Useful Feature Identification

Orthonologists use this and that. We're taking an image recognition based approach

show spectrograms of various species to show differences

Variations in amplitude along the song are not taken into account but may be a useful feature to consider.

Direct spectral information such as mean energy per frequency bin is not taken although this can be a useful statistic to help identify the species.

3.3 Template Selection

3.3.1 Basic Elimination

Some effort is taken to reduce the number of templates extracted in the preprocessing stage to reduce the computational and storage overhead otherwise incurred. An increase in template count implies an increase in noise and inconsistencies in the semantics captured. **is semantics the right word?**

is RF good at ignoring noise? I think so.?

speculative section on what makes a good template

for extraction section: several extra templates are included around the template

The following criteria is used to select valid templates:

- dimensions within x
- uhh

These criteria was reached by empirical trial and error.

3.3.2 Guided Elimination

It is desireable to reduce the template count further by recognizing aspects which make for good templates. This subsection outlines some speculative options for selecting better templates, and reducing those which would have a low importance score after training and classification.

Image contrast

It can be argued that templates with low local contrast contain insufficient information to be meaningful in any way during template matching. Templates with a low contrast match against much of any image, resulting in an increase in noise. Implicitly **is implicitly the right word?** such templates have a high correlation with not only the species from which it was extracted but with all species. **show some images, maybe graphs of correlation**

Spatial inclusion

Due to the imperfect nature of the preprocessing methods used, gaps and inconsistencies in structure appear in the thresholded spectrogram. These inconsistencies are present also in repeated components in a bird song, at all levels of granularity. This causes multiple templates to be extracted for a single component in some instances, and single larger blocks to be extracted in others.

In many of these cases, one template's bounding box intersects or is contained entirely within another template's bounding box. Merging these templates by extracting the bounding box of the union of the two or more templates may result in more consistent extractions. **show some images**

Similarly, templates which are sufficiently close to eachother may be merged, but care must be taken not to form extremely large templates.

Variation in granularity

There exists a variance in granularity for the extracted templates, in which some sections of song are mostly connected to form a single template, and others are disconnected, leaving templates with single syllables **right word?** and templates with entire sections of song. This is a similar to the observation in **spatial inclusion**.

what to do about it is it a problem

inter-template correlation

some templates may correlate with eachother, should they be merged?

It can also be argued that templates with little to know intercorrelation may be independent anomalies such as noise in the signal or other sounds irrelevant to the subject species.

Species-specific template statistics

It may be possible to use information regarding average dimensionality and mean frequency information to determine the relevance likelyhood of a particular template. Such metrics require an existing set of validated templates, which may be gathered by filtering a non discriminated set of templates by their measures importances.

3.4 Feature Engineering

With the collection of extracted templates from each spectrogram, a model can now be constructed for a particular sample. This model is represented as a feature vector consisting of the maxima of each template cross correlation operation done on the spectrogram. This vector will then be compared to those of other samples using a classifier. This section describes in detail the operation of template matching to build this feature vector, as well as some notes on the time required.

3.4.1 Cross-correlation Mapping

Cross-correlation mapping, also referred to as template matching, is a method for determining the similarity of an image within another, often larger image. It is essentially a form of image recognition. The intuition is that songs from birds of the same species will have extremely similar spectral shapes.

Cross-correlation mapping works by convolving the template image over the target image and measuring the pixel similarities. check this.

The Open-CV library is used to perform template matching. Open-CV's implementation is highly optimized, and may be computed using a GPU. For details see appendix.

Results

The result of template matching is a cross-correlation mapping of the template against the target spectrogram. show an example of a template ccm against a spectrogram.

The data in the mapping can give us a rough estimate of how closely the template matches. Given this result, we store the global maxima of the mapping in a feature vector.

3.4.2 Feature Vector Construction

When classifying a particular sample, the spectrogram is cross-correlated with each template accumulated so far in the database. For each cross-correlation, the maxima of the result is taken and stored at the index in the vector correspoinding to the template that was used. We then refer to this index later for further analysis.

3.4.3 Computational Expense and Optimizations

Template matching is the most expensive operation in the program. Although the underlining algorithm is itself well optimized, further improvements can be made for marginal gains.

Time anal

Template matching takes approximately x minutes per template, given mean dimensions of mxn and ixj for spectrograms and templates respectively. Considering the quantity of templates stored in the database, the time required quickly compounds into the order of days. xx templates against yy spectrograms was measured at x days on a xyz machine. This stresses the requirement for optimization, which is the topic of section blah.

Implemented and proposed optimizations

Dimensionality reduction: Correlation area truncation:

CLASSIFICATION AND EVALUATION

4.1 Classification

briefly mention the data we are working with (feature vector). explain we need statistical method of categorizing each sample using our model.

4.1.1 Approaches to Classification

number of classifiers can be used skim through some strengths and weaknesses

4.1.2 Random Forest Classifier

we pick the random forest. explain why we use this classifier explain how it fits with the data we are working with

4.2 Accuracy Evaluation

a mention on differences in measuring accuracy for single-label vs multi label classifier

mention any metrics that we do not use but are standard, and explain why we do not use them.

4.2.1 Validation Strategy

10 times 10kfold

4.2.2 Accuracy Metrics

mean accuracy, f-score, etc results

4.2.3 AUC curve

we dont use this but if we can construct one then show how we do and what this tells us

4.2.4 Confusion Matrix

what it is, what it can show us what it actually shows us

4.3 Feature Importance

Without reduction, feature counts per species range from 2000 to 3000. Each new sample to be classified will involve template matching against all templates of all species. For the n selected this means n cross correlations, which takes approximately n time. Feature reduction is therefore an important aspect for performance improvement. It is also helpful in determining what kind of features are most helpful, which may be used to help eliminate features early on before they are used in any heavy computation.

4.3.1 In Random Forests

Mean decrease accuracy

One method for measuring the impact of each feature is through the mean decrease in accuracy. With this method, the impact of removing each feature one-by-one is measured. This method may be sensitive to the random nature of the classifier, so multiple runs may be required to eliminate any variance (is this true?). This can be prohibitively expensive.

Mean decrease impurity

Because the nodes of trees in a random forest correlate directly with a specific feature, it is possible to directly measure or estimate the importance of each feature by determining the probability of a node in a tree being traversed over the number of nodes in the forest. This is known as the mean decrease impurity, or gini impurity, and it can be measured directly after a forest has been trained, or estimated beforehand

(is this true?).

This project makes use of the .... measure.

4.3.2 Results and Analysis

Measuring feature importances exposes x of y features to be completely useless for specifying or discriminating classes of species. Removing these features results in no change in accuracy. Performance increases are observed in both feature extraction **(is it still extraction? build feature vector)** and classification. The greatest increase in performance was observed in the template matching stage with a mean decrease of x minutes.

graph of feature importances

to do: Further reductions were made to the feature set in effort to reduce the feature count as much as possible while retaining a consistent accuracy. etc etc was found. lots of graphs

4.4 Classifier Tuning

we havent done this. section talks about the importance of tuning the classifier for optimal performance. touches on issues of overfitting.

4.4.1 Gridsearch

explain gridsearch and how to use it. explain how we use it if we do.

4.4.2 Parameter Analysis

Analuse the results of parameter optimization also measure overfitting