Data-dependent evaluator modeling and its application to emotional valence classification from speech

Kartik Audhkhasi, Shrikanth S. Narayanan

Signal Analysis and Interpretation Laboratory (SAIL)
Electrical Engineering Department, University of Southern California, Los Angeles, CA
audhkhas@usc.edu, shri@sipi.usc.edu

Abstract

Practical supervised learning scenarios involving subjectively evaluated data have multiple evaluators, each giving their noisy version of the hidden ground truth. Majority logic combination of labels assumes equally skilled evaluators, and is generally suboptimal. Previously proposed models have assumed data independent evaluator behavior. This paper presents a data dependent evaluator model, and an algorithm to jointly learn evaluator behavior and a classifier. This model is based on the intuition that real world evaluators have varying performance depending on the data. Experiments on an emotional valence classification task show modest performance improvements of the proposed algorithm as compared to the majority logic baseline and a data independent evaluator model. But more critically, the algorithm also provides accurate estimates of individual evaluator performance, thus paving the way for incorporating active learning, evaluator feedback and unreliable data detection.

Index Terms: subjective evaluation, evaluator modeling, emotion classification

1. Introduction

A conventional approach to a binary supervised learning problem involves a set of training examples $(\mathbf{x}_i)_{i=1}^N$ ($\mathbf{x}_i \in \mathbb{R}^D$), and the associated labels $(y_i)_{i=1}^N$ ($y_i \in \{0,1\}$). In this framework, the training labels are assumed to be true and a correct characterization of the two classes. However, this assumption is a weak one from many perspectives. First, many real-world classification tasks involve ambiguous, ill-defined and fuzzy classes. For example, classical emotion recognition from speech typically quantizes the entire range of human emotions into a finite set of categories, such as for example: {Angry, Happy, Sad, Neutral}. However, this discretization is only an approximation to the continuum of human emotions [1]. Labeling the training data in such cases is a challenging task, even for trained evaluators, since many training examples lie at the boundaries of different classes. This invariably leads to labels which are not a true representation of the data. Moreover, the expertise level and consistency in labeling across evaluators is highly variable; training and using highly trained evaluators is often expensive, and not scalable to large databases.

One typical approach to deal with class uncertainty and limited availability of trained evaluators is to obtain labels from multiple evaluators, and use the majority logic or averaged labels as a proxy for the golden truth labels. In [2], the authors show that these averaged labels improve emotion classification performance. However, such an approach is clearly sub-optimal, since it gives equal weight to the opinion of each evaluator. Each evaluator has varying skill levels and offers a different perception of the various classes. In such situations, it is intuitive to give more weight to the labels assigned by reliable evaluators as compared to the unreliable ones.

Many works in the past have dealt with this multiple evaluator problem. In [3], the task considered was inferring the ground truth label for the presence of volcanoes in radar images. Each putative volcano region was categorized by evaluators into one of five categories. In addition, each category was associated with a probability of the actual presence of a volcano, given that the category was true. For a region, given the categorical labels assigned by multiple evaluators, the task was to estimate the posterior probability of the occurence of a volcano. The authors do this using the Expectation Maximization (EM) algorithm [4]. In [5], this probabilistic ground truth estimate is used to learn a classifier. The work by Dawid and Skeene in [6] also focused on estimating just the ground truth label.

Raykar et al. [7] present a slightly different approach, where they directly learn the classifier from the training data and the multiple evaluator labels. They propose a simple two-coin generative model for each evaluator. Let R be the number of evaluators. The j^{th} evaluator ($j = 1, ..., R$) is assumed to have two coins, with the following probabilities of turning heads: $\alpha^j = P(y^j = 1|y = 1)$ and $\beta^j = P(y^j = 0|y = 0)$. Here, y is the true hidden label and y^j is the label assigned by the j^{th} evaluator. α^j and β^j represent the sensitivity and specificity of the j^{th} evaluator. Thus, highly reliable evaluators are expected to have both α^j and β^j close to 1. In addition, they assume a logistic regression model for generation of the true hidden label y given the training example $\mathbf{x}$ and the classifier weight vector $\mathbf{w}$:

$$P(y = 1|\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{x}}} \quad (1)$$

The generation of y_i^j, i.e., the label assigned to $\mathbf{x}_i$ by the j^{th} evaluator is done as follows:

1. Each evaluator hypothetically looks at the true hidden label y_i of $\mathbf{x}_i$.
2. If $y_i = 1$, he flips the α^j coin, else the β^j one.
3. If the coin falls heads, he retains y_i as y_i^j, else sets $y_i^j = 1 - y_i$.

The authors use the EM algorithm to learn the parameters - the (α^j, β^j) pair for each evaluator $j = 1, ..., R$ and the logistic regression weight vector $\mathbf{w}$. As expected, the true labels $(y_i)_{i=1}^N$ are taken as the hidden variables in the formulation. This algorithm has been shown to perform better than majority logic based logistic regression on both synthetic and real datasets. In addition, it estimates evaluator sensitivity and specificity.

It is easy to see that this two-coin evaluator model is highly simplified. Particularly, according to this model, the evaluator does not use the data $\mathbf{x}_i$ while generating a label for it. This is

impractical, since real world evaluators base their decision on the example which has been given to them for labeling. For example, in the case of labeling emotional speech, it is unreasonable to expect evaluators to assign labels to sentences without listening to them. The main contribution of the present paper is to propose a model for this data-dependent nature of the subjective evaluation process. The proposed model is based on the intuition that real world evaluators have different sensitivity and specificity for different regions of the entire feature space.

2. Data-dependent evaluator model

The training data of the proposed algorithm consists of N independent D-dimensional examples $(\mathbf{x}_i)_{i=1}^N$ and R binary labels for each example, $(y_i^j)_{j=1}^R$, one from each evaluator. An M-mixture component Gaussian Mixture Model (GMM) is used to approximate the distribution of the training examples. To model the data-dependent behavior of evaluators, we hypothesize that each evaluator has a pair of coins for each mixture component. Let these coins be defined by probabilities α_m^j and β_m^j of turning heads ($j = 1, ..., R;\ m = 1, ..., M$), defined as follows:

$$\alpha_m^j = P(y_i^j = 1 | y_i = 1, m^{th} \text{ Gaussian gives } \mathbf{x}_i) \quad (2)$$

$$\beta_m^j = P(y_i^j = 0 | y_i = 0, m^{th} \text{ Gaussian gives } \mathbf{x}_i) \quad (3)$$

We hypothesize that evaluators generate labels as follows:

1. Given an example $\mathbf{x}_i$ to label, the j^{th} evaluator finds the mixture component which generated that example. Let it be the m^{th} mixture component.
2. Based on whether the true label y_i is 1 or 0, he flips the α_m^j or β_m^j coin respectively.
3. If the coin falls heads, he retains y_i as y_i^j. Else he sets $y_i^j = 1 - y_i$.

Note that for $M = 1$, this algorithm becomes similar in nature to the one proposed in [7], where the evaluators become totally oblivious to the presented example while generating their labels. Figures 1 and 2 contrast the graphical model representations of the evaluator model in [7] and the proposed model, respectively. The dependence of the evaluator labels y^j on the data $\mathbf{x}$ is evident through the hidden random vector $\mathbf{z}$. This M-dimensional random vector has value 1 in exactly one of its components, rest being 0. $z_m = 1$ indicates that the m^{th} Gaussian was used for generating $\mathbf{x}$. Next, we discuss the algorithm for estimation of various parameters of the model in the EM framework.

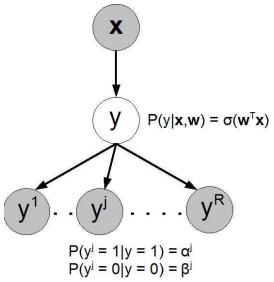

Figure 1: *Graphical model representation of the evaluator model in [7].* $\mathbf{x}$ *is the observed data,* y *is the hidden label and* y^j *is the label assigned to* $\mathbf{x}$ *by the* j^{th} *evaluator.*

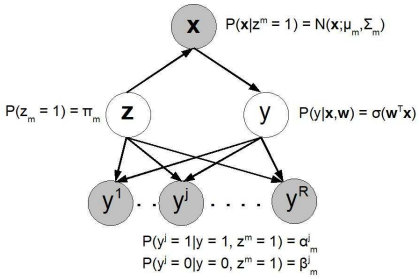

Figure 2: *Graphical model representation of the proposed model.* $\mathbf{z}$ *is the indicator vector for GMM mapping of* $\mathbf{x}$.

3. Parameter estimation

The parameters of the evaluator model which have to be estimated are $\mathbf{w}$, $(\pi_m, \mu_m, \Sigma_m)_{m=1}^M$ (mixture weights, means and covariances of the GMM) and $((\alpha_m^j, \beta_m^j)_{m=1}^M)_{j=1}^R$. Let these parameters be denoted by Θ. Finding analytic expressions for ML estimates of these parameters is intractable by direct maximization of the likelihood of the training data, i.e., $P((\mathbf{x}_i)_{i=1}^N, ((y_i^j)_{j=1}^R)_{i=1}^N | \Theta)$. Thus, we resort to the EM algorithm for performing this optimization. The hidden variables are $(y_i)_{i=1}^N$ (the true labels) and $(\mathbf{z}_i)_{i=1}^N$ (the 1-in-M encoding of the mixture component index which emitted $\mathbf{x}_i$).

3.1. The E step

The complete data likelihood is the joint probability density function (pdf) of the observed and hidden data given the parameters, and it can be factored as follows:

$$P((\mathbf{x}_i)_{i=1}^N, ((y_i^j)_{j=1}^R)_{i=1}^N, (y_i)_{i=1}^N, (\mathbf{z}_i)_{i=1}^N | \Theta) = \prod_{i=1}^N P(\mathbf{x}_i, \mathbf{z}_i | \Theta) P((y_i^j)_{j=1}^R, y_i | \mathbf{x}_i, \mathbf{z}_i, \Theta) \quad (4)$$

Now, $P(\mathbf{x}_i, \mathbf{z}_i | \Theta)$ can be written as:

$$P(\mathbf{x}_i, \mathbf{z}_i | \Theta) = \prod_{m=1}^M (\pi_m N(\mathbf{x}_i | \mu_m, \Sigma_m))^{z_{im}} \quad (5)$$

where $N(\mathbf{x}_i | \mu_m, \Sigma_m)$ is the Gaussian pdf and z_{im} is the m^{th} component of $\mathbf{z}_i$. Consider the term $P((y_i^j)_{j=1}^R, y_i | \mathbf{x}_i, \mathbf{z}_i, \Theta)$ in (4). Since y_i is a Bernoulli random variable, it can be written as follows:

$$P((y_i^j)_{j=1}^R, y_i | \mathbf{x}_i, \mathbf{z}_i, \Theta) = P((y_i^j)_{j=1}^R, y_i = 1 | \mathbf{x}_i, \mathbf{z}_i, \Theta)^{y_i} P((y_i^j)_{j=1}^R, y_i = 0 | \mathbf{x}_i, \mathbf{z}_i, \Theta)^{1-y_i} \quad (6)$$

Upon some calculations, we can express (6) as:

$$P((y_i^j)_{j=1}^R, y_i | \mathbf{x}_i, \mathbf{z}_i, \Theta) = (p_i \prod_{m=1}^M a_{im}^{z_{im}})^{y_i} ((1 - p_i) \prod_{m=1}^M b_{im}^{z_{im}})^{1-y_i} \quad (7)$$

where $p_i = \sigma(\mathbf{w}^T \mathbf{x}_i)$, $a_{im} = \prod_{j=1}^R (\alpha_m^j)^{y_i^j} (1 - \alpha_m^j)^{1-y_i^j}$ and $b_{im} = \prod_{j=1}^R (1 - \beta_m^j)^{y_i^j} (\beta_m^j)^{1-y_i^j}$.

Therefore, using (4), the complete data log-likelihood can be expressed as:

$$\log P((\mathbf{x}_i)_{i=1}^N, ((y_i^j)_{j=1}^R)_{i=1}^N, (y_i)_{i=1}^N, (\mathbf{z}_i)_{i=1}^N|\Theta) = \sum_{i=1}^N \Big[\sum_{m=1}^M z_{im} \Big(\log \pi_m + \log N(\mathbf{x}_i|\mu_m, \Sigma_m) \Big) + y_i \Big(\log p_i + \sum_{m=1}^M z_{im} \log a_{im} \Big) + (1-y_i) \Big(\log(1-p_i) + \sum_{m=1}^M z_{im} \log b_{im} \Big) \Big] \quad (8)$$

Next, the posterior pdf of the hidden variables given the observed variables and parameters can be written as:

$$P((y_i)_{i=1}^N, (\mathbf{z}_i)_{i=1}^N|(\mathbf{x}_i)_{i=1}^N, ((y_i^j)_{j=1}^R)_{i=1}^N, \Theta) = \prod_{i=1}^N P(\mathbf{z}_i|(y_i^j)_{j=1}^R, \mathbf{x}_i, \Theta) P(y_i|\mathbf{z}_i, (y_i^j)_{j=1}^R, \mathbf{x}_i, \Theta) \quad (9)$$

So, while computing the expectation of the complete data log-likelihood with respect to this posterior pdf, we first compute the expectation with respect to $P(y_i|\mathbf{z}_i, (y_i^j)_{j=1}^R, \mathbf{x}_i, \Theta)$, and then the expectation of the resulting function of $\mathbf{z}_i$ with respect to $P(\mathbf{z}_i|(y_i^j)_{j=1}^R, \mathbf{x}_i, \Theta)$. After some calculations, we get the following expression for the expectation:

$$\mathbb{E} = \sum_{i=1}^N \Big[\sum_{m=1}^M \gamma_{im} \Big(\log \pi_m + \log N(\mathbf{x}_i|\mu_m, \Sigma_m) \Big) + \eta_i \log p_i + \sum_{m=1}^M \zeta_{im} \log a_{im} + (1-\eta_i) \log(1-p_i) + \sum_{m=1}^M (\gamma_{im} - \zeta_{im}) \log b_{im} \Big] \quad (10)$$

where we define the following quantities:

$$\kappa_i = \frac{p_i \prod_{m=1}^M a_{im}^{z_{im}}}{p_i \prod_{m=1}^M a_{im}^{z_{im}} + (1-p_i) \prod_{m=1}^M b_{im}^{z_{im}}} \quad (11)$$

$$\gamma_{im} = \frac{\pi_m N(\mathbf{x}_i|\mu_m, \Sigma_m)}{\sum_{m'=1}^M \pi_{m'} N(\mathbf{x}_i|\mu_{m'}, \Sigma_{m'})} \quad (12)$$

$$\eta_i = E_{P(\mathbf{z}_i|(y_i^j)_{j=1}^R, \mathbf{x}_i, \Theta)}\{\kappa_i\} \quad (13)$$

$$\zeta_{im} = E_{P(\mathbf{z}_i|(y_i^j)_{j=1}^R, \mathbf{x}_i, \Theta)}\{\kappa_i z_{im}\} \quad (14)$$

Note that κ_i is a function of $\mathbf{z}_i$, and the expectations in (13) and (14) can be computed easily using the fact that the M-dimensional random vector $\mathbf{z}_i$ has M possible values.

3.2. The M step

As can be seen from (10), the GMM parameters are decoupled from the rest of the parameters. Hence, we obtain the standard EM re-estimation equations for the GMM parameters. Re-estimation equations for all α_m^j and β_m^j can be found out by setting the partial derivative of $\mathbb{E}$ to 0.

$$\alpha_m^j = \frac{\sum_{i=1}^N \zeta_{im} y_i^j}{\sum_{i=1}^N \zeta_{im}} \quad (15)$$

$$\beta_m^j = \frac{\sum_{i=1}^N (\gamma_{im} - \zeta_{im})(1-y_i^j)}{\sum_{i=1}^N (\gamma_{im} - \zeta_{im})} \quad (16)$$

We note that $\mathbf{w}$ appears in $\mathbb{E}$ in a sigmoid function, due to which we resort to the Newton-Raphson method to estimate it:

$$\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - \delta \mathbf{H}^{-1} \mathbf{g} \quad (17)$$

where $\mathbf{g} = \sum_{i=1}^N (\eta_i - \sigma(\mathbf{w}^t \mathbf{x}_i)) \mathbf{x}_i$ is the gradient, $\mathbf{H} = -\sum_{i=1}^N \sigma(\mathbf{w}^t \mathbf{x}_i)(1 - \sigma(\mathbf{w}^t \mathbf{x}_i)) \mathbf{x}_i \mathbf{x}_i^{\mathbf{T}}$ is the Hessian and δ is the step size.

For initialization, we set $\mathbf{w} \leftarrow 0$, obtain π_m, μ_m and Σ_m through K-means clustering of $(\mathbf{x}_i)_{i=1}^N$, set $\kappa_i \leftarrow \frac{1}{R} \sum_{j=1}^R y_i^j$, $\eta_i \leftarrow \kappa_i$ and $\zeta_{im} \leftarrow \kappa_i \gamma_{im}$. The E-step involves computing (11)-(14), followed by the M-step, where the GMM parameter update and (15)- (17) are computed. The EM iterations continue till increase in the log-likelihood of the training data $(\mathbf{x}_i, (y_i^j)_{j=1}^R)_{i=1}^N$ becomes smaller than a chosen threshold, $\epsilon > 0$.

4. Experiments and Results

4.1. Emotional speech database and features

The performance of three algorithms - logistic regression with majority logic labels, the data independent algorithm from [7] and the proposed approach, was compared on an emotional valence classification task from speech using the EMA database [8]. This database has 3 subjects who read 10 sentences five times each, portraying four emotional states: anger, happiness, sadness and neutrality. Each sentence has been evaluated by 4 human evaluators. This database was chosen since the ground truth labels are known (and used in published studies), which facilitates easy evaluation of the learnt classifier. For valence (i.e., how positive or negative an emotion is) classification, {happy, neutral} were assigned to class 0, and {sad, angry} were assigned to class 1. The Fleiss kappa statistic [9] for the set of 4 evaluator labels was 0.7112, indicating a high inter-evaluator agreement.

We extracted 13 component Mel Filter Bank (MFB) features from the speech signal over 25 ms frames with 10 ms overlap. The component-wise mean and standard deviation of this vector are computed over the entire utterance, resulting in two 13-dimensional feature vectors. We found out that components of these mean and standard deviation vectors were highly correlated across utterances. To prevent multi-collinearity problems in logistic regression, we applied discrete cosine transform (DCT) to these vectors separately. The first 2 dimensions of the resulting vectors were retained, as they contained more than 99% of the energy of the entire signal. Concatenation resulted in a 4-dimensional feature vector for each utterance.

4.2. Classification results

The step size for Newton-Raphson iterations in the two multiple label algorithms was empirically set to 1, and the threshold for termination of the EM iterations to 0.01 (fraction change in log-likelihood of the training data). A 10-fold cross-validation was performed, with 8 folds for training and 1 fold each for development and testing. In each validation cycle, the number of mixture components was set to the value which gave the maximum area under the Receiver Operating Characteristic (ROC) curve for the development set. The area under ROC for the test set for the 9 possible configurations is shown in Table 1. As can be observed, the proposed algorithm leads to an improvement of 1.4% relative to the majority logic based method, significant at 5% level. The simple data independent algorithm from [7] leads to an improvement of 0.7% over the majority logic baseline, but it is not statistically significant.

Table 1: *Area under ROC for the 3 algorithms on the test set.*

Fold no.	Majority logic based algorithm	Simple multiple label based algorithm	Proposed algorithm
1	0.6835	0.6886	0.6869
2	0.7202	0.7366	0.7440
3	0.7470	0.7515	0.7485
4	0.6566	0.6532	0.6582
5	0.5658	0.5705	0.5784
6	0.7252	0.7438	0.7438
7	0.6528	0.6466	0.6497
8	0.7441	0.7458	0.7576
9	0.5952	0.5982	0.6086
Mean	**0.6767**	**0.6817**	**0.6862**
p-value [10]	reference	**0.1641**	**0.0195**

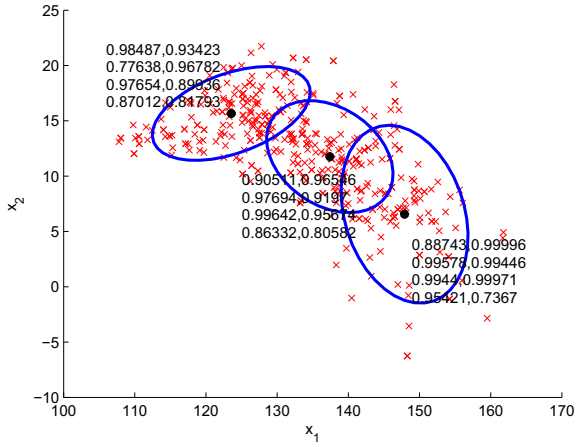

Figure 3: *3 mixture component GMM and the estimated* (α, β) *pairs (one row per evaluator) for each component.*

Figure 3 shows the scatter plot of the first two dimensions of the training data in the first cross-validation cycle. Also shown is the learnt 3 mixture component GMM, and the estimated (α, β) pairs for each evaluator. The true (α, β) pairs for each evaluator at every mixture component can be found out using the reference ground truth labels, and assigning each training example to the mixture component having the maximum posterior probability of emitting it. The total squared error between the true and estimated parameter values for all evaluators over all mixture components was found to be 0.0092 for α and 0.0090 for β, indicating highly accurate estimates.

4.3. Additional insights

In addition to a modest (but statistically significant) improvement above the majority logic based method, one is able to obtain valuable evaluator performance information from the proposed algorithm. As we can see from figure 3, evaluators don't give equally reliable labels in all clusters (e.g. evaluator 4 has lower reliability while labeling class 0 examples in the rightmost cluster, indicated by a β of 0.7367). Such data-dependent evaluator reliability information can be used in a variety of ways. First, it can be used for the training of human evaluators, by informing them about the set of examples which they labeled unreliably. Second, it can help screen out unreliable evaluators, by putting a threshold on the α and β values. Both the above steps can help improve the overall quality of subjectively obtained labels. In addition, knowledge of clusters of data on which all evaluators gave unreliable labels can be used as an indicator of unreliable data itself.

5. Conclusion and scope of future work

This paper presented a data-dependent multi-coin evaluator model, motivated by the observation that evaluators assign labels only after taking the example presented to them into consideration. A GMM is used to model the training examples. Each evaluator's labeling behavior for a particular mixture component is modeled by a pair of biased coins associated with that component. The EM algorithm is used to learn the parameters of this model. Valence classification experiments show a statistically significant improvement over the commonly used majority logic baseline, and the simple data-independent evaluator model presented in [7]. The estimated evaluator reliability parameters are highly accurate in the squared-error sense. More importantly, this model gives valuable insights into evaluator reliability, which can be used for active learning, feedback and data/evaluator selection. One of the major directions of future work is developing more realistic evaluator models. For example, labels assigned by a particular evaluator are not independent. Evaluator behavior is continuously evolving and many evaluations in the real world are serial (in the sense that the labels from a relatively naive evaluator are filtered by an expert). This is in contrast to the current framework, where each evaluator is assumed to be independent, and working in parallel.

6. Acknowledgements

This work was supported by the NSF, DARPA and US Army.

7. References

[1] Wollmer, M., Ebyen, F., Reiter, S., Schuller, B., Cox, C., Douglas-Cowie, E. and Cowie, R.,"Abandoning emotion classes - Towards continuous emotion recognition with modeling of long-range dependencies", Proc. Interspeech, Brisbane, Australia, 2008, pp. 597-600.

[2] Mower, E., Mataric, M. J. and Narayanan, S., "Evaulating evaluators: A case study in understanding the benefits and pitfalls of multi-evaluator modeling", Proc. Interspeech, Brighton, UK, 2009, pp. 1583-1586.

[3] Smyth, P., Fayyad, U., Burl, M., Perona, P. and Baldi, P., "Inferring ground truth from subjective labeling of Venus images", Advances in Neural Information Processing Systems, pp. 1085-1092, 1995.

[4] Dempster, A. P., Liard, N. M. and Rubin, D. B., "Maximum likelihood from incomplete data via the EM algorithm", Journal of Royal Statistical Society: Series B, vol. 39, pp. 1-38, 1977.

[5] Smyth, P, "Learning with probabilistic supervision", Computational Learning Theory and Natural Learning Systems, vol. 3, pp. 163-182, MIT Press.

[6] Dawid, A. P. and Skeene, A. M., "Maximum likelihood estimation of observed error-rates using the EM algorithm", Applied Statistics, vol. 28, pp. 20-28, 1979.

[7] Raykar, V. C., Yu, S., Zhao, L. H., Jerebko, A., Florin, C., Valadez, G. H., Bogoni, L. and Moy, L., "Supervised learning from multiple experts: Whom to trust when everyone lies a bit", Proc. ICML, 2009.

[8] Lee, S., Yildrim, S., Kazemzadeh, A. and Narayanan, S., "An articulatory study of emotional speech production", Proc. Eurospeech, Lisbon, Portugal, September 2005, pp. 497-500.

[9] Fleiss, J. L., "Measuring nominal scale agreement among many raters", Psychological Bulletin, vol. 76, no. 5, pp. 378-382, 1971.

[10] Mann, H. B. and Whitney, D. R., "On a test of whether one of two random variables is stochastically larger than the other", Annals of Mathematical Statistics, vol. 18, pp. 50-60, 1947.