Comparing SVM, Softmax, and shallow neural networks for eating condition classification

Thomas Pellegrini[1]

[1]Université de Toulouse; UPS; IRIT; Toulouse, France

thomas.pellegrini@irit.fr

Abstract

This paper reports experiments on Eating Condition (EC) classification in the context of the INTERSPEECH 2015 Paralinguistic EC sub-challenge. Several techniques were compared: Support Vector Machines, Softmax classifiers and single hidden-layer neural nets using the ReLu activation function. Although eating noise and speech overlap in the recordings most of the time, performance improvements were obtained with all the tested techniques, by using the baseline features augmented with the same features but extracted on audio frames with low energy only. This led to a total of 12K features. With the Softmax classifier, for instance, UAR increased from 58.3% to 64.3% in the Leave-One-Speaker-Out (LOSO) cross-validation configuration. As expected, the 'Biscuit' and 'Crisp' categories benefited the most from using low-energy frames, with UAR improvements between 10% and 15% absolute. Indeed, these noises are high-frequency noises with low energy. SVM and Softmax showed similar performance, with Softmax slightly outperforming SVMs. Our best performance of 68.4% UAR on the test set was obtained by averaging the scores of several neural nets trained in the LOSO configuration. We also report a performance comparison of three different weight update rules used with batch gradient descent: the sgd, momentum and rmsprop rules.

Index Terms: Eating Condition classification, shallow neural networks, Computational Paralinguistics Challenge

1. Introduction

The *Eating Condition (EC) Sub-Challenge* consists in automatically identifying whether a speaker is eating and classifying which type of food is involved. Six food types are considered: Apple, Banana, Biscuit, Crisps, Haribo (gummy bears), and Nectarine. There is little research related to this very specific task as stated in the challenge description article [1]. Applications, such as in automatic speech recognition and life-logging to name a few, are listed in [2]. The topic is similar to the one of Audio Event Detection (AED), where one wants to detect the occurring of events in audio or video recordings with the help of audio cues only. The same approaches may be used in EC and AED classification, with basically two main modules: one for acoustic feature extraction and one for audio event inference. Nevertheless, there are differences in the goals. By audio events, people usually mean non-speech events such as vehicles, animals or gun shots. Also, eating condition impacts speech production, hence we are interested in characterizing speech while eating. AED's goal is to detect but also to locate events in an audio stream. Most of the AED systems take decisions at frame-level. On the contrary, in the present challenge, a decision is taken at file-level and no notion of time is involved.

A variety of techniques were used in the literature related to AED: Gaussian mixture models for scream and gun shot detection [3], Support Vector Machines (SVM) for a hundred audio concept detection [4], hidden Markov models in the context of the CLEAR evaluation [5]. There is no optimal solution for AED, and authors often focus on specific cases. Other interesting approaches try to model acoustic events with atomic units of sound learned automatically such as audio unit descriptors [6] or spectro-temporal patch bases [7]. In this work, it seems more appropriate to use the first kind of AED approaches, i.e. acoustic feature extraction and classification, since we do not want to detect specific events but rather characterize speech altered by food consumption.

This paper is organized as follows. After briefly describing the challenge corpus and setup, Section 3 describes the methodology adopted in this work. Then, the addition of features extracted on low-energy frames is justified and presented. Section 6 reports ans discusses our results. Finally, conclusions and directions of future work are given.

2. Challenge material and setup overview

Participants were provided with speech audio files, two text files with train and test feature sets in the ARFF format from the WEKA platform, and several Bash and Perl scripts to reproduce experiments with a baseline system using SVM, more precisely the WEKA Sequential Minimal Optimization (SMO) implementation. Each audio file contains one or several utterances, with an average duration of 7.1s (STD=2.8s). The ARFF files contain 6,373 features per audio file that were extracted by the organizers with the OpenSMILE toolbox. The training and test sets are comprised of 945 and 469 utterances recorded by 20 and 10 speakers, respectively. Since no separate validation set was provided, prototyping tests are done with Leave-One-Speaker-Out cross-validation on the training set (LOSO-CV). To compete, participants were allowed to submit up to 5 trials on the challenge Website, in the form of an ARFF file with a list of predictions. Performance is evaluated in terms of Unweighted Average Recall (UAR). Baseline performance is 65.9% UAR. For more details on the corpus and baseline results, the reader may refer to [1].

3. Methodology

Similarly to AED systems, two main modules are needed: one for feature extraction eventually followed by feature selection, and one for EC inference. First, attention was paid to the acoustic features. As it will be described in the next section, the original feature set size was doubled, from 6K to 12K features, by extracting the baseline features on low-energy frames only, with the idea that eating noises do not always overlap speech. Since

classification decisions are based on single feature vectors, the machine learning techniques to be used do not need to have sequence modelling capabilities. For this reason, we restricted our tests to Softmax, SVM and artificial neural networks (ANN). Most of the preliminary tests were done with Softmax since it is fast and also because it can be seen as a single artificial neurone. We could have used SVM instead but our main objective was to gain expertise and practice in ANN.

An in-house implementation of a Softmax classifier was used. For SVM (SMO), we used the WEKA scripts provided by the organizers. For ANN, a code skeleton available from the Stanford cs231n course (http://cs231n.github.io/) was completed to test several architectures. Since the corpus is small in terms of the number of training examples (less than 10,000 cases), we used batch gradient descent to train the ANN. Two stopping criteria were tested: 'early-stopping', and 'best-LOSO'. 'Early-stopping' consists in stopping the training iterations when the cost function variation between two iterations is below a given threshold (1e-6). With the 'best-LOSO' option, training is stopped when a maximum number of iterations is reached (set to 100 epochs). The resulting model is the one that gave the best accuracy on a validation data subset no matter the iteration during which it was achieved. We also tested three weight update rules used during backpropagation: the standard, momentum and rmsprop rules. Finally, in order to improve performance, late fusion was applied by averaging prediction probabilities emitted with several models.

4. Feature extraction

4.1. Low-energy frames

The 6K feature set comprises first and second order statistical moments of acoustic features such as filter bank coefficients, MFCCs, harmonic to noise ratio, zero-crossing rate, etc. In [1], the organizers reported that they removed isolated eating noises from the recordings, otherwise the challenge would have been too easy. Nevertheless, there are almost always audio segments left with breathing, chewing, or biting noises that do not overlap with speech. Hence, we thought that detecting non-speech frames and extracting acoustic features on them could bring additional information. The extraction of low-energy frames was done by using a file-specific threshold equal to 10% times the median energy value. This threshold was empirically set by verifying that enough frames were selected in order to compute features on them. Non-overlapping frames of 64ms duration were selected if their energy was inferior to this threshold. They were concatenated to generate low-energy WAV files used in feature extraction. Then, the same features as the baseline ones were extracted, by using OpenSMILE and the 'IS13_ComParE.conf' configuration file. The duration of the original and low-energy training WAV files were 1h53min and 1h11min, respectively.

4.2. Feature selection

Features were standardized (zero-mean and unit-variance normalized). An epsilon value (2e-16) was added to the feature STD values to prevent zero-divisions as some features were constant over all the training examples. We could also have simply removed such features but both ways led to the same classification results as SMO and Softmax are not sensitive to non-informative features (it may slow down the converge time though). The mean and STD values obtained on the train subset were used to normalize the test set features.

Two feature selection techniques were tested: Correlation

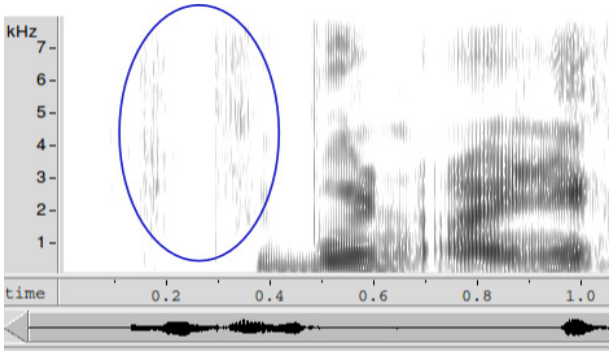

Figure 1: *Excerpt of the spectrogram of a 'Biscuit' audio file. The ellipsis denotes a part corresponding to biscuit eating noise with no overlapped speech.*

Feature Selection (CFS) and Singular Value Decomposition (SVD). CFS roughly consists in selecting features that have discriminant power relatively to the class labels. With CFS, feature dimension reduced to 327 and 533 features for the 6K and 12K feature sets, respectively.

With SVD, features are projected on a lower dimension subspace defined by the so-called right singular vectors. The dimension of the subspace is chosen by selecting the singular vectors that are associated to the largest singular values. With a threshold of 95% that corresponds to the percentage of energy ($\sum_i \sigma_i^2$ with σ_i the i^{th} singular value) kept when truncating the projection matrix, dimension reduced from 6K and 12K to 260 and 500, respectively.

5. ANN description

Several ANN architectures were tested. The one that worked best was a two-layer neural network with a single hidden layer with rectified linear activation functions (ReLu). This activation function is popular, the main reason being that it is not as prone to saturate as the Softmax or tanh functions are (output constant activations and almost-zero valued gradient) [8]. ANN with two hidden layers were tested but no improvement was achieved compared to using a single hidden layer. In terms of number of neurons, we used 20 units in the hidden layer, as using less units decreased accuracy and using more did not help (we tested up to 200 neurones). The 'normalized initialisation' proposed in [8] was used to initialize the weights:

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n^l + n^{l+1}}}, \frac{\sqrt{6}}{\sqrt{n^l + n^{l+1}}}\right]$$

where n^l and n^{l+1} are the number of units in consecutive layers l and $l+1$. Bias terms were 0-initialized.

The Softmax negative log-likelihood was used as loss function:

$$\mathcal{E}(W, D) = -\frac{1}{|D|}\sum_{i=1}^{|D|} \log(P(Y = y^{(i)}|x^{(i)}, W))$$

with $P(Y = y^{(i)}|x^{(i)}, W) = \text{softmax}_{y^{(i)}}(Wx^{(i)})$

An L_2 regularization term was added to this data loss term.

As mentioned earlier, batch gradient descent was used to train the networks, and gradients were calculated using the backpropagation algorithm. In this study, three popular weight update rules were compared:

- Standard 'SGD' rule. With this simple rule, weights are updated by adding to them the negative gradient of the loss function multiplied by a learning rate η:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \mathcal{E}(h_w(\mathbf{x}, y))$$

- Classical 'Momentum' rule [9]. Instead of updating the weights directly, the gradient is used to update the velocity v with which the weights are updated:

$$\mathbf{v} \leftarrow \mu \mathbf{v} - \eta \nabla \mathcal{E}(h_w(\mathbf{x}, y))$$
$$\mathbf{w} \leftarrow \mathbf{w} + \mathbf{v}$$

  where μ is an additional hyperparameter called 'momentum', with values within the $[0, 1]$ interval. A value of 0.9 is typically used. The name of this rule comes from the idea that gradient descent gains momentum when gradient directions persist across iterations [10]. It allows to pass a plateau faster than with the 'SGD' method.

- 'RMSprop' rule. This method keeps a running average of the squared gradient for each weight [11]. It is used to equally scale the gradient updating term among units:

$$\text{mSquare} \leftarrow 0.9\text{mSquare} + 0.1(\nabla \mathcal{E}(h_w \mathbf{x}, y))^2$$
$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla \mathcal{E}(h_w(\mathbf{x}, y)) / \sqrt{\text{mSquare}}$$

6. Results

All the results discussed in this section are reported in Table 1.

6.1. Impact of low-energy-based features

Performance was compared when using the baseline 6K and the augmented 12K feature sets, with SMO and Softmax. The first two rows of Table 1 named 'SMO-6K' and 'Softmax-6K' are two results obtained with the 6K baseline features. All the other results were achieved with the augmented feature set of 12K features. Significant performance improvements were obtained with both SMO and Softmax on LOSO-CV: 2.0% and 6.0% absolute, respectively. The optimal SMO complexity hyperparameter was $C = 10^{-3}$ in all cases. The optimal weight decay parameter for Softmax slightly increased from $4.0 * 10^{-3}$ with 6K to $6.6 * 10^{-3}$ with 12K features, which could be due to the fact that with more features, more regularization is needed. In the literature, Softmax and SVM are reported to achieve similar performance in general. It is interesting to observe, here, that Softmax outperformed SMO by 1.0% absolute when using 12K features, whereas it was the contrary with 6K features.

The example illustrated in Figure 1 was misclassified as Apple with 6K features, but it was well classified as Biscuit with 12K features. In this case, Apple and Biscuit probabilities changed from 0.550 and 0.314 to 0.064 and 0.770, respectively. Figure 2 is a bar plot illustrating recall values for each class when using 6K or 12K features. As one can observe on the graph, all the categories benefited from augmenting the feature set. The 'Biscuit' and 'Crisp' categories benefited the most from using low-energy frames, with UAR improvements between 10% and 15% absolute. Indeed, these noises are high-frequency noises with low energy that are well separated from speech when extracting low-energy frames.

Concerning feature selection, neither SVD nor CFS brought improvement with Softmax. SVD with an energy threshold of 95% achieved the same performance as the one obtained with all the features. CFS was carried out in a 6-fold CV mode on the training subset and the resulting 533 features led to a 64.1% UAR. For this reason, we decided to keep using all the 12K features in the remaining experiments.

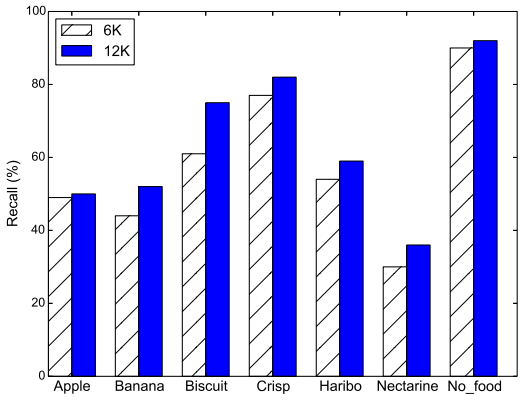

Figure 2: *Comparison of recalls obtained with 6K and 12K features.*

Table 1: Mean UAR(%) per speaker

Classifier	LOSO-CV	Test
SMO-6K (baseline)	61.3	65.9
Softmax-6K	58.3	_
SMO	63.3	_
Softmax	64.3	66.8
ANN-SGD	62.5	_
best-ANN-SGD	63.7	_
ANN-Momentum	64.5	65.6
best-ANN-Momentum	68.6	_
ANN-RMSprop	62.2	_
best-ANN-RMSprop	66.8	_
ANN-Momentum (20)	_	68.4
Best-ANN-Momentum (20)	_	67.6

6.2. Comparison between techniques

Since Softmax outperformed SMO with 12K features in cross-validation, a test trial was submitted with this configuration. A 66.8% UAR was obtained, slightly better than the 65.9% baseline UAR. All the remaining rows in the table report performance of different ANN configurations. The rows beginning with 'ANN' and 'best-ANN' correspond to the 'early-stopping' and 'best-LOSO' stopping criteria, respectively. Since they were chosen for this, the 'best-ANN' systems always show better LOSO-CV performance than their 'early-stopping' counterparts: 63.7% versus 62.5% for the ANN trained with the 'sgd' update rule, for instance. The best LOSO-CV performance was obtained by the ANN using the Momentum update rule, with a 68.6% UAR. An ANN using Momentum was trained on all the training data (no CV) with the 'early-stopping' criteria. This model achieved a 94.0% accuracy on the training data. A test trial with this model was submitted. A 65.6% performance revealed smaller than the Softmax and the baseline ones. This indicates a lack of generalization power due to overfitting, in other words the model suffers from high-variance. One way to limit overfitting is to increase weight regularization, nevertheless, LOSO-CV performance decreases and it is hard to set the level of regularization needed without a separate validation

classified as →	Ap	Ba	Bi	Cr	Ha	Ne	No
Apple	30	2	6	2	5	11	0
Banana	2	39	0	1	10	8	10
Biscuit	6	1	58	3	1	1	0
Crisp	1	1	10	53	2	1	2
Haribo	2	11	1	2	42	3	9
Nectarine	13	7	3	1	3	35	1
No_Food	1	1	0	0	1	0	67

Table 2: Confusion matrix obtained on the test set with the best system (UAR=68.4%).

set. Twenty ANNs can be obtained during the LOSO-CV training phase, with data from 19 speakers to train and data from the left-out speaker to tune each single model. After noticing that the predictions obtained with these ANNs differ in 10% of the test samples, it appeared natural to combine them in some way. The prediction discrepancy concerned cases in which the winning class had a low probability, often below 0.5. It happened mainly with the fruit classes for which confusions were frequent (Apple and Nectarine, typically). To combine the output of the 20 ANN trained with Momentum, prediction probabilities were averaged. As reported in the last two rows of the table, the best test performance was 68.4% UAR, 2.5% absolute above the baseline UAR, when using the 20 ANNs trained with the 'early stopping' criterion. It is interesting to observe that the 20 best-ANNs-Momentum performed slightly worse, with a 67.6% UAR. Once again, it is likely due to overfitting. Table 2 shows the confusion matrix obtained on the test set with the best configuration. As with the other techniques, the most frequent confusions occur between fruit classes: Apple and Nectarine, Haribo and Banana, and Banana with No_Food.

6.3. Comparison between update rules

In this section, details are given about the use of different weight update rules. Figure 3 shows the loss, train and cross-validation accuracy evolutions for the first 50 training epochs for a single speaker ('Prob01'). The sgd and momentum loss curves converge rapidly toward the same cost, whereas the rmsprop one converges more slowly. The train and validation accuracies (equal to UAR for a single speaker) are very similar. The hyperparameters of the different update methods were tuned for this speaker in order to obtain these curves. As one can expect, the tuned parameters are not necessarily the same for all the speakers as it is shown in Figure 4. This Figure illustrates the best validation accuracy obtained during CV for the 20 speakers, indicated from 1 to 20 on the X-axis and ordered according to a decreasing accuracy obtained with the baseline SMO classifier. For a given update rule, the same rule-specific hyperparameter values were used for all the speakers. The three curves are similar, but the Momentum curve with '*' points is almost always above the other ones.

7. Conclusions

In this paper, we described experiments carried out in the context of the Eating Condition classification challenge of INTERSPEECH 2015. The difficulty of the task lies in the fact that eating noise overlap speech most of the time. By listening to a few recordings, it clearly appeared that eating while speaking greatly impacts speech production and in different ways according to the type of food that is consumed. The 65.9% UAR

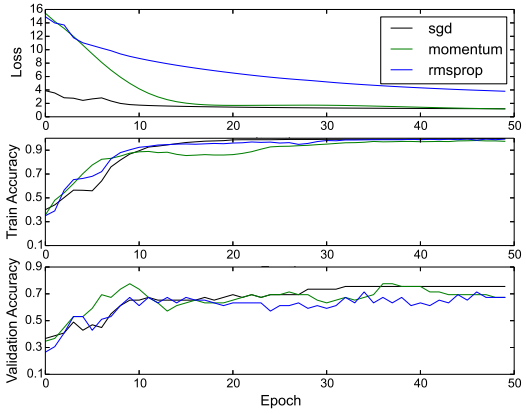

Figure 3: *Evolution of the loss, training and validation accuracy with the three different weight update rules for the first 50 training iterations.*

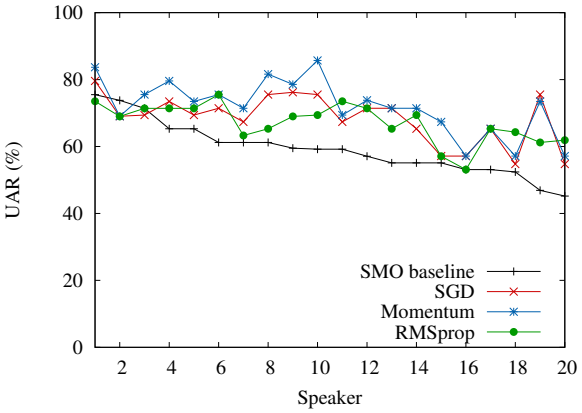

Figure 4: *LOSO-CV UAR for 20 speakers obtained with SMO and ANN with three different update rules: 'sgd', 'momentum', 'rmsprop'.*

performance of the baseline system provided by the organizers revealed difficult to beat. Improvements were achieved by augmenting the feature set with acoustic features extracted on low-energy frames. Although eating noise and speech overlap most of the time, low-energy frames are likely to contain eating noise, if any. Using these extra features led to a 2.0% absolute gain with the SMO baseline classifier in the 20-fold LOSO cross-validation setup. Using these features and a Softmax classifier led to a small gain of 0.9% absolute on the test set. Several types of neural networks were then tested and despite significant gains obtained in cross-validation experiments, no improvement on the test set was achieved due to overfitting. Late fusion of prediction probabilities obtained with several ANNs allowed to achieve a 68.4% UAR on the test set, corresponding to a 2.5% absolute gain over the baseline result. Among three popular weight update rules, the Momentum rule proved to be the best one. We plan to explore more sophisticated fusion methods, namely discriminative score calibration/fusion and to apply it to combine all the techniques that were used in this work.

8. References

[1] B. Schuller, S. Steidl, A. Batliner, S. Hantke, F. Honig, J. Orozco-Arroyave, E. Noth, Y. Zhang, and F. Weninger, "The INTERSPEECH 2015 Computational Paralinguistics Challenge: Nativeness, Parkinsons & Eating Condition," in *To appear in Proc. Interspeech*, Dresden, 2015.

[2] S. Hantke, F. Weninger, R. Kurle, A. Batliner, and B. Schuller, "I hear you eat and speak: automatic recognition of eating condition and food type," *to appear*, 2015.

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

[4] V. Barbosa, T. Pellegrini, M. Bugalho, and I. Trancoso, "Browsing videos by automatically detected audio events," in *Proc. EUROCON*, April 2011, pp. 1–4.

[5] X. Zhou, X. Zhuang, M. Liu, H. Tang, M. Hasegawa-Johnson, and T. Huang, "HMM-based acoustic event detection with adaboost feature selection," in *Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR 2007 and RT 2007*, Berlin, Heidelberg, 2008, p. 345353.

[6] A. Kumar, P. Dighe, R. Singh, S. Chaudhuri, and B. Raj, "Audio event detection from acoustic unit occurrence patterns," in *Proc. ICASSP*, March 2012, pp. 489–492.

[7] C. V. Cotton and D. P. W. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2011, New Paltz, NY, USA, October 16-19, 2011*, 2011, pp. 69–72.

[8] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc AISTATS, Society for Artificial Intelligence and Statistics*, 2010.

[9] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4(5), pp. 1–17, 1964.

[10] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. ICML*, vol. 28, Atlanta, 2013, p. 11391147.

[11] G. Hinton, N. Srivastava, and K. Swersky, "Lecture 6: Overview of mini-batch gradient descent," Coursera Lecture slides https://class.coursera.org/neuralnets-2012-001/lecture, [Online; accessed 20-March-2015].