Birdsong Classification

Advanced Computing - U. de Cantabria - 20/04/2015

Yael Gutiérrez *Ignacio Suárez* *Pablo de Castro*

Introduction

➔ <u>Aim of this project</u>

- *Develop a system capable of identifying bird species by the sounds they make*

➔ *<u>Motivation</u>*

- Interesting for bird-watchers and ornithologists
- Automatic acoustic monitoring system
- Obtain biodiversity estimators
- Ecological surveillance and conservation
- **Open problem in machine learning and signal processing**

Birdsong data sources

- ➔ Data is required to train and test any classification system
  - http://www.xeno-canto.org/ - repository of bird sounds around the world (~200000 recording of ~9000 species)
  - Curated datasets from bioacoustic classification challenges
    - ICML 2013 Bird Challenge ⇢ 35 species & cont. rec.
    - NIPS 2013 Bird Challenge ⇢ 87 species & cont. rec.
    - BirdCLEF 2014 ⇢ **501 species & 14027 recordings!**

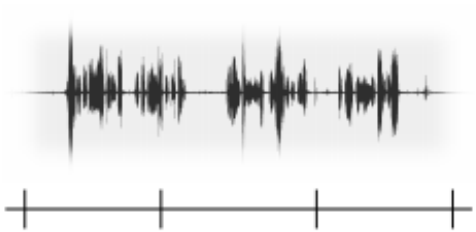
- ➔ Things to take into account
  - Recording and metadata quality
  - Number of recordings per species

BirdCLEF 2014

- ➔ <u>Task/Challenge overview</u>
  - ◆ Bird identification
  - ◆ Subset from *xeno-canto*
  - ◆ 501 species of Brazil area
- ➔ <u>Dataset characteristics</u>
  - ◆ One main bird species per recording (14027 total rec.)
  - ◆ Splitted in train (with labels) & test (no labels/not used)
  - ◆ 44.1 kHz norm. wav files
  - ◆ Metadata also provided

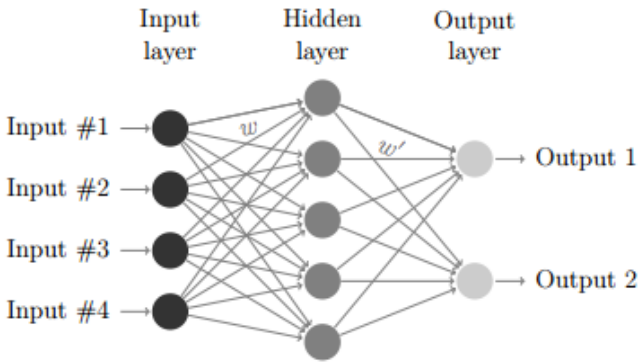
xeno-canto Pl@ntNet SABIOD.ORG Scaled Acoustic Biodiversity Mastodons Big Data

Breaking down the problem



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_d \end{bmatrix}$$

Feature vector



Data Reduction → Feature Engineering → Classification

Automatic Segmentation

Averaged MFCCs estimators

Neural Network (MLP)

Data Reduction: Segmentation

- **Problem:**
  - Most of the audio in the recording is not relevant (i.e. silence)
  - Background noise (e.g. other animals, wind or recording device hum)
  - However, we are only interested in birdsong for classification

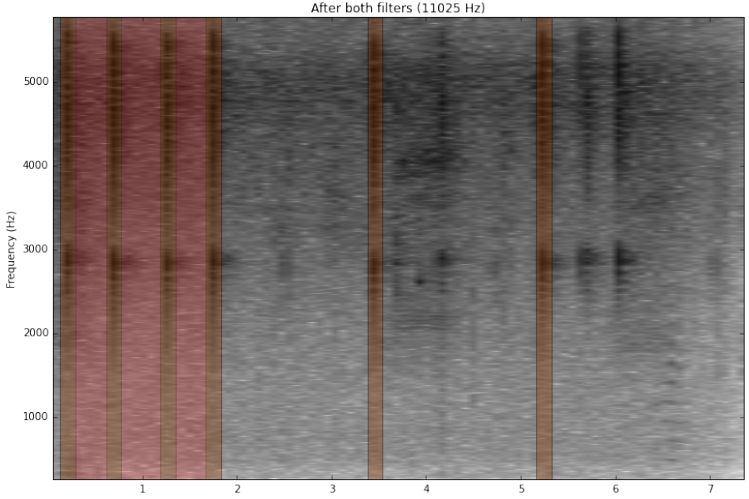
- **Solution:**
  - Find relevant segments with birdsong within each audio file
  - It can be done manually (but not to 14027 recordings)
  - Therefore, an algorithm for automatic segmentation is needed:
    - **Energy based (e.g. [Somervuo and Harma, 2004])**
    - Time-frequency based (e.g. [Neal et al, 2012])

Automatic Segmentation Procedure

1. **Audio Downsampling**
   - 44.1 kHz to 11.025 kHz
   - Faster processing (less data)
   - Lower Nyquist freq (~5 kHz)
2. **Filtering (noise removal)**
   - 10th order highpass filter (1 kHz)
   - Find fund. freq. f_0 (w/ FFT)
   - 10th order highpass filter ($0.6*f_0$)
3. **Find Syllables**
   - **Spectrogram (i.e. STFT)**
   - **Energy based algorithm**
4. **Cluster in Segments**
   - **Temporal gap-wise**

★ **Developed in Python**
- *NumPy (efficient array library)*
- *Scipy (filters, FFT and wav IO)*
- *matplotlib (visualization)*

★ [IPython Notebook Interactive Example]



Energy Based Segmentation

- After downsampling and filtering, the loudest parts of the recording will most likely correspond to birdsong.
- Based on [Somervuo and Harma, 2004] & [HV Koops, 2014]
- An spectrogram (short-time FFT) is computed for the filtered data, then:
  - Obtain maximum amplitude (log) per time bin A(t) (at a certain freq.)
  - Obtain the maximum A(t) and set a threshold (e.g. max(A) -17 dB)
  - Until there is a maximum in A(t) larger than threshold
    - Find max A(t) and trace peak until $\Delta A > 17$ dB
    - Get leftmost and rightmost limit and remove segment
  - After this, you have a list of small segments for each recording
- Birdsongs may have higher temporal structure, so segments are clustered if the temporal gap between them is smaller than 800 ms.

Feature Engineering: MFCCs

- **What are MFCCs?**
  - Audio representation that approximation human auditory response.
- **How are MFCCs calculated?**
  - Original signal transformed to the frequency domain ⇨ DFT
  - Frequency domain mapped into Mel scale ⇨ Auditory response
  - Mel values transformed to the frequency domain ⇨ DCT
  - Amplitudes of the spectrum ⇨ MFCCs
- **Why using MFCCs?**
  - Used with success for classification tasks in bio acoustic and music information retrieval.

Feature Engineering: MFCCs

rastamat lib - Matlab implementation for MFCC extraction from soundfiles (by Dan Ellis @ Columbia University).

- ➔ Draw spectrograms
- ➔ Supports many options:
  - ◆ Window length
  - ◆ Hoptime
  - ◆ Number of cepstra (16)
  - ◆ Max and min frequencies
  - ◆ ...
- ➔ Set Values: minimize the energy difference between audio files of a training set and the reconstructed signal from the calculated MFCC (by Hendrick V. Koops @ Utretch University).

Feature Engineering: Procedure

input

RASTAMAT

SEGMENT #1 → 16 MFCCs = [$^{1}MFCC_1$ $^{1}MFCC_2$ ... $^{1}MFCC_{16}$]

SEGMENT #2 → 16 MFCCs = [$^{2}MFCC_1$ $^{2}MFCC_2$ ... $^{2}MFCC_{16}$]

...

...

...

SEGMENT #N → 16 MFCCs = [$^{N}MFCC_1$ $^{N}MFCC_2$ ... $^{N}MFCC_{16}$]

Mean & Variance

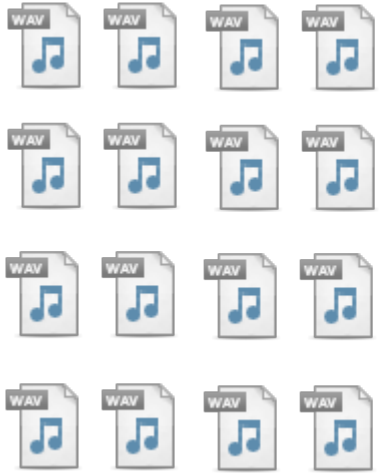
MFCCs MEANS = [<$MFCC_1$> <$MFCC_2$> ... <$MFCC_{16}$>]

MFCCs VARIANCES = [Var($MFCC_1$) Var($MFCC_2$) ... Var($MFCC_{16}$)]

output

FEATURES

Data Reduction: ACHIEVED



24 GB

9688 .wav files

Segmentation
&
Feature Extraction

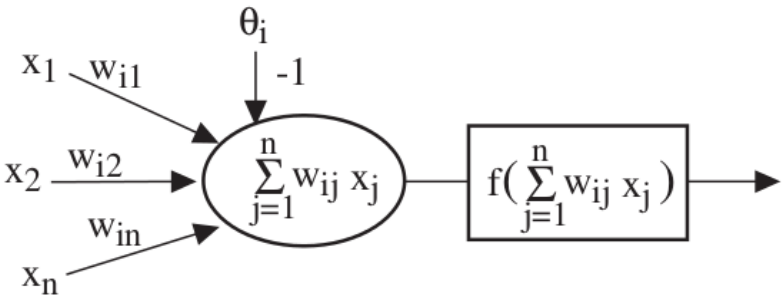
20 MB

Classification: Neural Networks

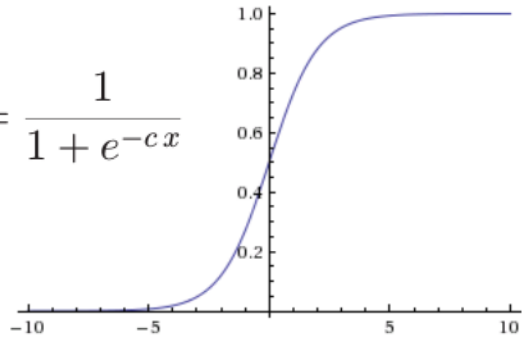
- **What are Artificial Neural Networks?**
  - Algorithms based on propagation of information in real-life neurons, used for supervised machine learning
- **Advantages:**
  - Able to identify and adapt to patterns according to input variables
  - Widely used for regression and classification
    - Many libraries available!
    - In our case, RSNNS package for R, adaptation of Stuttgart Neural Network Simulator (SNNS).
- **Disadvantages:**
  - Scaling, ‘black box’

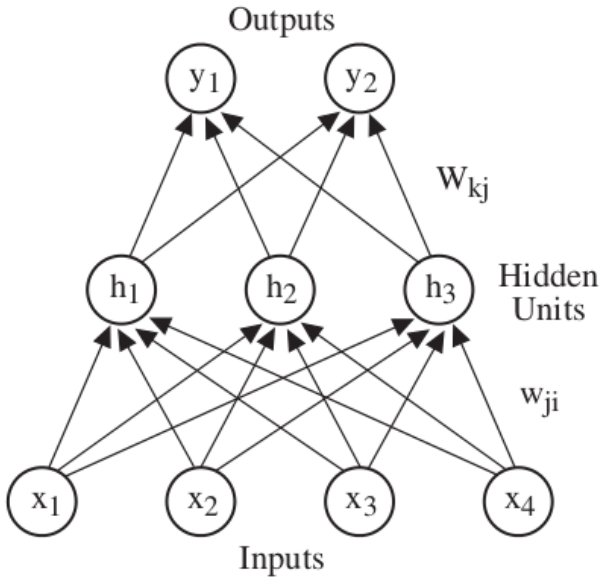
Multilayer Perceptron (MLP)

Perceptron (not enough!)



$$f_c(x) = \frac{1}{1 + e^{-c\,x}}$$



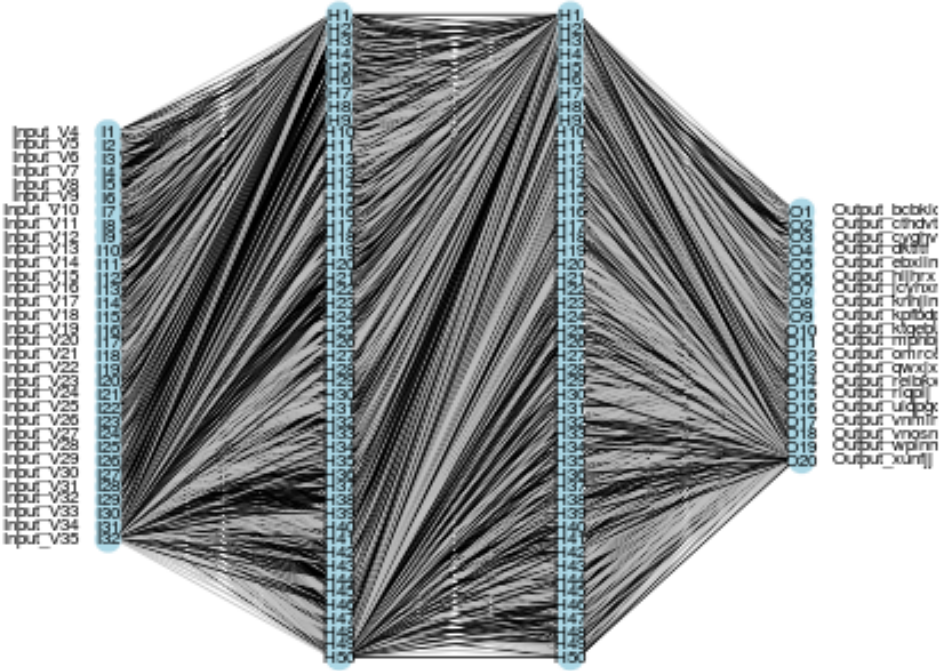


Weights updated in each iteration through error back-propagation and gradient descent methods for minimizing errors.

Our Artificial Neural Network

Input:

N x 32 matrix (MFCC means & variances)



Output:

N x C matrix (Non-binary, highest -> class)

N = Number of segments. Max: 46449
C = Number of bird species (classes). Max: 501

Results

20 species

Hidden layer: **[50 50]**

Train	Test
93.1%	71.1%

Hidden layer: **[100 200]**

Train	Test
94.5%	79.8%

50 species

Hidden layer: **[50 50]**

Train	Test
73.2%	53.2%

Hidden layer: **[100 200]**

Train	Test
87.3%	68.0%

(Only taking into account most likely species)

Difficulties Encountered

- **Scaling problems:**
  - Computation time for more classes or larger networks was exceedingly long, over 24 hours.

- **Solution? Parallelization**
  - Neural Network Toolbox for MATLAB has provided parallel and GPU computing support since version R2012b.

Conclusions

- ➔ A system for the classification of birdsongs from audio recordings has been successfully developed.
- ➔ The system includes energy based automatic segmentation algorithm, MFCCs feature generation and a powerful neural network classifier.
- ➔ We had some problems scaling the classifier to 501 classes and large numbers of hidden layer nodes. The use of GPUs for training could speed up this process.
- ➔ The accuracy of the will system could be for example further improved with more features (e.g. more MFCC estimators).

Project code available at GitHub

Animal sound recognition project for High Perfomance Computing course @ University of Cantabria — Edit

38 commits | 4 branches | 0 releases | 2 contributors

branch: master | pajaros / +

Merge branch 'nachosandres-master'

pablodecm authored 27 minutes ago — latest commit 4e23285657

data	Remove temporal files	28 minutes ago
features	features each line	10 days ago
nnetwork	Remove temporal files	28 minutes ago
rastamat	Merge from pull request branch	14 days ago
segmentation	Update import in ipynb and link in README	23 hours ago
.gitignore	Merge from pull request branch	14 days ago
README.md	Add link to Gitter chat	a month ago

https://github.com/pablodecm/pajaros.git