Pattern Recognition Letters 128 (2019) 385–392

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Generating labels for regression of subjective constructs using triplet embeddings☆

Karel Mundnich*, Brandon M. Booth, Benjamin Girault, Shrikanth Narayanan

Signal Analysis and Interpretation Lab, University of Southern California, 3740 McClintock Ave EEB 400, Los Angeles, CA 90089, USA

ARTICLE INFO

Article history:
Received 5 April 2019
Revised 29 September 2019
Accepted 2 October 2019
Available online 4 October 2019

Keywords:
Continuous-time annotations
Annotation fusion
Inter-rater agreement
Triplet embeddings
Ordinal embeddings

ABSTRACT

Human annotations serve an important role in computational models where the target constructs under study are hidden, such as dimensions of affect. This is especially relevant in machine learning, where subjective labels derived from related observable signals (e.g., audio, video, text) are needed to support model training and testing. Current research trends focus on correcting artifacts and biases introduced by annotators during the annotation process while fusing them into a single annotation. In this work, we propose a novel annotation approach using triplet embeddings. By replacing the absolute annotation process to relative annotations where the annotator compares individual target constructs in triplets, we leverage the accuracy of comparisons over absolute ratings by human annotators. We then build a 1-dimensional embedding in Euclidean space that is indexed in time and serves as a label for regression. In this setting, the annotation fusion occurs naturally as a union of sets of sampled triplet comparisons among different annotators. We show that by using our proposed sampling method to find an embedding, we are able to accurately represent synthetic hidden constructs in time under noisy sampling conditions. We further validate this approach using human annotations collected from Mechanical Turk and show that we can recover the underlying structure of the hidden construct up to bias and scaling factors.

1. Introduction

Continuous-time annotations are an essential resource for the computational study of hidden constructs such as human affect or behavioral traits over time. Indeed, the study of these hidden constructs is commonly tackled using regression techniques under a supervised learning framework, which heavily rely on accurately labeled features with respect to the constructs under study. Formally, regression problems deal with finding a mapping $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the *feature space*, and $\mathcal{Y}$ is the *label space*. Note that if $\boldsymbol{Y} \in \mathcal{Y}$ is indexed by time, then it is sometimes called a *continuous-time label*.[1] In this paper, we are interested in finding labels $\boldsymbol{Y} \in \mathcal{Y}$, such that $\boldsymbol{Y}$ is a good proxy for a hidden construct $\boldsymbol{Z} \in \mathcal{Z}$. As an example, in affective computing, $\boldsymbol{Z}$ is often a dimension of affect such as arousal (emotion intensity) or valence (emotion polarity), and it is assumed to be characterizable by data in the observation space $\mathcal{X}$ (e.g., audio, video, or bio-behavioral signals).

In the current literature, continuous-time labels in $\mathcal{Y} \subseteq \mathbb{R}^n$ are often generated from a set of continuous-time annotations acquired from a set of human raters or annotators $\mathcal{A}$. Each annotator $a \in \mathcal{A}$ uses perceptually interpretable features $\boldsymbol{X} \in \mathcal{X} \subseteq \mathbb{R}^{n \times q}$ to generate annotations $\boldsymbol{Y}_a \in \mathcal{Y}_a \subseteq \mathbb{R}^n$ about the construct $\boldsymbol{Z}$ [6,7,18]. In the sets above, n is the number of samples in time, and q represents the dimension of the set of perceptual features (e.g., audio levels, frames in a video) used for the real-time annotation acquisition. More generally, annotators are requested to do the mapping:

$$f_{\boldsymbol{Z}}^{a} : \mathcal{X} \to \mathcal{Y}_a, \tag{1}$$

$$\boldsymbol{X} \mapsto f_{\boldsymbol{Z}}^{a}(\boldsymbol{X}) = \boldsymbol{Y}_a, \tag{2}$$

where each $f_{\boldsymbol{Z}}^{a}$ is specific to annotator a for a construct $\boldsymbol{Z}$. Usually, several of these single annotations $\boldsymbol{Y}_a$ are collected from several annotators $a \in \mathcal{A}$, processed, and combined to create a single label $\boldsymbol{Y}$. This problem is called *annotation fusion*.

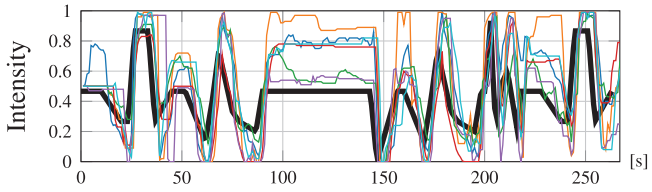
To train accurate statistical models, it is important that the labels $\boldsymbol{Y}$ used are precise and accurate, and properly reflect the variable $\boldsymbol{Z}$ under study [22]. Unfortunately, the annotation of hidden cues such as behavioral traits is a challenging problem due to several factors including diverse interpretations of the construct under study, differences in the perception of scale, improper design of the annotation-capturing tools, as well as disparate reaction times

☆ Haibin Yan, Ph.D.

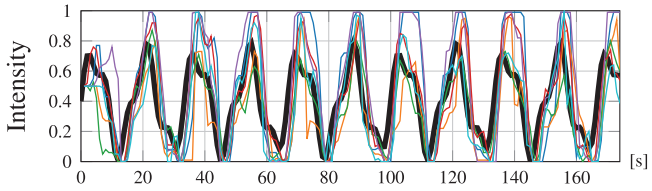
* Corresponding author.
E-mail address: mundnich@usc.edu (K. Mundnich).

[1] As opposed to discrete labels without time dependency.

https://doi.org/10.1016/j.patrec.2019.10.003
0167-8655/



(a) Task A: https://youtu.be/hOPfInpDD9E



(b) Task B: https://youtu.be/o3o5cUPBPAg

Fig. 1. Two real-time human annotation tasks with known ground truth $\boldsymbol{Z}$ (intensity of green over time, shown by the thick black lines). The annotators were presented with a user interface in which the video was shown, and they had to move a slider to match in real-time the current intensity of green observed. Six annotations are plotted in each task. Different colors represent each annotation $\boldsymbol{Y}_a$ done in real-time by a different annotator $a \in \mathcal{A}$ in a synthetic data experiment. For more details, please refer to [4]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

[4,14,18]. All of these affect the fidelity of individual annotations $\boldsymbol{Y}_a$.

To better study these challenges and the efficacy of algorithms to generate $\boldsymbol{Y}$, we build upon perceptual annotation tasks proposed previously in [4] where the ground truth $\boldsymbol{Z}$ is known, as a way to evaluate annotation fusion and correction algorithms. We proposed these tasks to decouple the problems of annotations themselves and the interpretation of hidden constructs. Fig. 1 shows the outcome of these experiments in [4], where nine human annotators were asked to annotate the intensity of green color (varying continuously between 0 and 1) in two different tasks (A and B) by moving a slider while watching the videos to match the intensity they were observing. We invite the readers to look at the videos referenced in the caption of Fig. 1 to directly experience what was presented to the annotators. More complex real-world scenarios with coupled problems will be the subject of a future communication. In Fig. 1, six annotations are plotted for clarity for each task. Fig. 1 exhibits many of the artifacts that complicate the fusion of continuous-time annotations: variable reaction times [15], overshooting fast changes, time-varying biases, disparate interpretations of scale, and difficulties in annotating constant intervals of the variable under study (mainly due to real-time corrections in the annotation process of the annotators themselves).

1.1. Related work

Related recent research has attempted to estimate an underlying construct $\boldsymbol{Z}$ by using continuous-time annotations $\boldsymbol{Y}_a$. Different works have addressed a subset of the aforementioned challenges (time lags, scale interpretations). For example, [14,15] study and model the reaction lag of annotators by using features from the data and shift each annotation before performing a simple average to fuse them, thus creating a unique label (EvalDep). Dynamic time warping (DTW) proposed by [19] is another popular time-alignment method that warps the signals in time to maximize time-alignment, which is usually combined with weighted averaging of signals. [23] proposes the use of a Long-Short-Term-Memory network (LSTM) to fuse asynchronous input annotations, by conducting time-alignment and de-biasing the different annotations. [10] presents a method for modeling multiple annotations over a continuous variable, and computes the ground truth by modeling annotator-specific distortions as filters whose parameters can be estimated jointly using Expectation-Maximization (EM). However, this work relies on heavy assumptions in the models for mathematical tractability, that do not necessarily reflect how annotators behave. All of the aforementioned works involve post-processing the raw continuous-time annotations, and performing the annotation fusion by averaging weighted signals in different (non)linear ways.

A different set of approaches is used to learn a warping function so that the fusion better correlates with associated features [11,21]. These spatial-warping methods can be combined with time warping [26,30,31]. All of these approaches rely on using a set of features.

In [4], we proposed a framework based on triplet embeddings to correct a continuous-time label generated by a fusion algorithm. This approach warps the fused label by selecting specific windows of it in time to collect extra information from human annotators through triplet comparisons. In [5], we also used triplet embeddings to fuse real-time annotations directly, by using majority voting to make a decision for each query. However, in these works the question of whether triplet comparisons *alone* can be used to generate the label $\boldsymbol{Y}$ is not studied.

A Triplet Embedding approach to learn metrics from multimodal data was first proposed in [17]. The authors develop an algorithm to account for noisy triplet labels (the notion of noisy labels was initially observed by [9] in music applications). In [17], the authors use their proposed algorithm to embed artists based on their (subjective) similarities. In [16] the authors introduce the idea of using ranking information extracted from metric leaning approaches for the comparison of music applied to recommender systems. However, none of these works use triplet embeddings to model the dynamics of subjective constructs over time. This is the topic of this paper.

1.2. Contributions

In this paper we study the performance of a new methodology to acquire and create a single label for regression by changing the sampling procedure of the latent construct. We sample this information by asking annotators questions of the form "is the signal in time-frame i more similar to the signal in time-frame j or k?" to build a 1-dimensional embedding $\boldsymbol{Y}$ in Euclidean space, where (i, j, k) forms a *triplet*. Fig. 2 shows an example of a query in the proposed sampling method where the comparison is based on the perceived shade (intensity) of the color.

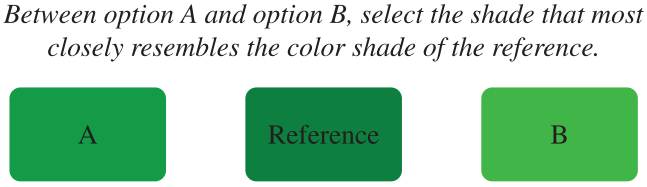


Fig. 2. Question design for queries in Mechanical Turk.

Formally, we propose that annotators perform the following mapping:

$$f^a_{\mathcal{Z}} : \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \times \boldsymbol{\mathcal{X}} \to \{-1, +1\}, \tag{3}$$

$$\left(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k\right) \mapsto \operatorname{sign}\left(d^a_{\mathcal{Z}}(\boldsymbol{x}_i, \boldsymbol{x}_k) - d^a_{\mathcal{Z}}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right) = w^a_t, \tag{4}$$

where $d^a_{\mathcal{Z}}$ is the perceived dissimilarity of construct $\mathcal{Z}$ by annotator a. We use a set of queried triplets $\{(i, j, k)\}$ and the corresponding annotations $\{w^a_t = \operatorname{sign}(d^a_{\mathcal{Z}}(\boldsymbol{x}_i, \boldsymbol{x}_k) - d^a_{\mathcal{Z}}(\boldsymbol{x}_i, \boldsymbol{x}_j))\}$ to calculate the embedding $\boldsymbol{Y}$.

We motivate this approach using three key observations. First, psychology and machine learning/signal processing studies have shown that people are better at comparing than rating items [18,20,28,29], so this sampling mechanism is easier for annotators than requesting absolute ratings in real-time. Second, the use of triplet embeddings naturally solves the annotation fusion problem, since it is done by taking the union of sets (details in Section. 3). Third, triplet embeddings offer a simple way of verifying the agreement of the annotations, given by the number of triplet violations in the computed embedding.

We empirically show that it is possible to reconstruct the hidden green intensity signal (i.e., recover the metric information) of tasks A and B in Fig. 1 under different synthetic noise scenarios in the triplet labeling stage. These reconstructions are accurate up to a scaling and bias factor but do not suffer from artifacts such as time-lags present in real-time annotations. Moreover, to test our approach, we gather triplet comparisons for the same experiments from human annotators in Amazon Mechanical Turk and show that it is possible to reconstruct the hidden green intensity values over time up to scaling and bias factors when humans perform the triplet comparisons. Finally, we compare our results to two continuous-annotation fusion algorithms recently proposed in the literature to show the strengths of our method.

2. Background: triplet embeddings

We first recall the general setting of Triplet Embeddings from a probabilistic perspective [12]. Let $\boldsymbol{z}_1, \dots, \boldsymbol{z}_n$ be items that we want to represent through points $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n \in \mathbb{R}^m$, respectively, with $[\boldsymbol{y}_1 \dots \boldsymbol{y}_n] = \boldsymbol{Y} \in \mathbb{R}^{m \times n}$. We assume that the items $\{\boldsymbol{z}_i\}$ lie in a metric space, and the Euclidean distances between them are given by $\boldsymbol{D}^*_{ij} = \left\|\boldsymbol{z}_i - \boldsymbol{z}_j\right\|_2^2$. We also assume that we have access to noisy distance comparisons, denoted by $d(\boldsymbol{z}_i, \boldsymbol{z}_j)$. These noisy distances may be perceptual, such as comparisons of expressed affect in the context of affective computing. We use these noisy distances to examine comparisons of the form:

$$d(\boldsymbol{z}_i, \boldsymbol{z}_j) \overset{?}{\lesseqgtr} d(\boldsymbol{z}_i, \boldsymbol{z}_k) \tag{5}$$

to find the embedding $\boldsymbol{Y}$.

Formally, let $\mathcal{T}$ be the set of all possible unique triplets for n items:

$$\mathcal{T} = \{(i, j, k) \mid i \neq j < k \neq i, 1 \leq i, j, k \leq n\}. \tag{6}$$

Note that $|\mathcal{T}| = n\binom{n-1}{2} = \mathcal{O}(n^3)$, which may be a very large set. We *observe* a set of triplets $\mathcal{S}$, such that $\mathcal{S} \subseteq \mathcal{T}$, and corresponding realizations of the random variables w_t, where $t = (i, j, k) \in \mathcal{S}$, such that:

$$w_t = \begin{cases} -1, & \text{w.p. } f\left(\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}\right) \\ +1, & \text{w.p. } 1 - f\left(\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}\right). \end{cases} \tag{7}$$

Here, $f : \mathbb{R} \to [0, 1]$ is a function that behaves as a cumulative distribution function [8] (sometimes called link function), and therefore has the property that $f(-x) = 1 - f(x)$. Hence, the w_t's indicate if i is closer to j than k, with a probability depending on the difference $\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}$ (or the difficulty of the annotation task).

Let $\boldsymbol{G} = \boldsymbol{Y}^\top \boldsymbol{Y}$ be the Gram matrix of the embedding. We can estimate $\boldsymbol{G}$ (and hence $\boldsymbol{Y}$) by minimizing the empirical risk:

$$\widehat{R}_{\mathcal{S}}(\boldsymbol{G}) = \frac{1}{|\mathcal{S}|} \sum_{t \in \mathcal{S}} \ell(w_t \langle \boldsymbol{\mathcal{L}}_t, \boldsymbol{G} \rangle_F), \tag{8}$$

where ℓ is a (margin-based) loss function and $\boldsymbol{\mathcal{L}}_t$ is defined as:

$$\boldsymbol{\mathcal{L}}_t = \begin{array}{c} \\ i \\ j \\ k \end{array} \begin{array}{c} \begin{array}{ccc} i & j & k \end{array} \\ \begin{bmatrix} 0 & -1 & 1 \\ -1 & 1 & 0 \\ 1 & 0 & -1 \end{bmatrix} \end{array}, \tag{9}$$

and zeros everywhere else, so that the Frobenius inner product $\langle \boldsymbol{\mathcal{L}}_t, \boldsymbol{G} \rangle_F = \|\boldsymbol{y}_i - \boldsymbol{y}_k\|_2^2 - \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2$ (and therefore, w_t contributes only a sign). After minimizing Eq. 8, we can recover $\boldsymbol{Y}$ from $\boldsymbol{G}$ up to a rigid transformation using the SVD.

In a maximum likelihood framework, ℓ is induced by our choice of f, assuming that the w_t are independent. For example, if f is the logistic function $f(x) = 1/(1 + \exp(-x))$, the induced loss is the logistic loss $\ell(x) = \log(1 + \exp(x))$ [12]. This setup is equivalent to Stochastic Triplet Embeddings [27], since the logistic loss and softmax are equivalent.

[12] proves that the error $R(\hat{\boldsymbol{G}}) - R(\boldsymbol{G}^*)$ (where $\boldsymbol{G}^*$ is the true underlying Gram matrix associated to $\boldsymbol{D}^*$) is bounded with high probability if $|\mathcal{S}| = \mathcal{O}(mn \log(n))$ and consequently, $\left\|\hat{\boldsymbol{D}} - \boldsymbol{D}^*\right\|_F$ is also bounded. Therefore, the practical number of triplets that need to be queried is $\mathcal{O}(mn \log(n))$ instead of $\mathcal{O}(n^3)$.

When computing a 1-dimensional embedding (i.e., $m = 1$), each $y_i \in \mathbb{R}$ can be interpreted as the value that the embedding takes at time index i, therefore representing a time series.

3. Labeling triplets with multiple annotators

Eq. 7 shows a way to encode the decision of a single annotator when queried for a decision as in Eq. 5. However, for multiple annotators we need to extend this model. Let $\mathcal{A}$ be a set of annotators. We define $\mathcal{S}_a$ as the set of triplets annotated by annotator $a \in \mathcal{A}$, so we observe a random variable w_t^a for each $t \in \mathcal{S}_a$. The labels are defined as:

$$w_t^a = \begin{cases} -1, & \text{w.p. } f_a\left(\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}\right) \\ +1, & \text{w.p. } 1 - f_a\left(\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}\right). \end{cases} \tag{10}$$

where f_a is the function that drives the probabilities for each annotator.

3.1. Annotation fusion

Due to annotation costs, we choose the sets $\mathcal{S}_a$ such that they are disjoint:

$$\mathcal{S} = \bigcup_{a \in \mathcal{A}} \mathcal{S}_a \quad \text{and} \quad \bigcap_{a \in \mathcal{A}} \mathcal{S}_a = \varnothing, \tag{11}$$

so that all queries are unique and any annotated triplet (i, j, k) is labeled by at most one annotator.

Note that the fusion process occurs in this step: The annotation fusion in a triplet embedding approach is done by taking the union of all the individually generated sets $\mathcal{S}_a$ to generate a single set of triplets $\mathcal{S}$, and using all corresponding labels w_t^a, defined for each annotator and each corresponding triplet $t \in \mathcal{S}$.

One difficulty of this multi-annotator model is that the distribution of w_t^a depends on the annotators through f_a, and, hence, the loss function is annotator-dependent. Fortunately, in our experiments, we can assume $f_a = f$, as we show experimentally in Fig. 4. We will extend this to annotator-dependent distributions in a future communication.

3.2. Triplet violations and annotation agreements

Triplet violations occur when a given triplet $t = (i, j, k) \in \mathcal{S}$ does not follow the calculated embedding $\boldsymbol{Y}$:

$$\|\boldsymbol{y}_i - \boldsymbol{y}_k\|_2 < \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2, \quad (i, j, k) \in \mathcal{S}. \tag{12}$$

Therefore, we can count the fraction of triplet violations using:

$$\tau_v = \frac{1}{|\mathcal{S}|} \sum_{(i,j,k) \in \mathcal{S}} \delta\left[\|\boldsymbol{y}_i - \boldsymbol{y}_k\|_2 < \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2\right], \tag{13}$$

where $\delta[\cdot]$ is Kronecker's delta.

To compute the expected number of correctly labeled triplets in $\mathcal{S}$, we can derive another random variable that models the correct annotation of triplet $t = (i, j, k)$ based on f:

$$c_t = \begin{cases} 0, & \text{w.p. } 1 - f\left(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|\right) \\ 1, & \text{w.p. } f\left(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|\right), \end{cases} \tag{14}$$

where $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ is the probability of successfully annotating triplet (i, j, k).

Using Eq. 14 we can model the number of correctly labeled triplets as a Poisson binomial random variable C:

$$C = \sum_{t=(i,j,k)\in\mathcal{S}} c_t \sim \text{PBD}\left(f\left(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|\right), |\mathcal{S}|\right). \tag{15}$$

Its expected value is the sum of the success probabilities:

$$\mathbb{E}[C] = \sum_{t=(i,j,k)\in\mathcal{S}} f\left(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|\right). \tag{16}$$

After computing $\boldsymbol{Y}$ from $\mathcal{S}$, and assuming that the optimization routine has found the best possible embedding $\boldsymbol{Y}$ for $\mathcal{S}$, then the fraction of triplet violations τ_v in $\boldsymbol{Y}$ is linearly related to C by:

$$\tau_v = 1 - C/|\mathcal{S}|, \text{ or } \mathbb{E}[\tau_v] = 1 - \mathbb{E}[C]/|\mathcal{S}|. \tag{17}$$

$\tau_v \in [0, 1]$ is a measure of disagreement between all triplets used to compute the embedding $\boldsymbol{Y}$. $\tau_v = 0$ means that all used triplets agree with the computed embedding $\boldsymbol{Y}$, meaning that all triplet labels agree with each other.

4. Experiments

We conduct two simulation experiments and one human annotation experiment using Mechanical Turk to verify the efficacy of our approach. We use the two synthetic data sets proposed in [4], for which the values for $\boldsymbol{Z}$ are known. We use this data because the reconstruction errors can be computed and we can assess the quality of the resulting labels, in contrast to experiments with affect, where the underlying signal is unknown. The two tasks correspond to videos of green frames with varying intensity of color over time and where the hidden construct $\boldsymbol{Z}$ is the intensity of green color (shown in thick black lines in Fig. 1). The video in task A is 267s long, and 178s long in task B.

To construct our triplet problem we first downsample the videos to 1Hz, so that the number of frames n equals the length of the video in seconds to reduce the number of unique triplets. We also set the dimension m to 1, since we want to find a 1-dimensional embedding that represents the intensity of green color over time.

Our experiments are implemented in Julia v1.0 [3], and available at www.github.com/kmundnic/PRL2019.

4.1. Synthetic triplet annotations

We simulate the annotation procedure by comparing the scalar green intensity values of frames of the video using the absolute value of the difference between points. Hence, the dissimilarity for Eq. 5 is $d(z_i, z_j) = |z_i - z_j|$, where i and j are time indices.

We generate a list of noisy triplets $\mathcal{S}$ by randomly and uniformly selecting each triplet (i, j, k) from the pool of all possible unique triplets. Each triplet $t = (i, j, k)$ is *correctly* labeled by w_t with probability $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$.

We test eight different fractions of the total possible number of triplets $|\mathcal{T}|$ using logarithmic increments such that $|\mathcal{S}| = \{0.0005, \ldots, 0.1077\}|\mathcal{T}|$, which goes from 0.05% to 10.77% of $|\mathcal{T}|$. We use a logarithmic scale to have more resolution for smaller percentages of the total number of possible unique triplets. Note that for 267 frames (task A), the total number of unique triplets is 9,410,415. The queried triplets are randomly and uniformly sampled from all possible unique triplets, since there is no guarantee of better performance for active sampling algorithms in this problem [13].

We use various algorithms available in the literature to solve the triplet embedding problem: Stochastic Triplet Embeddings (STE) [27] (with $\sigma = 1/\sqrt{2}$) and t-Student Stochastic Triplet Embeddings (tSTE) [27] (with $\alpha \in \{2, 10\}$), Generalized Non-metric Multidimensional Scaling (GNMDS) [1] (parameter-free) with hinge loss, and Crowd Kernel Learning (CKL) [24] (with $\mu \in \{2, 10\}$). We use gradient descent to optimize all the loss functions proposed by the algorithms. Note that STE and GNMDS pose convex problems, while tSTE and CKL pose non-convex problems, and therefore we perform 30 different random starts for each set of parameters.

We now describe the three experimental settings we use to validate our approach.

Simulation 1: Constant success probabilities We choose $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ to be approximately constant, such that the probability $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|) = \mu + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.01$ (to add small variations). We run three different experiments for $\mu \in \{0.7, 0.8, 0.9\}$.

Picking the values of $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ randomly affects our calculation of $\mathbb{E}[C]$ (Eq. 16), but we will assume that these have been fixed *a priori*, meaning that the annotation process has a fixed probability for labeling any triplet (i, j, k).

Simulation 2: Logistic probabilities A more realistic simulation is given by labeling the triplets in $\mathcal{S}$ according to the following probabilities:

$$f\left(\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}\right) = \frac{1}{1 + \exp\left(-\sigma\left(\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}\right)\right)}, \tag{18}$$

which is the logistic function. We use different values for $\sigma = \{2, 6, 20\}$. Intuitively, the triplets with smaller differences between $\boldsymbol{D}^*_{ij}$ and $\boldsymbol{D}^*_{ik}$ should be harder to label, and a more realistic noise model than constant errors independent of the difficulty of the task. Note that this noise model induces the logistic loss used in STE.

4.2. Mechanical Turk triplet annotations

Using the list of images generated earlier we sample 0.5% of the total number of triplets of images randomly and uniformly. In this setting, we sample approximately $Kn\log(n)$ triplets, with $K = 31.5$ for task A, and $K = 15$ for task B. To compute the embedding we use STE with parameter $\sigma = 1/\sqrt{2}$.

To obtain the list of annotated triplets, we show the annotators options A and B against a reference, and instructions as in Fig. 2. We do not provide further instructions for the case where $\boldsymbol{D}^*_{Reference,A} \approx \boldsymbol{D}^*_{Reference,B}$. For this task, we paid the annotators $0.02 per answered query.

4.3. Error measure

We use the error measure proposed in [25], and compute the error by first solving the following optimization problem:

$$\text{MSE} = \inf_{a,b} \frac{1}{n} \|a\boldsymbol{Y} - b\boldsymbol{1} - \boldsymbol{Z}\|_2^2, \tag{19}$$

where $a, b \in \mathbb{R}$ are the scaling and bias factors, and n is the length of $\boldsymbol{Y}$. We use this MSE and not a naive MSE between the ground truth $\boldsymbol{Z}$ and the reconstructed label $\boldsymbol{Y}$ because the embeddings are optimal only up to scaling and bias factors. Hence, this approach yields a more fair assessment of the quality of the embedding.

We also report Pearson's correlation ρ between the ground truth and the estimated embedding, to compare our method with other proposed algorithms in a scale-free manner.

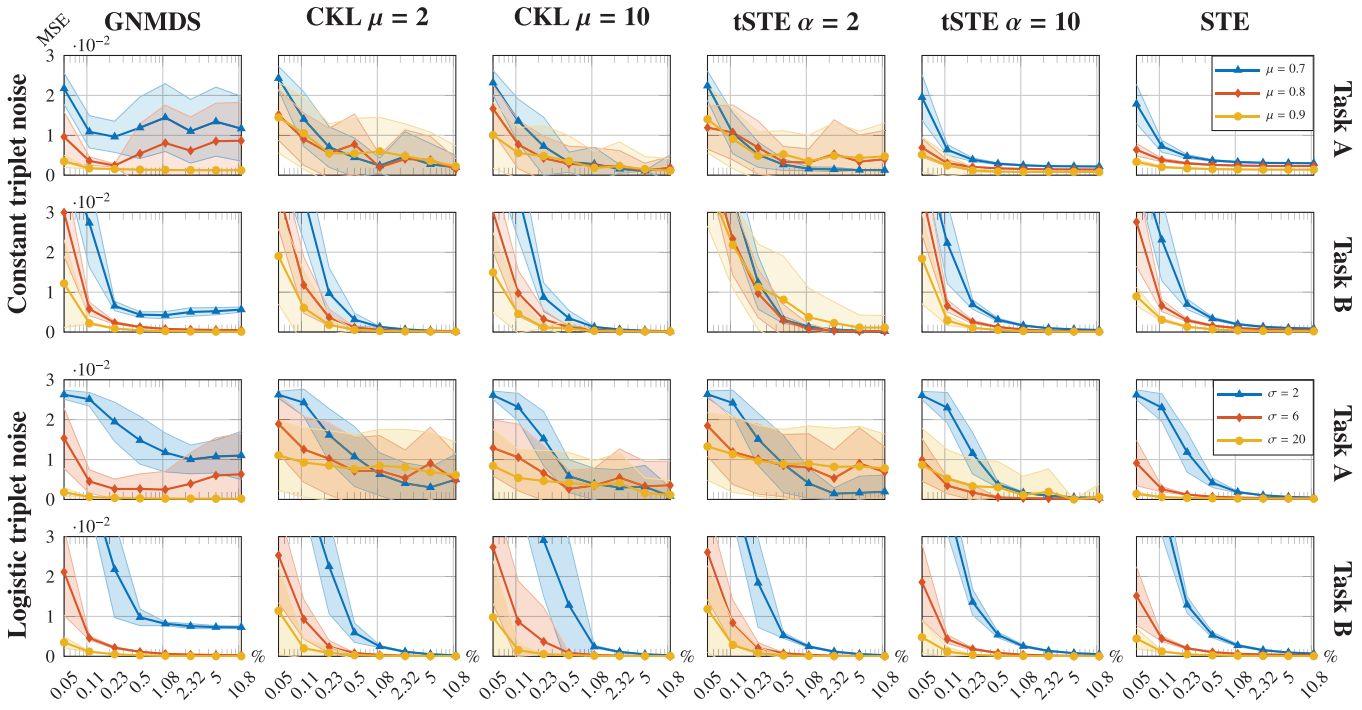


Fig. 3. MSE as a function of the number of observed triplets $|\mathcal{S}|$ with constant and logistic noise in triplet labels. Each point in the plots represents the mean over 30 random trials, while the shaded areas represent one standard deviation from the average MSE values.

4.4. Comparison to other methods

We compare the proposed annotation and fusion framework with two different approaches using real-time annotations: EvalDep [15] and the EM-based approach (after time-alignment using EvalDep's method) from [10] with window lengths of 4, 8, 16, and 32.

5. Results and analysis

5.1. Synthetic annotations

Fig. 3 shows the MSEs as a function of $|\mathcal{S}|/|\mathcal{T}| \times 100$ for both synthetic experiments. For both constant and logistic noise in tasks A and B we generally obtain a better performance as the amount of noise in the triplet annotation process is reduced (larger μ or σ). This is not always true in the algorithms that propose non-convex loss functions (tSTE, CKL), where sometimes more noise generates better embeddings. We *hypothesize* that these algorithms sometimes find better local minima under noisier conditions.

The MSE in Fig. 3 typically becomes smaller as $|\mathcal{S}|$ increases. This is true (generally) for tSTE, STE, and CKL. GNMDS does not always produce a better embedding by increasing the number of triplets employed.

We also note that the embedding in task B is easier to compute than that of Task A. We observe two possible reasons for this: (1) Task A has constant intervals while task B has none (and constant regions may be harder to compute in noisy conditions), and (2) the extreme values in task A seem harder to estimate, since these occur for very short intervals of time that are less likely to be sampled.

Overall, STE is the best-performing algorithm independent of noise or task. We note that tSTE with $\alpha = 10$ *approaches* STE in many of the presented scenarios. In fact, tSTE becomes STE with $\sigma \to 1$ as $\alpha \to \infty$, so these results are expected (we do not include the proof due to space restrictions).

5.2. Mechanical Turk triplet annotations

5.2.1. Annotator noise

In the Mechanical Turk experiments, 170 annotators annotated triplets in task A, and 153 in task B. To understand the difficulty of the tasks and the noise distributions for the annotators, we estimate the probabilities of success $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ for both tasks, using the top three annotators.

To estimate $f(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$, we partition the triplets based on $|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|$ into intervals with the same number of triplets. For each interval, we compute the average distance of the triplets. For each triplet $(i, j, k) \in \mathcal{I}$, we know the outcome (realization) of the random variable w^a_{ijk} since we know the hidden construct $\boldsymbol{Z}$. We assume that the success probability $f_a(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ is constant in this interval, so that $C \sim \text{Binomial}(f_a(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|))$. Finally, we use the maximum likelihood estimator for success probabilities for each interval:

$$\hat{f}_{I^a}\left(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|\right) = \frac{1}{|\mathcal{I}^a|} \sum_{(i,j,k)\in\mathcal{I}^a} c^a_{ijk}. \tag{20}$$

In Fig. 4, we show the function $\hat{f}_a(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ for each of the top annotators with the most answered queries and compare it to the logistic function with $\sigma = 20$. The comparison between the estimated probabilities of success and the logistic function shows that this is a very good noise model for this annotation task, while also telling us that we should expect the best results from STE when computing the embedding from the crowd-sourced triplet annotations. *Noticeably, our initial assumption of an annotator-independent noise model is verified.*

5.2.2. Mechanical Turk embedding

We present in Fig. 5 the results for the reconstructed embeddings using triplets generated by annotators via Mechanical Turk. We show the reconstructed embeddings obtained using 0.5% of the total number of triplets $|\mathcal{T}|$ for each task. Although there is some visible error, we are able to capture the trends and overall shape

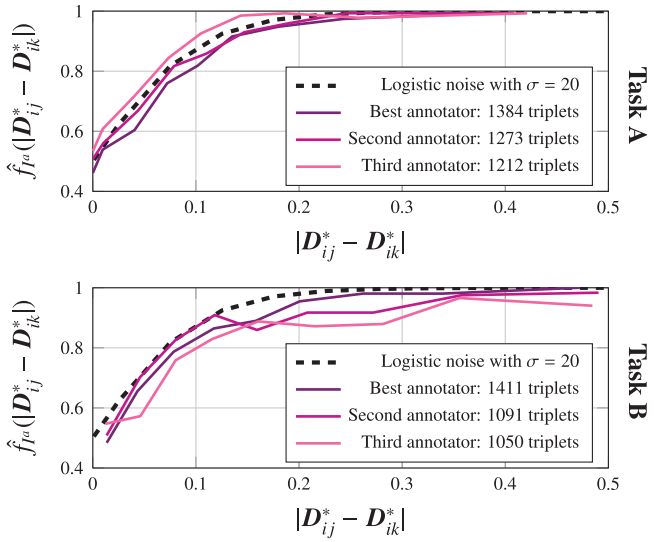


Fig. 4. Probabilities of success $\hat{f}_{I^\alpha}(|\boldsymbol{D}^*_{ij} - \boldsymbol{D}^*_{ik}|)$ as a function of the distance from the reference i to frames j and k. Only the top annotators have been included.

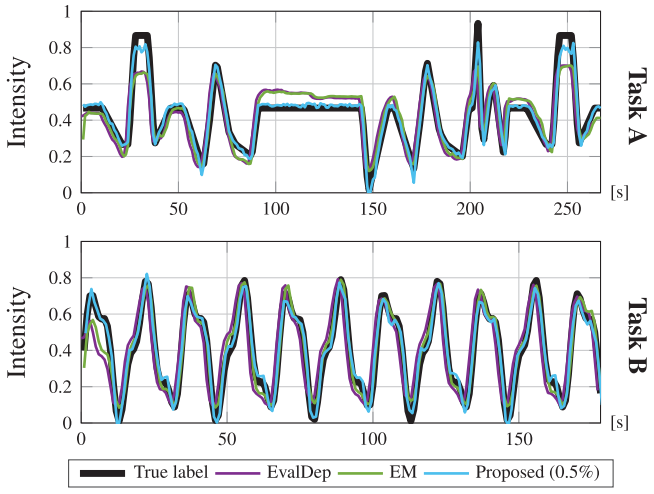


Fig. 5. Results for Mechanical Turk annotations. The computed embeddings have been scaled to fit the true labels $\boldsymbol{Z}$ (Eq. 19). The embedding in task A uses 0.5% (47,052) of all possible triplet comparisons $|\mathcal{T}|$. The embedding in task B uses 0.5% (13,862) of all possible triplet comparisons $|\mathcal{T}|$. In both tasks, the estimated green intensity is sometimes less than zero due to scaling. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the underlying construct with only 0.5% or fewer of all possible triplets for both tasks. We also plot a scaled version (according to Eq. 19) of the fused annotation obtained using the EvalDep method and using the continuous-time annotations from [4] (Fig. 1). In this figure, we observe that fusion methods based on continuous-time annotations are not able to debias the annotations in windows of time that are biased. Our proposed method does not suffer from this issue.

We show in Table. 1 the MSE for each task, where percentages again represent the number of triplets employed. We have also included the MSE and ρ for the embedding produced with 0.25% of the triplets (not included in Fig. 5 due to the high overlap between the 0.25% and 0.5% embeddings). We observe that the MSE is lower for a higher number of labeled triplets used. This is expected: there is more information about the embedding as we increase the number of triplets that we feed into the optimization routine, therefore producing a higher quality embedding. We also

Table 1
MSE and Pearson's correlation ρ for the proposed method and state of the art continuous-time fusion techniques against ground truth. For our method, percentage is with respect to the total number of triplets.

Task	Fusion technique	MSE	ρ
A	EvalDep [15]	0.00489	0.906
	EM [10] (best, window length: 16)	0.00494	0.903
	Proposed (0.25%)	0.00145	0.973
	Proposed (0.50%)	**0.00132**	**0.975**
B	EvalDep [15]	0.00304	0.969
	EM [10] (best, window length: 32)	**0.00241**	**0.975**
	Proposed (0.25%)	0.00305	0.969
	Proposed (0.50%)	0.00285	0.971

Table 2
Triplet violations τ_v for the Mechanical Turk experiment. Percentages correspond to percentage of total triplets observed. We include the fraction of triplet violations as computed by the labels generated with EvalDep.

Task	Triplet violations τ_v				
	MTurk	Y (0.25%)	Y (0.5%)	EvalDep [15]	EM [10]
A	0.122	0.159	0.161	0.262	0.259
B	0.179	0.146	0.129	0.139	0.124

show a scale-free comparison through Pearson's correlation, which captures how signals vary over time and neglects differences in scale and bias. In task A, our approach improves upon previous work by a large margin. In task B, our approach performs comparably to the EM-based method. Our understanding suggests that the EM-based algorithm benefits from a smooth ground truth, given their filter-modeling approach on a given window size.

5.2.3. Triplet violations and annotator agreement

Table. 2 displays the number of triplet violations for each task. We record the true percentage of triplet violations according to our ground truth (generated using distances $d(z_i, z_j)$ and $d(z_i, z_k)$, as in Eq. 12) and then compare them to the annotation responses. We also display the number of triplet violations according to the computed embeddings $\boldsymbol{Y}$. We see that the percentage of triplet violations according to our ground truth and the triplet violations calculated from the embeddings $\boldsymbol{Y}$ is not the same, being overestimated in task A and underestimated in task B. We also observe that even if the number of violations increases in task A, the MSE is reduced with a larger number of triplets. This happens because a higher number of triplet constraints more easily define an embedding.

6. Discussion

Section. 5 shows that it is possible to use triplet embeddings to find a 1-dimensional embedding that resembles the true underlying construct up to scaling and bias factors. There are several additional considerations for our proposed method.

Annotation costs One of the challenging aspects of using triplet embeddings is the $\mathcal{O}(n^3)$ growth of the number of unique triplets for n objects or frames. As mentioned earlier, the results by [12] suggest however that the theoretical number of triplets needed scales with $\mathcal{O}(mn\log(n))$. In our experiments, we use $Kn\log(n)$ triplets with $K = 31.5$ for task A and $K = 15$ for task B to achieve equivalent or better approximations of the underlying ground truth compared to the state-of-the-art.

Computational costs Triplet embeddings are computationally cheap in comparison to the other methods employed in this paper, since they can be efficiently estimated using gradient-based methods to minimize the loss function. Moreover, in our current implementation, the calculation of the gradient is parallelized, and the average time to estimate the embeddings for task A and B using 0.5% of $|\mathcal{T}|$ in Fig. 5 is 125 ms and 44 ms respectively over 100

trials using 10 threads on a laptop with an Intel i7-8850H processor and 32Gb of RAM.

The evaluation of the gradient for a fixed dimension m of the embedding scales linearly with the number of triplets employed (where the number of triplets needed is $\mathcal{O}(n \log n)$. This is computationally cheap in comparison to other methods considered: for example, time-alignment (needed in all continuous-time fusion approaches) is an expensive operation. For a signal of length n, alignment using Dynamic Time Warping is $\mathcal{O}(n^2)$, and EvalDep [14,15] needs to compute the determinants of two $n \times n$ matrices and one $2n \times 2n$ matrix to estimate the mutual information, each being at least $\mathcal{O}(n^{2.373})$ (by using Fast Matrix Multiplication [2]). The EM-based approach also requires inverting an $n \times n$ matrix. As examples, EvalDep takes 8s in task A and 6s in task B, while EM takes on average 20 min for both tasks and different window lengths.

Embedding quality The embeddings reconstructed are more accurate than the method proposed in [15]. Moreover, no time-alignment is needed since the annotation process does not suffer from reaction times. It is also important to note that sharp edges (high frequency regions of the construct) are most appropriately represented and do not get smoothed out, as with averaging-based annotation fusion techniques (where annotation devices such as mice or joysticks and user interfaces perform low-pass filtering).

In terms of reconstruction, the scaling factor is an open challenge. We see two possible ways to work with the differences in scaling when the underlying construct is unknown: (1) Learn the scaling in a machine learning pipeline that uses these labels to create a statistical model of the hidden construct, or (2) normalize the embedding $\boldsymbol{Y}$ such that $\bar{\boldsymbol{Y}} = 0$ and $\sigma_{\boldsymbol{Y}} = 1$, and train the models using either these labels or the derivatives $d\boldsymbol{Y}/dt$. However, we note that continuous-time annotations do suffer from the same loss of scaling and bias, since both techniques are trying to solve an inverse problem where the scale is not accessible.

Feature sub-sampling for triplet comparisons In the experiments of this paper, we sub-sample the videos to 1Hz so that we have a manageable number of frames n. Down-sampling is possible due to the nature of the synthetic experiment we have created, but may not be suitable for other constructs such as affect in real world data, where annotation of single frames might lose important contextual information. In these scenarios, further investigation is needed to understand how to properly sub-sample more complex annotation tasks.

7. Conclusion

In this paper, we present a new sampling methodology based on triplet comparisons to produce continuous-time labels of hidden constructs. To study the proposed methodology, we use two experiments previously proposed in [4] and show that it is possible to recover the structure of the underlying hidden signals in simulation studies using human annotators to perform the triplet comparisons. These labels for the hidden signals are accurate up to scaling and bias factors.

Our method performs annotator fusion seamlessly as a union of sets of queried triplets $\mathcal{S}_a$, which greatly simplifies the fusion approach compared to existing approaches which directly combine real-time signals. Moreover, our approach does not need post-processing such as time-alignments or averaging.

Some challenges for the proposed method include dealing with the annotation costs given the number of triplets that needs to be sampled, and also learning the unknown scaling and bias factors.

As future directions, we are interested in several paths. We believe it is necessary to further study the proposed method for labeling constructs where the ground truth cannot be validated, as is the case of human emotions, and contrast the effects of using triplet comparisons to annotate individual frames and using triplet comparisons to annotate over frame sequences.

Declaration of Competing Interest

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgments

This work was supported by the National Science Foundation grant number 151454. We thank Anil Ramakrishna for sharing with us the code for the EM-based approach.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2019.10.003.

References

[1] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, S. Belongie, Generalized non-metric multidimensional scaling, in: Artificial Intelligence and Statistics, 2007, pp. 11–18.
[2] A.V. Aho, J.E. Hopcroft, J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.
[3] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: a fresh approach to numerical computing, SIAM Rev. 59 (2017) 65–98.
[4] B.M. Booth, K. Mundnich, S. Narayanan, A novel method for human bias correction of continuous-time annotations, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018a, pp. 3091–3095.
[5] B.M. Booth, K. Mundnich, S. Narayanan, Fusing annotations with majority vote triplet embeddings, in: Proceedings of the 2018 Audio/Visual Emotion Challenge and Workshop, ACM, 2018b, pp. 83–89.
[6] C. Busso, M. Bulut, S. Narayanan, Toward Effective Automatic Recognition Systems of Emotion in Speech. Social Emotions in Nature and Artifact: Emotions in Human and Human–Computer Interaction, J. Gratch and S. Marsella, 2013, pp. 110–127. Eds.
[7] R. Cowie, R.R. Cornelius, Describing the emotional states that are expressed in speech, Speech Commun. 40 (2003) 5–32.
[8] M.A. Davenport, Y. Plan, E. Van Den Berg, M. Wootters, 1-bit matrix completion, Inf. Inference 3 (2014) 189–223.
[9] D.P.W. Ellis, B. Whitman, A. Berenzweig, S. Lawrence, The quest for ground truth in musical artist similarity, in: In Proceedings of the International Symposium on Music Information Retrieval (ISMIR 2002), 2002, pp. 170–177.
[10] R. Gupta, K. Audhkhasi, Z. Jacokes, A. Rozga, S. Narayanan, Modeling multiple time series annotations as noisy distortions of the ground truth: an expectation-maximization approach, IEEE Trans. Affect. Comput. 9 (2018) 76–89, doi:10.1109/TAFFC.2016.2592918.
[11] H. Hotelling, Relations between two sets of variates, Biometrika 28 (1936) 321–377.
[12] L. Jain, K.G. Jamieson, R.D. Nowak, Finite sample prediction and recovery bounds for ordinal embedding, in: Advances in Neural Information Processing Systems, 2016, pp. 2711–2719.
[13] K.G. Jamieson, L. Jain, C. Fernandez, N.J. Glattard, R.D. Nowak, NEXT: a system for real-world development, evaluation, and application of active learning, in: Advances in Neural Information Processing Systems, 2015, pp. 2656–2664.
[14] S. Mariooryad, C. Busso, Analysis and compensation of the reaction lag of evaluators in continuous emotional annotations, in: Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on, IEEE, 2013, pp. 85–90.
[15] S. Mariooryad, C. Busso, Correcting time-continuous emotional labels by modeling the reaction lag of evaluators, IEEE Trans. Affect. Comput. 6 (2015) 97–108.
[16] B. McFee, L. Barrington, G. Lanckriet, Learning content similarity for music recommendation, IEEE Trans. Audio Speech Lang. Process. 20 (2012) 2207–2218.
[17] B. McFee, G. Lanckriet, Learning multi-modal similarity, J. Mach. Learn. Res. 12 (2011) 491–523.
[18] A. Metallinou, S. Narayanan, Annotation and processing of continuous emotional attributes: challenges and opportunities, in: Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on, IEEE, 2013, pp. 1–8.

[19] M. Müller, Dynamic time warping, Inf. Retr. Music Motion (2007) 69–84.
[20] N.S. Chater, G.D.A. Brown, Nick, Absolute identification by relative judgement, Psychol. Rev. 112 (2005) 881–911.
[21] M.A. Nicolaou, S. Zafeiriou, M. Pantic, Correlated-spaces regression for learning continuous emotion dimensions, in: Proceedings of the 21st ACM International Conference on Multimedia, ACM, 2013, pp. 773–776.
[22] V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, J. Mach. Learn. Res. 11 (2010) 1297–1322.
[23] F. Ringeval, F. Eyben, E. Kroupi, A. Yuce, J.P. Thiran, T. Ebrahimi, D. Lalanne, B.W. Schuller, Prediction of asynchronous dimensional emotion ratings from audiovisual and physiological data, Pattern Recognit. Lett. 66 (2015) 22–30.
[24] O. Tamuz, C. Liu, S. Belongie, O. Shamir, A.T. Kalai, Adaptively learning the crowd kernel, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 673–680.
[25] Y. Terada, U. von Luxburg, Local ordinal embedding, in: International Conference on Machine Learning, 2014, pp. 847–855.
[26] G. Trigeorgis, M.A. Nicolaou, B.W. Schuller, S. Zafeiriou, Deep canonical time warping for simultaneous alignment and representation learning of sequences, IEEE Trans. Pattern Anal. Mach.Intell. 40 (2018) 1128–1138.
[27] L. Van Der Maaten, K. Weinberger, Stochastic triplet embedding, in: Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on, IEEE, 2012, pp. 1–6.
[28] G.N. Yannakakis, J. Hallam, Ranking vs. preference: a comparative study of self-reporting, Affect. Comput. Intell. Interact. (2011) 437–446.
[29] G.N. Yannakakis, H.P. Martínez, Ratings are overrated!, Front. ICT 2 (2015) 13.
[30] F. Zhou, F. De la Torre, Canonical time warping for alignment of human behavior, in: Advances in Neural Information Processing Systems, 2009, pp. 2286–2294.
[31] F. Zhou, F. De la Torre, Generalized time warping for multi-modal alignment of human motion, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, 2012, pp. 1282–1289.