Interpretable Machine Learning for Generating Semantically Meaningful Formative Feedback

Nese Alyuz
Intel Corporation, Intel Labs
Hillsboro, OR, USA
nese.alyuz.civitci@intel.com

Tevfik Metin Sezgin
Koc University
Istanbul, TURKEY
mtsezgin@ku.edu.tr

Abstract

We express our emotional state through a range of expressive modalities such as facial expressions, vocal cues, or body gestures. However, children on the Autism Spectrum experience difficulties in expressing and recognizing emotions with the accuracy of their neurotypical peers. Research shows that children on the Autism Spectrum can be trained to recognize and express emotions if they are given supportive and constructive feedback. In particular, providing formative feedback, (e.g., feedback given by an expert describing how they need to modify their behavior to improve their expressiveness), has been found valuable in rehabilitation. Unfortunately, generating such formative feedback requires constant supervision of an expert who assesses each instance of emotional display. In this work, we describe a system for automatic formative assessment integrated into an automatic emotion recognition setup. Our system is built on an interpretable machine learning framework that answers the question of what needs to be modified in human behavior to achieve a desired expressive display. It propagates the desired changes to human-understandable attributes through explanation vectors operating on a shared low level feature space. We report experiments conducted on a childrens voice data set with expression variations, showing that the proposed mechanism generates formative feedback aligned with the expectations reported from a clinical perspective.

1. Introduction

The neuro-developmental conditions affecting communication and behavioral skills are referred to as Autism Spectrum Conditions (ASC). The main characteristics accompanied with ASC can be listed as having difficulty in social communication and interaction with other people, having restrictive interest, repetitive behaviors, and showing symptoms that restrict these individuals from functioning properly at any areas of life such as work or school [5]. Research shows that individuals with ASC particularly struggle with recognizing affective or mental states expressed by others and expressing their own inner states [7]. These deficiencies in recognition and expression of affect act as social communication barriers for individuals with ASC, keep them from developing healthy social relationships, and lead to social exclusion.

There have been numerous technological developments that target ASC population to teach them emotion-related skills. Technologies such as *ICPS - I Can Problem-Solve* [3], *Emotion Trainer* [2], *Lets Face It* [9], or *Mindreading* [4] are examples that target to train individuals on emotion recognition and social communication skills. However, there is a lack of interactive tools and technologies for assisting individuals on the Autism spectrum in their quest to improve their skills for expressing emotions.

This work, is part of the European Union project called *ASC-Inclusion-Enlarged* [1], which aims to address difficulties associated with training ASC children to better express their own emotions and assess others' emotions. When assessing the emotional expressiveness of an individual, automated emotion recognition technologies through any or all of the expressive modalities of face, voice, or gestures can be used. The emotion performed by the subject can be evaluated through inference algorithms and the output can be used to inform whether the attempt of displaying a particular emotion was successful or not. However, such an approach lacks any feedback on what they can do to improve their expressiveness. In this work, we propose a formative assessment approach to include a feedback mechanism which would provide the subject with specific and comprehensible guidance for improved performance. By receiving easy-to-understand and targeted corrective feedback, individuals have the opportunity to learn how they should adjust their facial, vocal, or gestural behavior to show prototypical emotions.

The paper is organized as follows: First, in Section 2, the proposed formative assessment approach is explained. Sec-

tion 3 summarized the experimental results on the sample voice data. Section 4 concludes the paper, outlining future directions as well.

2. Proposed Formative Feedback Generation Scheme

The typical scenario for formative feedback involves a person displaying a performance, with a target label in mind. In the case of Autism rehabilitation, this is a child who attempts to display a target emotion through a variety of channels (facial displays, voice, body language, gestures). In the typical scenario, the performer fails to express the emotion successfully (hence the need for the formative feedback). The goal of the formative feedback module is to generate human understandable explanations of what behavioral changes would move the performed instance closer to the desired target.

Formative feedback is generated on a per-instance basis. Hence, there are two main inputs to the formative feedback module: features extracted from the performance instance, and the label of the expected (target) class. The overall scheme describing how these two inputs are converted into formative feedback is outlined in Figure 1. Implicitly, there is a third input: the classifier that classifies user input. In general, this classifier can come in any form (e.g., an SVM, a PGM, or a Deep Network), and our scheme does not restrict the system designer to adopt a particular kind of classifier. Hence, to generalize the scheme to any kind of classifier, a probabilistic black-box classification module is obtained from the input classifier through Parzen window estimation (top left in Figure 1), which serves as the third input to the formative feedback system. Using the idea of the explanation vectors of Baehrens *et al*. [6], these three inputs are used to obtain feature modifications required to achieve the target classification on an instance basis.

Using these explanations, the original feature can be modified to convert the input instance into one whose prediction decision matches the target. The difference between the original and the modified feature sets can be used to describe the necessary corrections. However, providing corrections at the feature level will not be comprehensible from a user perspective. Therefore, it is necessary to define high-level attributes and obtain the necessary alterations in terms of these attributes. To generate the formative feedback, attribute extraction is run on both the original and the modified feature sets, and the difference between the original attributes (from original feature set) and the modified attributes (from the modified feature set) is used to generate the semantically meaningful formative feedback to the subject.

The two main contributions of this work are as follows: First of all, the features are updated in an iterative manner to find the minimum necessary alteration. Secondly, for the feedback to be semantically meaningful, the necessary feature modifications are mapped to a higher-level attribute space. In the following three subsections, we describe the details of how we generate explanation vectors, modify the features iteratively until convergence, and obtain semantically meaningful corrections.

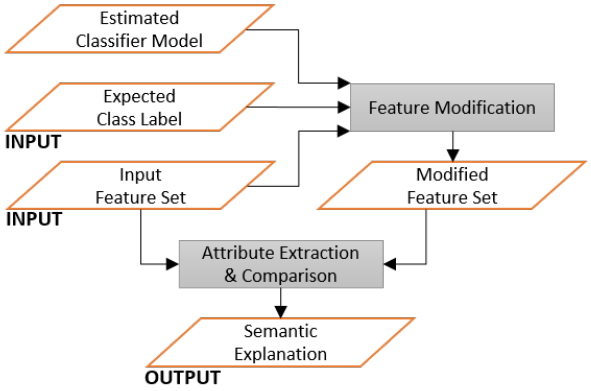

Figure 1. General scheme for the formative feedback generation approach.

2.1. Explanation Vector Generation

Let's assume that we have a training set of d-dimensional points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ with class labels $Y = \{y_1, y_2, \ldots, y_n\} \in \{1, \ldots, C\}$, where we have C distinct classes of output labels and the joint distribution $P(X, Y)$ is unknown. The explanation vector of a given instance $\mathbf{x}_0$ for a target class label c can be computed as the derivative of the conditional probability of the given class label for the given input instance:

$$\zeta_c(\mathbf{x}_0) = \frac{\partial}{\partial \mathbf{x}} P(Y \neq c | \mathbf{X} = \mathbf{x})\Big|_{\mathbf{x}=\mathbf{x}_0} \tag{1}$$

Here, ζ_c is a d-dimensional vector like the original input instance, defining the flow away from the corresponding class: The entries with high absolute values will point out features that have high influence on the classification decision, where positive and negative signs indicate individual features whose values should be decreased or increased, respectively, to better resemble the target class.

To generate explanations for an unknown classifier $g(\cdot)$, first of all we have to estimate the classifier. Then the explanation vector will be computed using the estimation $\hat{g}(\cdot)$ and the class label given by the classifier. In this work, we have considered Kernel Density Estimation [8] to estimate the joint probability for the given class label:

$$\hat{p}_\sigma(\mathbf{x}, y = c) = \frac{1}{n} \sum_{i \in I_c} k_\sigma(\mathbf{x} - \mathbf{x}_i) \tag{2}$$

where $k_\sigma(\cdot)$ is the kernel function and I_c is the index set for

the given class. Here, we employed Gaussian kernel and estimated the conditional probability distribution as follows:

$$\hat{p}_\sigma(y = c|\mathbf{x}) \approx \frac{\sum_{i \in I_c} k_\sigma(\mathbf{x} - \mathbf{x}_i)}{\sum_i k_\sigma(\mathbf{x} - \mathbf{x}_i)} \quad (3)$$

The explanation vector for instance $\mathbf{z}$ and expected class label c can be defined as follows:

$$\hat{\zeta}_c(\mathbf{z}) = \frac{\partial}{\partial \mathbf{x}} \hat{p}_\sigma(y \neq g(\mathbf{z})|\mathbf{x})\Big|_{\mathbf{x}=\mathbf{z}} \quad (4)$$

Using $g(\mathbf{z})$ outputs instead of c class labels allows us to interpret the classification decisions of our black-box classifier (assuming $\hat{g}(\cdot)$ approximates $g(\cdot)$ well).

2.2. Iterative Feature Modification

The explanation vector defines the direction of the flow away from the corresponding class. However, the magnitude of the vector does not embody any information about how much change is needed to resemble the target class better. As in the gradient descent algorithm, the explanation vector can be computed in an iterative manner until convergence to the target class is achieved. Moreover, a predefined step size can be employed for each iteration. As a fixed step size, we compute the minimum inter sample distance:

$$d_s = \min_{i \in \{1,\ldots,n\}, j \in \{1,\ldots,n\} \setminus i} ||\mathbf{x}_i - \mathbf{x}_j|| \quad (5)$$

At each iteration, using the directional information residing in the explanation vector and the calculated step size, the input instance (feature) is modified:

$$\mathbf{z}_{(t+1)} = \mathbf{z}_{(t)} - \frac{d_s \cdot \hat{\zeta}_c(\mathbf{z}_{(t)})}{|\hat{\zeta}_c(\mathbf{z}_{(t)})|} \quad (6)$$

After each iterative modification, the decision of the classifier (either $g(\cdot)$ or $\hat{g}(\cdot)$) will be checked to see if convergence to the target class is achieved. Once convergence is reached, iterative computation of the explanation vectors and feature updates will result in a modified feature set comprising the minimal amount of change needed.

2.3. Attribute-based Feedback Generation

Once convergence at the feature space is obtained, the difference between the original and the modified feature set gives the minimum change needed for the instance to be classified from the target class. However, computing this difference at the feature space and providing it as a feedback to the user will not be comprehensible as the feature space can be very high dimensional and even the little changes needed at feature level would fire up in the difference vector. In this work, we propose to switch to a high-level attribute level which would enable us to provide semantically meaningful feedback to the user. The high-level attribute space can either be defined by careful selection of meaningful set of features, or it can require to switch to this high-level space by running pre-trained attribute extractor. It should be noted that defining semantically meaningful attributes is highly dependent on the feature space and the classification problem.

3. Experimental Results on Voice Modality

3.1. Voice Database and Emotional-Attributes

In order to test our proposed formative feedback generation approach, we experimented with a sample emotional voice dataset, consisting of 1534 instances labeled for 27 emotional classes. For each instance, 6373 vocal features were available.

As an initial step, we defined possible forms of semantically meaningful attributes for the voice modality considering a variety of affective states. We mainly referred to the relations shortlisted by our clinical partners, where modality-specific characteristics are given for different emotions. The attributes shortlisted are as follows: (1) pitch, (2) pitch variation, (3) loudness, and (4) speech rate. For our initial analysis, we focused on two basic emotions of *Happy* and *Sad*. Voice attributes and their expected associations to these two basic emotions are given in Table 1. For the samples of these two emotions in our database, we binarized attributes (High or Low) considering their respective location to the attribute mean over all emotional instances. The histogram for the attribute combinations is plotted in Figure 2. As can be seen from this figure, the sample instances do not always resemble the attribute-emotion associations, where it would be expected to have most of the sad examples at the left extreme, whereas the happy samples would gather at the right side.

Table 1. Attribute candidates and attribute-emotion associations for the voice modality. (H and L stand for high, and low values.)

	Happy	Sad
Pitch	H	L
Pitch Variation	H	L
Loudness	H	L
Speech Rate	H	L

3.2. Classifier Approximation with Kernel Density Estimation

As visualized in Figure 2, a simple thresholding approach for formative assessment, where we would decide on the correctness of an attribute using its respective location to the threshold (e.g. attribute mean), would fail. The proposed approach based on explanation vector generation is expected to yield more accurate explanations related to classifier's decisions.

For our initial experiments, we focused on the two-class classification problem (Happy vs. Sad), and utilized only

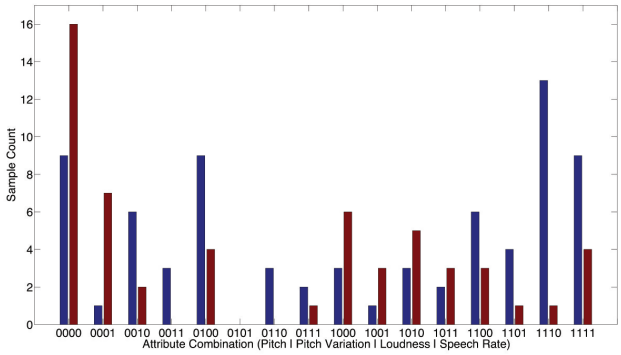

Figure 2. Histogram for binarized attribute combinations is given, where the attributes are decoded in the order of pitch, pitch variation, loudness, and speech rate. Instances of happy and sad emotion labels are shown with blue and red bars, respectively.

the instances of these two classes. Keeping the ratio of the two emotion instances approximately the same, we have spared 14 happy and 11 sad instances for testing. Since we have a limited number of training examples (60 happy and 45 sad), we have employed Leave-One-Out cross validation for hyper-parameter optimization. This randomized separation of training and test sets is handled for 20 repetitions. The mean and standard deviation of the optimum hyper-parameter and the test accuracy obtained with the KDE are summarized in the first two columns of Table 2.

3.3. Explanation Vectors and Modified Features

After the classifier is estimated with the KDE, the explanation vectors for a single iteration can be computed as given in Equation (4). The explanation vector generation and feature modification steps will be handled in an iterative manner, where the iteration structure and the convergence rule is given in Section 2.2. For our preliminary experiments, where we have considered two emotion classes, we have considered each instance of a class as a misexpressed instance of the opposite class: For example, for a happy instance, the explanation vector generation and feature modification iterations are run, where the convergence rule is being classified as from the sad class. The mean and standard deviation of the average number of modification steps required are given in the last column of Table 2 (calculated over 20 repetitions). As these results indicate, a sample from one class can be modified to be classified as from the other class in 5.82 steps in average.

For the voice modality, the attributes that we want to give feedback on are directly included in the input features. Therefore, we directly investigated the alterations caused on these four attributes. Once again, the attribute-specific means of all instances from all six basic emotions are used as thresholds to binarize the attribute values. In Figure 3, each row gives the bar plots of *Happy* (left) and *Sad* (right)

Table 2. The mean and standard deviation of the optimum hyper-parameter, the test accuracy obtained with the KDE, and the average number of steps required to move to the opposite class (calculated over 20 repetitions).

	optimum σ	test accuracy (%)	avg. step count
μ	8.06	73.60	5.50
σ	0.74	11.78	0.25

instances, when the attribute values are considered as either Low or High when compared with the attribute-specific threshold. For example, let's consider the top-left plot: For *Happy* instances, the features are modified so that each instance is classified as from the *Sad* class. Here, the bar plots for the *pitch* attribute are given for *before* modification and for *after* modification. As expected, the number of instances with low pitch is increased, whereas the number of instances with high pitch value is decreased. This trend is opposite for the *Sad* instances (top-right plot). Moreover, we see similar trends for the other attributes.

4. Conclusion and Future Directions

In this work, we propose a method to generate comprehensible corrective feedback to guide children with ASC in expressing their emotions better. Based on the explanation vector generation approach of Baehrens *et al*. [6], we propose to modify features in an iterative manner until they resemble the target class, and the minimum required alterations are expressed in terms of high-level attributes to provide semantically meaningful corrections. The initial experiments on the voice modality showed that we were able to generate feedback aligned with the expectations from a clinical point of view.

As future directions, we target to evaluate the generated formative feedback from a user perspective, assessing whether they are semantically meaningful. For performance improvement, use of a larger training set is necessary. Moreover, the current module can be extended to work on a larger set of emotions. Moreover, a separate set of high-level attributes that do not reside in the feature space can be considered when providing the corrective feedback.

References

[1] Asc-inclusion: Interactive emotion games. http://www.geniiz.com/, Mar. 2019.

[2] Emotion trainer. http://www.appyautism.com/en/app/emotion-trainer/, Mar. 2019.

[3] I can problem solve. http://www.icanproblemsolve.info, Mar. 2019.

[4] Mind reading. https://www.jkp.com/uk/mindreading, Mar. 2019.

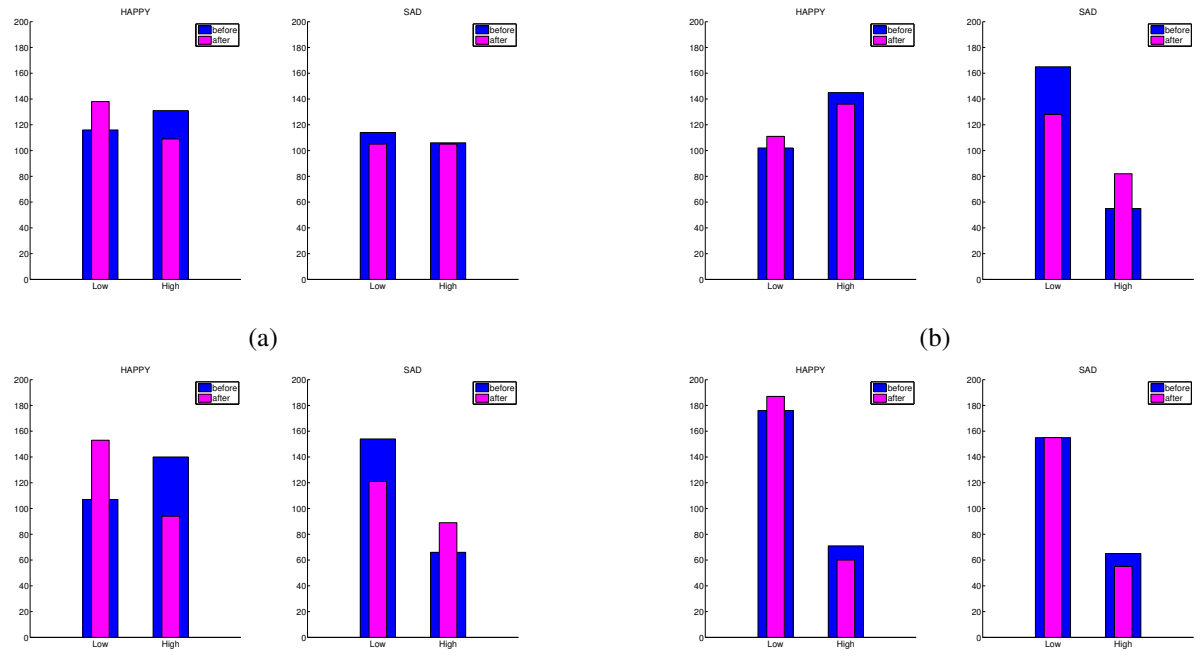

Figure 3. Histograms of binarized attribute values are given for before (blue) and after (pink) modification states for: (a) pitch, (b) pitch variation, (c) loudness, (d) speech rate attributes. Plots on left and right are for happy and sad instances, respectively.

[5] American Psychiatric Association et al. *Diagnostic and statistical manual of mental disorders (DSM-5)*. American Psychiatric Pub, 2013.

[6] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decision. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

[7] Simon Baron-Cohen. *Mindblindness: An essay on autism and theory of mind*. MIT press, 1997.

[8] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[9] J. W Tanaka, J.M. Wolf, C. Klaiman, K. Koenig, J. Cockburn, L. Herlihy, C. Brown, S. Stahl, M.D. Kaiser, and R.T. Schultz. Using computerized games to teach face recognition skills to children with autism spectrum disorder: the lets face it! program. *Journal of Child Psychology and Psychiatry*, 51(8):944–952, 2010.