A mixture model-based real-time audio sources classification method

Maxime Baelde, Christophe Biernacki, Raphaël Greff

A MIXTURE MODEL-BASED REAL-TIME AUDIO SOURCES CLASSIFICATION METHOD

Maxime Baelde[1,2], Christophe Biernacki[1], Raphaël Greff[2]

[1]University Lille 1, CNRS, Inria
[2]A-Volute

ABSTRACT

Recent research on machine learning focuses on audio source identification in complex environments. They rely on extracting features from audio signals and use machine learning techniques to model the sound classes. However, such techniques are often not optimized for a real-time implementation and in multi-source conditions. We propose a new real-time audio single-source classification method based on a dictionary of sound models (that can be extended to a multi-source setting). The sound spectrums are modeled with mixture models and form a dictionary. The classification is based on a comparison with all the elements of the dictionary by computing likelihoods and the best match is used as a result. We found that this technique outperforms classic methods within a temporal horizon of 0.5s per decision (achieved 6% of errors on a database composed of 50 classes). Future works will focus on the multi-sources classification and reduce the computational load.

***Index Terms*—** real-time, audio identification, statistical learning, mixture models, sound classification.

1. INTRODUCTION

Audio source classification is a vast and trendy topic in machine learning, and can be divided into three groups. Music Information Retrieval (MIR) aims at recognizing musical instruments [1] or musical genres [2] from musics. Automatic Speech Recognition (ASR) aims at detecting and identifying speakers in audio recordings [3]. Environmental Sound Recognition (ESR) aims at recognizing classes of sounds that are neither music nor speech [4]: for instance, airplanes or gunshots.

Typical audio source classification methods include two stages. The first one consists in extracting features from the signals using audio descriptors [5]. Common features are temporal (energy, zero crossing rate,...), spectral (Mel Frequency Cepstral Coefficient, centroid,...) or harmonic (fundamental frequency, inharmonicity,...). This step extracts relevant information from the signal and can be seen as a dimension reduction. The second one uses machine learning algorithm to model the sound classes based on the previous features. Commonly used algorithms are Gaussian Mixture Models (GMM) [6, 7], Support Vector Machines (SVM) [8, 9], Hidden Markov Models (HMM) [10, 11] or Neural Networks (NN) [12, 13]. More recently, research focuses on neural networks and uses them for features extraction and classification at the same time. These networks often involve convolutional layers and deep architecture (Deep Convolutional Neural Network, DCNN) to model fine details in signals [3, 14].

In this study, we develop an audio single-source classification system based on a dictionary of models in a probabilistic framework, using the mixture model theory [15]. Indeed, the future goal of this research is to identify several sound sources at the same time, which is merely impossible with the current techniques. The use of mixture models allows to deal with mixture of sounds and therefore to identify simultaneous audio sources. Each sound spectrum is modeled by a mixture model and all the models constitute a dictionary. The classification is performed by comparing an unknown signal with the elements of the dictionary by computing likelihoods, aggregating these probabilities and taking the Maximum A Posteriori (MAP).

The rest of the paper is organized as follows. The creation of the dictionary is presented in Section 2. The single-source identification procedure that uses the previous dictionary is detailed in Section 3. The experiments carried out to assess the performance of the method are presented in Section 4. Section 5 presents the results of the experiments. Finally, a brief discussion is presented in Section 6 and concludes the paper.

2. CREATION OF THE DICTIONARY

The classification algorithm deals with multiple classes of sounds, denoted G_i, $i = 1, ..., I$, and a mono-channel stream. For instance, the classes can be $G_1 = \text{airplane}$, $G_2 = \text{gunshot}$. In each group G_i there are multiple sounds, labeled C_{ij}, $j = 1, ..., J_i$. For instance in G_1, $C_{11} = \text{airplane\_1}$, and in G_2, $C_{23} = \text{gunshot\_3}$. As a typical signal processing step for real-time application, the sounds C_{ij} are splitted into buffers, labeled c_{ijk}, $k = 1, ..., K_{ij}$ of size T samples with a time shift of D samples (see **Fig.** 1). In a signal notation, $\mathbf{c}_{ijk} = \mathbf{C}_{ij}[kD : kD + T]$. In real world applications, the signal does not necessarily fit correctly into the buffer. To tackle this problem, Gaussian white noise is added at the beginning and the end of each sound.

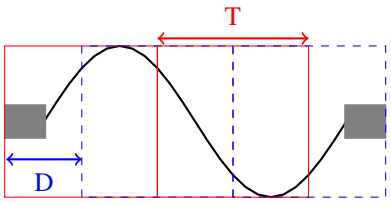

Fig. 1. A signal (black plain line) splitted into buffers of size T samples (red empty square) with a time shift of D samples (blue empty dashed square). Gaussian white noise is added at the beginning and the end (gray full block).

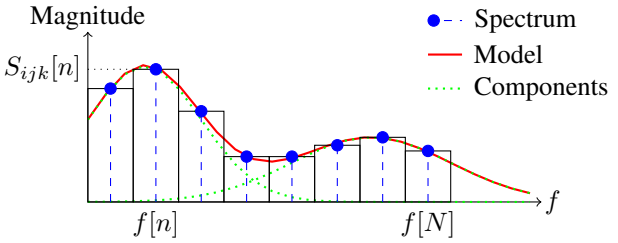

Fig. 2. Modeling of the spectrums as histograms. The spectrum is represented by the blue dots, the mixture model by a red plain line and its components by green dotted lines.

The Fourier spectrum of each buffer is computed, denoted by $\mathbf{s}_{ijk}$. Only the first N bins in the spectrum are kept, because most of the information is included in the low-frequency content of the signal. As our algorithm is designed to classify multiple sounds present at the same time, the energy spectrum is used. Indeed, when two uncorrelated signals are mixed, their energy spectrums are mixed in the same proportion. The spectrums are normalized so that they sum up to N:

$$S_{ijk}[n] = N \frac{\left|s_{ijk}[n]\right|^2}{\sum_{p=1}^{N} \left|s_{ijk}[p]\right|^2}, \tag{2.1}$$

where $|z|$ denotes the modulus of the complex number z. The square values are considered since we want to keep the additivity of the spectrums. These spectrums are considered as histograms and are modeled using a mixture model for binned data [16] (see **Fig. 2**). For the sake of clarity, we will drop the indices (i, j, k) until the end of this section. A mixture model [15] is a mixture of several probability density functions (pdf), the components, which represents the distribution of a random variable. Here the random variable is the frequency f at which the spectrum is computed, and the pdf is parameterized by a set of parameters $\boldsymbol{\theta} = (\pi_m, \mu_m, \sigma_m^2)_{m=1,..,M}$:

$$p\left(f|\boldsymbol{\theta}\right) = \sum_{m=1}^{M} \pi_m \mathcal{N}\left(f\left|\mu_m, \sigma_m^2\right.\right), \tag{2.2}$$

where $p(f|\boldsymbol{\theta})$ means the probability of f parameterized by $\boldsymbol{\theta}$, π_m the mixing coefficients ($\sum_m \pi_m = 1, \pi_m > 0$) and M is the number of components. $\mathcal{N}\left(f\left|\mu_m, \sigma_m^2\right.\right)$ is the density of the univariate normal distribution of mean μ_m and variance σ_m^2, taken at point f:

$$\mathcal{N}\left(f\middle|\mu_m, \sigma_m^2\right) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{1}{2}\left(\frac{f-\mu_m}{\sigma_m}\right)^2\right).$$

The frequency f is not known precisely: we know only how many frequencies $S[n]$ fall into the frequency range $\left[f[n], f[n+1]\right[$. This framework is close to the one of McLachlan [16] except that $S[n]$ is not an integer. Thus, given that the fs are independent and identically distributed (i.i.d.), the probability that one sample falls into $\left[f[n], f[n+1]\right[$ is:

$$p\left(S[n]|\boldsymbol{\theta}\right) = \left(\int_{f[n]}^{f[n+1]} p(f|\boldsymbol{\theta}) df\right)^{S[n]}. \tag{2.3}$$

The Expectation-Maximization (EM) algorithm of Dempster [17] is used to estimate the set of parameters $\boldsymbol{\theta}$. This algorithm finds the parameters that maximize the likelihood of the model parameterized by $\boldsymbol{\theta}$. Given that the $S[n]$s are i.i.d., the likelihood is:

$$\mathcal{L}(\boldsymbol{\theta}) = p(\mathbf{S}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p\left(S[n]|\boldsymbol{\theta}\right), \tag{2.4}$$

where $\mathbf{S} = \left[S[1], ..., S[N]\right]$. Starting with an initial parameter estimate $\boldsymbol{\theta}^{(0)}$, the EM algorithm iterates the two following steps until convergence:

(E) Compute: $Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(p)}\right) = \mathbb{E}\left[\log \mathcal{L}(\boldsymbol{\theta})\middle|\mathbf{S}, \boldsymbol{\theta}^{(p)}\right]$,

(M) Maximize: $\boldsymbol{\theta}^{(p+1)} = \text{argmax}_{\boldsymbol{\theta}}\, Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(p)}\right)$.

This model and the related algorithm require the number of components M in the mixture to be specified. For each buffer, the optimal number of components is chosen by minimizing the Bayesian Information Criterion (BIC) of Schwarz [18]:

$$BIC(M) = -2\log \mathcal{L}\left(\widehat{\boldsymbol{\theta}}\right) + d\log N, \tag{2.5}$$

where $\widehat{\boldsymbol{\theta}}$ denotes the Maximum Likelihood (ML) value of $\boldsymbol{\theta}$ and $d = 3M - 1$ is the dimension of $\boldsymbol{\theta}$, with M varying from 1 to 20 in this case. The estimated parameters are gathered into $\left\{\widehat{\boldsymbol{\theta}}_{ijk}\right\}$, with $k = 1, ..., K_{ij}$, $j = 1, ..., J_i$, $i = 1, ..., I$, which is named the *dictionary*.

3. IDENTIFICATION PROCEDURE

Once the dictionary is computed (this corresponds to the learning step), it can be used to identify unknown sounds. Suppose that at a time r there is a new buffer which is transformed into a normalized energy spectrum $\mathbf{S}^r$. The identification process

consists in finding the correct label G_i^r by aggregating the following probabilities:

1. Compute the likelihoods of all the models of the dictionary $p\left(\mathbf{S}^r \middle| c_{ijk}^r\right) = p\left(\mathbf{S}^r \middle| \widehat{\boldsymbol{\theta}}_{ijk}\right)$, for $k = 1, ..., K_{ij}$, $j = 1, ..., J_i$ and $i = 1, ..., I$ using Eq. (2.4).

2. Compute the likelihood of the sound C_{ij}^r, for $j = 1, ..., J_i$ and $i = 1, ..., I$:

$$p(\mathbf{S}^r|C_{ij}^r) = \sum_{k=1}^{K_{ij}} p(\mathbf{S}^r|c_{ijk}^r)p(c_{ijk}^r|C_{ij}^r). \quad (3.1)$$

3. Compute the likelihood the group G_i^r, for $i = 1, ..., I$:

$$p(\mathbf{S}^r|G_i^r) = \sum_{j=1}^{J_i} p(\mathbf{S}^r|C_{ij}^r)p(C_{ij}^r|G_i^r). \quad (3.2)$$

4. Finally compute the conditional probability of the group G_i^r for $i = 1, ..., I$:

$$p(G_i^r|\mathbf{S}^r) = \frac{p(\mathbf{S}^r|G_i^r)p(G_i^r)}{\sum_h p(\mathbf{S}^r|G_h^r)p(G_h^r)}. \quad (3.3)$$

The decision rule uses R buffers (as in [6]) and is computed as follows. For each buffer r and each class the classifier computes a probability $p(G_i^r|\mathbf{S}^r)$. Then these probabilities are multiplied so as to aggregate the results over a larger temporal horizon:

$$p\left(G_i|\mathbf{S}\right) = \prod_{r=1}^{R} p\left(G_i^r \middle| \mathbf{S}^r\right), \quad (3.4)$$

where $\mathbf{S} = \left[\mathbf{S}^1, ..., \mathbf{S}^R\right]$. The MAP estimate is finally taken on these probabilities: $\widehat{G}_i = \mathrm{argmax}_{G_i}\, p\left(G_i|\mathbf{S}\right)$.

4. EXPERIMENTS

4.1. Setup of the experiments

Several experiments were carried out to assess the quality of our sound sources classification algorithm. Three databases were considered for these experiments. One database (from A-Volute) was composed of 704 video games sounds divided into 9 classes. The ESC databases [19] (50 and 10) were also considered, composed of 2000 environmental sounds from 50 classes for the former and 400 sounds from 10 classes for the latter. Each sound was resampled to 44.1kHz and the mean was subtracted to the signal because it caused instability in the fitting process. Several values were considered for the window size T and the time shift D, mainly $T = [512, 1024, 2048]$ samples and $D = 512$ samples. The number of kept bins N was set to $T/5$ which corresponded to approximately 8kHz in our setting. R was set to 10, that corresponds approximately to 0.5s.

Each dataset were splitted into a training and a test set for cross-validation procedure. We used 80% of the set for training and the remainder for testing. The division was done in a v-fold manner, with $v = 5$. The recognition rate, defined by the number of buffers correctly labeled over the total number of buffers, was used to assess the performance of the system. In a cross-validation setting, a recognition rate for each fold was computed and the cross-validation recognition rate was simply the mean of the v folds recognition rates.

Three probabilities were introduced in Section 3 and have to be specified:

- The probability of a buffer c_{ijk} given a sound C_{ij} was set to 1: $p(c_{ijk}|C_{ij}) = 1$ for $k = 1, ..., K_{ij}$, $j = 1, ..., J_i$ and $i = 1, ..., I$.
- The probability of a sound C_{ij} given a class G_i was uniformly distributed over the class G_i: $p(C_{ij}|G_i) = 1/\#\{G_i\}$ for $j = 1, ..., J_i$ and $i = 1, ..., I$, where $\#\{A\}$ means the number of elements in A.
- The probability of a class G_i was set to the ratio between the number of sounds in G_i and the total number of sounds: $p(G_i) = \#\{G_i\}/\sum_h \#\{G_h\}$ for $i = 1, ..., I$.

4.2. Comparison with other techniques

We compared our algorithm with other state-of-the-art techniques. The first one was a parametric method inspired from Clavel [6]. Acoustic features were extracted from the signals: energy, 8 MFCC, spectral centroid, spectral spread, plus the first and second derivatives. A Principal Component Analysis step was applied on these descriptors and only the first 13 principal components were kept. The classifier was a standard GMM.

The second one was a non-parametric method: a DCNN. A neural network is a set of neurons (computation units) stacked together in multiple layers that can performs classification or regression. The input of a layer l is denoted by $\mathbf{h}_{l-1}$, and the convolutional kernel $\mathbf{W}_l$. The network computes an output $\mathbf{h}_l$ by taking an activation function g: $\mathbf{h}_l = g(\mathbf{W}_l \star \mathbf{h}_{l-1})$, where $\star$ is the convolution operation. Recently, the most popular activation function is the ReLU (Rectified Linear Unit), which is simply $g(\mathbf{X}) = \max(\mathbf{X}, 0)$ element-wise. After a convolutional layer, a max-pooling operation consisting in merging adjacent cells by taking the maximum value is applied. We used the network developed by Piczak [14]. The input was a log-mel spectrogram with its delta, considered as a 2-channel images of size 60×41. The first convolutional layer consisted in 80 filters of size 57×6, with a max-pooling of size 4×3. The second convolutional layer consisted in 80 filters of size 1×3, with a max-pooling of size 1×3. Finally, two fully connected layers composed of 5000 units each and a softmax layer ended the network.

Table 1. The recognition rates in % of the considered methods on the different datasets (A-Volute, ESC-50 and ESC-10).

	A-Volute	ESC-50	ESC-10
Parametric method	73.6	45.5	73.5
Non-parametric method	46.6	53.2	76.0
Our algorithm	96.5	94.0	96.0
Human	91.8	81.3	95.7

For each previous technique, the hyperparameters were adapted so as to have comparable results.

4.3. Human listening tests

Because a human score was available for the ESC dataset [19], we carried out a listening test on the A-Volute dataset. 21 participants were selected and had to classify 10 sounds from the 9 classes of the dataset. It gave a rough estimate of the human good classification rate on this dataset.

5. RESULTS

5.1. Recognition rate

The experiments were carried out for all the values of T. However, only the results for $T = 2048$ are shown because they are the best, and are presented in **Table 1**. The results for the A-Volute's database are the following. The parametric method achieves 73.6% and the non-parametric method 46.6%. Our algorithm outperforms the current methods by achieving a recognition rate of 96.5%. Concerning the ESC-50 database, the parametric method achieves 44%, the non-parametric method 53.2% and our algorithm 94.0%. Finally, on the ESC-10 dataset, the parametric method achieves 73.5%, the non-parametric method 76.0% and our algorithm 96.0%. It is worth noticing that a human can achieve 91.8% on the A-Vote dataset, 94.0% on the ESC-50 database and 95.7% on the ESC-10 database [19]. Our algorithm outperforms state-of-the-art methods and humans on these databases.

5.2. Complexity and execution time

We evaluate the complexity of the previous algorithms at the identification step. Our algorithm is mainly concerned with multiplication. Indeed, by computing the log-probability, computations are just additions and multiplications. All the operations needed to compute the decision for one buffer can be resumed as a $O(\#\{\mathcal{D}\}T/5)$. The parametric method relies mainly on computing exponentials and matrix operations (inversion and determinant), so the complexity is roughly $O(d^3)$ where d is the dimension of the feature vector. The neural network uses convolutions which are the more expensive operations when evaluating the decision. The complexity is thus $O\left(\sum_{l=1}^{L} n_{l-1}(s_l^1 s_l^2) n_l (m_l^1 m_l^2)\right)$, with L the number of layers, n_{l-1} the number of input channels for layer l, s_l^i the dimensions of the input, n_l the number of filters of the layer l and m_l^i the size of the output.

Considering the settings of the experiments, the numbers of operations needed for one buffer for our algorithm are $O(28 \cdot 10^6)$ (A-Volute database), $O(63 \cdot 10^6)$ (ESC-10) and $O(120 \cdot 10^6)$ (ESC-50), for the parametric method $O(2 \cdot 10^3)$ and for the neural network $O(14 \cdot 10^6)$.

We also report the time needed to compute the decision. The machine used a Intel®Core™i7-5820K CPU @3.30GHz. Using a dictionary composed of approximately 70,000 models, it takes 185ms to the system to infer the decision for one buffer. By using an NVidia GeForce®GTX 960 to compute the multiplications, this computational time can be reduced to 35ms.

6. DISCUSSION AND CONCLUSION

The aim of this algorithm is to perform real-time audio multi-source identification. Up to now, this study examined only the single-source case. As we can see in the results, our technique outperforms standard methods (GMM), more recent algorithms (DCNN) and even humans on both benchmark [19] and industrial databases. Despite these good recognition rates, the main advantage of our algorithm is that it can theoretically handle multi-source conditions. The learning algorithm of classic techniques would have to learn every combination of sound classes, which is practically impossible because of the combinatorial and the amount of data required for training. Some research on neural networks begins to study polyphonic identification, as in [20].

The energy spectrum is chosen so as to extend this technique to the multi-source setting. Indeed, we make the assumption that the additivity of two energy spectrums is preserved (uncorrelated sources). Mixture models were employed because they allow a flexible modeling of any signal. Moreover, in a multi-source setting, the signal model is a mixture of mixture, which is well defined in the mixture model framework.

The real-time constraint requires that the signals are not known by advances. Besides, we want the system to have a low latency. This is why such values of T and D were chosen. However, it is not reach yet. Indeed, the larger the dictionary the longer it takes to compute the decision. This is why future work will try to reduce the computation time, for instance by organizing the dictionary in a binary tree.

7. REFERENCES

[1] Cyril Joder, Slim Essid, and Gal Richard, "Temporal Integration for Audio Classification With Application to Musical Instrument Classification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 174–186, Jan. 2009.

[2] Gursimran Kour and Neha Mehan, "Music Genre Classification using MFCC, SVM and BPNN," *International Journal of Computer Applications*, vol. 112, no. 6, pp. 12–14, Feb. 2015.

[3] Dimitri Palaz, Mathew Magimai Doss, and Ronan Collobert, "Convolutional Neural Networks-based continuous speech recognition using raw speech signal," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 4295–4299.

[4] Sachin Chachada and C. C. Jay Kuo, "Environmental sound recognition: A survey," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, Oct. 2013, pp. 1–9.

[5] Geoffroy Peeters, Bruno L. Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams, "The Timbre Toolbox: extracting audio descriptors from musical signals," *The Journal of the Acoustical Society of America*, vol. 130, no. 5, pp. 2902–2916, Nov. 2011.

[6] Chlo Clavel, T. Ehrette, and G. Richard, "Events Detection for an Audio-Based Surveillance System," in *2005 IEEE International Conference on Multimedia and Expo*, July 2005, pp. 1306–1309.

[7] Jae-Hun Choi and Joon-Hyuk Chang, "On using acoustic environment classification for statistical model-based speech enhancement," *Speech Communication*, vol. 54, no. 3, pp. 477–490, Mar. 2012.

[8] Sbastien Lecomte, Rgis Lengell, Cdric Richard, Francois Capman, and Bertrand Ravera, "Abnormal events detection using unsupervised One-Class SVM - Application to audio surveillance and evaluation -," in *2011 8th IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Aug. 2011, pp. 124–129.

[9] Souli Sameh and Lachiri Zied, "On the Use of Time-Frequency Reassignment and SVM-Based Classifier for Audio Surveillance Applications," *International Journal of Image, Graphics and Signal Processing*, vol. 6, no. 12, pp. 17–25, Nov. 2014.

[10] Toni Heittola, Annamaria Mesaros, Antti Eronen, and Tuomas Virtanen, "Audio context recognition using audio event histograms," in *18th European Signal Processing Conference*, Aug. 2010, pp. 1272–1276.

[11] Alberto Bietti, Francis Bach, and Arshia Cont, "An online em algorithm in hidden (semi-)Markov models for audio segmentation and clustering," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 1881–1885.

[12] Robin Biondi, Gareth Dys, Gilles Ferone, Thibault Renard, and Morgan Zysman, "Low Cost Real Time Robust Identification of Impulsive Signals," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 8, no. 9, pp. 1653–1656, 2014.

[13] Cristina P. Dadula and Elmer P. Dadios, "Neural network classification for detecting abnormal events in a public transport vehicle," in *2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, Dec. 2015, pp. 1–6.

[14] Karol J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept. 2015, pp. 1–6.

[15] Geoffrey McLachlan and David Peel, *Finite Mixture Models*, Wiley, 2000.

[16] G. J. McLachlan and P. N. Jones, "Fitting Mixture Models to Grouped and Truncated Data via the EM Algorithm," *Biometrics*, vol. 44, no. 2, pp. 571–578, 1988.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[18] Gideon Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, Mar. 1978.

[19] Karol J. Piczak, "ESC: Dataset for Environmental Sound Classification," 2015, pp. 1015–1018, ACM Press.

[20] Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen, "Recurrent Neural Networks for Polyphonic Sound Event Detection in Real Life Recordings," *arXiv:1604.00861 [cs]*, Apr. 2016, arXiv: 1604.00861.