Monitoring asthma medication adherence through content based audio classification

Stavros Nousias*, John Lakoumentas*, Aris Lalos*,Dimitrios Kikidis†,
Konstantinos Moustakas*, Konstantinos Votis†, Dimitrios Tzovaras†

*Department of Electrical and Computer Engineering, University of Patras, Greece
Emails: {Nousias, Lakoumentas, Aris.Lalos, Moustakas}@ece.upatras.gr

†Centre for Research and Technology-Hellas (CERTH), Thessaloniki, Greece
Email: {DKikidis, KVotis, Tzovaras}@iti.gr

***Abstract*—Chronic respiratory diseases, such as asthma, are very common around the world and have been shown to have a significant effect on the quality of life of patients. A crucial component for the effective management of asthma is the adherence of patients to their medication prescription, which can be separated into two distinct and equally important components, i) the adherence of patients to the time schedule for the use of their inhaled medication and ii) their competence in using the inhaler correctly and effectively. Aiming in this direction the current paper investigates three different algorithmic approaches not only for the detection of Metered Dose Inhaler actuations but for the understanding of the overall inhaler technique of patients. More specifically, Short Time Fourier Transform is used as the basis for the extraction of features that are then used for the classification of 4 events (inhaler actuation, patient inhalation, patient exhalation, background noise) using three distinct algorithmic approaches (Support Vector Machines, Random Forests and AdaBoost). The final experimental results demonstrate that Adaboost outperforms the alternative approaches leading to accuracies above 96%.**

***Index Terms*—Asthma, medication adherence, pMDI correct usage, time-frequency analysis, classification.**

1. Introduction

Asthma is a chronic disease of the airways that affects more than 235 million people worldwide [1] including a continuously increasing number of children [2]. In the region of Europe more than 30 million adults suffer from asthma [3] creating a number of difficulties in the quality of life of patients and their families and affecting the overall efficiency of the healthcare system as a whole [4]. The fact that asthma is present in a such a diverse and global scale [5] reveals the importance for new and innovative approaches that can help patients irrespective of their cultural and educational background to manage their asthma and avoid dangerous exacerbation events [6].

One of the most important components for the effective management of asthma is the adherence of patients to their medication prescription both in terms of following the schedule that is proposed by the responsible doctor, and using the inhaler device correctly and effectively. Reduced adherence of inhaled medication has been already associated with asthma attack incidents and patient hospitalizations [7], indicating that the effective monitoring of the medication adherence can facilitate the management of these respiratory diseases.

Recent studies of inhaler based monitoring devices have revealed the significance of such solutions for the effective management of respiratory diseases and outlined some important common characteristics and disadvantages of current approaches [8], [9], [10], [11]. The majority of devices presented are based on electromechanical sensing capabilities, ranging from simple push buttons attached on the top of the inhaler's canister through integrated counters [12] to force sensing elements attached on the back of the inhaler's plastic casing. The second most prevalent sensing approach adopted by electronic inhaler devices is the use of microphones [13] [14], the measurements of which are locally processed and used to indicate inhaler actuations. Nevertheless, only few devices have demonstrated functionalities for the assessment of inhaler technique and especially the timing of inhaler actuation in regards to the inhale and exhale events.

The objective of this work is to evaluate three different algorithmic approaches in this direction and in order to identify and evaluate the proper use of pressurized Metered Dose Inhalers (pMDI) by patients and in real life conditions. More specifically the correct usage of a pMDI is defined according to clinical expert suggestions by the following steps [15]: a)Remove the cap b)Breathe out, away from your inhaler c)Bring the inhaler to your mouth. Place it in your mouth between your teeth and close you mouth around it. d)Start to breathe in slowly. Press the top of you inhaler once and keep breathing in slowly until you have taken a full breath. e)Remove the inhaler from your mouth, and hold your breath for about 10 seconds, then breathe out.

Towards this direction, a first step of analysis is to achieve to differentiate individual sounds. The individual sounds we process are breathe in , breathe out , pMDI inhaler actuations, and background/environmental sounds. In the future, we intend to utilize more sophisticated procedures in order to classify the whole process of pMDI inhaling as correct or incorrect, using audio processing as well as pattern classification schemes. The rest of the paper is outlined as follows: Section 2 presents the methods for data acquisition and processing of the audio samples. Section 3 presents the audio feature extraction. Section 4, describes the adopted classification approaches. In Section 5, we present the results . Finally, Section 6 concludes this paper.

2. Materials and Methods

2.1. Data Acquisition and Construction of Datasets

We recorded a set of sounds in indoor and outdoor environments that were classified into inhaler actuations, breathe in, breathe out, and noise referring to environmental sounds. 5 healthy people participated in the experiments and recording was performed with the use of a recording device composed of a wireless Bluetooth microphone attached to the pMDI and a Smartphone. The dataset that was produced was comprised of 280 samples per sound class. Thus, totally 1120 sound samples were available. Each sound sample was of 0.5 seconds duration, with 4 kHz sampling rate and 4-bit depth.

2.2. Signal processing approach

The objective of the described approach is to classify correctly those sound samples. 3 problems were considered for analysis. The first problem was of 2-class nature: we aimed to distinguish only inhaler actuations from noise. The second was of 3-class nature: we placed into a new bin 280 mixed sound samples from breathe in and breathe out and attempted to distinguish from inhaler actuations and noise simultaneously. In the third problem, we utilized all the dataset and performed classification of 4-class nature: inhaler actuations vs. breathe in vs. breathe out vs. noise. In the following paragraphs we mention technical details of the whole process, which is comprised of feature extraction and pattern classification.

As a first step of processing, STFT was applied for feature extraction of the recorded sounds. Each sound sample consists of 2000 samples and the length of the sliding window is 128 samples with a sliding step of 32 samples The resulting spectrogram consists of 59 columns containing an estimate of the short-term, time-localized frequency content consisting of 257 frequency components. In order to derive the frequency vector we reduce the dimensionality of the spectrogram removing the time domain. Finally, the subsampled frequency vector produces the feature vector consisting of 40 features in order to be used as input for the classification problem.

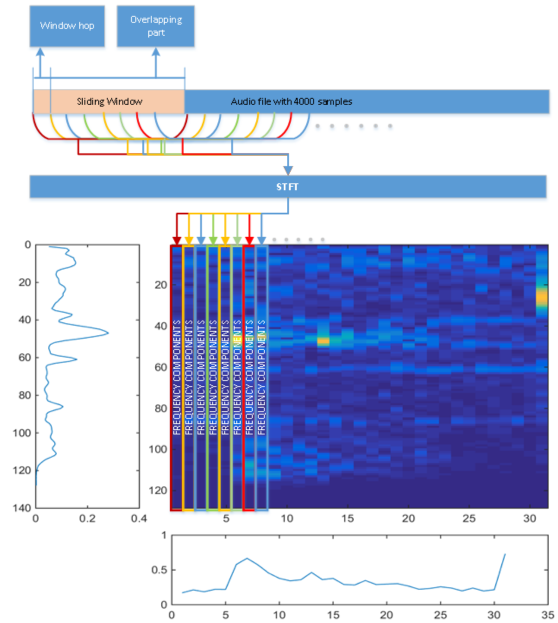


Figure 1. *Short time fourier transform of the audio samples*

Following feature extraction, pattern classification was performed in order to differentiate the sound samples into the 4 aforementioned types. 3 classifiers were tested: support vector machines, random forests, and AdaBoost . We did not follow the approach of a single testing and training test; instead, we utilized 10-fold cross validation for evaluation of the classifiers, which we take as a most robust approach. Classification accuracy is defined as the percent of the samples that were finally classified correctly.

3. Audio Features Extraction

The acoustic sensor (source node), records a real time audio sound and transmits to the BNC (e.g., smartphone) the recorded samples. At the BNC, the features are extracted from the encoded audio samples, which are then used for the audio classification problem. Specifically , the audio signal is recorded by the microphone, digitized, and divided into segments of $N = f_s/2$ samples that correspond to 0.5 sec of audio, e.g., $\mathbf{x} = [x_1, \ldots, x_N]^T$, where $x_i \in \mathbb{R}$.

Existing acoustic approaches [13] suggest the use of time-frequency analysis from pMDI audio recordings to automatically detect pMDI actuations. Thus we derive the audio features from the spectrogram of the audio samples that is generated by applying a short time Fourier Transform (STFT). To be more specific, the processed audio frames are separated to overlapping parts (so as to reduce artifacts at the boundary) with each part Fourier transformed.The complex result is added to a matrix, which records magnitude and phase for each point in time and frequency. This can be expressed as:

$$\mathbf{STFT}\{x[n]\}(m,\omega) = \sum_{n=0}^{N} x[n]w[n-m]e^{-j\omega n} \quad (1)$$

where x[n] is the signal, w[n] the window and N the number of samples.The magnitude squared of the STFT yields the spectrogram:

$$\text{spectrogram}\{x[n]\}(m,\omega) \equiv |STFT\{x[n]\}(m,\omega)|^2 \quad (2)$$

Thus the audio features $\mathbf{f}_i$ are derived by adding the frequency magnitude for every time windows for each frequency component resulting in a subsampled one dimensional vector containing a summation of all frequency components at a given time index i.

4. Feature Classification for Medication Adherence

Pattern classification in machine learning is described as the following problem [16]. There is available a set of entities, where each entity is described by an attribute vector and a special attribute called the class. While the attributes of the attribute vector can be either discrete or continuous, the class attribute is discrete. Pattern classification is the issue of estimating a function f that maps the attributes vector to the class attribute. Such a function is referred to as a classification model. A classification model is useful for the reason that it explains in what way the attributes vector distinguishes a set of entities into diverse classes. However, its most widespread objective is predictive. It can be applied to differentiate into classes new entities, which have known attributes but unknown classes. Pattern classification is the most frequently seen problem in supervised learning.

A classification algorithm or classifier is a systematic approach of constructing classification models from data. Each classifier applies a learning technique, aiming to identify the model that fits to the data best (in particular, to identify the dependencies between the attributes vector and the class). The resulted model must be able to fit to the data optimally, but also to predict entities with unknown classes as well. Thus, it is required to dispose generalization properties. A classification problem is comprised by the following steps. Initially, a training dataset is given, where entities have known classes, and a classification algorithm is applied to produce a classification model. Then, the generated model is applied to a testing dataset, where entities have unknown classes, in order to predict their classes. In some cases, there exists no testing dataset, but only training. In such a case, a classification model is evaluated with k-fold cross validation: the training dataset is split into k parts (usually 10), each of the parts sequentially forms the testing set and all the others form the training set, finally the k results are combined.

In the next few paragraphs, we describe 3 well-known, sophisticated, and relatively recently invented classification algorithms: support vector machines, random forests, and AdaBoost. We will be referring to the vector of attributes of entity i as x_i (assuming cardinality p) and to its class value c_i (taking integer values from 1 to K). Thus, K is the number of classes.

4.1. Support vector machines

Support vector machines [17] are a well known and sophisticated method for supervised learning. In describing SVMs, we will first assume that the number of classes is equal to two (K=2); that is $c_i = \pm 1$. At the end, we will generalize to more distinct class values. Furthermore, let us assume that all entities in the training dataset are linearly separable in the space of their attributes (p-dimensional). This fact denotes that there exists a separating hyper plane and consequently a vector of weights w and number b that satisfy the following conditions: $b + wx_i \geq 1$ if $c_i = 1$, and $b + wx_i \leq -1$ if $c_i = -1$. Equivalently, $c_i(b + wx_i) \geq 1$ for every entity of the dataset. In that simple situation of linearly separable entities, SVMs effort to satisfy the aforementioned condition while minimizing the quantity $\frac{1}{2}\left\|w\right\|^2$. This is a curve quadratic programming problem, that is solved approximately. If the separating hyper plane is recognized, its boundaries are referred to as support vectors and the solution can be represented simply by a linear combination of those. Unfortunately, in real life situations the entities of a dataset are rarely linearly separable. In such a case, there are two ways to deal with the issue, that can be used simultaneously. The first one is that one can leave a margin ξ for misclassified entities. In quadratic programming form this equals to adding a slack variable in the constraint equation: $c_i\left(b + wx_i\right) \geq 1 - \xi$ and minimizing the quantity $\frac{1}{2}\left\|w\right\|^2 + C\xi$, where C is a trade-off constant. The second one is to attempt to separate the entities linearly in another space of higher dimensionality. For that purpose, a kernel function is used that transforms the inner product of any two entities to the new space. Common kernel functions are:

- The polynomial of degree q:
$$K\left(x_i, x_j\right) = \left(x_i x_j + 1\right)^q \quad (3)$$
- The RBF with parameter q:
$$K\left(x_i, x_j\right) = e^{-q\left\|x_i - x_j\right\|^2} \quad (4)$$
- The sigmoid of order q:
$$K\left(x_i, x_j\right) = tanh(\kappa x_1 x_2 - \delta)^q \quad (5)$$

For multi-class problems ($K \geq 2$), there are 2 ways to deal with. One way is the one versus all approach. Here, we create K SVM classifiers, and for each classifier, we are attempting to distinguish one particular class from all the rest. To determine the optimal class to pick, we assign the class for which the observation produces the highest distance from the separating hyper plane, therefore lying farthest away from all other classes. An alternative approach is known as the one versus one approach. We create a classifier for all possible pairs of output classes. We then classify our observation with every one of these classifiers and tally up the totals for every winning class. Finally, we pick the class that has the most votes.

4.2. Random Forests

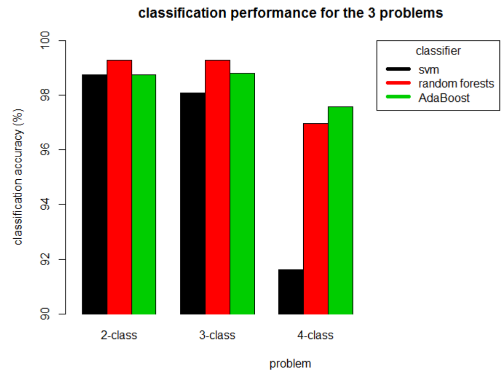
Random forests [18] are the most widespread paradigm for the concept of classification bagging. Bagging (bootstrap aggregating) is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms. It is based on the idea of combining classifications of randomly generated training sets. In the case of random forests, the individual classifiers used in parallel are small classification trees and they are applied to different bootstrap samples of the training dataset. The number of trees is selected to be 500 or 1000 usually. More specifically, random forests grow a lot of classification trees. To classify a new entity with an attributes vector, put the attributes vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). Each tree is growing as follows:

- If the number of entities in the training dataset is n, sample n entities at random but with replacement. This sample will be the training set for growing the tree.
- If the cardinality of x_i is p, a number of $\sqrt{p}$ attributes is selected at random for each node of the tree, and the best split on these is used to split the node.
- Each tree is grown to the largest extent possible. There is no pruning.

The forest error rate (accuracy) depends on two things: (a) the correlation between any two trees in the forest. Increasing the correlation increases the forest error rate. (b) the strength of each individual tree in the forest. Increasing the strength of the individual trees decreases the forest error rate. An advantage of random forests is that they do not need k-fold cross validation. Instead, the out-of-bag (OOB) error estimate can be computed. More specifically, each tree is built using a different bootstrap sample from the original data. About one-third of the cases are left out of the bootstrap sample and not used in the construction of each tree. Put every entity left out in the construction of each tree down the tree to get a classification. In this way, a test set is obtained for each entity in about one-third of the trees. The OOB error is estimated in that test set.

4.3. AdaBoost

AdaBoost [19] is the most widely used classification boosting algorithm. Boosting answers to the following question: Can a set of weak learners create a single strong learner? A weak learner is defined to be a classifier that is slightly correlated with true classification (for example, it can label entities better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification. As it is in the case of random forests, in AdaBoost the weak learners are simple classification trees. More specifically, the AdaBoost algorithm is an iterative process that attempts to combine a set of weak classifiers. Starting with the unweighted training dataset, the AdaBoost builds a classifier, for example a classification tree, that produces class labels for the entities. Then, if an entity is misclassified, the weight of that entity is increased (boosted). Another classifier is then built with the new weights, which are no longer equal. Once more, the misclassified entities have their weights boosted and the process is repeated. Typically, one may build 500 or 1000 classifiers this way. A score is assigned to each one classifier, and the final classifier is defined as the linear combination of the classifiers from each stage. Specifically, let T be a weak multi-class classifier (classification tree).



(a)

Figure 2. *Classification accuracy for the three problems*

- Initialize the weights for all the entities as $w_i = \frac{1}{n}$.
- Repeat the following for, say, 500 or 1000 times, indexed by j:
  - Fit a classification tree $T^j(x)$ to the training data using weights w_i.
  - Compute the error :

$$err^j = \frac{\sum_{i=1}^{n} w_i I(c_i \neq T^j(x_i))}{\sum_{i=1}^{n} w_i} \quad (6)$$

  - Compute the coefficient :

$$a^j = \log\left(\frac{1 - err^j}{err^j}\right) + \log(K-1) \quad (7)$$

  - Update the weights :

$$w_i = w_i e^{a^j I\left(c_i \neq T^j(x_i)\right)} \quad (8)$$

  - Renormalize the weights.
- Output $C(x) = argmax_k \sum_j a^j I\left(T^j(x) = k\right)$.

5. Results

The classification results achieved are presented in figure 2. One can observe that in all 3 considered problems we achieve classification accuracy of 90% and over. However, in all cases minus one (support vector machines in the 4-class problem) the accuracy is more than 96%. More specifically, considering the 2-class problem, support vector machines, random forests, and AdaBoost produce accuracies of 98.75%, 99.29%, and 98.75%. Considering the 3-class problem, the same classifiers produce accuracies of 98.10%, 99.29%, and 98.81%, respectively. Considering the (most difficult) 4-class problem, they are accurate at the 91.61%, 96.96%, and 97.59% of the instances, respectively. Random forests seem to overcome the other 2 classifiers, except from the 4-class problem where AdaBoost is preferable.

6. Conclusion and future work

Wireless monitoring of pMDI inhaler medication adherence from acoustic sounds facilitates the early diagnosis and management of chronic inflammatory diseases of the airways, such as asthma, but introduces challenges related to accurate classification of the audio signals. To this end, we attempt to implement a method that categorizes as correct or incorrect the usage of a pMDI inhaler, according to medical experts suggestions, utilizing techniques from the domains of audio signal processing and machine learning. As an initial step towards that direction, we achieved to classify four different types of sounds , including breath in , breath out , inhaler actuations and noise , with accuracies of more than 96%, using state-of-the-art classification methods, along with feature extraction from audio signals . 3 classification problems were taken into account, a 2-class, a 3-class, and a 4-class, and in almost all cases the three classifiers , support vector machines, random forests and AdaBoost, performed adequately well. The presented schemes will be implemented on Smartphones that will be running the Android operating system. In the future, we intend to utilize more sophisticated procedures in order to classify the whole process of pMDI inhaling as correct or incorrect, using audio processing as well as pattern classification schemes.

Acknowledgments

This work has been funded by the H2020-PHC-2014-2015 Project MyAirCoach (Grant Agreement No. 643607).

References

[1] World health organization asthma. fact sheet no 307. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs307/en/

[2] European lung white book. childhood asthma. [Online]. Available: http://www.erswhitebook.org/chapters/childhood-asthma/

[3] European lung white book. adult asthma. [Online]. Available: http://www.erswhitebook.org/chapters/adult-asthma/

[4] M. Masoli, D. Fabian, S. Holt, and R. Beasley, "The global burden of asthma: executive summary of the gina dissemination committee report," *Allergy*, vol. 59, no. 5, pp. 469–478, 2004.

[5] M. I. Asher, S. Montefort, B. Björkstén, C. K. Lai, D. P. Strachan, S. K. Weiland, H. Williams, I. P. T. S. Group *et al.*, "Worldwide time trends in the prevalence of symptoms of asthma, allergic rhinoconjunctivitis, and eczema in childhood: Isaac phases one and three repeat multicountry cross-sectional surveys," *The Lancet*, vol. 368, no. 9537, pp. 733–743, 2006.

[6] L. J. Akinbami, J. E. Moorman, C. Bailey, H. S. Zahran, M. King, C. A. Johnson, X. Liu *et al.*, "Trends in asthma prevalence, health care use, and mortality in the united states, 2001–2010," *NCHS data brief*, vol. 94, no. 94, pp. 1–8, 2012.

[7] A. C. Murphy, A. Proeschal, C. E. Brightling, A. J. Wardlaw, I. Pavord, P. Bradding, and R. H. Green, "The relationship between clinical outcomes and medication adherence in difficult-to-control asthma," *Thorax*, pp. thoraxjnl–2011, 2012.

[8] A. S. Lalos and K. Moustakas, "Energy efficient telemonitoring of wheezes," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 539–543.

[9] A. S. Lalos, J. Lakoumentas, A. Dimas, and K. Moustakas, "Energy efficient monitoring of metered dose inhaler usage," *Journal of Medical Systems*, vol. 40, no. 12, p. 285, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10916-016-0642-y

[10] D. Kikidis, V. Konstantinos, D. Tzovaras, and O. S. Usmani, "The digital asthma patient: The history and future of inhaler based health monitoring devices," *Journal of aerosol medicine and pulmonary drug delivery*, vol. 29, no. 3, p. 219232, June 2016. [Online]. Available: http://dx.doi.org/10.1089/jamp.2015.1267

[11] S. Howard, A. Lang, M. Patel, S. Sharples, and D. Shaw, "Electronic monitoring of adherence to inhaled medication in asthma," *Current Respiratory Medicine Reviews*, vol. 10, no. 1, pp. 50–63, 2014.

[12] J. Pilcher, M. Holliday, S. Ebmeier, S. McKinstry, F. Messaoudi, M. Weatherall, and R. Beasley, "Validation of a metered dose inhaler electronic monitoring device: implications for asthma clinical trial use," *BMJ open respiratory research*, vol. 3, no. 1, p. e000128, 2016.

[13] T. E. Taylor, M. S. Holmes, I. Sulaiman, S. D'Arcy, R. W. Costello, and R. B. Reilly, "An acoustic method to automatically detect pressurized metered dose inhaler actuations," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 4611–4614.

[14] S. D'Arcy, E. MacHale, J. Seheult, M. S. Holmes, C. Hughes, I. Sulaiman, D. Hyland, C. O'Reilly, S. Glynn, T. Al-Zaabi *et al.*, "A method to assess adherence in inhaler use through analysis of acoustic recordings of inhaler events," *PloS one*, vol. 9, no. 6, p. e98701, 2014.

[15] The asthma society of canada - how to use a metered-dose inhaler. [Online]. Available: http://www.asthma.ca/adults/treatment/meteredDoseInhaler.php

[16] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[17] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[18] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] J. Zhu, H. Zou, S. Rosset, and T. Hastie, "Multi-class adaboost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.