Semi-supervised context-aware discovery of unknown audio concepts

Antonio Juarez[1], *Bhiksha Raj*[2], *Rita Singh*[2]

[1]Machine Learning Department, Carnegie Mellon University
[2]Language Technologies Institute, Carnegie Mellon University
{ajuarez,bhiksha,rsingh}@cs.cmu.edu

Abstract

Both defining new audio categories and annotating data that belong to these is a problem yet hardly tackled, much less resolved. These problems need to be solved, however, if we are to escalate the labeling of audio data from subjective manual annotation onto automatic audio discovery upon the vast amounts of audio data available today. The lack of work on this matter is understandable: audio data overlaps different semantic categories through a single channel, it is most often noisy, and any application that works on large datasets needs to deal with these problems. Additionally, relating semantic concepts to audio data is a problem in itself: how can a system associate newly-encountered acoustic data to concepts of which there is no data available? In this paper we describe how we used a labeled dataset to train reliable concept detectors for several semantic categories, how we augmented unlabeled data with contextual features through co-occurrence to and duration of known concepts, and results that indicate the feasibility of this task. We believe the design presented can serve as a general discovery framework for audio-like sequential data in general.

Index Terms: audio, detection, discovery, clustering

1. Introduction

The research field of general audio understanding (as opposed to speech-oriented applications) is maturing only recently, so it is difficult to get ahold of data with good training labels. The majority of annotations in available datasets are specific to a problem or domain, so they tend to not generalize nor easily merge with one another. In addition, the set of labels that have been the focus of past annotations is very small relative to the entire space of semantic audio categories. Notice this presents challenges beyond those encountered in traditional speech recognition, where the phoneme category space is finite and known, and a ground-truth language model reduces the space further by defining a subset of valid sound sequences.

While the problem of detection for specific domains have shown great progress[12][15][17], their results are solving real-world problems[10][11][13][14], and researchers have attempted to model and parse the complete audio domain[6][7][8][9][16], general audio understanding is still a naturally noisy and ill-defined problem — there exists no universal audio model that lists the possible categories to recognize, nor is there a concept of validity in general sound sequences, unlike those of a specific domain (i.e. speech [18][19]). Nevertheless, relationships between sounds exist in our world. Sounds correspond to local physical phenomena which necessarily affect each other, and many sounds are often indicative of environments in which only a subset of sound categories are likely to be present.

We think it is possible to apply the contextual information provided by an existing concept detector (for known audio concepts) onto the discovery of unknown semantic audio concepts. To the best of our knowledge, this has not yet been attempted by other researchers. If this task is feasible, we can learn to recognize new audio concepts, collect instances belonging to them, insert them back into the detector, and extend the set of concepts and contexts we are able to recognize. The vision of this project is a system that takes in an initial set of binary detectors and explores any amount of unlabeled audio data to automatically discover semantically meaningful audio concepts.

2. Modeling the problem

We take two main steps in our approach to Audio Discovery. We first design a general-purpose audio detector for which there is any positive training data available, and test its effectiveness on an isolated validation set. We aim to create detectors with high precision, as their results are later used as inputs to the audio discovery system. These detectors are implemented as a sequential graph search where the nodes are audio segments, with edges between consecutive segments. This algorithm scores audio segments with a classifier trained over all known concepts.

Our second step explores the discovery process of new audio concepts through clustering with different feature sets. We apply the concept detector trained in the previous step on a validation set, which yields a full segmentation into known concepts and unrecognized segments. The latter are clustered with a well-known algorithm, and we measure the purity of the clusters with respect to the true labels of their elements with an entropy score. We represent the unrecognized audio segments either with the acoustic feature set alone, or by appending two types of contextual features to them. We then compare the scores achieved by the different feature sets, and analyze the results to determine whether concept co-occurrence can present meaningful information to achieve a more correct clustering, and thus discover data for new audio concepts.

3. Dataset and Experiments

3.1. Dataset description

3.1.1. Data Source

The dataset used in this project is a subsample of 108 tracks of up to 32 seconds each from the BBC Sound Effects Library[1], which contains sounds from all around the world and in different contexts. Track categories in the set include "Africa, the Natural World", "Schools and Crowds", and "Livestock". These 108 tracks were segmented into short clips of 2 second durations.

All together we collected 5911 short audio segments, which

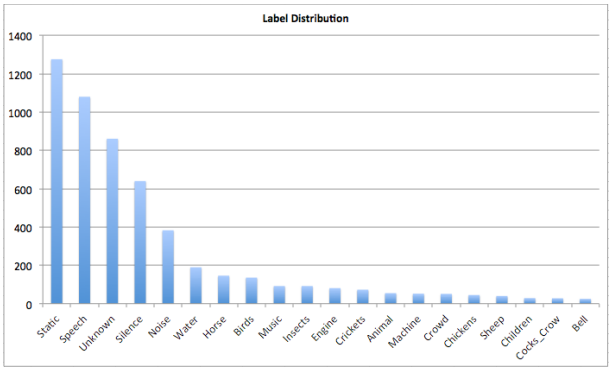

Figure 1: *Label distribution in dataset*

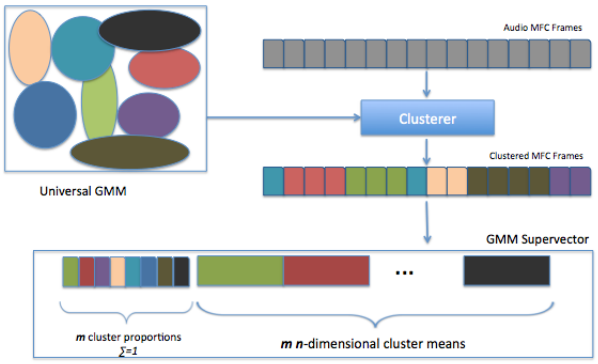

Figure 2: *GMM Featurization*

were crowd-annotated on Amazon Mechanical Turk. The labels obtained had a typical heavy concentration on the most frequent labels; their distribution is shown in Figure 1. We chose the 20 most frequent labels (out of over 300) as our training concepts.

The audio segments were then converted into standard MFC format. All audio was standardized to mono 32Kb/s, and Fourier transforms were computed on 0.025s Hamming-window frames with 60% overlap. The magnitude values in the 50Hz-10kHz frequency range were passed through 60 triangular filters to obtain the 13 most significant MFC coefficients.

3.1.2. Signal Featurization

To convert varying-length MFC sequences into fixed-length vectors appropriate for traditional classification, we trained a 64-dimensional Gaussian Mixture Model as a Universal Audio Model (UAM) on a large unlabeled audio dataset[2] containing 112 hours of unlabeled audio from varied categories and in different recording environments. We then converted our labeled MFC sequences into fixed-length feature vectors by creating GMM supervectors, which represent the audio sequences in terms of our UAM. This approach is based on the one used by [16]. All MFC frames in an audio sequence are clustered by assigning each frame to the UAM cluster from which it was most likely generated. We calculate the mean vector and the proportion of MFC frames assigned to each mode, and we concatenate these values into a single feature vector. For 13 dimensions and 64 modes, this resulted in a feature set of dimensionality 896. A diagram of the featurization process is depicted in Figure 2.

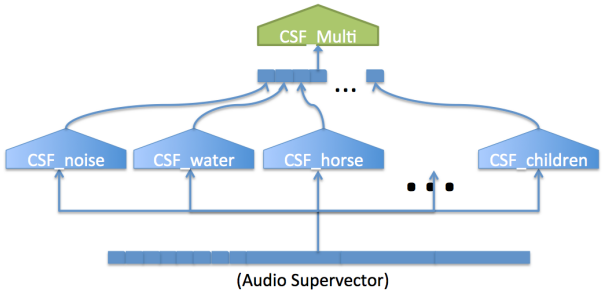

Figure 3: *Bagging Multi-Label Classifier*

3.2. Classification, Segmentation, and Clustering

3.2.1. Classification

We trained a binary 200-tree random forest classifier[20] for each of the 20 labels shown in Figure 1. We then trained a 21-label random forest classifier with 40 trees upon the 20-dimensional feature space consisting of the scores output by the 20 binary classifiers upon an instance. The possible output classes are the original set of 20, plus one labeled ⟨OTHER⟩, to which all other less-occurring labels belong. The ⟨OTHER⟩ label serves later as the identifier for unrecognized segments.

This two-level classifier design is similar to the approach described in [3], pg. 1765, and is depicted in Figure 3. We chose this bagging design because it allows for subsequent introduction of new concepts without full re-training, which we envision the system to do on a regular basis. We evaluated the performance of this classifier through 2-fold cross-validation on the dataset; the results are shown in table 1.

Table 1: *Multi-label performance, 2-fold Cross-Validation*

Label	Accuracy	Precision	Recall	F1	Total
water	1.0000	1.0000	1.0000	1.0000	0.0934
music	0.9993	0.8312	0.8312	0.8312	0.0073
static	0.9994	0.9835	0.9835	0.9835	0.0635
piano	0.9999	0.9982	1.0000	0.9991	0.2144
engine	0.9998	1.0000	0.8605	0.9250	0.0041
waves	1.0000	1.0000	1.0000	1.0000	0.5209
horse	0.9999	1.0000	0.9783	0.9890	0.0132
unknown	0.9996	0.9872	0.9809	0.9841	0.0450
storm	1.0000	1.0000	1.0000	1.0000	1.0000
insects	1.0000	1.0000	1.0000	1.0000	0.0182
birds	1.0000	1.0000	1.0000	1.0000	0.2801
fire	1.0000	1.0000	1.0000	1.0000	0.8572
noise	0.9996	0.9583	0.9673	0.9628	0.0204
speech	0.9986	0.9552	0.9600	0.9576	0.0572
traffic	0.9999	1.0000	0.9968	0.9984	0.1208
violin	1.0000	1.0000	1.0000	1.0000	0.0614
<OTHER>	0.9993	0.9487	0.9873	0.9676	0.0375
silence	0.9999	0.9972	0.9972	0.9972	0.0340
crickets	1.0000	1.0000	1.0000	1.0000	0.1366
sheep	0.9999	1.0000	0.9057	0.9505	0.0051
crowd	0.9998	0.9429	0.8462	0.8919	0.0037

3.2.2. Segmentation

We then built a concept detector with the trained multi-label classifier as the core scoring mechanism. The recognition mechanism is a graph search similar to that of a typical speech recognizer, but it uses fixed-length feature classifiers in place

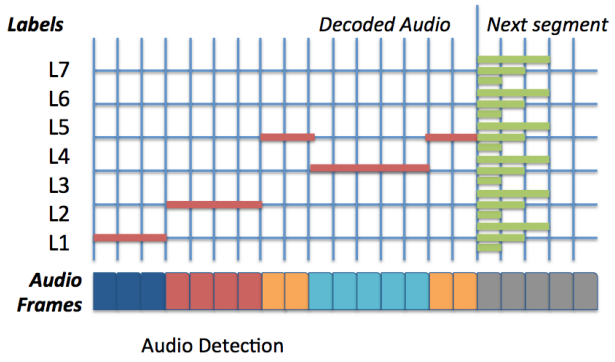

Figure 4: *An example output of the concept detector*

```
segment(audioSequence,scorer)

  //The set of current best sequence candidates
  bestSequences = new SequenceCandidateList()

  currentEndPosition=0
  while currentEndPosition<length(audioSequence)
    //At each time step,
    currentEndPosition += stepSize
    for each newStartPos in bestSequences.endPositions()
      //For each previous end position
      clipSequence = audioSequence[newStartPos:currentEndPosition]
      for each sequence in bestSequences.sequencesWithEndPos(segendPosition)
        scores = scorer.score(sequence)
        for each label in scorer.allLabels()
          newSegment = new Segment(label,scores[label],newStartPos,currentEndPos)
          newSequence = sequence.copy()
          newSequence.append(newSegment)
          bestSequences.add(newSequence)
        end for
      end for
    end for
    bestSequences.pruneLowScoringSequence()
  end while
end segment
```

Figure 5: *Segmentation pseudocode*

of HMMs. The segmentation algorithm is as follows: we keep a set of the best sequence candidates accumulated so far, from which we regularly prune the lowest-scoring elements. At each time point, we take the set of ending times of our best sequences, and use each of them as the starting point of a new clip that ends at the current time frame. Each of these clips is scored with the 21-label classifier, and appended to the existing best sequences. The lowest-scoring complete sequences are then pruned out. A depiction of one such candidate sequence is shown on Figure 4, and pseudo-code for the algorithm is in Figure 5.

We applied this recognizer on an isolated subset of our data, with a step size of 1 second and a maximum number of best candidate sequences of 40. We evaluate track-level precision and recall scores for the set: a true positive occurs if a category was recognized in a track that actually contained that label, a false positive if a recognized category is not contained in that track, and a false negative if a category in a track was not recognized at all. Results for this evaluation are shown in table 2.

3.2.3. *Clustering*

We ran clustering experiments on the data to explore whether clustering an audio set would naturally find semantic categories. We clustered this same dataset (unlabeled) with K-Means — chosen due to its sensitivity to scaling — and scored the different clusterings by their cluster entropy score. We define the cluster entropy score for a clustering C and labels L as follows:

$$Score(C) = \sum_{c \in C} p(c) * H(L|c) \quad (1)$$

Table 2: *Track-level segmentation performance*

Label	Count	Matches	Errors	Misses	Precision	Recall
water	11	3	0	8	**1.0000**	0.2727
music	3	0	1	2	0.0000	0.0000
static	45	31	2	12	**0.9394**	0.7209
engine	6	1	0	5	**1.0000**	0.1667
horse	7	2	0	5	**1.0000**	0.2857
unknown	40	9	0	31	**1.0000**	0.2250
cheering	1	0	1	0	0.0000	0.0000
insects	2	0	0	2	0.0000	0.0000
machine	7	0	0	7	0.0000	0.0000
birds	8	0	0	8	0.0000	0.0000
noise	29	2	1	26	0.6667	0.0714
bell	1	0	0	1	0.0000	0.0000
speech	22	17	1	4	**0.9444**	0.8095
<OTHER>	46	34	10	2	0.7727	0.9444
silence	38	15	14	9	0.5172	0.6250
sheep	2	0	0	2	0.0000	0.0000
chickens	2	0	0	2	0.0000	0.0000
crickets	2	0	0	2	0.0000	0.0000
crowd	6	0	0	6	0.0000	0.0000
animal	2	0	0	2	0.0000	0.0000
TOTAL	**280**	**114**	**30**	**136**	**0.791667**	**0.456**

where H is the binary entropy function. This score represents the expected entropy of a cluster with respect to the labels of the segments it contains. A perfect clustering, where each cluster corresponds to one label and viceversa, would incur a score of 0, while a random clustering with cluster sizes equal to the true label distributions represents the baseline entropy to improve upon.

3.2.4. *Contextual Augmentation*

To evaluate the validity of augmenting the acoustic feature set with contextual features, we define two simple contextual feature vectors based on concept co-occurrence: $S1$ and $S2$. The former holds only binary co-occurrence information, while the values of $S2$ are proportional to the durations of other known labels. We also define a weight w, which determines the relative importance between contextual and acoustic features.

Let L be the set of known labels, and SEG the set of ($clip$,$label$) segments output by our detector. Then the coefficients of our contextual feature vectors $S1$ and $S2$, each of length |L|, for any isolated audio segment are:

$$S1_i = \sum_{seg \in SEG} w * I(seg_{label} * L_i) \quad (2)$$

$$S2_i = \sum_{seg \in SEG} w * I(seg_{label} * L_i) * Duration(seg_{clip}) \quad (3)$$

We call the above the context-binary and context-duration features, respectively, and we append them to the original acoustic features for our clustering experiments. Since the effects of this linear weighting method will vary according to the clustering algorithm used, for our experiments in this section we only used the K-Means algorithm, where K was set to 1.2 times the number of true labels.

We used the concept detector described in the previous section to segment an isolated validation set, and we extract those segments recognized as ⟨OTHER⟩ as the unrecognized labels in our set. We then featurized these segments as GMM supervectors as described in 3.1.2, normalized them to zero-mean and one-variance, and appended the contextual features ($S1$ or $S2$) to create our augmented feature set. Altogether we collected three different feature sets: acoustic-only or unaugmented, context-binary, and context-duration.

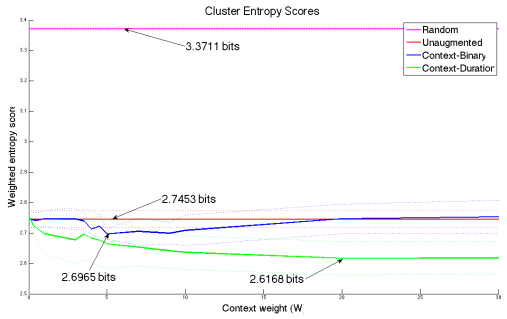

Figure 6: *Entropy scores for feature sets with/without context*

Table 3: *p-values for the Wilcoxon test between feature sets*

$H_1 : ScoreDist(A) > ScoreDist(B)$		p-value
A	B	
Unaugmented	Context-Binary	9.21E-3
Context-Binary	Context-Duration	3.60E-5
Unaugmented	Context-Duration	4.06E-5

We considered the most frequent 40 labels (as the first 20 are known categories), clustered and evaluated them as described in section 3.2.3, and compared their scores against the acoustic-only feature sets. Figure 6 plots the scores obtained for both context-binary and context-duration feature sets against different values of w, and compares them against the scores for the unaugmented feature sets and the random baseline.

The significance of the score differences between the feature sets obtained was evaluated through a Wilcoxon signed-rank test[21] at a significance level of $\alpha = 0.05$, pairwise between all three feature sets. The p-values obtained for all three pairwise comparisons are summarized in table 3.

4. Data Analysis

4.1. Classification results

Cross-validation results (displayed in table 1) over the original 5911 segmented audio clips display excellent discrimination power between the 21 classes, showing that the classifier manages to correctly extract the discriminatory features from the dataset. This allows us to apply them confidently on the subsequent steps of the project.

4.2. Segmentation results

The results on table 2 show that meaningful classes such as water, horse, and engine can be correctly recognized in the absence of segment boundary information. We notice precision tends to score significantly higher than recall — in the absence of insufficient discriminatory information, the recognizer prefers to assign labels with a high prior probability, like static or ⟨OTHER⟩. The speech category seems to score high across all different parameterizations, which is not surprising, as speech is known in the audio field to be highly recognizable.

These results are encouraging, as they align with our goals. The system that we envision will discover new semantic audio concepts by recognizing known ones, but it need not recognize all of them — it is more important that the detections be reliable.

4.3. Clustering results

As seen in figure 6, the mean entropy score is diminished from the random baseline by 0.63 bits through acoustic clustering alone. This reduction is furthered by the addition of contextual features, both context-binary and context-duration. Context-binary features contribute up to a 0.05 bit decrease, and though the error margins' sizes are comparable to the difference, the 0.00921 p-value output by the Wilcoxon test indicates that co-occurrence alone is a meaningful feature for concept discovery.

Context-duration features draw a clearer trend. Their entropy scores are consistently lower than either acoustic-only or context-binary features, and the Wilcoxon test confirms the observation. This difference grows with the contextual weight up to $w = 20$, then stabilizes at 0.13 bits lower than the unaugmented score. This tells us that concept duration features contain information meaningful to audio discovery, more so that co-occurrence alone, and encourages us to investigate further.

5. Conclusions and Future Work

We have presented a potential framework for the discovery of novel audio categories in unlabeled data. The first part of this framework is the design of a reliable concept detector for general-purpose audio, given sufficient training data for each category. These data are assumed to be fairly precise, especially if positive examples are not abundant. Experiments show that this detector incurs in high-precision performance, which allows for high reliability on a meaningful subset of categories, and appropriate behavior for our clustering experiments.

The second part of this framework, which attempts discovery with data clustering, are in line with intuition: signals from different audio categories will tend to be clustered separately, and they lower the entropy score by as much as 0.63 bits. Contextual features take this a notch further, and reduce the score by up to 0.13 bits. Both these results strengthen the belief that discovery of unknown audio concepts is a feasible task, that concept co-occurrence and duration are indicative of an audio clips category, and encourage us to further investigation. These findings strengthen the hypothesis that discovery of unknown audio concepts is a feasible task, and that the framework and techniques presented may help achieve it.

The next steps to take in this line of research are twofold. Firstly, a larger labeled dataset is required to validate the results expressed in this paper. The concept detector is expected to display high track-level precision, and the cluster entropy score is expected to decrease when contextual information is added to the acoustic feature sets, more so with context-duration features.

Secondly, the parameter space of the system could use further exploration. The parameters chosen so far are a result of casual search, and a systematic approach would perhaps elucidate trends and optimum values for the parameters in this framework. Parameters to be considered include the kinds of classifier used, step size and pruning values for the detector, and the definition of contextual feature sets. While the mentioned contextual feature sets possess meaningful information for an audio clip, more sophisticated features can be designed with information like concept proximity and leverage of ontologies. This broader scope of work was not yet attempted due to a lack of sufficient data to reliably extrapolate upon.

We hope that the steps mentioned above help lay the foundation for creating a completely automated system capable of not only recognizing known labels in audio, but also of discovering new categories of audio in an unsupervised fashion.

6. References

[1] BBC Sound Effects Library (http://www.sound-ideas.com/sound-effects/bbc-1-40-cds-sound-effects-library.html).

[2] TRECVid Multimedia Event Detection Evaluation Track (http://www.nist.gov/itl/iad/mig/med.cfm).

[3] Mikel Galar, Alberto Fernndez, Edurne Barrenechea, Humberto Bustince, Francisco Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition, Volume 44, Issue 8, August 2011, Pages 17611776. Online: http://www.sciencedirect.com/science/article/pii/S0031320311000458

[4] Tommi Jaakkola, Mark Diekhans, David Haussler. Using the Fisher kernel method to detect remote protein homologies. From: ISMB-99 Proceedings, 1999.

[5] Pedro J. Moreno, Purdy P. Ho, Nuno Vasconcelos. A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications. HPL-2004-4, HP Laboratories Cambridge

[6] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, B. Raj, Audio event detection from acoustic unit occurrence patterns, in ICASSP, 2012.

[7] S. Chaudhuri, M. Harvilla, and B. Raj, Unsupervised learning of acoustic unit descriptors for audio content representation and classification, in Interspeech, 2011, pp. 717720.

[8] S. Chaudhuri, R. Singh, and B. Raj, Data driven acoustic units for audio classification. Submitted to ICASSP, 2011.

[9] S. Chaudhuri and B. Raj, Learning contextual relevance of audio segments using discriminative models over aud sequences, in WASPAA, 2011.

[10] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, Scream and gunshot detection and localization for audio-surveillance systems, in IEEE Conf. on advanced Video and Signal Based Surveillance, 2008, pp. 2126.

[11] A. Pikrakis, T. Giannakopoulos, and S. Theodoridis, Gunshot detection in audio streams from movies by means of dynamic programming and bayesian networks, in IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, 2008.

[12] K. Lee, D. Ellis, and A. Loui, Detecting local semantic concepts in environmental sounds using markov model based clustering, in ICASSP, 2010.

[13] K. Lee and D. Ellis, Audio-based semantic concept classification for consumer video, IEEE Trans. on Audio, Speech and Language Processing, vol. 18:6, pp. 14061416, 2010.

[14] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, Audio based event detection for multimedia surveillance, in IEEE Intl. Conf. on Acoustics Speech and Signal Processing, 2006.

[15] X. Zhuang, X. Zhou, T. S. Huang, and M. Hasegawa-Johnson, Feature analysis and selection for acoustic event detection, in IEEE Intl. Conf. on Acoustics, Speech and Sig. Proc., 2008.

[16] X. Zhuang, J. Juang, G. Potamianos, and M. Hasegawa-Johnson, Acoustic fall detection using gaussian mixture models and gmm supervectors, in IEEE Intl. Conf. on Acoustics, Speech and Signal Processing, 2009.

[17] L. Lu, F. Ge, Q. Zhao, and Y. Yan, A svm-based audio event detection system, in Intl. Conf. on Electrical and Control Engineering, 2010.

[18] M. Bacchiani, Speech recognition system design based on automatically derived units, PhD Thesis, 1999.

[19] R. Singh, B. Raj, and R. Stern, Automatic generation of subword units for speech recognition systems, IEEE Trans. on Speech and Audio Processing, vol. 10:2, pp. 8999, 2002.

[20] L. Breiman, Random forests, Machine Learning, vol. 45, pp.532, 2001.

[21] Wilcoxon, Frank, "Individual comparisons by ranking methods". Biometrics Bulletin 1 (6): 8083 (Dec 1945).