Quantification of similarity between dickcissel song dialects

Patrik Andrén

July 5th 2020

Abstract

In this thesis, five methods of scoring the similarity of phrases from dickcissel birds are explored. The methods are tested on real and simulated data. The method that is found to perform the best is a variation of the spectrographic cross-correlation method that also looks at correlation in the frequency domain. The second best method is based on singular value decomposition of the spectrogram of the phrase and extracting density-based features. While the methods are not good enough to give a reliable similarity between two phrases, they are able to find structure in a larger data set.

1 Introduction

When analyzing large sets of bird songs a quantitative method of assessing the similarity of songs is often useful. Automated analysis of bird songs and calls is an active area of research with many possible applications. For example in [1] Kwan et. al. uses automated recognition of birds to prevent collisions between birds and aircraft by using audio recordings to monitor bird activity around airports and in [2] Lee et. al. uses it to do automated recognition of bird species.

Timothy H. Parker is an Associate Professor at the department of biology at Whtiman College, Walla Walla, and has collected the songs of dickcissel birds over the course of several years in a couple of different locations. He wants to analyze them to find out more about how the songs differ in order to better understand their culture is shared. Culture in this context means behaviours learned from other birds of the same species. Examples of questions to be explored is how the songs differ across space and time and what impact the environment has on songs. In order to answer these questions a method for quantifying the similarity of songs is required. The goal of this thesis is to try and find a automated method of scoring song similarity that works well for the dickcissel bird songs.

Many general methods of solving this problem exist. One common method used in [3] by Cartopassi and Bradbury is spectrographic cross-correlation (SPCC). This method is based on transforming a birdsong into the spectrogram and doing cross-correlation. Another common method is found in [2] where Lee et. al. uses a feature extraction method called Mel-frequency cepstrum coefficients, or MFCC, to classify signals. MFCC is based on a frequency representation that is meant to align with the way humans perceive different frequencies of sound. There have also been some attempts at using neural networks for example by Sprengel et. al. in [4]. General methods often work decently among a large variety of songs but when working with specific birds you can often achieve better results by writing a more specialized algorithm.

A full analysis of a bird song usually includes some prepossessing such as denoising and segmentation of the song into shorter sections, called phrases. In this case that is already done by Timothy. The phrases of the dickcissel songs have also been classified into three different classes: cissel, dick and trill, both based on the shape of the phrase and the context of the phrase. Because the full similarity score only compares phrases of the same class, this thesis will do the same and phrases of different classes will be treated as completely different objects and will not be compared. When comparing full songs you would compare the composition of the song by different phrase classes in combination with the similarity of phrases of the same class.

The thesis is organized as follows: In chapter 2 the methods are described, in chapter 3 the data used to evaluate the methods is presented, in section 4 the methods are tested on simulated data, in section 5 the methods are tested on real data and in section 6 the results are discussed.

2 Model and Methods

2.1 Model

The signals are zero mean and non-stationary. The phrases have different parts where a tone or a chirp is sustained for some time, these different parts are called notes. The signals vary in amplitude by a large amount, both between signals and between different notes in the signal. The signals are generally low in noise but not free from it. Since the signals are recorded in nature the primary source of noise seems to be things happening in the background. The cissel class phrases vary between 105 ms and 439 ms in length. The dick class phrases are between 40 ms and 1264 ms in length and the trill class phrases are between 130 ms and 615 ms. An example signal looks like figure 1 in the time domain and like figure 2 in the time-frequency domain.

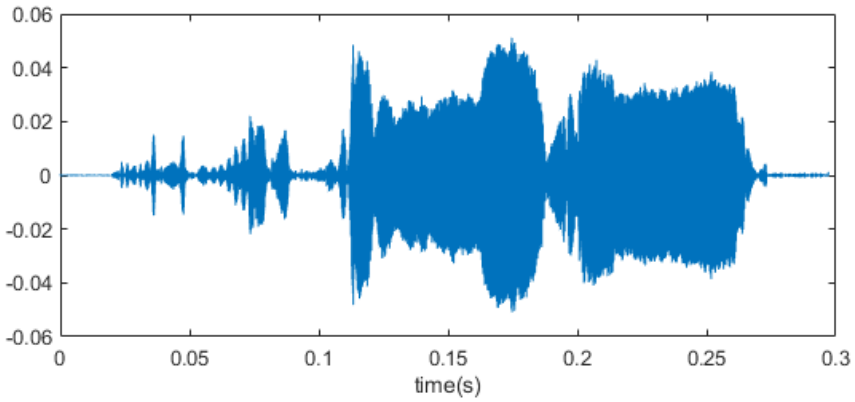

Figure 1: The time domain representation of a phrase.

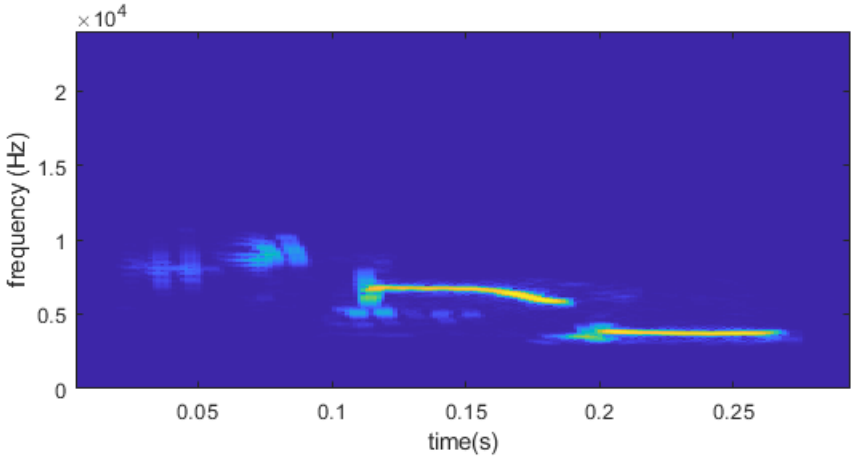

Figure 2: The time-frequency domain representation of a phrase.

The goal of a method is to take two signals x and y and make some sort of statement about the similarity or dissimilarity between those two signals. A signal will consist of one phrase of one class, either dick cissel or trill. A method will generally consist of two components: A feature extraction

part that reduces the signal down to some informative characteristics and a distance measurement that takes two sets of characteristics from different signals and compares them.

2.2 Spectrographic cross-correlation (SPCC)

This method is a slight variation on the *spectrographic cross-correlation (SPCC)* method described by K. A. Cortopassi and J. W. Bradbury in [3]. This method was chosen because it has seen wide usage and success on many different species of birds. The method, as most methods in this thesis, starts from *the spectrogram*. The spectrograms in this thesis are made using MATLAB's *pspectrum* command which uses Kaiser windows. Additionally the command was configured to use a window size of 10 ms, 85 % overlap and 0.9 leakage. This results in a spectrogram image with 1024 values in the frequency range $(0, 24000)$, one value every $\Delta f = 23.4375$ Hz. The image has a time value every $\Delta t = 1.5$ ms in the variable size time range $(0, t_{end})$ seconds. The time indexing of the spectrogram does not match exactly with the time indexing of the original signal to make space for the entire window. The spectrogram image can be represented both as a 1024-by-$t_{end}/\Delta t$ matrix and as a function. In this thesis the spectrogram image function will be denoted by $P(f, t)$ and the spectrogram image matrix will be denoted by $A(i, j)$. They are connected by the vectors $F = \begin{bmatrix} 0 & \Delta f & 2\Delta f & \dots & 24000 \end{bmatrix}$ and $T = \begin{bmatrix} 0 & \Delta t & 2\Delta t & \dots & t_{end} \end{bmatrix}$ such that

$$P(F(i), T(j)) = A(i, j). \tag{1}$$

It is to be understood that taking the sum over f or t represents letting them take all the values in F and T respectively.

A way to do dimensional reduction in a spectrogram is to take the sum in the time and frequency directions, reducing the spectrogram to two one-dimensional signals that can be plotted along the marginals. For example the signal in figure 3 can be reduced to the marginals in figure 4.

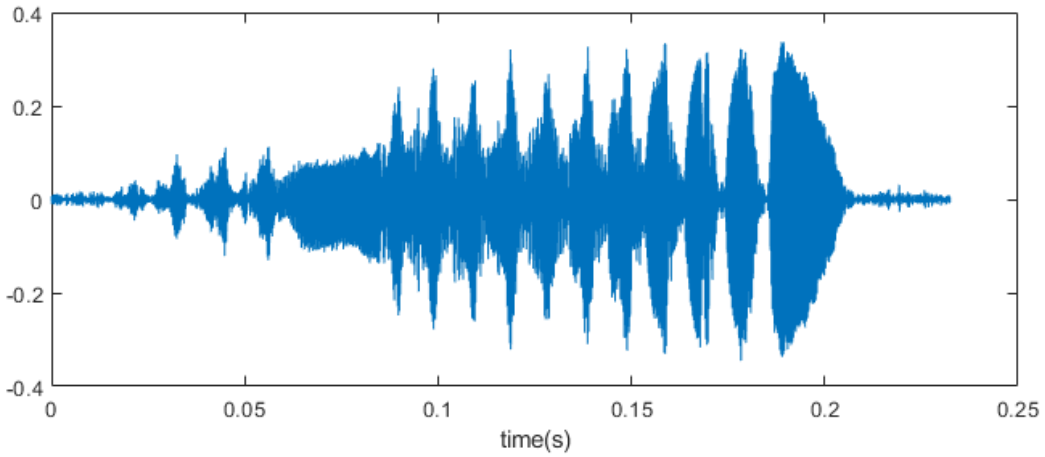

Figure 3: The time domain representation of a phrase.

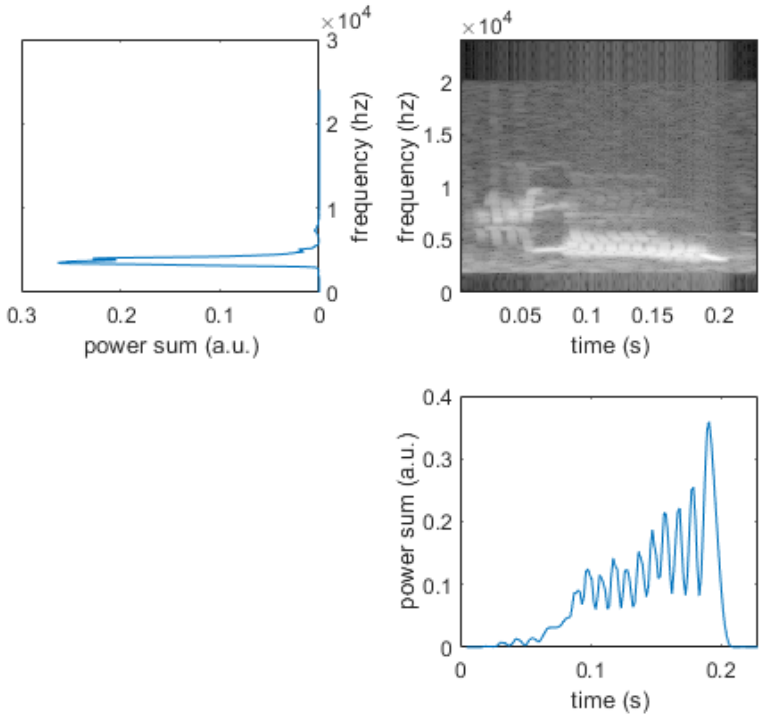

Figure 4: The spectrogram of a phrase together with the time and frequency marginals.

Let $A(i, j)$ be the spectrogram image value at frequency index i and time index j, the marginals can then be written as:

$$m_f(i) = \sum_j A(i, j) \tag{2}$$

$$m_t(j) = \sum_i A(i, j) \tag{3}$$

The signal m_f is called the *frequency marginal* and vice versa. The main idea behind this method is to directly compare these marginals between signals using a *normalized cross-correlation*. Cross-correlation gives a measure of signal similarity for each time lag. The normalized cross-correlation between signal x and y at lag τ used is:

$$xcorr_{x,y}(\tau) = \frac{1}{N} \sum_t \frac{(x(t) - \bar{x})}{s_x} \frac{(y(t + \tau) - \bar{y})}{s_y} \tag{4}$$

where $\bar{x}$ is the mean value of signal x and s_x is the sample standard deviation of signal x and N is the length of the signals. When x and y are frequency marginals then τ represents a shift in alignment in frequency and when they are time marginals τ represents a shift in time alignment, one signal is shifted in time.

Taking the correlation at zero time lag as the measure of similarity seems natural but would lead to a sensitivity in how each phrase is cut from the whole song and would classify two phrases

identical in structure but shifted in frequency as two completely different phrases. An alternative method would be to take the correlation at the time lag that would maximize correlation. This would however run the risk of creating false positives where different parts of two signals have similar structure. Visual inspections of the signal suggest that the value that maximizes the time correlation can be used but that the frequency correlation needs some sort of compromise.

The method uses the following compromise: create a new signal shaped like a normal distribution centered at zero time lag and with standard deviation σ_f, then do point-wise multiplication between the cross-correlation and the normal distribution and take the largest value of the resulting signal as the similarity score. This means that if the signals are shifted in frequency some penalty to the similarity is given but they would not be classified as completely different. The similarity scores for the time and frequency marginals are multiplied to make a single similarity score.

The method is summarized in the following equation:

$$SPCC_{x,y} = \sup_{\tau}\left(xcorr_{m_t^x, m_t^y}(\tau)\right) \cdot \sup_{\nu}\left(xcorr_{m_f^x, m_f^y}(\nu) * \frac{1}{\sigma_f\sqrt{2\pi}} e^{-\frac{\nu^2}{2\sigma_f^2}}\right) \tag{5}$$

where $*$ denotes point-wise multiplication, m_t^x is the time marginal for phrase x, m_f^x is the frequency marginal for phrase x, and σ_f is a tune-able parameter.

2.3 SVD as probability distributions, the Groutage method

This method uses feature extraction found in an article by Dale Groutage and David Bennink [5] which uses the matrix decomposition *Singular Value Decomposition (SVD)* and will be referred to as *the Groutage method*. Using SVD they decompose the spectrogram image matrix into a sum of rank one matrices:

$$A(i,j) = \sum_k \beta_k A_k(i,j) = \sum_k \beta_k u_k v_k \tag{6}$$

where u_k is a column vector of size 1024 and v_k is a row vector of size $t_{end}/\Delta t$. The values β_k are called the singular values and show how much the matrix A_k contributes to the full matrix.

With the goal of being able to treat the matrices A_i as probability distributions new matrices $\tilde{A}_i$ are created that are positive and normalized so that:

$$\sum_i \sum_j \tilde{A}_k(i,j) = 1 \tag{7}$$

Because the vectors u and v are orthonormal this is achieved by element-wise squaring the vectors before multiplying them:

$$\tilde{A}_k(i,j) = u_k(i)^2 v_k(j)^2 \tag{8}$$

Normalizing this way means the vectors $u_k(i)^2$ and $v_k(j)^2$ are the time and frequency marginals of the matrix and the spectral moments can be calculated directly from them. Since the matrices are all rank one the marginals here contain the full information of the matrix A_k. The drawback is that since the decomposition creates so many matrices the cross-correlation can not be used here. Instead features are extracted from each marginal and compares across signals using a metric between sets.

Since $\tilde{A_k}$ is treated as a probability distribution the expected value and variance can be calculated in both the time and frequency direction creating a feature vector with four features. The hope is that these four features will accurately describe where things are happening in the signal in both the time and frequency domains.

Using the assumption that matrices A_k with small associated singular values β_k represent noise only the matrices with singular values $\beta_k > \frac{\beta_1}{100}$ are included in the method. This effectively acts as de-noising the signal.

The features of the Groutage method can be conceptualized as rectangles with centers at the expected values and sides the size of the standard deviation. This is shown for an example phrase in figure 5.

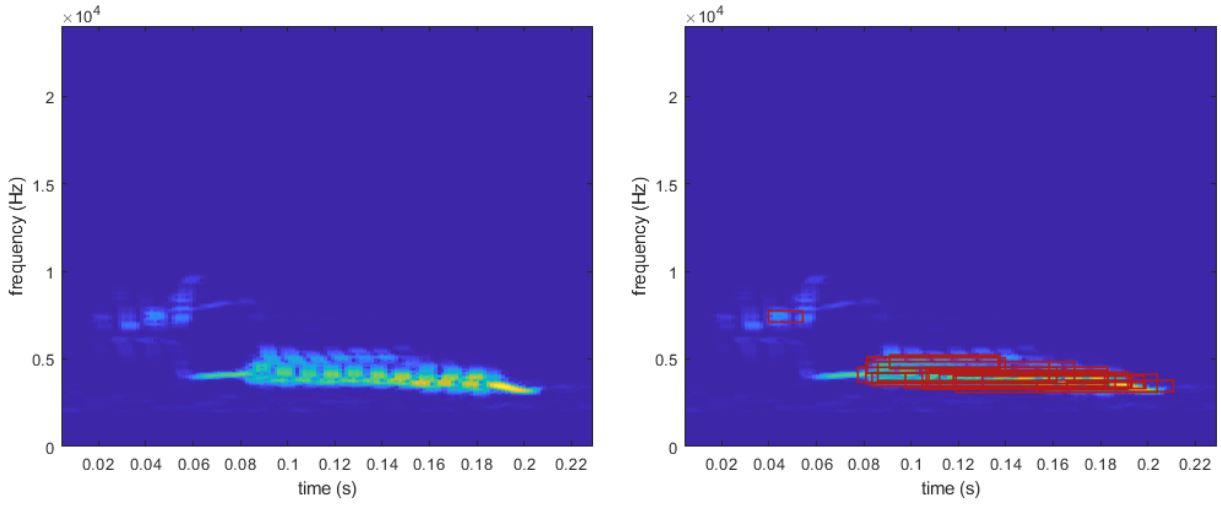

Figure 5: Left: Spectrogram of an example phrase. Right: The same spectrogram but with features extracted by the Groutage method drawn on top.

In order to turn the feature vectors into a workable metric the modified Hausdorff distance described in section 2.7 is used. The weight of a feature vector is the singular value β_k of the corresponding matrix A_k.

2.4 Clustered Laplacian of Gaussian (LoG)

This method extracts features from a song by combining two methods from image analysis: the *Laplacian of Gaussian* and the *DBSCAN* algorithm. They can be found in books on computer vision such as [6]. It then uses those features together with the modified Hausdorff distance which is a metric between sets in order to create a metric between two songs.
The method will be referred to as *the Laplacian of Gaussian method*, or *the LoG method* for short. A number of different image analysis techniques have been explored and this one seems to be the most promising. Because the method is an image analysis technique the spectrogram will be treated as an image with pixels rather than a function of time and frequency. The main idea is to find points in the spectrogram where a smoothing operation would have an especially large effect

with the hope that this will allow us to find where the notes are located in the image with less sensitivity to amplitude variations than a simple thresholding operation. This is achieved by taking the *Laplacian* of the *Gaussian*. Those locations will then be clustered and features will be extracted.

The Gaussian function in two dimensions is defined as:

$$G(i,j) = \frac{1}{\sqrt{2\pi\sigma_g}} e^{-\frac{i^2+j^2}{2\sigma_g^2}} \tag{9}$$

where σ_g is a parameter. The function is applied to an image by reducing the function into a smaller matrix, called a kernel, with the center of the kernel being $(i = 0, j = 0)$. The kernel is then convoluted with an image resulting in a blurred image. In this application the kernel size is $5x5$ and $\sigma_g = 4$. These values were arrived at by applying the LoG to different signals and visually inspecting the results.

The Laplacian is a differential operator defined in two dimensions as:

$$\Delta P(f,t) = \frac{\partial^2 P}{\partial f^2} + \frac{\partial^2 P}{\partial t^2} \tag{10}$$

and can be interpreted as the relative local rate of change.

The Laplacian of Gaussian is applying the Laplacian to the Gaussian function before reducing it into the kernel. The LoG results in a new value for every pixel in the spectrogram representing how much is happening in that pixel, these values are expected to be high on notes and low on the background of the spectrogram. This new value is thresholded based on a parameter α_{log}, setting every pixel larger than the maximum value times α_{log} to one and every pixel smaller than the maximum value α_{log} to zero. The pixels with value one are then simply referred to as the points.

The clustering algorithm "Density-based spatial clustering of applications with noise", or DBSCAN, is used to cluster the points into groups. The DBSCAN algorithm works as follows:

1. Specify a radius ϵ and a number of points $minPts$.
2. Count the number of neighbours in a distance ϵ from every point. Designate each point with more than or equal to $minPts$ neighbours as a "core point".
3. Make clusters of core points using the rule that if two core points are neighbours they are in the same cluster.
4. Each non core point is assigned to the cluster of its nearest neighbour cluster point, if the point has no cluster points as neighbours it is assigned as noise.

In this method the radius ϵ is set to 10 pixels and the number of points $minPts$ is set to 50.

The DBSCAN method was chosen because it has some desirable properties. First it doesn't require specifying the number of clusters, which is necessary since the number of notes differs between phrases. Second it does not assign every point to a cluster, allowing noise to be filtered out. Third the fact that it is density-based means that it can deal with the fact that the shapes of the notes

are often quite oblong.

The resulting clusters for an example phrase using the LoG method is shown in figure 6.

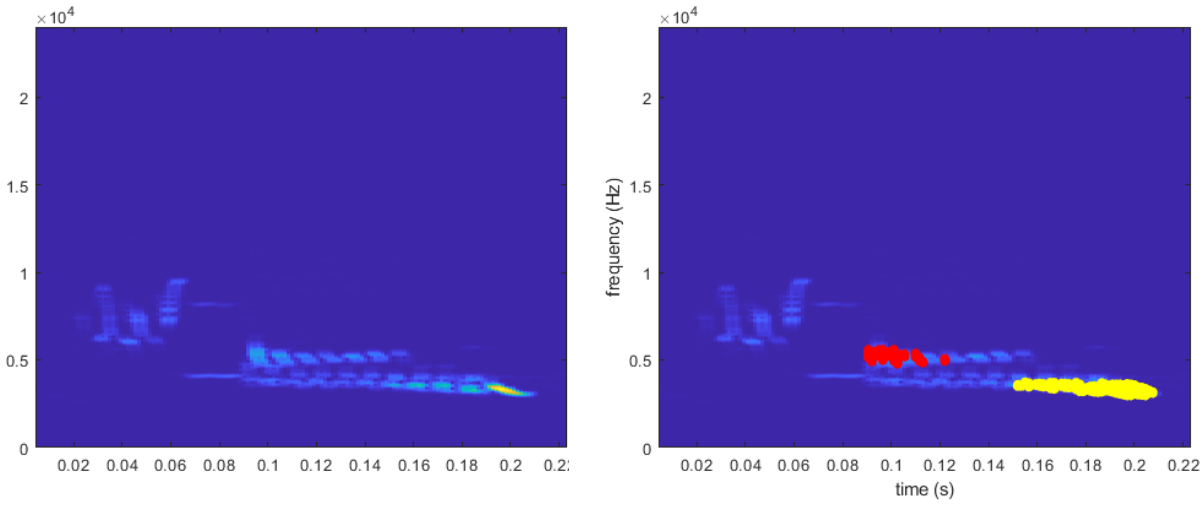

Figure 6: Left: Spectrogram of a phrase. Right: The same spectrogram but with the two clusters found by the LoG method drawn on top.

After clustering, the first two moments in both time and frequency are extracted as feature vectors for each cluster. Aside for some trilling the shapes of the notes seem simple so the hope is that the first and second moment encodes all the important information of each note. The modified Hausdorff distance described in section 2.7 is then used to create a distance between two phrases. The weight of a feature vector is the number of points in the associated cluster.

2.5 Peak frequency contours

The main idea behind this method is to trace the peaks of shapes in the spectrogram and extract features from those contours. The method is based in part on the method used by Tchernichovski et. al. in [7] and will be referred to as *the Peak Frequency Contour method*, or *the PFC method* for short. While Tchernichovski et. al. do not assume any shape of the contours this method does and will try to fit them to straight lines. This method also extracts a single feature vector per line while the method in [7] they extract features that vary over time.

The first part of the method is to apply the Gaussian smoothing described in section 2.4. This eliminates many small peaks created by noise while preserving the larger peaks. After that the peaks at every slice of time are picked out and assigned as candidate peaks. A point $A(i, j)$ is a candidate peak if

$$A(i-1, j) < A(i, j) > A(i+1, j) \tag{11}$$

Any peak with a larger peak in the same time slice and within 700 Hz in frequency is removed from the candidates list. This is done to eliminate sub-peaks that often appear next to large peaks in the spectrogram. In the next step any peak with a value that is smaller than the maximum peak times the tune-able parameter α_{pfc} is removed from the candidates list, this is also done to remove

any peaks that has appeared in the background noise.

The peaks are made into contours by fitting them to lines in the spectrogram. This is done iteratively by the following process:

1. Set $t = t_0$
2. Calculate the distance to the closest contour for each candidate peak in the time slice $P(\cdot, t)$
3. For every peak where the distance is less than 700 Hz, add the peak to the contour.
4. For every peak that did not get added to a contour, create a new contour with the peak as a member.
5. If a contour has not had a line added to it for five time slices, terminate the contour.
6. Refit the contours to fit with the now points using linear regression. A contour with only one point is defined as a horizontal line.
7. Set $t = t + \Delta t$
8. If $t > t_{max}$, terminate, otherwise go to step 2.

After the contours are found, any contour spanning less than 15ms is removed to remove false contours.

The peaks making up a line using the PFC method in an example phrase show in in figure 7

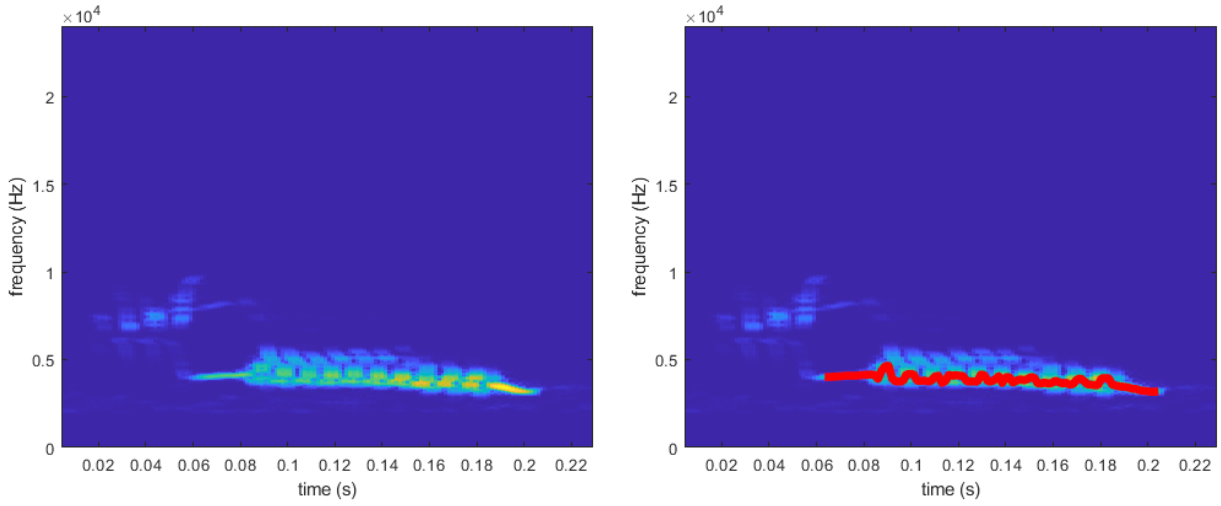

Figure 7: Left: Spectrogram of a phrase. Right: The same spectrogram but with the peaks making up a single contour drawn on top.

A feature vector consisting of the center of the line in time and frequency as well as the angle of the line is extracted as the feature vector for each line. The modified Hausdorff distance described in section 2.7 is used to create the distance between two sounds. The weight is the mean intensity among the peaks making up the contour.

2.6 The filtered ambiguity spectrum

This method is taken from Große Ruse et. al. [8] and will be referred to as *the Ambiguity method*. The main idea behind the method is to move to the *filtered ambiguity spectrum* and extract features by way of SVD.

The filtered ambiguity spectrum is a spectral representation of the signal that is both invariant to time and frequency shifts and arises by taking the Fourier transform in both the time and frequency directions on the spectrogram. The SVD is applied to the ambiguity spectrum and the first orthonormal vectors u_1 and v_1 are extracted as features.

The spectrogram and ambiguity spectrum for an example phrase is shown in figure 8.

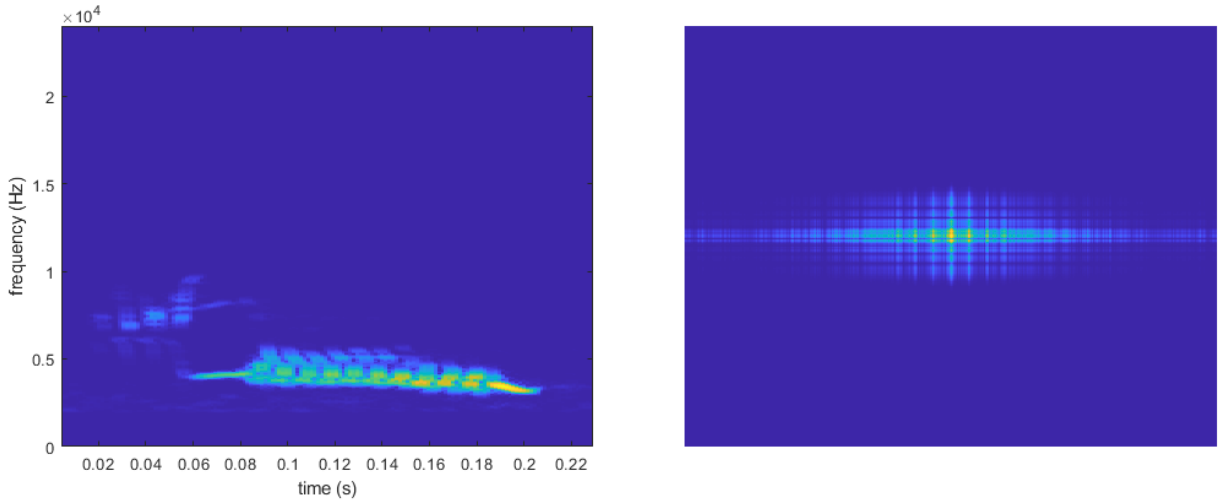

Figure 8: Left: Spectrogram of a phrase. Right: The filtered ambiguity spectrum of the same phrase.

The similarity score between two phrases is:

$$Ambig(x, y) = min\left(\langle u_1^x, u_1^y\rangle, \langle v_1^x, v_1^y\rangle\right) \tag{12}$$

The inner product is used because it gives the similarity score a nice normalization: If the vectors are the same then the score will be 1 and if they are orthogonal the score will be 0. The minimum is used to make sure two signals are only measured as similar if they are similar in both the time and the frequency directions.

2.7 Modified Hausdorff distance

Some of the methods in section 2 produce a variable number of feature vectors for each phrase. In order to compare two sets of feature vectors and get a single number a metric between sets found in a survey by A. Conci and C. S. Kubrusly [9] is used. The metric is a variation on the popular *Hausdorff distance* that is more robust against outliers than the Hausdorff distance. The survey contains many different metrics that could have been used but this one was was picked because of

its robustness and because a set with a large amount of elements is not going to give a consistently low or high similarity. The metric between two sets A and B is defined as:

$$H'(A,B) = \frac{1}{\#A}\sum_{a\in A} d(a,B) + \frac{1}{\#B}\sum_{b\in B} d(b,A) \tag{13}$$

where $\#A$ is the number of features in set A and $d(a,B)$ is defined as:

$$d(a,B) = \inf_{b\in B}(d(a,b)) \tag{14}$$

where $d(a,b)$ is a metric between feature vectors. In this thesis the metric between feature vectors is always the euclidean distance normalized by a precalculated standard deviation for the feature. To clarify: the standard deviation of each feature needs to be calculated before the method is used on a set of signals that is representative of the signals being analyzed. The training set can include any signals being analyzed.

The methods have a natural way to assign a weight to each feature vector in a set so the metric is extended to be weighted by:

$$H''(A,B) = \frac{1}{\#A}\sum_{a\in A} d(a,B)w_a + \frac{1}{\#B}\sum_{b\in B} d(b,A)w_b \tag{15}$$

where w_a is the weight of feature vector a. The weights are selected differently for each method: The LoG method uses number of points in each cluster, the Groutage method uses singular values and the PFC method uses mean intensity among the points making up each line. The weights are always normalized so that $\sum_{a\in A} w_a = 1$. Additionally, for all a: $w_a > 0$.

3 Data

3.1 Simulated data

The simulated signals are created by sampling the function:

$$s(t) = \sum_{i=1}^{n} A_i sin(2\pi t\omega_i + m_i(t - t_i^0)^2) \cdot G(0, t - t_i^0) \tag{16}$$

at 15000 points evenly distributed in the interval $[0, 0.3125]$ where A_i, ω_i, t_i^0 and m_i are parameters and G is the Gaussian kernel described in equation 9 where the standard deviation σ_g is also a parameter dependent on i. Five different types of signals are created using parameters sampled from different distributions. The distributions are presented in table 1 letting $U(a, b)$ represent the uniform distribution in the interval $[a, b]$.

Param	**Type 1**	**Type 2**	**Type 3**	**Type 4**	**Type 5**
n	3	3	3	3	4
A_1	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$
A_2	$U(0.005, 0.015)$	$U(0.005, 0.015)$	$U(0.005, 0.015)$	$U(0.005, 0.015)$	$U(0.005, 0.015)$
A_3	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$	$U(0.0125, 0.0375)$
A_4	-	-	-	-	$U(0.0025, 0.0075)$
ω_1	$U(3800, 4200)$	$U(3800, 4200)$	$U(3800, 4200)$	$U(4300, 4700)$	$U(3800, 4200)$
ω_2	$U(6650, 7350)$	$U(6650, 7350)$	$U(6650, 7350)$	$U(7150, 7850)$	$U(6650, 7350)$
ω_3	$U(6175, 6825)$	$U(6175, 6825)$	$U(6175, 6825)$	$U(6675, 7325)$	$U(6175, 6825)$
ω_4	-	-	-	-	$U(380, 420)$
m_1	0	0	0	0	0
m_2	0	0	0	0	0
m_3	$-10^{-6}U(9.5, 1.05)$	$-10^{-6}U(9.5, 1.05)$	$-10^{-6}U(9.5, 1.05)$	$-10^{-6}U(9.5, 1.05)$	$-10^{-6}U(8.75, 11.25)$
β_4	-	-	-	-	0
t_1^0	$U(0.209, 0.231)$	$U(0.259, 0.281)$	$U(0.259, 0.281)$	$U(0.209, 0.231)$	$U(0.209, 0.231)$
t_2^0	$U(0.114, 0.126)$	$U(0.114, 0.126)$	$U(0.164, 0.176)$	$U(0.114, 0.126)$	$U(0.114, 0.126)$
t_3^0	$U(0.152, 0.168)$	$U(0.152, 0.168)$	$U(0.202, 0.218)$	$U(0.152, 0.168)$	$U(0.152, 0.168)$
t_4^0	-	-	-	-	$U(0.0475, 0.0525)$
σ_{g1}	0.01	0.01	0.01	0.01	0.01
σ_{g2}	0.015	0.015	0.015	0.015	0.015
σ_{g3}	0.005	0.005	0.005	0.005	0.005
σ_{g4}	-	-	-	-	0.01

Table 1: The distributions parameters for equation 16 are drawn from when creating a realization of a simulated phrase

The goal of the simulated signals is to create one type that qualitatively approximates the real signals, the other types are then variations of that base type that are different in some predictable way. The base type can then be compared to the variations to see how well the methods can recognize the type of change present in each variation. The signals are primarily modeled after the example signal shown in section 2.1 and have some important characteristics: They contain several distinct components with a variable amount of definition, they are not as distinct in every realization. They also contain a chirp, a section with continuously decreasing frequency. They also have a large variation in amplitude. The simulated signals are not necessarily that similar the the real signals in a quantitative sense, the primary differences is that the real signals have more complexity and structural noise. The real signals also vary in more complex ways than the simulated signals.

Realizations of the five types of simulated signals are shown in figure 9. Type 1 is meant to act as the base type. Type 2 is the same as type 1 except the final envelope has been moved to peak slightly later. Type 3 is the same as type 1 except it is entirely shifted in time by about 50 ms. Type 4 is the same as type 1 except it is entirely shifted in frequency by about 5 kHz. Type 5 is the same as type 1 except it adds another enveloped sinusoidal signal at the start of the phrase.

Each phrase simulated is realized a little differently no matter what type it is. The centers of the envelopes varies a little in time, the sinusoidal signals varies a little in frequency and the amplitude of each envelope varies a lot. The signals are also corrupted by Gaussian white noise with standard deviation 0.0005 which is approximately the noise encountered in the real signals.

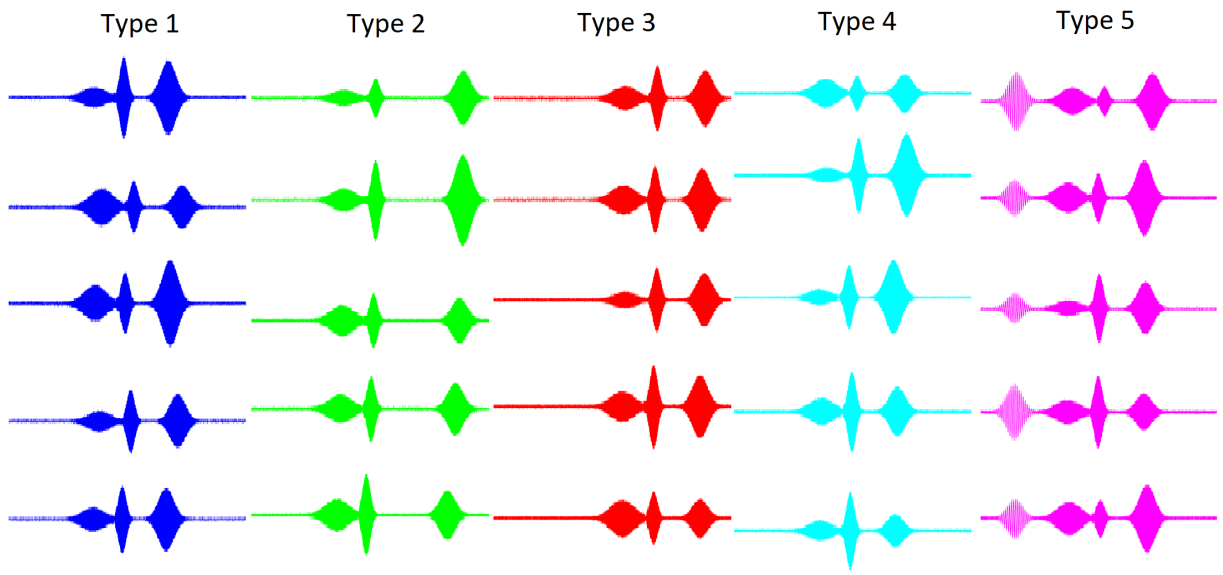

Figure 9: Simulated phrases separated by type.

3.2 Real data

The data set of recorded birds consists of 3583 sounds of which 1722 are of type cissel, 1634 are of type dick and 184 are of type trill. Each phrase has an associated bird name and date. Most of the sounds also have a location associated with it. Some of the birds are marked and can be tracked across time while some are unmarked and can have different names for different dates. The sounds are recorded between 2006 and 2014.

The phrases from real birds are shown in figure 10 and figure 11. The sounds in figure 10 are from the cissel class and from four different birds recorded in four different locations. The sounds in figure 11 are from all classes and from a single bird.

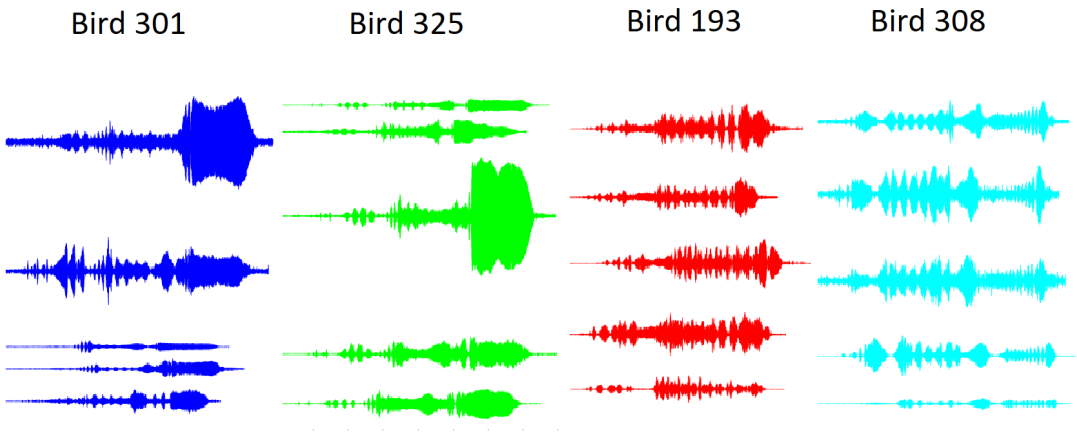

Figure 10: Real phrases of the cissel class from four different birds, separated by bird

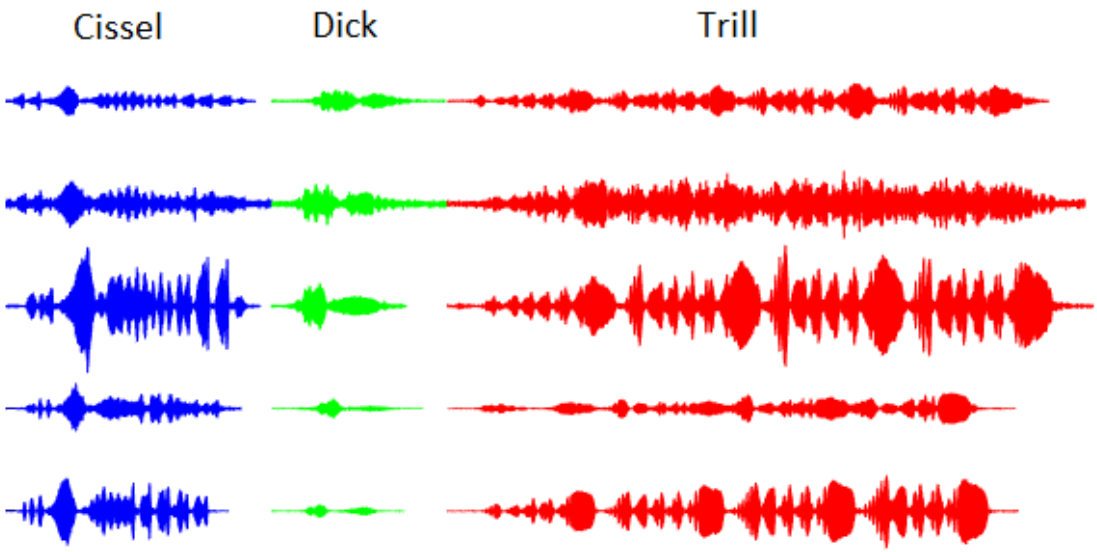

Figure 11: Real phrases of different classes from bird 319, separated by class

The data is partly collected at seven different sites around Manhattan, Kansas. Four of the locations are grassland locations and three are cropland. The approximate locations are plotted on a map from OpenStreetMap in figure 12. Yellow arrows are cropland while green arrows are grassland.

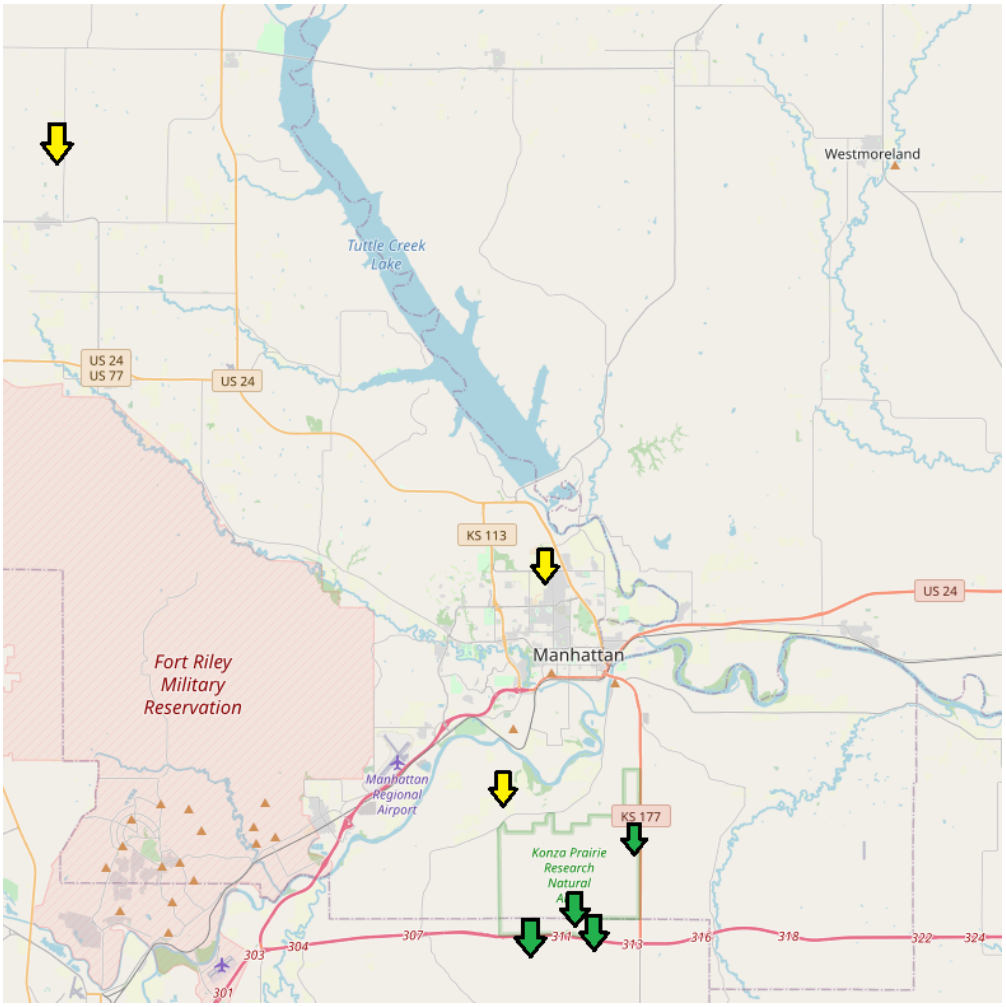

Figure 12: Map of sites where sounds have been collected. Yellow arrows mark cropland sites while green arrows mark grassland sites

3.3 Parameter tuning

Some of the algorithms have parameters that need tuning. An initial tuning was made by visual inspection. A better tuning was then found by testing the accuracy of several candidate parameter setups and picking the one with with highest accuracy on a single grassland site and class cissel after $N = 2000$ tests using the following algortihm:

1. set $n = 0, acc = 0$
2. Pick a phrase at random from the testing set, call that phrase "phrase 1".
3. Pick a different phrase at random from the same bird that made phrase 1, call that phrase "phrase 2".
4. Pick two new phrases at random from the testing set, call them "phrase 3" and "phrase 4".
5. If phrase 3 and phrase 4 were recorded from the same bird, go to step 4, otherwise continue.

6. Calculate the distance/similarity between phrase 1 and phrase 2 and between phrase 3 and phrase 4.
7. If the distance is smaller or the similarity is larger between phrase 1 and phrase 2 than between phrase 3 and phrase 4, set $acc = acc + 1$.
8. Set $n = n + 1$
9. If $n < N$, go to step 2, otherwise continue
10. Calculate the accuracy as acc/N

The parameter setups found are presented in table 2.

Method	**Parameter**	**Value**
SPCC	σ_f	67 hz
LoG	α_{log}	10 %
PFC	α_{pfc}	10 %

Table 2: Parameter tunings

4 Performance on simulated data

4.1 Accuracy

In order to test how sensitive the methods are to different types of changes in a signal the methods are given two pairs of signals. One pair has two simulated phrases of type 1 and one pair has one phrase of type 1 and one phrase of a different type. The methods are then asked which pair is which. This is done $N = 2000$ times for each method and type using the algorithm in section 3.3. Each test has signals generated for it and then thrown away so that every test has a unique realization of the signals. The results are presented in table 3. A 95 % confidence interval is approximately ± 1 %.

Method	**Type 1vs2**	**Type 1vs3**	**Type 1vs4**	**Type 1vs5**
SPCC	61 %	51 %	93 %	71 %
Groutage	99 %	50 %	84 %	100 %
PFC	92 %	95 %	67 %	83 %
LoG	61 %	75 %	55 %	66 %
Ambiguity	51 %	90 %	66 %	90 %

Table 3: Accuracy of classifying whether a pair of signals both come from type 1 simulated signals or whether one of them has a different type, for each column the type that is not type 1 changes

A high accuracy identifying type 2 sounds represent a high sensitivity to changes timing of notes in the signal relative to each-other. A high accuracy is found by the Groutage and PFC methods. A high accuracy identifying type 3 sounds represent a high sensitivity to how the phrase is segmented from the larger signal and a lower accuracy is better. The PFC, LoG and Ambiguity methods have some sensitivity to this while the SPCC and Groutage methods are completely invariant to zero padding at the start of the phrase. A high accuracy identifying type 4 sounds represent a high sensitivity to frequency shifts of the entire signal. The SPCC and Groutage methods have a high sensitivity. A high accuracy identifying type 5 sounds represent a high sensitivity to extra components in the signal. All methods have some sensitivity to this but the Groutage method stands out as the most sensitive.

The results here seem to indicate that the Groutage method performs well due top its high accuracy on type 2 and 5. It also has the nice property of being completely invariant to the phrase position in the signal which it shares with the SPCC method. It can also be seen that the SPCC method has a very high sensitivity to frequency shifts die to its high accuracy on type 4. Because the frequency sensitivity of the SPCC method is a tuneable parameter that is tuned to give good results on the real data this indicates that a high frequency sensitivity will correlate to good results on the real data.

4.2 Noise tolerance

In order to test how well the methods work in noisier environments the methods are given two pairs of signals. One pair has two simulated phrases of type 1 and one pair has one phrase of type 1 and one phrase of type 5. Type 5 was chosen as the comparison type because most of the methods can

tell it apart from type 1 fairly well in low noise conditions and this will make the noise sensitivity tests more accurate because the difference between high and low noise accuracy will be bigger. The methods are then asked which pair is which. This is done $N = 10000$ times using the algorithm in section 3.3 with different amounts of pink noise added to the signals. The pink noise was generated using MATLAB's *pinknoise* command and the noise level was adjusted by multiplying the resulting pink noise vector by a scalar. The noise level is measured using *Signal to Noise Ratio (SNR)* wich models the signal as a signal part y and a noise part ω such that $x = y + \omega$. The SNR is then:

$$SNR_x = \frac{\sum_i y_i^2}{\sum_i \omega_i^2}. \tag{17}$$

The accuracy at each level is evaluated and shown in figure 13.

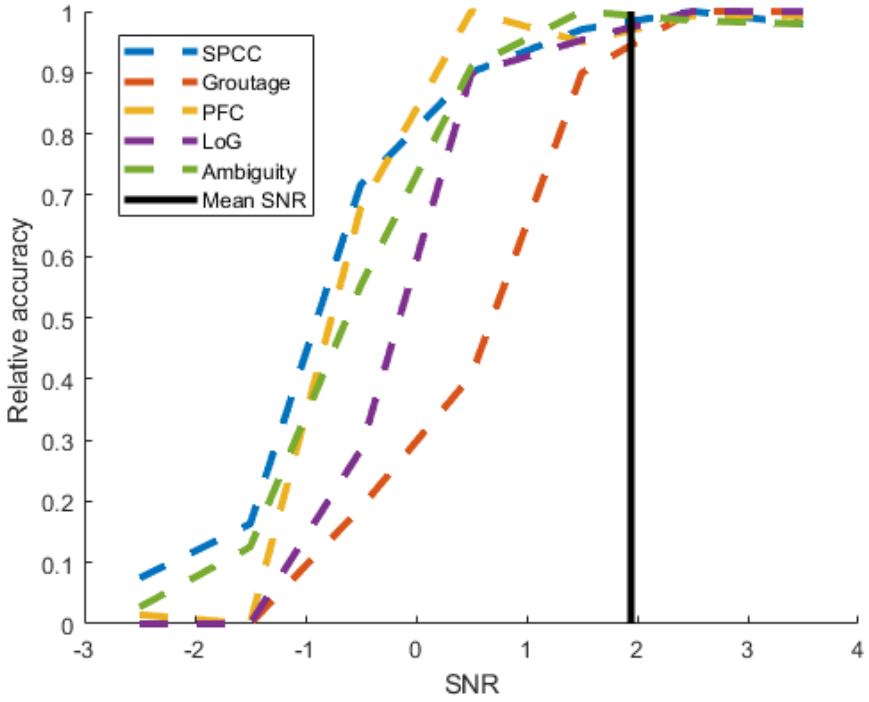

Figure 13: Accuracy with different levels of pink noise for the different methods and the mean SNR in the real signals

The figure shows that the Groutage method has a much lower noise tolerance than the other methods while the SPCC method has a consistently good noise tolerance. The figure also shows the estimated mean noise level in the real signals. The noise estimate was collected as the mean power in the first and last 50 samples of each phrase. The mean SNR in the signals was estimated to be 1.94 and 95 % of the signals had an estimated SNR in the range $(0.9, 3.0)$. In this range the difference between the methods is negligible aside from the Groutage method which is worse in the lower SNR range. This means that for the recordings used in this thesis noise tolerance will not be a differentiating factor except for the Groutage method which will perform worse due to the noise level of some of the recordings.

4.3 Continuity

In order to test if the methods actually provide a metric and not just a binary yes/no two sounds were simulated at the same time but different frequencies. The lower frequency sound was then shifted in time and the distance between the unshifted signal and the shifted signal were calculated for increasing amounts of shifting using every method. The setup is shown graphically using the spectrogram of the signal in figure 14. The equation describing the signal is:

$$s_{\Delta t}(t) = sin(\pi t \cdot 10^4) \cdot G(0, t - (0.1 + \Delta t)) + sin(3\pi t \cdot 10^4) \cdot G(0, t - 0.1) \tag{18}$$

where $G(\cdot, \cdot)$ is the Gaussian kernel from equation 9 with $\sigma_g = 0.01$. The result is presented in figure 15.

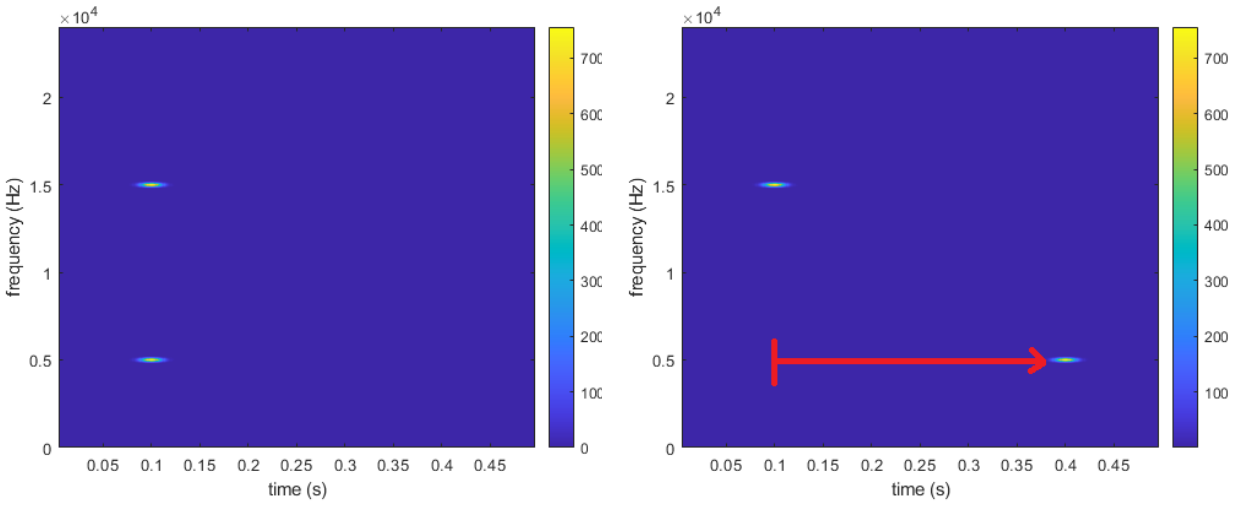

Figure 14: Left: Spectrogram of the base signal. Right: Spectrogram of the signal with a time shift of 300 ms

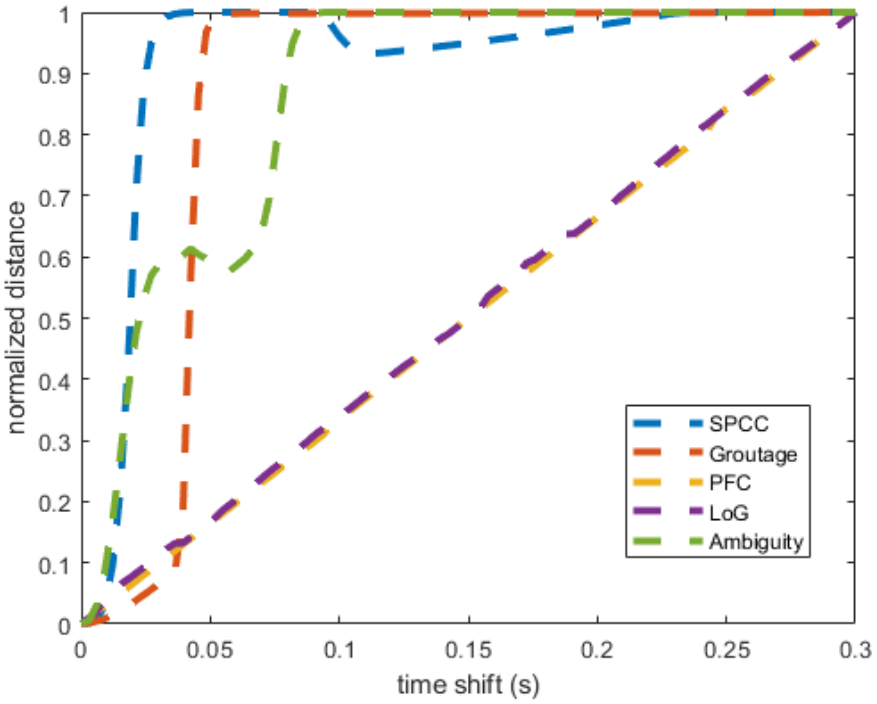

Figure 15: Normalized distance between the base sound and a sound that has had a component shifted in time

The distance is normalized using subtraction and division so that the lowest distance is zero and the highest distance is one. All of the methods rise smoothly but the SPCC, Groutage and Ambiguity methods reach a point of saturation in the difference, the PFC and LoG methods perform the best in this respect. There is also a bump down in the Ambiguity method and the SPCC method. The cause of this is unknown but probably related to how the cross-correlation and ambiguity spectrum works. The purpose of this test is more as a sanity check rather than something to rank the methods by. Seen in this light the inconsistencies in the Ambiguity and SPCC methods are so small that all methods pass this test.

5 Performance on real data

5.1 Accuracy

In order to test the methods performance on the real world data each method was given two pairs of phrases. One pair had both phrases from the same bird and the other pair had two phrases from different birds. The methods were then asked to classify which one was which. This was repeated $N = 2000$ times for each method and phrase type using the algorithm in section 3.3. The signals were chosen randomly from the entire data set and replaced between repetitions so a signal has the same chance of being chosen every repetition. The result are shown in table 4. A 95 % confidence interval is approximately ±1 %.

Method	**Cissel**	**Dick**	**Trill**	**Average**
SPCC	80 %	73 %	75 %	76.1 %
Groutage	77 %	73 %	70 %	73.3 %
PFC	73 %	66 %	66 %	68.4 %
LoG	67 %	73 %	61 %	67.0 %
Ambiguity	73 %	77 %	66 %	71.8 %

Table 4: Classification accuracy on different types and classes

The SPCC method performs well on every class and is the best on both cissel and trill, it also has the highest average accuracy. The Groutage method performs the second best or tied second best on all classes, it also has the second highest average accuracy. The Ambiguity spectrum method performs the best for the dick class but performs poorly on the trill class. The PFC and LoG methods both perform decently on one class but poorly on the others.

5.2 Correlation with distance

Further analysis will be limited to the two methods with highest total accuracy on the previous test, Groutage and SPCC, in order to be able to look more closely at the results. These two are picked because they can be seen as the most successful methods and therefore the most interesting to use.

Since the region a phrase was recorded in is known a test is made to see if the distance between recording sites correlates to an acoustic distance. To do this the mean acoustic distance between every pair of sites is calculated and plotted against the geographic distance between those sites for the cissel class. While the sites are not single geographical points the distance between the sites is usually larger than the distance within sites. The difference between sites will also be accentuated by averaging the distances between sites. The point of this test is to see if the methods can find structure in the data and if they do to see if the results differ. This test in particular is useful because it tests the method on a regression task, the methods need to give levels of difference rather than just saying if two signals are the same or different. The results are presented in figure 16.

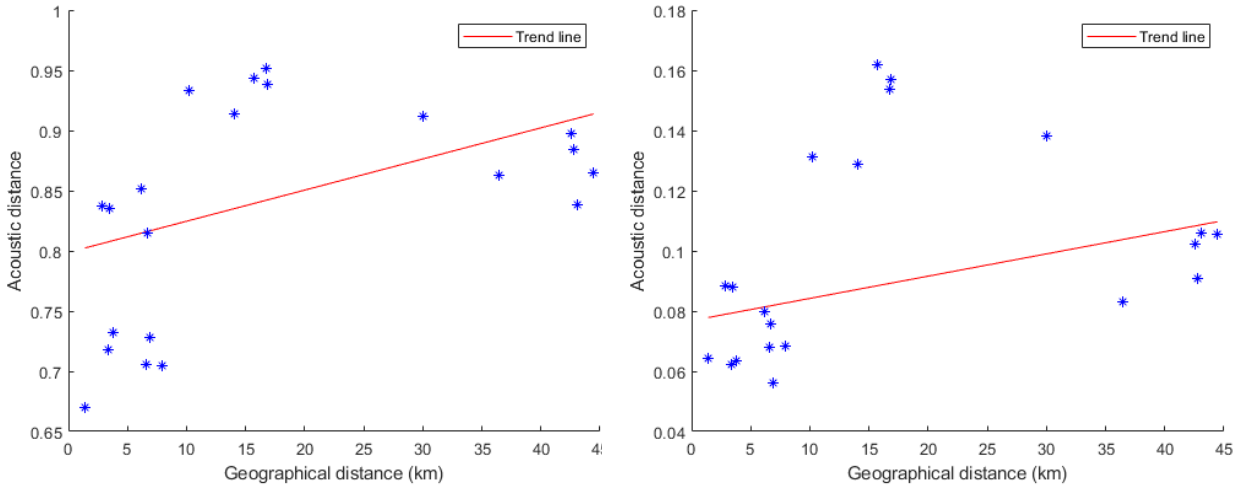

Figure 16: Left: Similarity over geographical distance using SPCC Right: Acoustic distance over geographical distance using the Groutage method. Trend lines are fit using robust linear least squares.

Both method seem to show that phrases are more similar between sites with a shorter distance but the correlation is much stronger using the SPCC method, $r^2 = 0.22$ for SPCC and $r^2 = 0.10$ for Groutage. The similarity produced by the SPCC method has been transformed into a distance by taking $1 - similarity$ to make the results easier to compare. This works because the similarity only gives values in the range $(0, 1)$. Both methods seem to give the same general shape to the data.

5.3 Homogeneity

A different type of structure that might be found in the data is a difference of homogeneity in the songs between grassland and cropland sites. This is tested for by taking two random birds from a single site and calculating the similarity/distance between them. This is repeated $N = 2000$ times for each type of site. The distributions are then presented as box-plots in figure 17.

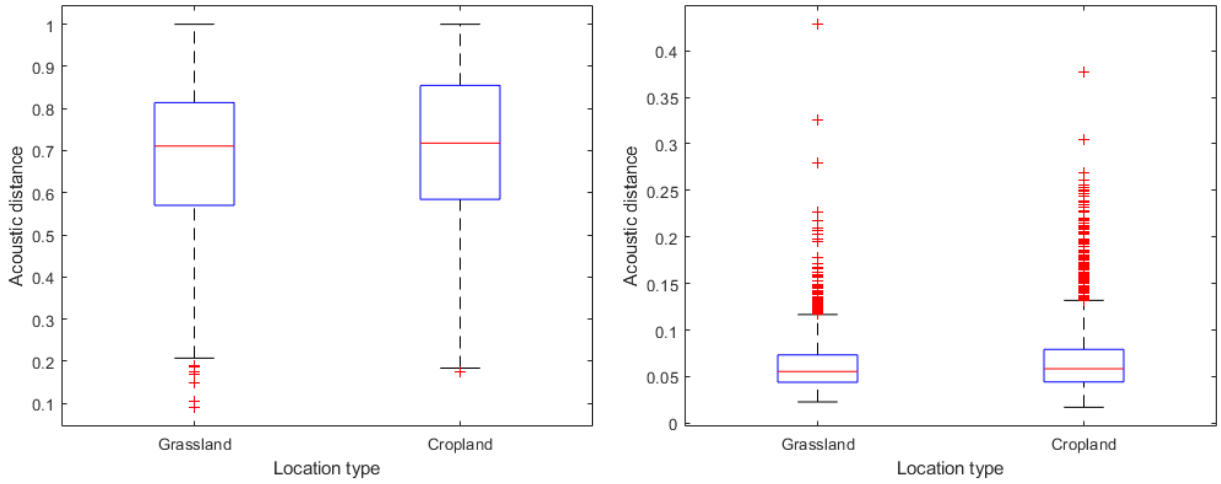

Figure 17: Left: Similarity within sites for the grassland sites vs the cropland sites using SPCC Right: Similarity within sites for the grassland sites vs the cropland sites using the Groutage method.

The p-value for the means being smaller in the grasslands is smaller than 0.001 for both methods. The SPCC method has again had its result transformed using $1 - similarity$ to create acoustic distance. This test reaffirms what could be seen in the previous test that the two different methods let us see the same structures when analyzing the data. A clear difference is however that the distributions look different, the Groutage method has large tails towards larger acoustic distance.

6 Discussion

The SPCC method seems to work the best the tests performed here since it has the highest accuracy on the real data. It also has the advantage over the method with the second highest accuracy, the Groutage method, of not requiring any training data aside from the data used to tune its parameter. The noise resistance tests seem to indicate that the Groutage method suffers a bot from the noise in the signals, even at the relatively low noise level in the testing data. This means that the Groutage method should not be used for noisier signal but might be useful in even less noisy environments, such as in a lab.

The accuracy of the methods against simulated types 2 and 5 do not seem to correlate to a high accuracy on real data. This is likely due to the higher complexity in the real data or that the variation withing birds in the real data is different from the variation between realizations of a simulated phrase. The properties found such as continuity and sensitivity to noise and time shifts should still be correct however. The continuity test should be viewed as a sanity test that all methods passed rather than something to rank the methods by since any real signal is going to be far more complex. The bumps in the ambiguity and SPCC methods are small enough to ignore.

No claims about the real world based on the analysis made here, only about the methods and the data set. For example: is the distance relation because the similarity correlates to distance in reality or is it just a difference between grasslands and croplands? Is there some other explanation? Based on only this analysis we cannot know.

One avenue of possible future research for the methods is to look at different metrics other than the modified Hausdorff distance. The metric was chosen because of its relative low sensitivity to outliers but evaluating the performance of other metrics could possibly uncover a better choice. It would also be interesting to see if a summary statistic of the feature vectors could be used instead of a metric between sets of feature vectors, a sort of feature of features. The PFC method is probably the most pliable of the methods. One way it could maybe be improved is to lower the acceptance rate of peaks and lines and then include the ones that has the best fit. You could also do things like increase the acceptance radius as the line becomes longer. The Groutage method suffers a lot when the matrices in the decomposition has multiple modes, this might be something that could be addressed by smoothing and splitting the matrices to only have a single mode per decomposed matrix.

The performance evaluation is primarily composed of a classification task while the problem that is being solved is a regression problem which can lead to things being missed in the performance evaluation. For instance the large tails in the Groutage method could be indications that when the method misclassifies two signals the error is very large. I would therefore err on the side of making the methods too robust rather than being too sensitive when implementing them in different contexts. For example the parameter σ_f for the SPCC method should rather be too big than too small and the same goes for the cutoff for singular values in the Groutage method. I could also see some merit in exchanging the Gaussian kernel in the SPCC method for a more fat-tailed distribution.

On the other hand: When making analyses of the real data of the type the methods were built to do there is a negligible difference between the two most promising methods. The methods also find structure in the data set when performing regression tasks. This indicates that the risk of having

created classification methods instead of regression methods might not be that big.

It is not easy to say whether the goal set in the introduction of finding a good method for comparing the songs of the dickcissel is met as it depends on what we mean by good. No method examined is good enough to give a reliable similarity between any two songs, however the SPCC method seems to be good enough to find relationships and test hypotheses in a larger data set, such as the one collected by Timothy Parker.

References

[1] C. Kwan, K.C. Ho, G. Mei, Y. Li, Z. Ren, R. Xu, Y. Zhang, D. Lao, M. Stevenson, V. Stanford, and C. Rochet. An automated acoustic system to monitor and classify birds. *EURASIP Journal on Applied Signal Processing*, 2006:52–52, 01 2006.

[2] C. Lee, Y. Lee, and R. Huang. Automatic recognition of bird songs using cepstral coefficients. *Journal of Information Technology and Applications*, 1(1):17–23, 2006.

[3] K. A. Cortopassi and J. W. Bradbury. The comparison of harmonically rich sounds using spectrographic cross-correlation and principal coordinates analysis. *Bioacoustics*, 11(2):89–127, 2000.

[4] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann. Audio based bird species identification using deep learning techniques. In *CLEF*, 2016.

[5] D. Groutage and D. Bennink. Feature sets for nonstationary signals derived from moments of the singular value decomposition of cohen-posch (positive time-frequency) distributions. *IEEE Transactions on Signal Processing*, 48(5):1498–1503, 2000.

[6] R Szeliski. *Computer Vision*. Springer, 2011.

[7] O. Tchernichovski, F. Nottebohm, C. E. Ho, B. Pesaran, and P. P. Mitra. A procedure for an automated measurement of song similarity. *Animal Behaviour*, 59(6):1167 – 1176, 2000.

[8] M. Große Ruse, D. Hasselquist, B. Hansson, M. Tarka, and M. Sandsten. Automated analysis of song structure in complex birdsongs. *Animal Behaviour*, 112:39 – 51, 2016.

[9] A. Conci and C. S. Kubrusly. Distance between sets - a survey. *Advances in Mathematical Sciences and Applications*, 26:1–18, 2017.

Popular abstract

A method for comparing the similarity of dickcissel songs

In this thesis some methods for automatically deciding the similarity of dickcissel songs are explored and compared. The method found to be working the best is a previously known method called the SPCC method.

Using computers to analyze bird songs has given us great possibilities in recent times. Not only does it reduce the workload of researches looking through audio files for the presence of birds it also allows for tasks that would be impossible otherwise such as computing the pairwise similarity between thousands of songs or tracking the positions of birds in real time. One particularly interesting application tracks the locations of birds around airports to prevent collisions between birds and aircraft.

While big steps have been made in the past there are still many obstacles left and all known methods have some downside that makes them unsuitable for some tasks. What is often done these days is to develop a method specifically for the bird species analyzed. Timothy H. Parker of Whtiman College, Walla Walla, has collected dickcissel songs and wants to find an algorithm that can compare two songs and say how similar they are. The goal is to see how bird culture changes over time and space and what the effect of different environments are on songs. In the thesis a method is found that is specialized towards the similarity of dickcissel songs.

Several methods are discussed and some are examined in greater detail. To test the methods some artificial songs are synthesized and by analyzing the way the methods react to different signals properties can be derived. For example some of the methods are very sensitive to shifts in the pitch of the signal while some are not. The methods are also evaluated on the real songs collected in Kansas, USA. Based on how accurate the methods are on these songs the performance can be evaluated.

The method that ends up performing the best is a well known general method called spectrographic cross-correlation. It works by seeing how well the signals correlate in time and frequency. Using this method some structure can be found in the data set but further analysis is needed to make statements about the birds themselves.