RESEARCH ARTICLE

Machine Learning Algorithms for Automatic Classification of Marmoset Vocalizations

Hjalmar K. Turesson[1]*, Sidarta Ribeiro[1], Danillo R. Pereira[2], João P. Papa[2], Victor Hugo C. de Albuquerque[3]

1 Instituto do Cérebro, Universidade Federal do Rio Grande do Norte, Natal, Brazil, **2** Departamento de Computação, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, São Paulo, Brazil, **3** Programa de Pós-Graduação em Informática Aplicada, Laboratório de Bioinformática, Universidade de Fortaleza, Fortaleza, CE, Brazil

* turesson@neuro.ufrn.br

Citation: Turesson HK, Ribeiro S, Pereira DR, Papa JP, de Albuquerque VHC (2016) Machine Learning Algorithms for Automatic Classification of Marmoset Vocalizations. PLoS ONE 11(9): e0163041. doi:10.1371/journal.pone.0163041

Editor: Michael Smotherman, Texas A&M University College Station, UNITED STATES

Received: March 3, 2016

Accepted: September 1, 2016

Published: September 21, 2016

Data Availability Statement: The dataset is available at https://osf.io/yqpvk/ and http://neuro.ufrn.br/data/marmosetvocalizations.

Funding: This work was supported by National Council for Scientific and Technological Development 402422/2012-0 (http://cnpq.br/) HKT and SR; National Council for Scientific and Technological Development 470501/2013-8 and 301928/2014-2 (http://cnpq.br/) VHCA; São Paulo Research Foundation 2013/07699-0 (http://www.fapesp.br) SR; São Paulo Research Foundation 2014/16250-9 and 2015/50319-9 (http://www.fapesp.br) JPP; and National Council for Scientific

Abstract

Automatic classification of vocalization type could potentially become a useful tool for acoustic the monitoring of captive colonies of highly vocal primates. However, for classification to be useful in practice, a reliable algorithm that can be successfully trained on small datasets is necessary. In this work, we consider seven different classification algorithms with the goal of finding a robust classifier that can be successfully trained on small datasets. We found good classification performance (accuracy > 0.83 and F_1-score > 0.84) using the Optimum Path Forest classifier. Dataset and algorithms are made publicly available.

Introduction

The common marmoset (*Callithrix jacchus*) is a species of arboreal New World monkeys native to the northeast region of Brazil. This species is becoming an increasingly important primate model of a number of human diseases, as well as for basic research in neuroscience [1, 2] and genetics [3]. Examples of the species' use as a disease model come from multiple sclerosis [4], herpes virus [5] and tuberculosis [6] research. Some of the reasons for this popularity are that marmosets have a similar disease susceptibility profile to humans, are relatively easy to handle, have a high reproductive rate, and that important genetic and neuroscience research tools already exist [7]. In particular, marmosets are an excellent model for the neurophysiological study of vocal communication [8]. A pubmed search on "Callithrix jacchus" shows between 113 and 164 publications per year during the last 10 years, with most of the publications in biomedicine. Given the widespread use of marmosets in laboratories, methods to reliably monitor health and behavior in captive colonies are of a high priority.

As typical for Neotropical arboreal primates, marmosets rely heavily on acoustic communication. This together with cooperative breeding and the complex social organization that follows, means that marmosets have a large vocal repertoire, with 9–13 different types of vocalizations reported [9–13]. The vocalizations produced by a group of marmosets provide a rich source of information about their activities and well-being [14]. However, although informative, it is practically untenable to manually track the vocalizations of a colony over any

and Technological Development 470571/2013-6 and 306166/2014-3 (http://cnpq.br/) JPP. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

longer period of time. Manually classifying calls require expert knowledge and daily hours of work, making it too time-consuming and prone to inconsistencies among researchers. Thus, a reliable automatic method for call identification is necessary.

Much work has been done on the automatic classification of animal vocalizations, especially bird song [15–19], but also mammalian [20–22] and amphibian calls [23–25]. However, only a few studies have addressed non-human primates (hereafter primates). Pertinent to the current study, there are only three other studies in which different primate vocalizations were analyzed. Mielke and Zuberbühler (2013) used artificial neural networks (ANNs) with Mel-Frequency Cepstral Coefficients features to classify blue monkey call types (*Cercopithecus mitis stuhlmanni*). Furthermore, they predicted caller identity from the alarm call, and identified the blue monkey alarm call among the alarm calls of other sympatric species [26].

Pozzi, Gamba and Giacoma (2010) also used ANNs, but with hand-designed features derived from fundamental frequency and formants to classify call type among seven distinct types made by the black lemur (*Eulemur macaco*) [27]. In a later study, the same group used the long grunt (a vocalization included in the repertoire of all lemurs) and similar analytical methods to classify species among the five species in the *Eulemur* genus [28]. Therefore, to our knowledge, there is no previous work on classifying vocalization types from Neotropical primates.

In contrast, several studies have explored the related question of how specific to the caller are the acoustic properties of individual vocalizations, that is, how well caller identity can be predicted from a given call. These studies have all relied on manually designed features and linear discriminant analysis (LDA). Particularly relevant are Jones et al. (1993) and Miller et al. (2010), who both studied the common marmoset's Phee call to classify caller individual [29, 30]. Both studies found that the call is highly caller-specific. Similar studies have been done on Japanese macaque (*Macaca fuscata*) [31], blue monkey [32], ring-tailed lemur (*Lemur catta*) [33], and cotton-top tamarin (*Saguinus oedipus*) [34].

Algorithms for the automatic classification of vocalizations learn the mapping from input (call features) to label (call type). Therefore, a dataset with labeled calls is necessary to train the algorithms. In general, classification performance increases with the amount of training data. However, in ethology, large sets of labeled data are often hard to obtain. The amount of data may be limited because data collection is labor-intensive, or the data of interest are inherently scarce because they are produced by animals passing through transitory learning or developmental stages. In the latter case, the amount of possible data is strictly limited. Thus, a method that achieves high accuracy with a relatively small number of labeled call exemplars is highly desirable.

To address the need for automatic call classification given the aforementioned constraints, we compared seven different types of classification algorithms with the goal of finding a reliable method.

To extract acoustic features, we used Linear Predictive Coding (LPC), a method commonly used for speech processing [35]. For classification purposes, we used seven different algorithms: (i) Optimum Path Forest (OPF), (ii) Bayesian Classifier, (ii) Multilayer Artificial Neural Network (MLP), (iv) Support Vector Machines (SVM), (v) k-Nearest Neighbors (k-NN), (vi) Logistic regression, and (vii) AdaBoost.

Materials and Methods

Dataset description

The subjects were five captive-born adult common marmosets (two females and three males), housed at the Instituto do Cérebro, Universidade Federal do Rio Grande do Norte. The marmosets where housed socially in two wire mesh enclosures (1.20 x 1.50 x 2.45 m), enriched with tree branches, ropes, plants, hammocks and nesting tubes. The animals were fed twice daily with fresh and dried fruit, nuts, egg and chicken, and had *ad libitum* access to water. The colony was maintained outdoors protected by a roof allowing daily sunbaths in natural light. The animals were housed in compliance with SISBIO permit 18394, and the experiment was approved by the ethics committee of Universidade Federal do Rio Grande do Norte with CEUA permit 11/2016. No animal was sacrificed at the end of the experiment.

A directional microphone (ECM-CG50 Pro Shotgun Microphone, Sony, Tokyo, Japan) was placed at a distance of 10 cm above the home cage and connected to a computer. The microphone signal was streamed to a computer where a custom-written Python script segmented the incoming signal. The signal was bandpass filtered between 4 and 10 kHz, and when the amplitude exceeded a threshold a segment beginning 0.5 s before first threshold crossing and ending 0.5 s after the last threshold crossing was saved. Sound was sampled at 44.1 kHz.

From the raw recordings, we selected and manually labeled 27–30 exemplars per class of marmoset vocalization. We attempted to cover the marmoset's vocal repertoire of approximately nine [36] to 13 [37] distinct vocalizations. The number of exemplars of each of the 11 types investigated in this work is listed in Table 1, and spectrograms of representative exemplars of each type are shown in Fig 1. All vocalizations were produced spontaneously, that is, without any intervention from the experimenter. The recordings were done over a period of two months. The dataset is available at https://osf.io/yqpvk/ and http://neuro.ufrn.br/data/marmosetvocalizations.

Feature extraction

The success of a signal classification system depends on the choice of features used to characterize the raw signals. In this work, we used LPC, a method commonly used for analysis and compression of speech and animal vocalizations [35, 38]. The linear prediction filter coefficients were used as input features to the classification algorithms. Those are the coefficients of an n^{th}-order linear finite impulse response filter that predicts the current value of the vocalization from past samples [39]. The number of features extracted from each call was set to 20 after experimenting with filter orders from 10 to 25, in steps of 5.

Classification algorithms

Optimum-Path Forest. The OPF classifier models the problem of pattern recognition as a graph partition task, in which a predefined set of samples from each class (i.e. *prototypes*)

Table 1. The number of calls considering each class.

Vocalization type	Exemplars per class
Alarm	30
Chirp	30
Loud-shrill	27
Phee-2	30
Phee-3	30
Phee-4	30
Seep	30
Trill	27
Tsik	27
Tsik-ek	30
Twitter	30

doi:10.1371/journal.pone.0163041.t001

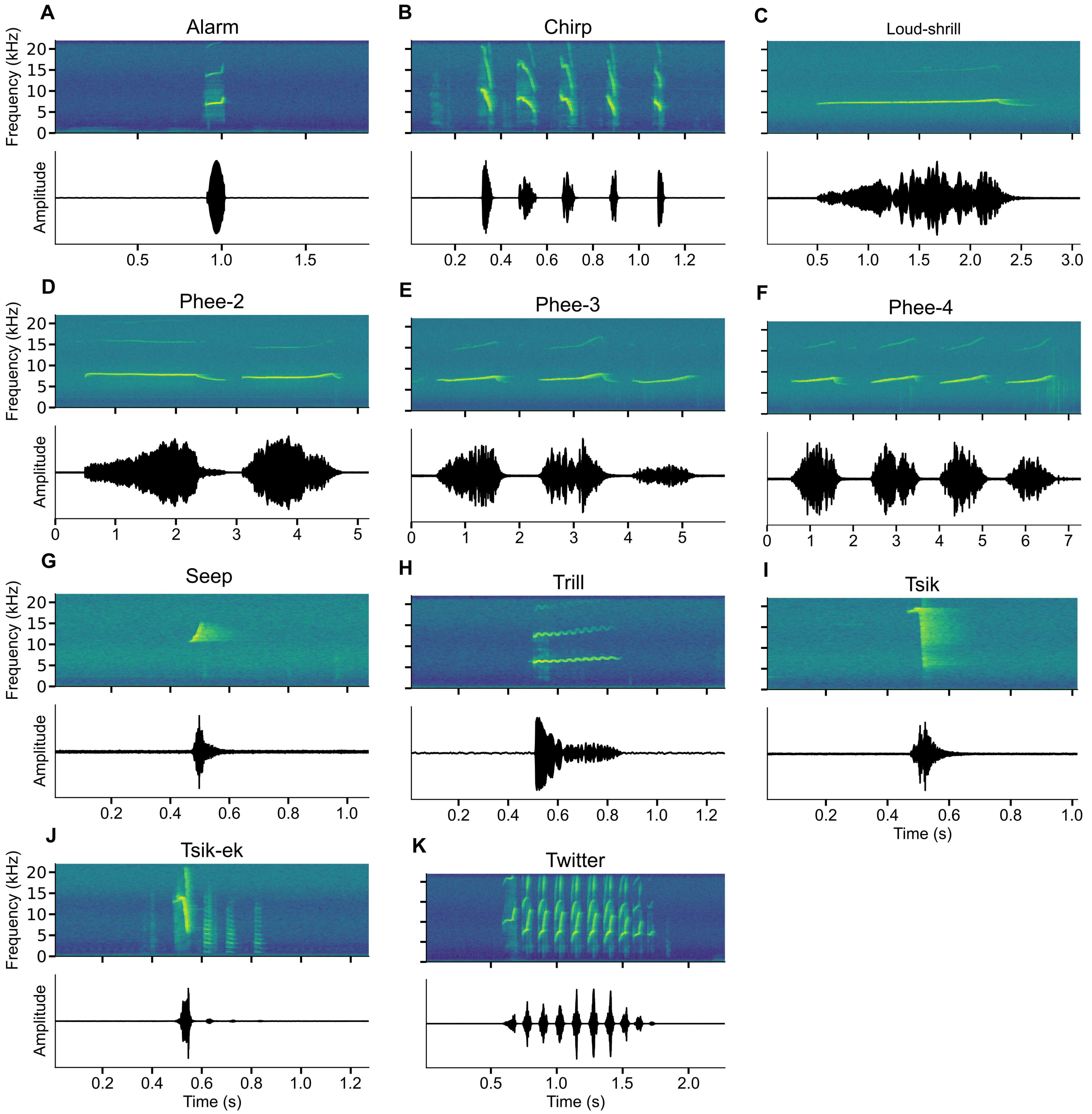

Fig 1. Vocalization exemplars. Amplitude and time-frequency spectrograms are shown for representative exemplars of the marmoset call types considered in this study. A: Alarm, B: Chirp, C: Loud shrill, D: Phee-2, E: Phee-3, F: Phee-4, G: Seep, H: Trill, I: Tsik, J: Tsik-Ek, K: Twitter.

doi:10.1371/journal.pone.0163041.g001

compete for minimal path cost to the rest of the samples. This results in a collection of optimum-path trees rooted at the prototype nodes, building an optimum-path forest considering from all training samples. Test samples are classified through incrementally evaluating the optimum paths from the prototypes, as though they were part of the forest, and assigning the labels of the most strongly connected roots. The notion of optimum-path connectivity comes from the minimization of a path-cost function [40]. An OPF classifier can be designed as long as we use a smooth path-cost function. Although there are two different versions of the supervised OPF classifier [41–43], in this paper we make use of the former and most widely used approach, as described below.

The OPF with complete graph was first proposed by Papa et al. [41]. Later on, Papa et al. [42] presented an improved version with a more efficient classification step. In this section, we present the OPF algorithm as described by those authors, using the same formalism.

Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ be a dataset partitioned into a training ($\mathcal{D}_1$) and a test ($\mathcal{D}_2$) set. In addition, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ be the graph originated from $\mathcal{D}_1$, such that $\mathcal{V} = \mathcal{D}_1$ and $\mathcal{E}$ stands for an adjacency relation that defines a full connectedness graph, that is, a graph where each pair of nodes is connected to each other. The arcs are weighted by the distance between their corresponding nodes. Each graph node $\mathbf{s} \in \mathcal{V}$ is modeled as an n-dimensional feature vector, and w $(\mathbf{s},\mathbf{t})$ is a weight (distance) between two graph nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ used to weight the arc $\langle \mathbf{s}, \mathbf{t} \rangle \in \mathcal{E}$. Mathematically, w is a function that takes two graph nodes and returns the distance between them, that is, $w \rightarrow \mathcal{V} \times \mathcal{V} : \Re^+$. In this work, a node is the set of n features extracted from a particular marmoset vocalization.

As aforementioned, the training step of the OPF classifier aims at building an optimum-path forest rooted in a set of prototype samples. Let $\mathcal{S}$ be that set, such that $\mathcal{S} \subseteq \mathcal{V}$. A very common and easy way to obtain $\mathcal{S}$ would be though a random sampling over the training samples. However, two requirements need to be met: (i) each class should be represented by, at least, one prototype node, and (ii) the prototype's distribution should cover different regions of the feature space. These requirements make the naïve approach too time-consuming. In order to circumvent this problem, Papa et al. [41] proposed to place prototypes in the regions more prone to errors, that is, nearby the frontier of the classes. The idea is to compute a Minimum Spanning Tree (MST) over the training set, and then mark the connected samples from different classes as prototypes. We say that $\mathcal{S}^*$ is an optimum set of prototypes when the training step minimizes the classification errors in $\mathcal{D}_1$ [44]. The optimum prototypes are the closest samples of the MST with different labels in $\mathcal{D}_1$.

Given the prototypes, the next step is to find the smallest path-cost from the prototypes to the remaining training samples in $\mathcal{S}^*$ to the remaining nodes in $\mathcal{V}/\mathcal{S}^*$. In this work, we adopted the same path-cost function as Papa et al. [41, 42], which computes the maximum arc-weight along a path, as follows:

$$\begin{aligned} f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in \mathcal{S}^*, \\ +\infty & \text{otherwise.} \end{cases} \\ f_{max}(\pi \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi), d(s,t)\}, \end{aligned} \tag{1}$$

where $\pi \cdot \langle s, t \rangle$ stands for the concatenation of path π and the arc $\langle s, t \rangle$. A path is defined as a sequence of distinct and adjacent samples.

After computing the optimum-path forest, unseen samples from $\mathcal{D}_2$ can be classified. For each test sample, the classification step consists of connecting the sample to all training nodes, evaluating which training sample offers the optimum-path cost C according to f_{max} defined in

Eq 1, that is:

$$C(t) = \min_{\forall s \in \mathcal{D}_1} \{\max\{C(s), d(s,t)\}\}. \tag{2}$$

Let s^* be the training sample that satisfies the above equation. Then, test sample t will be assigned the same label as sample s^*. For the current experiments we used the LibOPF library available at https://github.com/LibOPF/LibOPF.

Bayesian Classifier. A Bayesian Classifier estimates the probability that a given vocalization belongs to a certain class. This probability can be derived from Bayes' Theorem [45]:

$$p(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_i)P(\omega_i)}{p(\mathbf{x})}, \tag{3}$$

where $p(\mathbf{x}|\omega_i)$ denotes the probability of observing feature vector $\mathbf{x}$ given the class ω_i, $P(\omega_i)$ is the prior probability of class ω_i, and $p(\mathbf{x})$ is the probability of $\mathbf{x}$. In order to estimate $p(\mathbf{x}|\omega_i)$ we assumed that the likelihood function is Gaussian, and could thus estimate its parameters from the dataset [46].

Multilayer Artificial Neural Network. An MLP classifier is a feedforward neural network composed of several neuron layers aiming to solve multiclass problems [47]. The input to each layer is a weighted sum of the output from the previous layer. The number of neurons in the first layer equals the number of features of the input, while the number of neurons in the last layer is equal to the number of classes. The neural network assigns a feature vector extracted from a vocalization $\mathbf{x}$ to the class ω_q if the q-th output neuron has the highest activation. We used the MLP implementation from scikit-learn [48], with two hidden layers of eight and 16 neurons. The network was trained using backpropagation and the Limited-memory BFGS optimization algorithm [49] to update the weights. The learning rate was set to 0.001.

Support Vector Machines. While the learning of MLP is based on the principle of empirical risk minimization, the SVM induction process is rooted in the principle of structural risk minimization [50–52], aiming at establishing an optimal discriminative function between two classes of patterns while accomplishing the trade-off between generalization and overfitting. The SVM training algorithm constructs the optimal hyperplane separating the two classes [50]. In order to extend from linear to nonlinear classification the *kernel trick* is used [51], where kernel functions nonlinearly map input data into high-dimensional feature spaces in a computationally-efficient manner.

For classification problems with multiple classes, two approaches are commonly used for binary SVMs, *one-against-one* and *one-against-all* [53]. Both strategies tend to lead to similar results in terms of classification accuracy, but the former, which was the one adopted here usually requires shorter training time, although incurring a higher number of binary decompositions.

For the current experiments, we used the LibSVM library [54] (available at http://www.csie.ntu.edu.tw/$\sim$cjlin/libsvm). The hyperparameters C and σ of the SVM classifier were determined via a 5-fold cross-validation grid-search in the ranges $[2^{-5}, 2^{15}]$ and $[2^{-15}, 2^{3}]$, changing the exponent in steps of two.

k-Nearest Neighbors. k-NN is a very simple algorithm that works well in many different applications. In contrast to the OPF, the k-NN uses all training samples as prototypes.

The k-NN requires an input parameter k setting the number of neighbors that contribute to the classification of a sample [55, 56]. In order to classify a test sample t, the majority label in a region (e.g. sphere or hypercube) containing k training samples and centered at t, determines t's label. Note that for $k = 1$, the testing sample t is classified as the class of the closest training sample. For the current experiments, we defined the value of k as the best value of a grid-search in the range $\left[1, \lfloor \frac{m}{5} \rfloor\right]$ in steps of two; where m is the number of training samples.

Logistic regression. The logistic regression classifier is essentially a one-layer artificial neural network where the weighted input features are feed to the logistic function. Here, we trained the classifier under a one-against-all scheme, regularized by the L2 norm of the classifier weights. We used the LIBLINEAR [57] implementation of logistic regression.

AdaBoost. AdaBoost, short for Adaptive Boosting, is a meta-classifier that trains an ensemble of weak classifiers. It iteratively trains classifiers on the same dataset while adjusting the weights of incorrectly classified samples such that subsequent classifiers focus more on those difficult samples. The resulting ensemble classifier tends to be less susceptible to over-fitting than other classifiers, but it is also known to be sensitive to noisy data and outliers [58]. Here, we used the AdaBoost-SAMME.R algorithm [59] from the from scikit-learn package [48].

Statistical evaluation metrics

In regard to the recognition rate, we used an accuracy measure proposed by Papa et al. [41], which is similar to the Kappa index [60], but more restrictive. If, for example, there are two classes of vocalizations with very different sizes and a classifier always assigns the label of the largest class, the average number of correct assignments will be deceivingly high. A better accuracy measure should take into account the high error rate of the smallest class. The accuracy used here is measured by taking into account that classes may have different sizes in $\mathcal{D}_2$. Let us define:

$$e_{i,1} = \frac{FP_i}{|\mathcal{D}_2| - |\mathcal{D}_2^i|} \tag{4}$$

and

$$e_{i,2} = \frac{FN_i}{|\mathcal{D}_2^i|}, \; i = 1, 2, \ldots, K, \tag{5}$$

where K stands for the number of classes, $|\mathcal{D}_2^i|$ concerns the number of samples in $\mathcal{D}_2$ that come from vocalization class i, and FP_i and FN_i stand for the numbers of false positives and false negatives for class i, respectively. That is, FP_i is the number of samples from other vocalization classes that were classified as being from the class i in $\mathcal{D}_2$, and FN_i is the number of samples from the class i that were incorrectly classified as being from other classes in $\mathcal{D}_2$. The error terms $e_{i,1}$ and $e_{i,2}$ are then used to define the total error from class i:

$$E_i = e_{i,1} + e_{i,2}. \tag{6}$$

Finally, the accuracy Acc is then defined as follows:

$$Acc = 1 - \frac{\sum_{i=1}^{K} E_i}{2K}. \tag{7}$$

Sensitivity (Se), often called recall, is the ratio of the number of correctly classified vocalizations from a given class and the total number of vocalizations in that class (including misclassified vocalizations):

$$Se = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{8}$$

where TP and FN are the number of vocalizations from a given class that were correctly or incorrectly classified, respectively.

Positive predictive value (*PPV*), often called precision, is the ratio between the correctly classified vocalizations from a given class and the total number of vocalizations classified as pertaining to that class:

$$PPV = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \tag{9}$$

where FP denotes the number of vocalizations incorrectly classified as belonging to the considered class.

Finally, as a more global performance metric, we calculated the averaged F_1-score for all eight classes. The F_1-score for a given class is calculated as the harmonic mean of the *Se* and *PPV* values for that class:

$$F_1 - score = 2\left(\frac{Se \times PPV}{Se + PPV}\right). \tag{10}$$

These four metrics allow us to reliably evaluate the performance of the classification algorithms considered in this work. The performance metrics are reported as the averages over 100 repetitions of classifier training and testing. All training and testing of the classification algorithms was done on a computer with an Intel i7 5500U processor with 8GB of RAM using Linux as operational system. A Python script to reproduce the results is available at https://github.com/kalleknast/call_class.

Results

In order to find a robust method for the automatic classification of marmoset vocalizations, we compared the classification performance of seven different algorithms. In addition, we further explored different configurations of both OPF and SVM. OPF was tested using the following distance metrics: Euclidean, Manhattan, Canberra, Chi-Square and Bray-Curtis, and SVM was tested with linear, radial basis function and polynomial kernels. The goal was to find an algorithm that could be successfully trained on small sets of primate vocalization data. For this reason, we split our original dataset into training sets of increasing sizes, ranging from 10% to 90% of the original dataset. We found good performance of all algorithms except AdaBoost, Naive Bayes, SVM when using linear and polynomial kernels, and OPF when using Chi-Square, Bray-Curtis and Canberra distance metrics (see Table 2 and Fig 2). These poorly performing algorithms were excluded from further consideration. SVM, k-NN and OPF using Euclidean and Manhattan distances performed similarly, and well above chance (accuracy $\approx$ 0.5 and F_1-score $\approx$ 0.5) using as little as 10% of the original dataset, and reaching an accuracy around 0.8 when trained on 90% of the data. However, in spite of the good performance, Fig 2 shows that classification accuracy keeps improving with training set size, suggesting that adding even more training data would further improve performance.

The OPF algorithm configured with the Euclidean distance metric was selected for further analysis since it yielded better or comparable classification performance on both the smallest and the largest datasets (top accuracy $\approx$ 0.83 and F_1-score $\approx$ 0.84). It is parameter free and thus, easy to train, and computation time is an order of magnitude less than for SVM and k-NN (see Table 2). Table 3 presents the results as a confusion matrix.

We opted for a hierarchical strategy to investigate how well all 11 types of vocalization in the dataset could be classified. Such approach was performed through sorting the Phee and Tsik classes into three and two sub-classes, respectively. The Phee class was sorted into Phee-2, Phee-3 and Phee-4 depending on how many separate whistles the call contained (Fig 1(d), 1(e) and 1(f), respectively). The Tsik call was sorted into the sub-classes Tsik and Tsik-ek depending

Table 2. Classification of all eight classes of vocalizations using different algorithms. 90% of the samples were used for the training set. Time refers to the time required to classify one sample in milliseconds.

Method	F_1-score		Accuracy		Time (ms)
	Mean	SEM	Mean	SEM	Mean
MLP	0.757	0.010	0.744	0.011	145.6
OPF–Chi-Square	0.257	0.009	0.197	0.011	24.4
OPF–Bray Curtis	0.466	0.010	0.421	0.011	6.9
OPF–Canberra	0.622	0.010	0.595	0.011	24.5
OPF–Euclidean	0.840	0.007	0.832	0.008	7.7
OPF–Manhattan	0.818	0.008	0.805	0.009	10.7
k-NN	0.842	0.007	0.833	0.008	465.1
SVM	0.852	0.008	0.843	0.009	115.2
Naive Bayes	0.507	0.010	0.475	0.011	252.0
AdaBoost	0.603	0.011	0.571	0.013	10173.1
Logistic regression	0.744	0.010	0.741	0.011	89.6

doi:10.1371/journal.pone.0163041.t002

on the presence of the harmonic "ek" component following the "tsik" in the Tsik-ek calls [37] (Fig 1(i) and 1(j), respectively).

Table 4 presents a confusion matrix of the results from the Phee calls using OPF with Euclidean distance metric. Overall, the classification accuracy (56%) was not quite as good as when classifying the eight principal classes. Phee-2 was correctly classified in 70% of the cases, and Phee-3 was classified at 46%, where the remaining 64% were misclassified as Phee-2 15% or Phee-3 39%. Finally, Phee-4 was accurately classified in 52% of the samples, where the

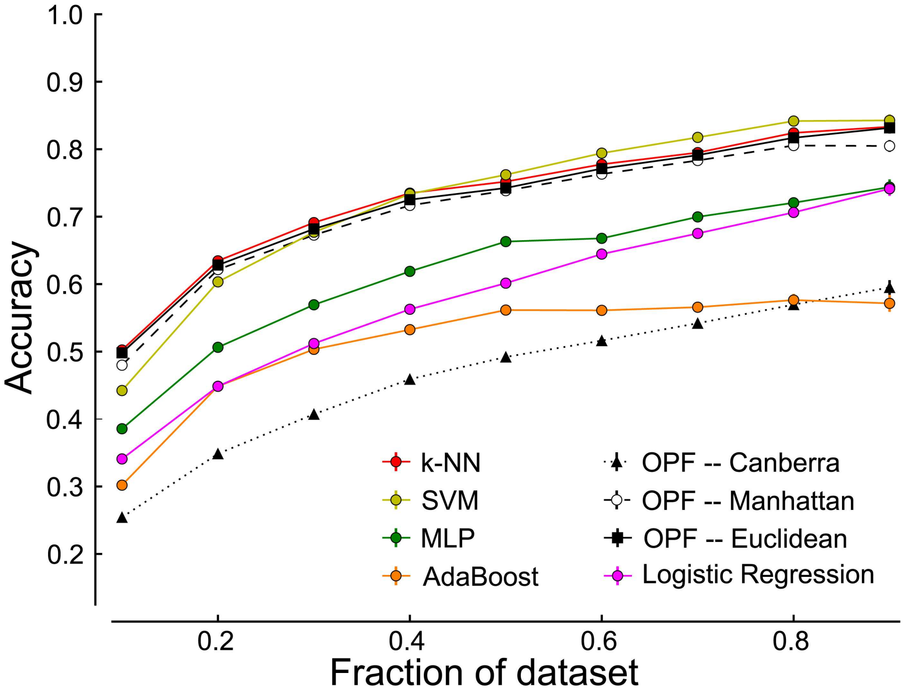

Fig 2. The effect of training set size on classification performance. For the sake of visual clarity, the results of OPF using the distance metrics Bray-Curtis and Chi-Square, and SVM using linear and polynomial kernels are excluded.

doi:10.1371/journal.pone.0163041.g002

Table 3. Confusion matrix considering the classification of all eight classes of vocalizations using OPF with Manhattan distance and 90% of the samples for training set.

		True class[%]							
		Alarm	**Chirp**	**Loud-shrill**	**Phee**	**Seep**	**Trill**	**Tsik**	**Twitter**
Classified as[%]	**Alarm**	78.4	0.0	4.7	4.1	4.3	0.7	2.5	0.0
	Chirp	3.9	89.5	0.0	1.6	0.0	0.0	4.4	7.0
	Loud-shrill	5.9	0.0	75.3	5.9	0.0	19.2	0.0	0.4
	Phee	4.3	0.1	4.3	83.0	0.0	0.9	0.0	0.5
	Seep	5.7	0.0	0.0	0.0	90.6	0.0	8.3	0.0
	Trill	1.6	0.0	15.7	0.0	0.0	74.0	0.0	0.0
	Tsik	0.2	1.0	0.0	0.0	5.1	0.0	77.5	0.0
	Twitter	0.0	9.4	0.0	5.4	0.0	5.2	7.3	92.1

doi:10.1371/journal.pone.0163041.t003

Table 4. Confusion matrix considering the classification of the principal Phee class into sub-classes using OPF with Euclidean distance metric and 90% of the samples for training set.

		True class[%]		
		Phee-2	**Phee-3**	**Phee-4**
Classified as[%]	**Phee-2**	70.4	19.8	9.7
	Phee-3	14.8	46.2	39.1
	Phee-4	14.8	34.0	51.2

doi:10.1371/journal.pone.0163041.t004

remaining 48% were misclassified as Phee-3 34% or Phee-2 15%. Classification of the Phee sub-class from the eight principal classes resulted in a compounded accuracy of 58% for Phee-2, 38% for Phee-3, and 42% for Phee-4, since accuracy for the principal Phee class was 83% (see Tables 3 and 4).

Table 5 presents a confusion matrix of the results using OPF with Euclidean distance metric for Tsik vocalizations. The overall accuracy was 83%, where the accuracy for sub-class Tsik was 91%, and that of Tsik-ek was 76%. The compounded accuracies for the sub-classes Tsik and Tsik-ek were 70% and 58%, respectively (see Tables 3 and 5).

Discussion

In this work, we presented methods for the automatic classification of commonly occurring vocalizations of the common marmoset. The provided dataset can be used for acoustic analysis, further algorithm development and playback experiments. The method presented should enable the online monitoring of vocal activity in colonies of captive marmosets, so as to provide valuable information about the colony's health and well-being. Further, the method allows for interactive experimental designs, in which different actions can be triggered depending on the vocal behavior of the subjects.

Table 5. Confusion matrix for the classification of the principal Tsik class into sub-classes using OPF with Manhattan distance metric and 90% of the samples for training set.

		True class[%]	
		Tsik	**Tsik-ek**
Classified as[%]	**Tsik**	90.5	24.5
	Tsik-ek	9.5	75.5

doi:10.1371/journal.pone.0163041.t005

Since recording and manually labeling data is both labor-intensive and time-consuming, the most important factor when selecting a classification algorithm is how well it performs on a small amount of data. Fig 2 demonstrates clear advantages for the k-NN, SVM and OPF (using Euclidean and Manhattan distances) algorithms. These algorithms perform best on both the least and the greatest amount of data.

Another factor to consider is how easy an algorithm is to use. Most algorithms have hyperparameters that require careful optimization for good performance. The standard way of doing this is to repeatedly re-train the algorithm on a manually specified range of hyperparameter values, each time evaluating classification performance on data that were not included among the training data. Both the k-NN and SVM algorithms require such hyperparameter optimization, whereas the OPF algorithm is parameterless, making it easier to use.

Under some circumstances, the time required for classification can become important. Algorithms that do not require much computational resources are valuable when real-time classification is necessary, and especially when the computations are performed on small single board computers or embedded devices with limited capacity. Examples of this are on-site portable audio acquisition and analysis [61], or home-cage vocal conditioning systems [62]. The OPF algorithm requires an order of magnitude less time than comparably performing k-NN and SVM algorithms and is thus suitable for these goals.

The proposed methods are mainly suited for laboratory conditions where audio can be recorded under consistent conditions, and with high signal-to-noise ratio. Such conditions are unlikely to be available under field conditions. Robust classification within field conditions would probable require much improved pre-processing before the classification step.

Acknowledgments

We thank Daniel Y. Takahashi for comments on the manuscript, and Thamiris Botelho, William Wood, Josy Pontes, Janaina Figueiredo, Bruno Soares, Maria Bernardete de Souza and the Primatology Center of UFRN for help with various aspects of housing & specialized marmoset care. Research equipment and HKT's stipend were funded by CNPq/BJT grant # 402422/2012-0 to SR and HKT. The last author thanks CNPq for providing financial support through a grants # 470501/2013-8 and # 301928/2014-2. The second author thanks the FAPESP Research, Innovation and Dissemination Center for Neuromathematics (grant # 2013/07699-0, S. Paulo Research Foundation). The fourth author thanks FAPESP grants # 2014/16250-9 and # 2015/50319-9, as well as CNPq grants # 470571/2013-6 and # 306166/2014-3.

Author Contributions

Conceptualization: VHCA HKT.

Data curation: HKT.

Formal analysis: HKT DRP.

Funding acquisition: HKT VHCA SR.

Investigation: HKT.

Methodology: VHCA JPP HKT DRP.

Project administration: VHCA SR.

Resources: HKT SR.

Software: JPP HKT DRP.

Supervision: VHCA SR.

Validation: HKT.

Visualization: HKT.

Writing – original draft: HKT.

Writing – review & editing: HKT SR JPP.

References

1. Solomon SG, Rosa MG. A simpler primate brain: the visual system of the marmoset monkey, Front Neural Circuits. 2014; 8: 1–24. doi: 10.3389/fncir.2014.00096
2. Okano H, Miyawaki A, Kasai K. Brain/MINDS: Brain-mapping project in Japan, Philos Trans R Soc Lond B Biol Sci. 2015; 370: 1–9. doi: 10.1098/rstb.2014.0310
3. Kishi N, Sato K, Sasaki E, Okano H. Common marmoset as a new model animal for neuroscience research and genome editing technology. Dev Growth Differ. 2014; 56: 53–62. doi: 10.1111/dgd.12109 PMID: 24387631
4. Hart BA, van Kooyk Y, Geurts JJ, Gran B. The primate autoimmune encephalomyelitis model; a bridge between mouse and man. Ann Clin Transl Neurol. 2015; 2: 581–593. doi: 10.1002/acn3.194 PMID: 26000330
5. Horvat B, Berges BK, Lusso P. Recent developments in animal models for human herpesvirus 6A and 6B. Curr Opin Virol. 2014; 9: 97–103. doi: 10.1016/j.coviro.2014.09.012 PMID: 25462440
6. Scanga CA, Flynn JL. Modeling tuberculosis in nonhuman primates. Cold Spring Harb Perspect Med. 2014; 4: a018564. doi: 10.1101/cshperspect.a018564 PMID: 25213189
7. Okano H, Hikishima K, Iriki A, Sasaki E. The common marmoset as a novel animal model system for biomedical and neuroscience research applications. Semin Fetal Neonatal Med. 2012; 17: 336–340. doi: 10.1016/j.siny.2012.07.002 PMID: 22871417
8. Wang X. On cortical coding of vocal communication sounds in primates. Proc Natl Acad Sci U S A. 2000; 97: 11843–11849. doi: 10.1073/pnas.97.22.11843 PMID: 11050218
9. Snowdon CT. Social processes in the evolution of complex cognition and communication, In: Oller D, Griebel U, editors. Evolution of Communication Systems: A Comparative Approach, The Vienna series in theoretical biology. Cambridge: MIT Press; 2004. pp. 131–150.
10. Jones CB. Quantitative analysis of marmoset vocal communication. In: Pryce C., Scott L, Schnell C, editors. Marmosets and tamarins in biological and biomedical research: proceedings of a workshop. Salisbury: DSSD Imagery; 1997. pp. 145–151.
11. Bezerra BM. Vocalização do sagüi comum: influências sociais e ontogênicas em ambiente natural. Ph.D. Thesis, Universidade Federal de Pernambuco. 2006. Available: http://repositorio.ufpe.br/handle/123456789/666
12. Stevenson MF, Rylands AB. The marmosets, genus *callithrix*. In: Mittermeier RA, Rylands AB, Coimbra-Filho AF, da Fonseca GAB, editors. Ecology and Behavior of Neotropical Primates, Vol. 2. Washington, DC: World Wildlife Fund; 1988. pp. 131–222.
13. Winter M. Some aspects of the ontogeny of vocalizations of hand-reared common marmosets. In: Rothe H, Wolters HJ, Hearn JP, editors. Hearn Biology and behaviour of marmosets: Proceedings of the Marmoset Workshop. Gottingen: H. Rothe, Eigenverlag Hartmut; 1978. pp. 127–139.
14. Bezerra BM, Souto AS, Oliveira MAB, Halsey LG. Vocalisations of wild common marmosets are influenced by diurnal and ontogenetic factors. Primates. 2009; 50: 231–237. doi: 10.1007/s10329-009-0132-7 PMID: 19224328
15. Stowell D, Plumbley MD. Automatic large-scale classification of bird sounds is strongly improved by unsupervised feature learning. PeerJ. 2014; 2: e488. doi: 10.7717/peerj.488 PMID: 25083350
16. McIlraith A, Card H. Birdsong recognition using backpropagation and multivariate statistics. IEEE Trans Signal Process. 1997; 45: 2740–2748. doi: 10.1109/78.650100
17. Lakshminarayanan B, Raich R, Fern X. A syllable-level probabilistic framework for bird species identification. In: Machine Learning and Applications (ICMLA), International Conference on, IEEE; 2009. pp. 53–59.
18. Damoulas T, Henry S, Farnsworth A, Lanzone M, Gomes C. Bayesian classification of flight calls with a novel dynamic time warping kernel. In: Machine Learning and Applications (ICMLA), Ninth International Conference on, IEEE. 2010; pp. 424–429.

19. Briggs F, Lakshminarayanan B, Neal L, Fern XZ, Raich R, Hadley SJ, et al. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. J Acoust Soc Am. 2012; 131: 4640–4650. doi: 10.1121/1.4707424 PMID: 22712937

20. Molnár C, Kaplan F, Roy P, Pachet F, Pongrácz P, Dóka A, et al. Classification of dog barks: A machine learning approach. Anim Cogn. 2008; 11: 389–400. doi: 10.1007/s10071-007-0129-9 PMID: 18197442

21. Jarvis S, DiMarzio N, Morrissey R, Moretti D. A novel multi-class support vector machine classifier for automated classification of beaked whales and other small odontocetes. Can Acous. 2008; 36: 34–40.

22. Weninger F, Schuller B. Audio recognition in the wild: Static and dynamic classification on a real-world database of animal vocalizations. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2011. pp. 337–340.

23. Acevedo MA, Corrada-Bravo CJ, Corrada-Bravo H, Villanueva-Rivera LJ, Aide TM. Automated classification of bird and amphibian calls using machine learning: A comparison of methods. Ecol Inform. 2009; 4: 206–214. doi: 10.1016/j.ecoinf.2009.06.005

24. Huang CJ, Yang YJ, Yang DX, Chen YJ. Frog classification using machine learning techniques. Expert Syst Appl. 2009; 36: 3737–3743. doi: 10.1016/j.eswa.2008.02.059

25. Dayou J, Han NC, Mun HC, Ahmad AH, Muniandy SV, Dalimin MN. Classification and identification of frog sound based on entropy approach. In: 2011 International Conference on Life Science and Technology, Vol 3; 2011. pp. 184–187.

26. Mielke A, Zuberbühler K. A method for automated individual, species and call type recognition in free-ranging animals. Anim Behav. 2013; 86: 475–482. doi: 10.1016/j.anbehav.2013.04.017

27. Pozzi L, Gamba M, Giacoma C. The use of artificial neural networks to classify primate vocalizations: A pilot study on black lemurs. Am J Primatol. 2010; 72: 337–348. PMID: 20034021

28. Pozzi L, Gamba M, Giacoma C. Artificial neural networks: A new tool for studying lemur vocal communication. In: Leaping Ahead. New York: Springer; 2013. pp. 305–313.

29. Jones BS, Harris DHR, Catchpole CK. The stability of the vocal signature in phee calls of the common marmoset, *callithrix jacchus*. Am J Primatol. 1993; 31: 67–75. doi: 10.1002/ajp.1350310107

30. Miller CT, Mandel K, Wang X. The communicative content of the common marmoset phee call during antiphonal calling. Am J Primatol. 2010; 72: 974–980. doi: 10.1002/ajp.20854 PMID: 20549761

31. Ceugniet M, Izumi A. Individual vocal differences of the coo call in japanese monkeys. C R Biol. 2004; 327: 149–157. doi: 10.1016/j.crvi.2003.11.008 PMID: 15060986

32. Butynski TM, Chapman CA, Chapman LJ, Weary DM. Use of male blue monkey "pyow" calls for long-term individual identification. Am J Primatol. 1992; 28: 183–189. doi: 10.1002/ajp.1350280303

33. Macedonia JM. Individuality in a contact call of the ringtailed lemur (*lemur catta*). Am J Primatol. 1986; 11: 163–179. doi: 10.1002/ajp.1350110208

34. Snowdon CT, Cleveland J, French JA. Responses to context- and individual-specific cues in cotton-top tamarin long calls. Anim Behav. 1983; 31: 92–101. doi: 10.1016/S0003-3472(83)80177-8

35. Bradbury J. Linear predictive coding. 2000. Available: http://my.fit.edu/~vkepuska/ece5525/lpc_paper.pdf

36. Epple G. Comparative studies on vocalization in marmoset monkeys (*hapalidae*). Folia Primatol. 1968; 8: 1–40. doi: 10.1159/000155129 PMID: 4966050

37. Bezerra BM, Souto A. Structure and usage of the vocal repertoire of *callithrix jacchus*. Int J Primatol. 2008; 29: 671–701. doi: 10.1007/s10764-008-9250-0

38. Deng L, O'Shaughnessy D. Speech processing: A dynamic and optimization-oriented approach. Boca Raton: CRC Press; 2003.

39. Jackson LB. Digital filters and signal processing. New York: Springer; 1996.

40. Falcão AX, Stolfi J, Lotufo RA. The image foresting transform: Theory, algorithms, and applications. IEEE Trans Pattern Anal Mach Intell. 2004; 26: 19–29. doi: 10.1109/TPAMI.2004.1261076 PMID: 15382683

41. Papa JP, Falcão AX, Suzuki CTN. Supervised pattern classification based on Optimum-Path Forest. Int J Imaging Syst Technol. 2009; 19: 120–131. doi: 10.1002/ima.20188

42. Papa JP, Albuquerque VHC, Falcão AX, Tavares JMRS. Efficient supervised Optimum-Path Forest classification for large datasets. Pattern Recognit. 2012; 45: 512–520. doi: 10.1016/j.patcog.2011.07.013

43. Papa JP, Falcão AX. A new variant of the optimum-path forest classifier. In: Bebis G, Boyle R, Parvin B, Remagnino P, Porikli F, Peters J, et al., editors. Advances in visual computing. Berlin: Springer-Verlag; 2008. 935–944.

44. Allène C, Audibert JY, Couprie M, Keriven R. Some links between extremum spanning forests, watersheds and min-cuts. Image Vis Comput. 2010; 28: 1460–1471. doi: 10.1016/j.imavis.2009.06.017

45. Jaynes ET. Probability theory: The logic of science. Camebridge: Cambridge University Press. 2003.

46. Duda RO, Hart PE, Stork DG. Pattern classification. 2nd ed. New York: Wiley-Interscience Publication. 2000.

47. Haykin S. Neural networks: A comprehensive foundation. 2nd ed. Prentice Hall. 1999.

48. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res. 2011; 12: 2825–2830.

49. Byrd RH, Lu P, Nocedal J. A limited memory algorithm for bound constrained optimization. SIAM J Sci Comput. 1995; 16: 1190–1208. doi: 10.1137/0916069

50. Vapnik VN. An overview of statistical learning theory. IEEE Trans Neural Netw. 1999; 10: 988–999. doi: 10.1109/72.788640 PMID: 18252602

51. Schölkopf B, Smola AJ. Learning with kernels. Cambridge: MIT Press. 2002.

52. Cortes C, Vapnik V. Support vector networks. Mach Learn. 1995; 20: 273–297. doi: 10.1023/A:1022627411411

53. Witten IH, Frank E. Data mining: Practical machine learning tools and techniques, 2nd ed. Amsterdam: Morgan Kaufmann Publishers. 2005.

54. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol. 2011; 2: 1–27. doi: 10.1145/1961189.1961199

55. Coomans D, Massart D. Alternative k-nearest neighbour rules in supervised pattern recognition. Anal Chim Acta. 1982; 136: 15–27. doi: 10.1016/S0003-2670(01)95359-0

56. Hall P, Park BU, Samworth RJ. Choice of neighbor order in nearest-neighbor classification. Ann Statist. 2008; 36: 2135–2152. doi: 10.1214/07-AOS537

57. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification. J Mach Learn Res. 2008; 9: 1871–1874.

58. Kanamori T, Takenouchi T, Eguchi S, Murata N. The most robust loss function for boosting, in Neural Information Processing. Berlin: Springer. 2004; 496–501.

59. Zhu J, Zou H, Rosset S, Hastie T. Multi-class AdaBoost. Stat Interface. 2009; 2: 349–360. doi: 10.4310/SII.2009.v2.n3.a8

60. Cohen J. A coefficient of agreement for nominal scales. Educ Psychol Meas. 1960; 20: 37–46. doi: 10.1177/001316446002000104

61. Aide TM, Corrada-Bravo C, Campos-Cerqueira M, Milan C, Vega G, Alvarez R. Real-time bioacoustics monitoring and automated species identification. PeerJ. 2013; 1: e103. doi: 10.7717/peerj.103 PMID: 23882441

62. Turesson HK, Ribeiro S. Can vocal conditioning trigger a semiotic ratchet in marmosets? Front Psychol. 2015; 6: 1–11. doi: 10.3389/fpsyg.2015.01519