Acta Universitatis Sapientiae
Electrical and Mechanical Engineering, 6 (2014) 47–58

Estimating the Level of Conflict Based on Audio Information Using Inverse Distance Weighting

Gábor GOSZTOLYA

MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary,
e-mail: ggabor@inf.u-szeged.hu

Manuscript received July 30, 2015; revised October 19, 2015.

Abstract: In recent years it has become possible to extract non-trivial information from audio sources. One such task is to determine the intensity of conflicts arising in speech recordings, based solely on audio information sources. This intensity is expressed as a real number, therefore this task is essentially a regression one, the objective being to estimate a given numeric score. As the number of examples in these tasks are limited, a kNN-like solution may work well in these problems. Such an approach is the Inverse Distance Weighting (IDW) algorithm, which is also a suitable choice as it is computationally cheap. By applying this method on the conflict intensity estimation task using the SSPNet Conflict Corpus, we were able to reach the level of performance of baseline SVM.

Keywords: speech technology, conflict detection, regression, KNN, inverse distance weighting.

1. Introduction

In the past, within the field of speech technology, most of the researchers' efforts were devoted to speech recognition. But in recent years they have turned their attention to other areas as well like emotion detection [25, 10], speaker verification [17], speaker age estimation [5], detecting social signals like laughter and filler events [1, 10, 12], and estimating the amount of physical or cognitive load during speaking [20, 11, 14]. What these tasks have in common is that what is considered noise in speech recognition (i.e. non-verbal audio information) becomes important, while what was relevant in speech recognition (i.e. what the speaker actually said) becomes irrelevant.

Such a task is to determine the level of conflict from the audio. Conflicts influence the everyday lives of people to a significant extent, either in their public or personal lives, and they are one of the main causes of stress [23]. With the rise of socially intelligent technologies, the automatic detection of conflicts can be the first step of handling them properly.

In this study we focus on the automatic estimation of the level of conflict in televised political debates. This is mainly a regression task [2], i.e. we have to match a score as closely as possible, as the level of conflict is expressed as one numerical value. Of course, from an application point of view, a categorical approach looks more practical, where the question is whether there a conflict present or not, and if so, we want to know what its level is. This in fact means that the task is turned into a classification one [6]. However, this categorization may be readily performed by setting up intervals for the conflict score; therefore we approached this task mainly from a regression point of view.

Although such recordings can be obtained quite easily, their annotation can be rather expensive; hence it is preferable to use a machine learning method that works well for small-sized training sets. One such algorithm for classification is the K Nearest Neighbours method (*kNN*), where the label of the given utterance to be classified is determined by simple majority voting of its K nearest neighbours. Of course, the distance function used and the value of K have to be determined, but these are not major requirements (especially when compared to the parameters of other machine learning methods like Artificial Neural Networks (ANNs) [3] and Support Vector Machines (SVM) [19, 24]).

Another advantage of this method is its low computational cost if both the train and test sets consist of just a small number (e.g. hundreds) of examples − especially when compared to high-complexity approaches like SVM and AdaBoost [18, 4]. A similar approach for regression is Inverse Distance Weighting (IDW) [22]. In it, the function value of a given point is calculated by computing the weighted sum of the function value of the training points, where the weight of a training point is inversely proportional to its distance from the point to be evaluated.

The structure of this paper is as follows. First, we describe the audio corpus used for conflict intensity estimation, and the evaluation methodologies. Then we describe the original and an improved version of the IDW algorithm. After, we explain the slight modifications made that we felt necessary to use IDW for this task. Then we present and analyse our results got from applying them on the development and test sets. Lastly, we draw some conclusions and make some suggestions for future study.

2. The SSPNet Conflict Corpus

We performed our experiments on the (freely accessible) SSPNet Conflict Corpus [15]. It contains recordings of Swiss French political debates taken from the TV channel “Canal9”. It consists of 1430 recordings, 30 seconds each, making a total of 11 hours and 55 minutes. The ground truth level of conflicts was determined by manual annotation performed by volunteers not understanding French (French-speaking people were excluded from the list of annotators). Each 30-second long clip was tagged by 10 annotators, and in the end we got a score in the range [-10, 10], 10 meaning a high level of conflict and -10 meaning no conflict at all. The data was later used in the Conflict sub-challenge of the Interspeech 2013 ComParE Challenge [21].

The database contains both audio and video recordings, and the annotators were able to rely on both sources. In the latter experiments, however, attention was focused only on the audio information for a number of reasons. Firstly, the annotators judged the level of conflict in a similar way based on the two sources: the correlation of the scores was 0.95 [15]. Furthermore, in a television political debate, audio can be a more reliable indicator: the subjects can hear all the participants, but they can only see the one that the cameraman of the debate has chosen, which is not the one speaking in many cases (especially in the heat of a debate when several persons may be speaking at the same time).

3. Inverse Distance Weighting

Inverse Distance Weighting (IDW) was introduced by Shepard in 1968, originally for interpolating surfaces from irregularly-spaced data [22]. Later it was used for other interpolation tasks as well [9, 26]. This method (sometimes called “Shepard's algorithm”) estimates the target score of a given point by the weighted sum of the input scores, and the weight of a training point is inversely proportional to its distance. Given a set of sample points $x_1, \dots, x_N$, score values $f_1, \dots, f_N$ and a distance function $d(x,y)$, for a point $y \neq x_i$, its score $F(y)$ will be

$$F(y) = \sum_{i=1}^{N} w_i f_i, \tag{1}$$

where w_i is the weight of the ith input point. It is defined by

$$w_i = \frac{d(x_i, y)^{-c}}{\sum_{j=1}^{N} d(x_j, y)^{-c}}, \tag{2}$$

where $c > 0$. Inserting this into Eq. (1) we get

$$F(y) = \frac{\sum_{i=1}^{N} d(x_i, y)^{-c} f_i}{\sum_{i=1}^{N} d(x_i, y)^{-c}}. \tag{3}$$

The value of c regulates the relative importance of closer and more distant points: for larger values of c, the closer points are more important, while using smaller values of c tends to equalize the weights. It is a global method in the sense that to determine the score of a test example, all training points are used, no matter how far away they are. A simple extension to make this method local was suggested by Franke and Nielson [8], who introduced the limiting parameter R. Their formula for determining the weights is

$$w_i = \frac{\left(\frac{\left(R - d\left(x_i, y\right)\right)_+}{Rd\left(x_i, y\right)} \right)^c}{\sum_{j=1}^{N} \left(\frac{\left(R - d\left(x_j, y\right)\right)_+}{Rd\left(x_j, y\right)} \right)^c}, \tag{4}$$

where $(v)_+$ denotes max(v, 0).

4. Experimental setup

Speech recognition usually decomposes the speech signal of an utterance into small-equal sized parts (*frames*), from which it is easy to extract the same number of features for machine learning. In the current task, however, we have to estimate the level of conflict for the whole 30 second-long utterance, therefore features which describe the whole recording are preferred. A straightforward choice is to compute the standard features (e.g. MFCC and filter banks) for each frame, then calculate the minimum, maximum, mean and standard deviation of these values.

In our experiments we used the feature set introduced in [21]. It contained 6373 features overall, extracted by using the tool openSMILE [7]. The set includes energy, spectral, cepstral (MFCC) and voicing-related low-level descriptors (LLDs) as well as a few LLDs including logarithmic harmonic-to-noise ratio (HNR), spectral harmonicity and psychoacoustic spectral sharpness. Of course, as this is a quite general feature set, not all attributes are useful for our current task; now, however, we focused on the application of IDW, and did not experiment with any kind of feature selection.

Following standard machine learning practice, the available data was split into training, development and test sets. The first one was used for training purposes, i.e. IDW estimation was done using the points belonging to this set. The development set was used to find the meta-parameters of the learning algorithm, i.e. *c* and *R* by choosing the values which led to the best results by training on the training set and evaluating on the development one. Next, using the “optimal” *c* and *R* values, we evaluated our model on the test set; in this case we used the points of both the training and development sets as training points. We used the division described in [21], so 793 recordings were used for model training, whereas 240 and 397 were used for the development and test sets, respectively.

A straightforward choice for measuring the similarity of the reference and the estimated values is cross-correlation. For the signals $X = x_1, \ldots, x_n$, and $Y = y_1, \ldots, y_n$, it is defined as

$$CC(x,y) = \frac{\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)}{N\sigma_X \sigma_Y}, \tag{5}$$

where μ_X and μ_Y are the mean and σ_X and σ_Y are the standard deviation values of *X* and *Y*, respectively. Another choice for measuring the difference between the two series is the Root-Mean-Square Error (RMSE), defined as

$$RMSE(x,y) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}}. \tag{6}$$

While cross-correlation measures the tendency of the two signals, RMSE measures the actual difference between the values; this means that in a regression task it may be sensitive to the scaling of results.

Another possibility is to turn this task into a classification one. We also carried out experiments for this, following the setup described in [21], where non-negative conflict scores were considered as *high* ones, while negative ones were converted into the class label *low*. Methods applied on such two-class classification problems can be measured by a number of metrics, all of which are based on the values of the confusion matrix. There, T_P will be the number of true positives (i.e. the occurrences of class *high* that were classified correctly) and F_P the number of false positives (the *low* occurrences classified as *high*), while the values T_N (true negatives) and F_N (false negatives) are defined in a similar way. (The sum of the four values will be *n*.) Then accuracy will simply be the ratio of correctly classified examples, i.e.

$$Accuracy = \frac{T_P + T_N}{n}. \tag{7}$$

If we treat our task as an information retrieval one, meaning that we are interested in the detection of occurrences of the positive class only (in our case, class *high*), we can measure our performance by means of precision and recall. Precision measures how many of the identified examples actually belonged to this class, i.e.

$$\Pr ecision = \frac{T_P}{T_P + F_P}, \tag{8}$$

whereas recall expresses how many of the examples actually belonging to the positive class were found; i.e.

$$\mathrm{Re}\, call = \frac{T_P}{T_P + F_N}. \tag{9}$$

As there is clearly a tradeoff between these two scores, they are usually aggregated via F-measure (or F_1-score), defined as the harmonic mean of the two values, i.e.

$$F_1 = \frac{2 \cdot \Pr ecision \cdot \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call} = \frac{2 \cdot T_P}{2 \cdot T_P + F_N + F_P}. \tag{10}$$

Using the concept of recall, we can define another variant of accuracy, namely the Unweighted Average Recall (UAR) or True Positive Rate (TPR), expressed as the mean of the recall values for all the classes. In a two-class set-up it is equal to

$$UAR = \frac{\frac{T_P}{T_P + F_N} + \frac{T_N}{T_N + F_P}}{2}. \tag{11}$$

Accuracy is sensitive to class distribution, whereas UAR can be viewed as an accuracy which is balanced class-wise. For this task and this dataset in the past, regression metrics (especially cross-correlation) were used [15], and we also find this approach more logical, so we will follow this in our study. However, we will also view the task as a classification one, where we will primarily rely on the UAR score, just as it was common in some earlier studies on this dataset [21, 16].

5. Applying IDW for estimating the conflict scores

Shepard's algorithm and Franke’s modified version were developed for generating surfaces based on sparsely distributed input points in a two-dimensional space and a function value. In a large-dimension regression task they might require some minor changes in order to perform well (and in our case there were 6373 features). To achieve this, we included some minor pre-processing and post-processing steps, which we will now describe in detail.

First, we used the Euclidean distance metric; that is, for two points $y = y_1, y_2, \ldots, y_n$ and $z = z_1, z_2, \ldots, z_n$, their distance $d(y,z)$ was simply

$$d(y,z) = \sqrt{\sum_{i=1}^{n} (y_i - z_i)^2} \tag{12}$$

and in our preliminary tests we found that applying other distance functions yielded somewhat worse results. To prevent confusion caused by differently-scaled features (where a few of them might dominate the distance, whereas other, perhaps more important attributes might simply be ignored because of initial scaling), feature normalization was clearly required. For this reason, first all the vectors were normalized so that they had a standard deviation of 1. A couple of features had a standard deviation of 0, which were discarded, but this step clearly did not lead to any information loss (as it meant that the value of these features was the same for all examples).

After performing the IDW procedure, the resulting values were quite small compared to the real ones, perhaps because of the high dimensionality of the input data. To handle this issue, the resulting scores were also normalized: the mean was set to zero, and they were multiplied by a factor such that the standard deviation of the results became equal to the one of the scores of the training set. Next, scores falling below or above the limits of the scores of the training set were set to the minimum or maximum score, respectively, and each value was rounded to one decimal place.

Franke’s method has two parameters, namely c and the limit value R. As for the latter, we decided to express it via the function of $maxd = \max(d(x,y))$ for all possible values of x and y (of the training set); that is, $R = r \cdot maxd$. Eventually when $r = 1$, all the training points were considered, whereas for lower values the more distant points were ignored. When no training points were found in the R-sized neighbourhood, the conflict score of the closest training point was used. We optimized the parameters cross-correlation, for UAR and for F-measure; we used linear SVM in regression (the SMOReg method in Weka [13]) mode as the baseline. (Note that this method was used as the baseline approach for ComParE

2013 [21]; the only difference is that the *c* parameter was tuned to maximize UAR, while we optimized CC as well).

5. Results

The results when optimizing for cross-correlation can be seen in Table 1.

Table 1: Scores obtained by optimizing for cross-correlation.

Method		CC	RMSE	Acc.	UAR	F_1
dev	IDW, c = 13.56	0.805	2.390	80.83%	80.67%	79.28%
	IDW, r = 0.15	0.816	2.314	81.67%	81.46%	80.00%
	SVM	0.828	2.427	74.58%	73.40%	66.30%
test	IDW, c = 13.56	0.782	2.654	80.60%	80.47%	77.94%
	IDW, r = 0.15	0.768	2.727	79.35%	79.23%	76.57%
	SVM	0.804	2.414	83.63%	82.35%	79.37%

Here, IDW achieved practically the same level of performance as SVM for all the metrics on the development set; Franke's method was somewhat better than the basic IDW algorithm. On the test set, however, the standard IDW method proved to be more stable, and Franke's variation (case $r = 0.15$) showed signs of overfitting. Shepard's method performed slightly worse than the baseline SVM, but the difference is not that big.

Table 2: Scores obtained by optimizing for UAR.

Method		CC	RMSE	Acc.	UAR	F_1
dev	IDW, c = 7.22	0.801	2.430	82.08%	81.95%	80.72%
	IDW, r = 0.69	0.808	2.383	82.50%	82.39%	81.25%
	SVM	0.806	2.330	80.42%	79.55%	75.65%
test	IDW, c = 7.22	0.775	2.702	79.09%	79.29%	76.88%
	IDW, r = 0.69	0.765	2.725	80.86%	80.55%	77.91%
	SVM	0.826	2.271	84.64%	83.87%	81.46%

Upon examining the classification results (see Table 2), it can be seen that the IDW classification performance significantly exceeded that of SVM for the development set in its basic form, and using the variation developed by Franke and Nielsen (case r = 0.69) even surpassed this. (This variant also performed better judging from the regression scores.) However, on the test set the best variation with r = 0.69 performed slightly worse than the baseline SVM, although the difference is again not that big. Still, in our opinion even matching the score of the SVM is a good result for an algorithm that has such low computational requirements as IDW.

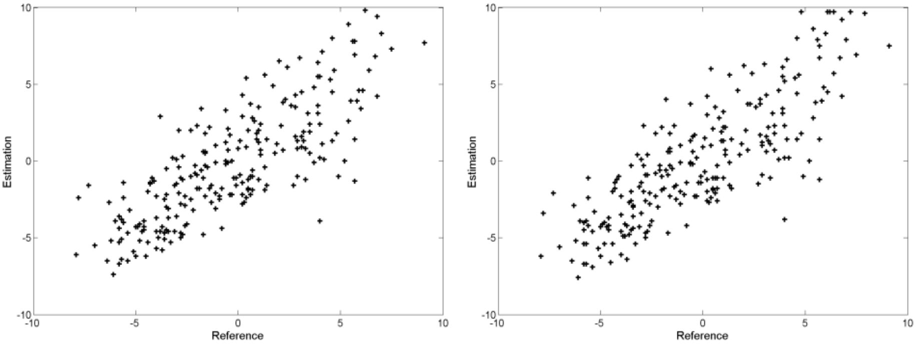

Figure 1: The estimated scores got as a function of the reference values, using IDW optimized for cross-correlation; Shephard's (left) and Franke's (right) methods.

Fig. 1 shows the regression scores in the function of the reference scores for the development set, obtained using the IDW algorithm with c = 13.56 (Shephard's method, left) and with $r = 0.15$ and $c = 7.68$ (Franke's algorithm, right). The strong correlation between the two values can clearly be seen; overall, the points produced by Franke's method seem a bit more packed, which is confirmed by both the higher CC and lower RMSE scores. It is understandable, though, as in this case we had one more parameter to set.

Fig. 2 shows the corresponding scores we got with the value c = 7.22 (Shephard's method, left) and with $r = 0.69$ and $c = 5.44$ (Franke's algorithm, right). This time we optimized for the UAR score, which is reflected in the lower cross-correlation value, resulting in somewhat more scattered points. The reason for this is that UAR only measures which point falls into which quarter of the chart (i.e. both the reference and the estimated scores are non-negative, both are negative, etc.), while the actual difference between the expected and the estimated scores is completely ignored.

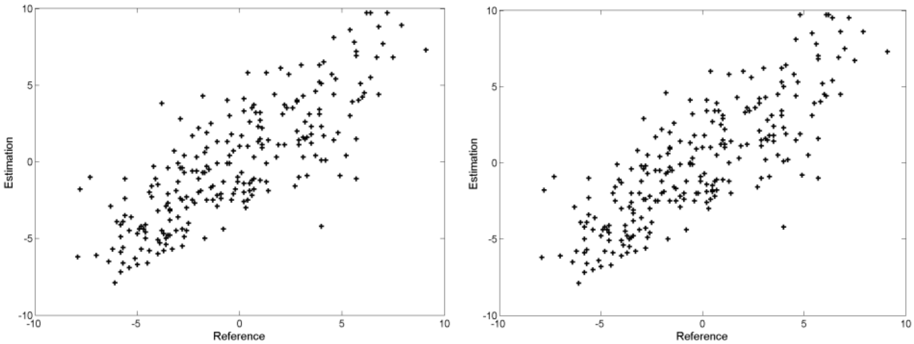

Figure 2: The estimated scores got as a function of the reference values, using IDW optimized for UAR; Shephard's (left) and Franke's (right) methods.

An interesting observation is that the optimal *c* values for Shepard's method were somewhat higher (13.56 and 7.2) than those of Franke's algorithm (7.68 and 5.44). This might be because for such a regression task training points which fall closer should be more important than those further away; this can be realized in the basic IDW method by using high values of c. When using the version developed by Franke and Nielsen, however, we can simply do this by choosing the right R value; then *c* can be set to a lower value as well.

Finally we should note that there were higher accuracy scores among the participants of ComParE 2013. (Although the cross-correlation scores were not always reported, since the official metric of the Challenge was UAR even for this regression task.) The more successful attempts, however, performed some kind of feature selection [16] or extracted new features from the utterances [12], while in this study we applied a different machine learning method for the regression task of conflict score estimation. Of course, it could be beneficial to use some kind of feature selection method for IDW as well, but this is clearly the subject of future work.

6. Conclusions

Regression tasks are quite rare in speech technology, but one exception is the detection of the intensity of conflicts based on speech recordings. We applied the Inverse Distance Weighting method to this task, which was originally developed for estimating surfaces on the basis of just a few sparsely and unevenly distributed reference points. After making a few minor alterations, this method outperformed the baseline SVM in terms of classification accuracy, and gave only slightly worse results in terms of regression scores. Taking into account the fact that IDW has low computational requirements and we can add further training points without having to retrain a complicated model, we think that this method is a valid tool for conflict intensity estimation in particular, and speech technology regression tasks in general.

Acknowledgements

This publication is supported by the European Union and co-funded by the European Social Fund. Project title: Telemedicine-focused research activities in the fields of mathematics, informatics and medical sciences. Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073.

References

[1] Beke, A., Neuberger, T. "Automatic laughter detection in Hungarian spontaneous speech using GMM/ANN hybrid method," in *Proc. SJUSK, Copenhagen, Denmark*, 2013

[2] Berk, R. A. "Statistical Learning from a Regression Perspective," Springer Verlag, 2008.

[3] Bishop, M. C., "Neural Networks for Pattern Recognition," Clarendon Press, Oxford, 1995.

[4] Busa-Fekete, R., Kégl, B., "Accelerating AdaBoost using UCB," in *Proc. KDDCup 2009 (JMLR W&CP), Paris, France*, 2009, pp. 111–122.

[5] Dobry, G., Hecht, R. M., Avigal, M., Zigel, Y., "Supervector dimension reduction for efficient speaker age estimation based on the acoustic speech signal," *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 19, no. 7, pp. 1975–1985, 2011.

[6] Duda, R. O., Hart, P. E., Stork, D. G. "Pattern Classification", John Wiley & Sons, 2001.

[7] Eyben, F., Wöllmer, M., Schuller, B., "Opensmile: The Munich versatile and fast open-source audio feature extractor," in *Proc. ACM Multimedia, Firenze, Italy,* 2010, pp. 1459–1462

[8] Franke, R., Nielson, G., "Smooth interpolation of large sets of scattered data," *Int. Jour. for Num. Meth. in Eng.*, vol. 15, pp. 1691–1704, 1980.

[9] Gemmer, M., Becker, S., Jiang, T., "Observed monthly precipitation trends in China 1951–2002," *Theor. and Appl. Climat.*, vol. 77, no. 1, pp. 39–45, 2004.

[10] Gosztolya, G., Busa-Fekete., R., Tóth, L., "Detecting autism, emotions and social signals using AdaBoost," in *Proc. Interspeech, Lyon, France*, 2013, pp. 220–224

[11] Gosztolya, G., Grósz, T., Busa-Fekete., R., Tóth, L., "Detecting the intensity of cognitive and physical load using AdaBoost and Deep Rectifier Neural Networks," in *Proc. Interspeech, Singapore, Singapore*, 2014, pp. 452–456

[12] Grézes, F., Richards, J., Rosenberg, A., "Let me finish: Automatic conflict detection using speaker overlap," in *Proc. Interspeech, Lyon, France,* 2014, pp. 200–204

[13] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[14] Kaya, H., Özkaptan, T., Salah, A. A., Gürgen S. F., "Canonical Correlation Analysis and Local Fisher Discriminant Analysis based multi-view acoustic feature reduction for physical load prediction," in *Proc. Interspeech, Singapore, Singapore,* 2014, pp. 442–446

[15] Kim, S., Valente, F., Filippone, M., Vinciarelli, A., "Predicting continuous conflict perception with Bayesian Gaussian Processes," *IEEE Trans. Aff. Comp.*, vol. 5, no. 2, pp. 187–200, May. 2014.

[16] Räsänen, O., Pohjalainen, J., “Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech,” in *Proc. Interspeech, Lyon, France*, 2013, pp. 210–214

[17] Reynolds, D. A., Quatieri, T. F., Dunn, R. B., “Speaker verification using adapted Gaussian Mixture Models,” *Dig. Sign. Proc.,* vol. 10, no. 1, pp. 19–41, 2000.

[18] Schapire, R. E., Singer, Y., “Improved boosting algorithms using confidence-rated predictions,” *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.

[19] Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., Williamson, R. C., “Estimating the support of a high-dimensional distribution,” *Neur. Comp.*, vol. 13, no. 7., pp. 1443–1471, 2001.

[20] Schuller, B., Steidl, S., Batliner, A., Epps, J., Eyben, F., Ringeval, F., Marchi, E., Zhang, Y.: “The INTERSPEECH 2014 computational paralinguistics challenge: Cognitive & Physical load,” in *Proc. Interspeech, Singapore, Singapore*, 2014, pp. 427–431.

[21] Schuller, B., Steidl, S., Batliner, A., Vinciarelli, A., Scherer, K., Ringeval, F., Chetouani, M., Weninger, F., Eyben, F., Marchi, E., Mortillaro, M., Salamin, H., Polychroniou, A., Valente, F., Kim, S., “The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism,” in *Proc. Interspeech, Lyon, France*, 2013, pp. 148–152.

[22] Shepard, D., “A two-dimensional interpolation function for irregularly-spaced data,” in *Proc. 23rd ACM Nat. Conf., New York, NY, USA*, 1968, pp. 517–524.

[23] Spector, P., Jex, S., “Development of four self-report measures of job stressors and strain: interpersonal conflict at work scale, organizational constraints scale, quantitative workload inventory, and physical symptoms inventory,” *Jour. of Occup. Health Psych.*, vol. 3, no. 4, pp. 356–367, 1998.

[24] Tax, D. M., Duin, R. P., “Support vector data description,” *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[25] Tóth, S. L., Sztahó, D., Vicsi, K., “Speech emotion perception by human and machine”, in *Proc. COST Action, Patras, Greece*, 2012, pp. 213–224.

[26] Verbunt, M., Gurtz, J., Jasper, K., Lang, H., Warmerdam, P., Zappa, M., “The hydrological role of snow and glaciers in alpine river basins and their distributed modeling,” *Jour. of Hydr.*, vol. 282, no. 1, pp. 36–55, 2003.