Identification of Songbird Species in Field Recordings

Hsiao-Yu Tung
Machine Learning Department
htung@andrew.cmu.edu

De-An Huang
Robotics Institute
deanh@andrew.cmu.edu

Xiao-Feng Xie
Robotics Institute
xfxie@cs.cmu.edu

Yurui Zhou
Electrical & Computer Engineering
yuruiz@andrew.cmu.edu

Joseph Russino
National Robotics Engineering Center
jrussino@rec.ri.cmu.edu

1 Introduction

It is important to gain a better understanding about the climate and ecological changes in the world. One way to address this is to study seasonal migration patterns in songbird populations, since birds respond quickly to environmental changes [29]. During migratory periods, many species of songbirds use flight calls, which are species-specific and are distinct from other vocalizations. Therefore, flight calls information can be used to determine the relative abundance of species and is important to understand long-term population trends. Due to costly human effort to collect data about birds in traditional methods, using machine learing (ML) methods to identify bird species from continuous audio recordings has been a hot topic in in recent conference competitions.[1] Although there are some recent advances [4, 20, 21, 26], it is still an open ML problem to reliably identify bird sounds in field recordings data due to simultaneously vocalizing birds and various background noise [6].

In this project, we will focus on critical aspects of this problem. We start from some existing classification methods, e.g., [3, 4, 7, 20, 21, 26], adapt components to the data we have, and finally develop a scalable software tool. The total process is divided into four steps. First, Audio data are first preprocessed into spectrograms. The spectrograms are further cleaned by applying background noise reduction and image processing techniques, and connected pixels (acoustic patterns) in the spectrograms are labeled into rectangle segments. Second, features are then extracted and selected from different sources, e.g., file statistics, segment statistics and probabilities, and mel-frequency cepstral coefficients (MFCCs). Third, the classification is then done by using multiple algorithms, e.g., naive Bayes, k-nearest neighbors (k-NN) [9], support vector machines (SVM) [8], etc. Finally, we will explore some ensemble methods [1,2,14,17,19,24,25] for further exploring some properties on overall performance by combining the predictions of models, as well as facilitating scalability in real-world usages. The software developed for this project will be used by the Carnegie Museum of Natural History, and possibly shared with other land managers, researchers, and educators to enhance the use of flight calls as a method to study the patterns of migratory songbirds.

2 Related work

As mentioned in the Introduction, the total process is divided into standard steps of ML components.

The first part is about *preprocessing and segmentation* audio data of songbirds. Normally audio files are first processed into grayscale image by applying the Fourier transform using a Hanning or Hamming window of samples with some overlap [5, 20]. Only relevant frequency range of the scope of domain interests are kept. The narrowed spectrograms are then treated as graysale images. To reduce the background noise, a *median clipping* method can be applied on each frequency band and time frame [5]. A spectrogram can also be processed using a series of sub-processes including

[1] ICML 2013: The Bird Challenge; NIPS 2013: Multi-label Bird Species Classification; MLSP 2013: Bird Classification Challenge

Gaussian filtering, local gradient, thresholding, and morphological nosing removal [13]. The resulting images can be further handled using standard image processing techniques such as dilution and median filter (e.g. using scikit-image). In [20], neighboring pixels exceeding certain spatial threshold (acoustic patterns) in the spectrograms are labeled into rectangle segments. In [13], small segments are discarded and remaining holes are filled. In [5], a supervised single-instance single-label classifier is used to label the probability of each pixel as bird sound or noise, and then obtain predicted segementations by applying a threshold.

The second part is about *feature extraction and selection*, which can have a large impact on later classification results. In [5], segment features are divided into two different categories - "mask descriptors" and "profile statistics". Histogram of gradients (HOG) have also been used [5]. In [13], features are obtained by applying the template matching function of scikit-image to compute the similarity at the maximum value of the normalized cross-correlation map with templates. In [20], features come from three different sources, i.e., file statistics, segment statistics, and segment probabilities. The template matching in OpenCV library is applied only on absolute-intensive spectrograms. Many existing work [7, 10, 21, 27] considers MFCCsas features. Features can also be extracted using unsupervised deep learning [22]. Additional methods, e.g., rescaling [5], concatenation [10], bag-of-words (BoW) model [12], and principal component analysis (PCA) [18], can also be used for feature engineering.

The third part is about *classification*. Typical methods include naive Bayes, neural networks, logistic regression, Gaussian mixture model, radial basis function (RBF), decision trees, k-NN [9], SVM [8], etc. Binary classifiers can be turned into multi-class ones by using general strategies, e.g., one-versus-all and pairwise decomposition, to classify instances into multiple classes. Some existing methods have been used for songbird identification. In [10], a LibSVM is used in a one-versus-all fashion, and best scores have been obtained with C-SVC SVM type and linear kernel function. In [22], pairwise SVM, LibSVM, decision trees, and neural networks are used, and the merged SVM and RDT often leads to better results. Mutiple-instance multi-label (MIML) classifiers, e.g., MIML-SVM, MIML-RBF, MIML-kNN are considered in [5].

Finally, *ensemble learning* methods have also been used for combining the predictions of several models. In theory, ensembles have more flexibility in the functions they can represent. Typical methods including gradient boosting (GB) [15], random forest (RF) [2], extremely randomized trees (ERT) [16], bootstrap aggregating (or bagging) [1], Bayesian model averaging (BMA) [17], and ensemble of classifier chains (ECC) [24]. Some of them, e.g., GB [7], RF [7, 13, 27], ERT [20], have been directly used for songbird identification. There are also some hybrid methods. In [7], GB and RF are combined by a simple linear blending method, where the final prediction is given by a linear combination of ensemble predictions, and the combination coefficients are determined by a Lasso and elastic-net regularization of generalized linear model. In [21], an ensemble of logistic regression and GB classifiers are considered. In [3], an ensemble of classifier chains with RF is applied.

Many classification and ensemble methods can be obtained in scikit-learn library.

3 Methods

3.1 Preprocessing

Figures 1 to 4 show the preprocessing steps. Similar to the methods used in [20], we first convert audio files into spectrogram images, and for each segments we use Hanning windows with 75% overlap. Notice the case that in a processed grayscale image most area was occupied by the random noise. What we want is to get rid of the background noise completely and increase the contrast between real signal and the background. Given the several different algorithm tested, the median clipping algorithm [5] works best because it not only removes most background noise, but also capture the sound feature clearly and precisely.

Such a algorithm, though perform well in noise removing and feature capturing, requires large amount of computation because the average and variance of each column of the row need to be calculated. Given the huge size of the processed grayscale image, we need a more efficient algorithm to segment the image. Further experiments is also needed to explore an balance point between noise reduction effectivity and efficiency.

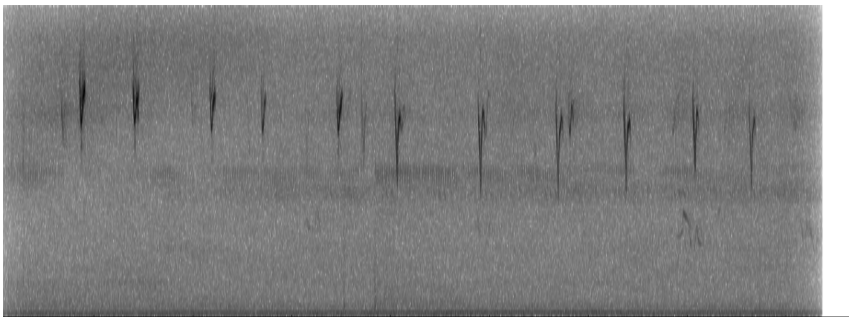


Figure 1: original spectrogram

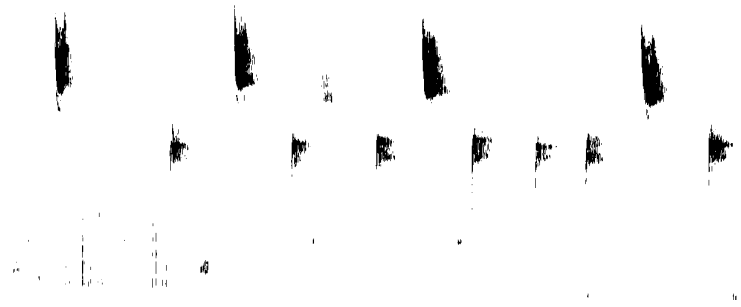


Figure 2: Median clipped spectrogram

Figure 3: Eroded and propagated

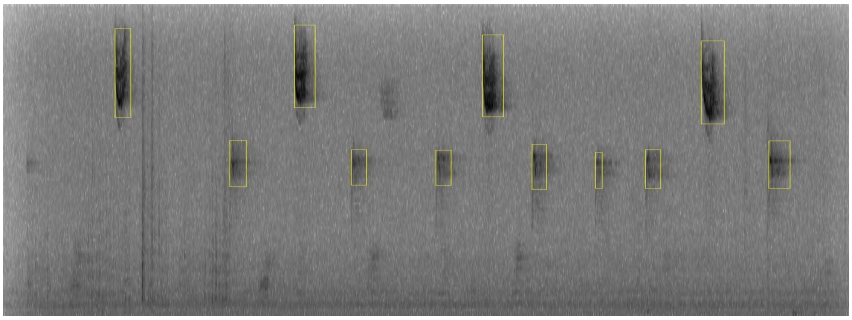


Figure 4: Segmentation

Finally, we apply standard image processing techniques to further reduce the residual noise dot. And then we would use find connected pixels from the image and label the segments, as in [20].

3.2 Features

Features can have a large impact on classification results. For this project, we have chosen to implement a large set of features used in previous successful attempts at birdsong classification. We will ultimately choose a smaller subset of these features that provide the most discriminatory power.

Mel-Frequency Cepstral Coefficients While flight calls recognition is a new application, audio recognition is a well developed area in signal processing. We will build features based on MFCCs, which are commonly used as features in speech recognition systems, but can be sensitive to the presence of noise [28]; for this reason, we expect that they may be useful features for classifying the clear lab data but less useful for the noisy outdoor data. A temporal signal is first transformed into a series of frames where each frame consists of a 13-dimensional MFCC vector, using Wojcicki's MATLAB implementation. [2] Each frame represents a duration of 12ms. The step size is 4ms. We also include the fist and second derivatives of MFCCs, which results in a 39 (13×3) dimensional vector for each frame.

Bag-of-Words Model over MFCCs In contrast to [11], which assumes that the number of frames is fixed for all audio segments to classify, we make no assumption on the length of the audio segment, since the length of flight calls might variate. In this case, we want a feature that is temporally scale invariant. Intuitively, even when a flight call is temporally scaled (extended or shorten), it should still be classified as the flight call of the same species. We leverage the progress in image classification and applied the bag-of-words (BoW) model [12] over our MFCCs. By treating audio features (MFCCs in our case) as 'words', each audio segment is represented by a sparse vector of occurrence counts of words in BoW model; that is, a sparse histogram over the vocabulary.

We learn the vocabulary, also called the codebook, by performing k-means clustering over sampled training MFCCs. Codewords are then defined as the centers of the learned clusters. In test time, we extract MFCCs from the test audio segment and quantize the 39 dimensional MFCCs to a learned codeword. The audio segment is represented by a K dimensional histogram over the codeword, where K is the size of the codebook.

One limitation of BoW model on audio segment is that the temporal information are lost in the model. However, as shown in the experiments, BoW is able to achieve satisfactory classification result on single-label single-instance case. We build higher order N-gram model on the learned vocabulary to capture the temporal information.

N-gram model One disadvantage of bag-of-words model is that the temporal information is ignored. We use the histogram of consecutive N words, which is of length K^N. Only the N-grams occurring more than three times in the training data are considered. Therefore, the actual length of the histogram is much smaller than K^N (4210 for $N = 2$ and 2084 for $N = 3$).

Denoising The first MFCC is the energy of the audio signal contained in the frame. Since the energy of noisy frames are lower than that of frames containing flight call signal. We can remove frames with lower energy and remove the noise.

Baseline The spectrogram based approach in [5] has been used as the baseline for the MLSP 2013 Bird Classification Challenge. We follow their approach and extract mask descriptors from the labelled regions shown in Figure 4 as our baseline.

3.3 Classifiers

We consider several classifiers/regressors.

Nearest Neighbor (NN). For each testing data, the NN classifier finds the nearest training data point and transfer the corresponding label. The NN classifier directly reflect the effect of our features and is used as baseline. Depends on the features, different distance metric should be used. We will compare the performance of euclidean and χ^2 distance on the BoW feature in the experiments.

[2] HTK MFCC MATLAB by K. Wojcicki: Mel frequency cepstral coefficient feature extraction that closely matches that of HTK's HCopy.

Support Vector Machine (SVM). The most common approach for multi-label classification is to use an ensemble of binary classifiers, where each classifier predicts if an instance belongs to one specific class or not. SVMs trained in a pairwise fashion has obtained state-of-the-art results on standard multi-label benchmark datasets [22]. Furthermore, we will be able to integrate different kernels to improve the performance using the SVM. For example, the χ^2 kernel will have a better performance for histogram than the linear kernel. We also consider support vector regression to produce soft output for ensemble learning.

Random Forest. Random forests has been widely used for multi-label classification [7, 20, 27, 31]. Random Forest is operated by constructing decision tree structure by the training examples. One of the popular algorithm is tree bagging, in which the training process includes repeatedly selecting a bootstrap sample of the training set and fitting the trees to them. After the training process, the label decision is made either on the majority of the votes or a weighted combination from individual trees.

Logistic Regression. Ensemble of logistic regression has been used for multi-label bird song classification [21] because of its simplicity. We also consider this method as the baseline for soft output.

3.4 Ensembles

An ensemble combines the prediction power of a set of individually trained classifiers H to produce a single classifier. Ensembles often are more accuracy as compared to base classifiers in the ensemble.

We focus on investigating *weighting* methods, as one of the two main methods discussed in [25], for combining the outputs of base classifiers. Let M be the set of examples, L be the set of labels, K be the set of classifiers, A_{M*} be the true labels, and A_{MK} be the labels assigned by K on M. We consider a generalized weighting scheme. Based on the labels A^{train}_{MK} on training examples, weights W and beliefs B are learned, where each $w_k \in W$ is the weight for $k \in K$, and each $b_{l_i l_j, k} \in B$ is the belief probability of true label l_i, given the label l_j assigned by k. For each testing example X, the ensemble classifier $\tilde{h}(X)$ assigns votes on labels L by the values WB for testing labels, and return the label or "UNKNOWN" using majority voting with a threshold value $\bar{V} \in [0, 1]$.

The weighting scheme is represented by a tuple $< F\text{-strategy}(\tilde{K}), C\text{-strategy}, B\text{-strategy}, \bar{V} >$.

The weight vector W can be obtained using a *two-step scheme*. First, F-strategy returns a binary array F about the chosen classifiers. Here a subset of top $\tilde{K}$ classifiers are picked with the highest diversity, which has shown to have positive relationship with the ensemble accuracy [19]. We consider nine measures of diversity [19], i.e., *disagreement*, *correlation coefficient*, *Q-statistic*, *double-fault*, *coincident failure diversity*, *entropy*, *interrater agreement*, *Kohavi-Wolpert*, and *generalized diversity*. Second, the weight coefficients C are assigned by C-strategy. The *equal assignment* just gives the same weight as in the canonical majority voting, where the *performance-based assignment* [23] assigns weights proportional to the accuracy performance of each classifier on M. The final weights is $W = F \cdot C$.

One can also obtain the weight coefficients C by optimizing in the configuration space. We consider DEPSO, a cooperative group optimizer [30], to obtain the optimal weights for K, and thus to know the lowest training error that can be achieved using weights on given A^{train}_{MK}. Without loss of generality, the weighting space is defined as $c_k \in [0, 1]$ for each weight coefficient on k.

B-strategy obtains the belief matrix B, by only considering forms that are independent on k. A *basic form* of B is a diagonal $L \times L$ sub-matrix for each k, which is equivalent to the one using in the canonical majority voting. A *new form*, which is not included in [25], obtains the beliefs as the frequency of (l_i, l_j) pairs, where $l_i = a_{mk} \in A^{\text{train}}_{Mk}$ and $l_j = a_{m*} \in A^{\text{train}}_{M*}$, on training examples.

In real-world usages of songbird identification, people would prefer a very higher precision rate, among the classified results on a big amount of testing examples. We accommodate the real-world requirement at the ensemble level, by defining one label as "UNKNOWN", to include those instances that are not surely classified. This is also has an implication for scalability, as the labels might change over time, and studies can be mainly performed on "UNKNOWN" instances.

4 Experiments

We first evaluate our system on single-label data to see if the data are separable using our method.

Data. We use the flight calls of songbirds manually segmented and labeled by Amy Tegeler, an Avian Ecologist from the Carnegie Museum of Natural History, to evaluate the performance of our system on indoor flight calls classification. We used the data from year 2008 to 2013. The total number of species is 32. But only 11 species has more than 100 data. Therefore, the experiment is performed on those 11 species, and 100 data points are randomly sampled from each species. The 100 data points are randomly partitioned into 20 testing data and 80 training data for each species.

Segmentation. Since the flight call segments are already manually identified, we do not perform segmentation in this experiment.

Features. We use the Mel-Frequency Cepstral Coefficient, BoW over MFCCs, denoised BoW over MFCCs, 2-Gram and 3-Gram in Section 3.2 as the features for this experiment. The size of the code book is 200, and the MFCCs consider the frequencies from 1500 Hz to 22000Hz.

Feature Analysis. Before beginning with the training process, we first examine the property of the features of BoW over MFCC. We can see that most of the value in the original data is zero. After taking the log value of the data, we can find that a small group of the data is separated from the others (see Figure 5). In some of the dimension, this small region contains only a number of specific species of birds, which might be helpful in the classification.

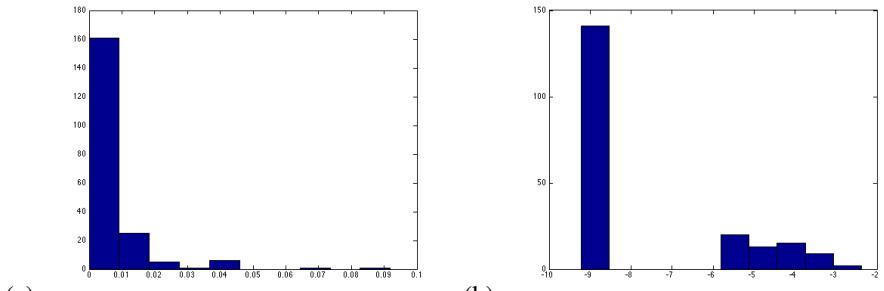
In the denoised version of BoW over MFCC and 2-Gram features, the features in some of the dimension can clearly separate a specific number of species only by zero and non-zero value. Figure 6 shows the 180^th features of denoised BoW over MFCC and the second feature in 2-Gram. In our experiments, we find that the species which can be separated very well by some of the dimensions of these features are classified more accurately than the others.

Feature Selection. Due to the limited size of our data, using all the features at the same time will lead to serious overfitting problem. So here feature selection is done by cross validation. After the process, we find that most of the classifiers have higher performance using the denoised version of BoW over MFCC and 2-Gram features. Taking a portion of the features does not help in increasing the performance. To demonstrate the difference between BoW over MFCC and denoised version of BoW over MFCC, we also report the performance of using the former.

Baseline. The accuarcy (using Random Forest) on mask descriptors by [5] is 0.25.

Classifiers. In Table 1, we compare the results using k-NN, SVM and random forest classifier.

First, taking the log over the features of BoW of MFCC decreases the accuracy in rbf SVM classifier. The features are expected to be more of a Gaussian distribution after taking log and can be separated better using rbf kernel. However, the performance decreases. The reason is that the transformed features are distributed in a skrewed Gaussian so that it can not be separated perfectly as well.



(a) (b)

Figure 5: (a) is the histogram of second feature. Most of the values are close to zero. (b) is the histogram of the original data after taking log.

Table 1: Accuracy of different classifier

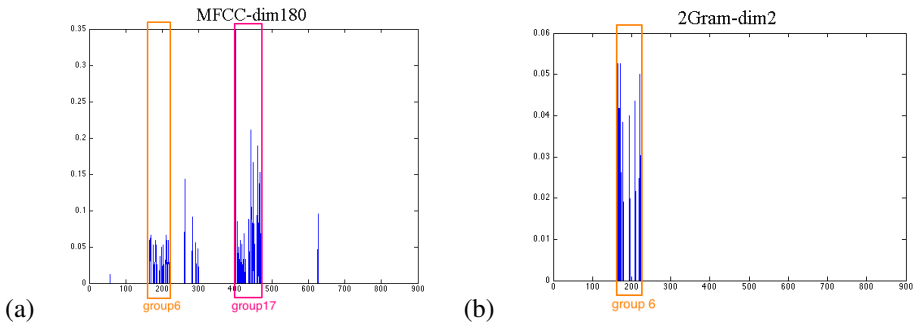
Classifier	Accuracy(%)	Features	Settings
linear SVM	67.7273/ 70.4545	BoW/ denoised	
poly SVM	69.0909/ 70.4545	BoW/ denoised	degree: 1
	70	denoised BoW	degree: 2
	70.4545	denoised BoW	degree: 3
rbf SVM	70/ 70.4545	BoW/ denoised	
	68	BoW (log)	
	76.8182	denoised BoW	$\gamma = 7.9433$
	78.1818	denoised BoW + 2gram	
sigmoid SVM	70.9091/70.4545	BoW/ denoised	
random forest	57/63	BoW/ denoised	100 trees; 5 splits
	63	denoised BoW	100 trees; 2 splits
NN-euclidean	54.09/62.73	BoW/ denoised	
NN-chisquare	62.27/71.82	BoW/ denoised	

Second, the performance of all kinds of classifier increases using the denoised features extracted from BoW of MFCC. Examine the features in images (see Figure 6) help us understand the properties of them. The denoised features concentrate on a small number of specific species.

Besides, among all the classifier, SVM classifiers have the highest performance in accuracy and random forest has the lowest performace. The reason why random forest does not perform well on our data is that the sample size is small. SVM with rbf kernel has the highest performance (accuracy: 0.7682) using denoised BoW. The accuracy reaches 0.7818 with denoised BoW and 2-Gram features. NN has higer performance using chisquare distance with denoised BoW (accuracy: 0.7182).

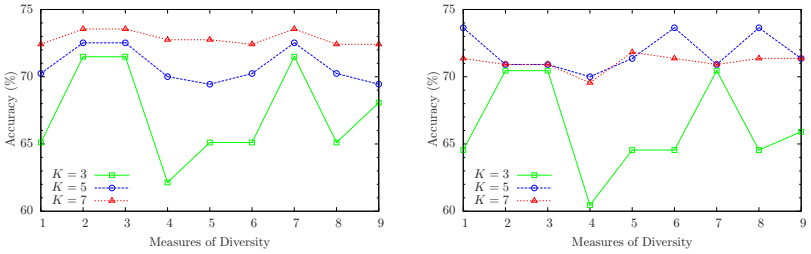
Ensembles. Ensemble learning is then performed on the labels by generated 8 classifers, including k-NN with euclidean, k-NN with chi-square, SVM-linear (-t 0), SVM-χ^2 (-t 5), SVM-RBF (-c 4000 -g 15 -t 2), linear SVM (-c 40), polynomial SVM (-c 1000 -g 1 -d 2 -t 1), sigmoid SVM (-c 1000 -g 1 -d 2 -t 3). We take 4/5 examples as training samples, and 1/5 examples as testing examples.

Figure 7 gives the training and testing accuracy for the nine measures of diversity (1. *disagreement*, 2. *correlation coefficient*, 3. *Q-statistic*, 4. *double-fault*, 5. *coincident failure diversity*, 6. *entropy*, 7. *interrater agreement*, 8. *Kohavi-Wolpert*, and 9. *generalized diversity*) in different $\tilde{K}$ values, where C-strategy is uniform, and B-strategy is the basic form. Accuracy increases as $\tilde{K}$ increases, which might due to K is not sufficently large. However, the test accuracy can be better as $\tilde{K} = 5$ for some measures of diversity.



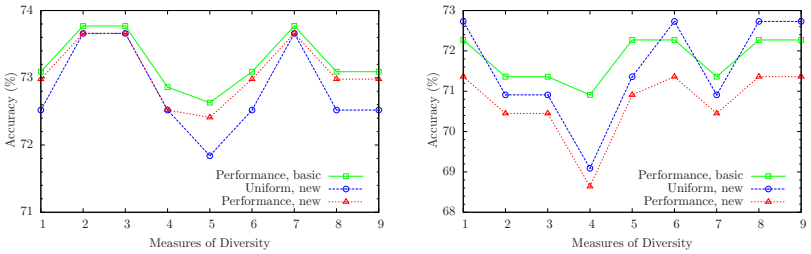
(a) (b)

Figure 6: (a) is the 180^th features of denoised BoW over MFCC. Most of the high values are around 2 species of birds. (b) shows the second dimension of the 2-Gram features. All the non-zero values appear only whithin one species.



(a) Training Accuracy (b) Testing Accuracy

Figure 7: Training and testing accuracy for different measures of diversity in different $\tilde{K}$ values.



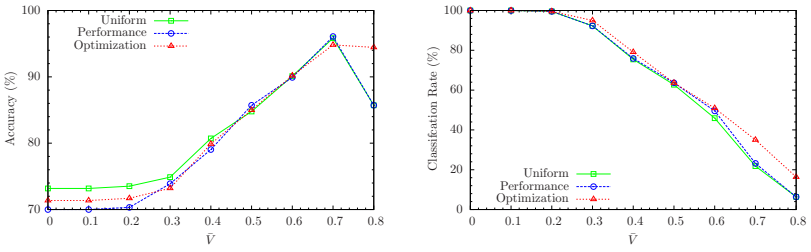
(a) Training Accuracy (b) Testing Accuracy

Figure 8: Training and testing accuracy for different measures of diversity in different $\tilde{K}$ values.

Figure 8 gives the training and testing accuracy for different measures of diversity in different C-strategy and B-strategy, where $\tilde{K} = 7$. The new form of B-strategy can lead to better testing accuracy for some measures of diversity, although does not help on training accuracy.

The optimization method using DEPSO is performed as $\tilde{K} = |K|$. For the basic and new forms of B-strategy, it respectively obtained 75.26% and 71.82% training and testing accuracy, and 75.26% and 71.36% training and testing accuracy. This gives the best training accuracy is 75.26%, and it does not lead to higher test accuracy.

Figure 9 gives the classification accuracy and rates for different C-strategy methods in different $\bar{V}$ values, where $\tilde{K} = |K|$, with the *new form* of B-strategy, on testing examples. In general, the accuracy accuracy increases and classification rate decreases, as $\bar{V}$ increases. However, there is a drop at $\bar{V} = 0.8$, which means the base classifers may reach agreements on some wrong labels. The optimizion method can keep the classification rate higher, especially in the high accuracy region. For example, for $\bar{V} = 0.7$, there is a nearly 95% accuracy rate with a 35% classification rate. This is critically useful for real-world applications based on statistics, as the sample size is huge.



(a) Classification Accuracy (b) Classification Rate

Figure 9: Classification accuracy and rates for different C-strategy methods in different $\bar{V}$ values.

References

[1] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] F. Briggs, X. Z. Fern, and J. Irvine. Multi-label classifier chains for bird sound. *arXiv:1304.5862*, (abs/1304.5862), 2013.

[4] F. Briggs, X. Z. Fern, R. Raich, and Q. Lou. Instance annotation for multi-instance multi-label learning. *ACM Transactions on Knowledge Discovery from Data*, 7(3):14, 2013.

[5] F. Briggs, B. Lakshminarayanan, L. Neal, X. Fern, R. Raich, S. J. K. Hadley, A. S. Hadley, and M. G. Betts. Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach. *Journal of the Acoustical Society of America*, 131:4640–4650, 2012.

[6] F. Briggs, R. Raich, K. Eftaxias, Z. Lei, and Y. Huang. The ninth annual mlsp competition: Overview. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2013.

[7] W. Chen, G. Zhao, and X. Li. A novel approach based on ensemble learning to nips4b challenge. In *Workshop on Neural Information Processing Scaled for Bioacoustics*, pages 195–197, 2013.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[10] O. Dufour, T. Artieres, H. Glotin, and P. Giraudet. Clusterized mel filter cepstral coefficients and support vector machines for bird song idenfication. In *Workshop on Machine Learning for Bioacoustics*, pages 89–92, 2013.

[11] O. Dufour, H. Glotin, T. Artires, Y. Bas, and P. Giraudet. Multi-instance multi-label acoustic classification of plurality of animals : birds, insects & amphibian. In *Workshop on Neural Information Processing Scaled for Bioacoustics*, pages 164–174, 2013.

[12] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531. IEEE, 2005.

[13] G. Fodor. The ninth annual mlsp competition: First place. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–2. IEEE, 2013.

[14] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[15] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[16] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[17] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: A tutorial. *Statistical science*, pages 382–401, 1999.

[18] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[19] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207, 2003.

[20] M. Lasseck. Bird song classification in field recordings: Winning solution for NIPS4B 2013 competition. In *Workshop on Neural Information Processing Scaled for Bioacoustics*, pages 176–181, 2013.

[21] L. Massaron. Ensemble logistic regression and gradient boosting classifiers for multilabel bird song classification in noise. In *Workshop on Neural Information Processing Scaled for Bioacoustics*, pages 190–194, 2013.

[22] E. L. Mencía, J. Nam, and D.-H. Lee. Learning multi-labeled bioacoustic samples with an unsupervised feature learning approach. In *Workshop on Neural Information Processing Scaled for Bioacoustics*, pages 184–189, 2013.

[23] D. W. Opitz and J. W. Shavlik. Generating accurate and diverse members of a neural-network ensemble. *Advances in neural information processing systems*, pages 535–541, 1996.

[24] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

[25] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010.

[26] E. Stattner, W. Segretier, M. Collard, P. Hunel, and N. Vidot. Song-based classification techniques for endangered bird conservation. In *Workshop on Machine Learning for Bioacoustics*, pages 67–73, 2013.

[27] D. Stowell and M. D. Plumbley. Feature design for multilabel bird song classification in noise. In *Workshop on Neural Information Processing Scaled for Bioacoustics*, pages 182–183, 2013.

[28] V. Tyagi and C. Wellekens. On desensitizing the mel-cepstrum to spurious spectral components for robust speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 529–532. IEEE, 2005.

[29] G.-R. Walther, E. Post, P. Convey, A. Menzel, C. Parmesan, T. J. Beebee, J.-M. Fromentin, O. Hoegh-Guldberg, and F. Bairlein. Ecological responses to recent climate change. *Nature*, 416(6879):389–395, 2002.

[30] X.-F. Xie, J. Liu, and Z.-J. Wang. A cooperative group optimization system. *Soft Computing*, 18(3):469–495, 2014.

[31] X. Zhang, Q. Yuan, S. Zhao, W. Fan, W. Zheng, and Z. Wang. Multi-label classification without the multi-label cost. In *SIAM International Conference on Data Mining*, pages 778–789. SIAM, 2010.