Action-Affect-Gender Classification using Multi-Task Representation Learning

Timothy J. Shields*, Mohamed R. Amer*, Max Ehrlich, and Amir Tamrakar
SRI International
firstName.lastName@sri.com

Abstract

Recent work in affective computing focused on affect from facial expressions, and not as much on body. This
work focuses on body affect. Affect does not occur in iso-
lation. Humans usually couple affect with an action; for
example, a person could be running and happy. Recogniz-
ing body affect in sequences requires efficient algorithms
to capture both the micro movements that differentiate be-
 tween happy and sad and the macro variations between dif-
ferent actions. We depart from traditional approaches for
time-series data analytics by proposing a multi-task learn-
ing model that learns a shared representation that is well-
suited for action-affect-gender classification. For this pa-
er we choose a probabilistic model, specifically Condi-
tional Restricted Boltzmann Machines, to be our building
block. We propose a new model that enhances the CRBM
model with a factored multi-task component that enables
scaling over larger number of classes without increasing
the number of parameters. We evaluate our approach on
two publicly available datasets, the Body Affect dataset and
the Tower Game dataset, and show superior classification
performance improvement over the state-of-the-art.

1. Introduction

Recent work in the field of affective computing [1] fo-
cus on face data [2], audiovisual data [3], and body data
[4]. One of the main challenges of affect analysis is that
it does not occur in isolation. Humans usually couple af-
f ect with an action in natural interactions; for example, a
person could be walking and happy, or knocking on a door
angrily as shown in Fig. 1. These activities are performed
differently given the gender of the actor. To recognize
body action-affect-gender, efficient temporal algorithms are
needed to capture the micro movements that differentiate
between happy and sad as well as capture the macro varia-
tions between the different actions. The focus of our work is

*Both authors equally contributed to this work
2. Prior work

In this section we first review literature on activity recognition in RGB-D and Motion Capture Sequences in Sec. 2.1; second we review Multi-Task Learning approaches in Sec. 2.2; finally we review temporal, energy-based, representation learning in Sec. 2.3.

2.1. Body Affect Analysis

Initial work on activity recognition in RGB-D sequences has been popular in recent years with the availability of cheap depth sensors. Since initial work [12], there have been an increasing number of approaches addressing the problem of activity recognition using skeletal data [9]. Prior to activity recognition in RGB-D sequences, datasets were captured using motion capture sensors. During that time, research focused on graphics applications such as generating animation and transitions between animations using signal processing techniques rather than machine learning or computer vision. Their main goal was to generate natural looking skeletons for animation. Some methods used knowledge of signal processing to transform a neutral skeleton pose to reflect a certain emotion [5]. These methods were very constrained to the type of motion and were engineered to reproduce the same motions. Other work used a language based modeling of affect [6] where they modeled actions (verbs) and affect (adverbs) using a graph. They were able to produce results using a combination of low level functions to interpolate between example motions. More recent work [13] modeling non-styled motion for affect communication used segmentation techniques which divided complex motions into a set of motion primitives that they used as dynamic features. Unlike our approach, their mid-level features were hand engineered rather than learned, which is very limited, does not scale and is prone to feature design flaws. More recent work such as [7] collected natural body affect datasets where they have varied identity, gender, emotion, and actions of the actors but not used it for classification.

2.2. Multi-Task Learning

Multi-task learning is a natural approach for problems that require simultaneous solutions of several related problems [14]. Multi-task learning approaches can be grouped into two main sets. The first set focuses on regularizing the parameter space. The main assumption is that there is an optimal shared parameter space for all tasks. These approaches regularize the parameter space by using a specific loss [15], methods that manually define relationships [16], or more automatic ways that estimate the latent structure of relationships between tasks [17, 18, 19, 20, 21]. The second set focuses on correlating relevant features jointly [22, 23, 24, 25]. Other work focused on the schedule of which tasks should be learned [26]. Multi-task learning achieved good results on vision problems such as: person re-identification [27], multiple attribute recognition [28], and tracking [29]. Recently, Deep Multi-Task Learning (DMTL) emerged with the rise of deep learning. Deep Neural Networks (DNNs) were used to address multi-task learning and were applied successfully to facial landmark detection [30], scene classification [31], object localization and segmentation [32] and attribute prediction [33]. Other work used multi-task autoencoders [34] for object recognition in a generalized domain [35], where the tasks were the different domains. Other work used multi-task RNNs for interaction prediction in still images [36].
Deep Multi-task Learning approaches only focused on using DNN-based models applied to still images. Our approach is the first DMTL for temporal and multimodal sequence analysis.

2.3. Representation Learning

Deep learning has been successfully applied to many problems [37]. Restricted Boltzmann Machines (RBMs) form the building blocks in energy-based deep networks [38, 39]. In [38, 39], the networks are trained using the Contrastive Divergence (CD) algorithm [40], which demonstrated the ability of deep networks to capture the distributions over the features efficiently and to learn complex representations. RBMs can be stacked together to form deeper networks known as Deep Boltzmann Machines (DBMs), which capture more complex representations. Recently, temporal models based on deep networks have been proposed, capable of modeling a rich set of time series analysis problems. These include Conditional RBMs (CRBMs) [41] and Temporal RBMs (TRBMs) [42, 43, 44]. CRBMs have been successfully used in both visual and audio domains. They have been used for modeling human motion [41], tracking 3D human pose [45], and phone recognition [46]. TRBMs have been applied for transferring 2D and 3D point clouds [47], and polyphonic music generation [48].

3. Model

Rather than immediately defining our Multi-Task CRBM (MT-CRBM) model, we discuss a sequence of models, gradually increasing in complexity, such that the different components of our final model can be understood in isolation. We start with the basic CRBM model (sec. 3.1), then we extend the CRBM to a new discriminative (D-CRBM) model (sec. 3.2), then we extend the D-CRBM to our main multi-task model (MT-CRBM) (sec. 3.3), and finally we define a multi-task multimodal model (MTM-CRBM) (sec. 3.4).

3.1. Conditional Restricted Boltzmann Machines (CRBMs)

CRBMs [41] are a natural extension of RBMs for modeling short term temporal dependencies. A CRBM, shown in Figure 2(a), is an RBM which takes into account history from the previous \(N\) time instances, \(t - N, \ldots, t - 1\), when considering time \(t\). This is done by treating the previous time instances as additional inputs. Doing so does not complicate inference. Some approximations have been made to facilitate efficient training and inference, more details are available in [41]. A CRBM defines a probability distribu-
tion $p_c$ as a Gibbs distribution (1).

$$
p_c(h_t, v_t | v_{<t}) = e^{-E_c(h_t, v_t | v_{<t})}/Z(\theta),
$$

$$
Z(\theta) = \sum_{h, v} e^{-E_c(h_t, v_t | v_{<t})},
$$

$$
\theta = \left[ \begin{array}{c}
\{ a, b \} - \text{bias}, \\
\{ A, B \} - \text{auto regressive}, \\
\{ W \} - \text{fully connected}
\end{array} \right].
$$

The visible vectors from the previous $N$ time instances, denoted as $v_{<t}$, influence the current visible and hidden vectors. The probability distributions are defined in (2).

$$
p_c(v_i | h_t, v_{<t}) = \mathcal{N}(c_i + \sum_j h_j w_{ij}, 1),
$$

$$
p_c(h_j = 1 | v_t, v_{<t}) = \sigma(d_j + \sum_i v_i w_{ij}),
$$

$$
c_i = a_i + \sum_p a_p v_{p,i,t}, \quad d_j = b_j + \sum_p B_{pj} v_{p,i,t}.
$$

The new energy function $E_c(h_t, v_t | v_{<t})$ in (3) is defined in a manner similar to that of the RBM.

$$
E_c(h_t, v_t | v_{<t}) = \sum_i (c_i - v_i)^2 - \sum_j d_j h_j - \sum v_i w_{ij} h_{j,t}.
$$

The probability distribution over the visible layer will follow the same distributions as in (2). The hidden layer $h$ is defined as a function of the labels $y$ and the visible nodes $v$. A new probability distribution for the classifier is defined to relate the label $y$ to the hidden nodes $h$ (5).

$$
p_d(y_{<t}, h_t, v_t | v_{<t}) = e^{-E_d(y_{<t}, h_t, v_t | v_{<t})}/Z(\theta)
$$

The new energy function $E_d$ is defined as in (6).

$$
E_d(y_{<t}, h_t, v_t | v_{<t}) = E_c(h_t, v_t | v_{<t}) + \sum_{k,l} s_{k,l} y_{k,t} - \sum_{j,k} h_{j,t} w_{j,k} y_{k,t}.
$$

### 3.3. Multi-Task CRBMs (MT-CRBMs)

In the same way the CRBMs can be extended to the DC-RBMs by adding a discriminative term to the model, we can extend the CRBMs to be multi-task MT-CRBMs Figure 2(c). MT-CRBMs define the probability distribution $p_{MT}$ as a Gibbs distribution (7). The MT-CRBMs learn a shared representation layer for all tasks.

$$
p_{MT}(y_t, h_t, v_t | v_{<t}) = e^{-E_{MT}(y_t, h_t, v_t | v_{<t})}/Z(\theta).
$$

### 3.4. Multimodal MT-CRBMs (MTM-CRBMs)

We can naturally extend MT-CRBMs to MTM-CRBMs. A MTM-CRBMs combines a collection of unimodal MT-CRBMs, one for each visible modality. The hidden representations produced by the unimodal MT-CRBMs are then treated as the visible vector of a single fusion MT-CRBMs. The result is a MTMCRBM model that relates multiple temporal modalities to multi-task classification labels. MTM-CRBMs define the probability distribution $p_{MTM}$ as a Gibbs distribution (10). The MTM-CRBMs learn an extra representation layer for each of the modalities, which learns a modality specific representation as well as the shared layer for all the tasks.

$$
p_{MTM}(y_t, h_t, v_{t:M} | v_{<t:M}) = \exp[-E_{MTM}(y_t, h_t, v_{t:M} | v_{<t:M})]/Z(\theta).
$$
The new energy function \( E_{\text{MTM}} \) is defined in (13) similar to that of the MT-CRBMs (7).

\[
E_{\text{MTM}}(y^L_t, h_t, h^{1:M}_t, v^{1:M}_t | v^{<t}_c) = \\
\sum_m E_{\text{MT}}(y^L_t, h^m_t, v^m_t | v^m_{<t}) \\
- \sum_j f_n h_{n,t} - \sum_{j,k,m} h^m_{j,t} v_{j,n} h_{n,t} \\
- \sum_{k,l} s^L_k y_{k,t} - \sum_{n,k,l} h_{n,t} u_{n,k} y_{k,t}
\]

4. Inference

We first discuss inference for the MTM-CRBM since it is the most general case. To perform classification at time \( t \) in the MTM-CRBM given \( v^{1:M}_t \) and \( v^{1:M}_t \), we use a bottom-up approach, computing the mean of each node given the activation coming from the nodes below it; that is, we compute the mean of \( h^m_t \) using \( v^{<t}_c \) and \( v^m_{<t} \) for each modality, then we compute the mean of \( h_t \) using \( h^{1:M}_t \), then we compute the mean of \( y^L_t \) for each task using \( h_t \), obtaining the classification probabilities for each task. Figure 3 illustrates our inference approach. Inference in the MTM-CRBM is the same as the MTM-CRBM, except there is only one modality, and inference in the D-CRBM is the same as the MT-CRBM, except there is only one task.

5. Learning

Learning our model is done using Contrastive Divergence (CD) [40], where \( \langle \cdot \rangle_{\text{data}} \) is the expectation with respect to the data and \( \langle \cdot \rangle_{\text{recon}} \) is the expectation with respect to the reconstruction. The learning is done using two steps: a bottom-up pass and a top-down pass using sampling equations from (5) for D-CRBM, (8) for MT-CRBM, and (11) for MTM-CRBM. In the bottom-up pass the reconstruction is generated by first sampling the unimodal layers \( p(h^m_{i,t} = 1 | y^m_{i,t}, v^{<t}_c, y_i) \) for all the hidden nodes in parallel. This is followed by sampling the fusion layer \( p(h_{i,t} = 1 | y^L_{i,t}, h^1:M_t, h^{1:M}_t) \). In the top-down pass the unimodal layer is generated using the activated fusion layer \( p(y^L_{i,t} = 1 | h_t, y^L_{i,t}) \). This is followed by sampling the visible nodes \( p(v^m_{i,t} | h^m_t, h^{1:M}_t) \) for all the visible nodes in parallel. The gradient updates are described in (14). Similarly
6. Experiments

We now describe the datasets in (sec 6.1), specify the implementation details in (sec 6.2), and present our quantitative results in (sec 6.3).

6.1. Datasets

Our problem is very particular in that we focus on multi-task learning for body affect. In the literature [4, 9] most of the datasets were either single task for activity recognition, not publicly available, too few instances, or only RGB-D without skeleton. We found two available datasets to evaluate our approach that are multi-task.

The first dataset is the Body Affect dataset [7], collected using a motion capture sensor, which consists of a set of actors performing several actions with different affects. The second dataset is the Tower Game dataset [8], collected using a Kinect sensor, which consists of an interaction between two humans performing a cooperative task, with the goal of classifying different components of entrainment.

In the following subsections we describe the datasets.

Body Affect Dataset: This dataset [7] consists of a library of human movements captured using a motion capture sensor, annotated with actor, action, affect, and gender. The dataset was collected for studying human behavior and personality properties from human movement. The data consists of 30 actors (15 female and 15 male) each performing four actions (walking, knocking, lifting, and throwing) with each of four affect styles (angry, happy, neutral, and sad). For each actor, there are 40 data instances: 8 instances of walking (2 directions x 4 affects), 8 instances of knocking (2 repetitions x 4 affects), 8 instances of lifting (2 repetitions x 4 affects), 8 instances of throwing (2 repetitions x 4 affects), and 8 instances of the sequences (2 repetitions x 4 affects). For knocking and lifting and throwing there were 5 repetitions per data instance. Thus, the 24 records of knocking, lifting, and throwing contain 120 separate instances, yielding a total of 136 instances per actor and a total of 4,080 instances. We split dataset into 50% training using 15 actors and 50% testing using the other 15 actors.

Tower Game Dataset: This dataset [8] is a simple game of tower building often used in social psychology to elicit different kinds of interactive behaviors from the participants. It is typically played between two people working with a small fixed number of simple toy blocks that can be stacked to form various kinds of towers. The data consists of 112 videos which were divided into 1213 10-second segments indicating the presence or absence of these behaviors in each segment. Entrainment is the alignment in the behavior of two individuals and it involves simultaneous movement, tempo similarity, and coordination. Each measure was rated low, medium, or high for the entire 10 seconds segment. 50% of that data was used for training and 50% were used for testing. In this dataset we call each person’s skeletal data a modality, where our goal is to model mocap-mocap representations.

6.2. Implementation Details

For pre-processing the Tower Game dataset, we followed the same approach as [50] by forming a body centric transformation of the skeletons generated by the Kinect sensors. We use the 11 joints from the upper body of the two players since the tower game almost entirely involves only upper body actions and gestures are done using the upper body. We used the raw joint locations normalized with respect to a selected origin point. We use the same descriptor provided by [51, 52]. The descriptor consists of 84 dimensions based on the normalized joint location, inclination angles formed by all triples of anatomically connected joints, azimuth angles between projections of the second bone and the vector on the plane perpendicular to the orientation of the first bone, bending angles between a basis vector, perpendicular to the torso, and joint positions. As for the Body Affect dataset we decided to use the full body centric representation [53] for motion capture sensors resulting in 42 dimensions per frame.

For the Body Affect dataset we trained a three-task model for the following three tasks: Action (AC) ∈ {Walking, Knocking, Lifting, Throwing}, Affect (AF) ∈ {Neutral, Happy, Sad, Angry}, Gender (G) ∈ {Male, Female}. The data is split into a training set consisting of 50% of the instances, and a test set consisting of the remaining 50%. For the Tower Game dataset we trained a three-task model for the following tasks: Tempo Similarity (TS), Coordination (C), and Simultaneous Movement (SM), each in {Low, Medium, High}. The data is split into a training set consisting of 50% of the instances, and a test set consisting of the remaining 50%.

We tuned our model parameters. For selecting the
model parameters we used a grid search. We varied the number of hidden nodes per layer in the range of \( \{10, 20, 30, 50, 70, 100, 200\} \), as well as the auto-regressive nodes in the range of \( \{5, 10\} \), resulting a total of 2744 trained models. The best performing model on the Body Affect dataset has the following configuration \( v = 42, h = 30, v_c = 42 \times 10 \) and the best performing model on the Tower Game dataset has the following configuration \( v^m = 84, h^m = 30, v_{c,t}^m = 10 \times 84 \) for each of the modalities and for the fusion layer in the Tower Game dataset \( h^{1:M} = 60, h = 60, h_{c,t}^{1:M} = 10 \times 60 \).

Note that in our MT-CRBM model, the tasks are assumed conditionally independent given the hidden representation. Thus the number of parameters needed for the hidden-label edges is \( H \cdot \sum_{k=1}^{L} Y_k \), where \( H \) is the dimensionality of the hidden layer and \( Y_k \) is the number of classes for task \( k \). Contrast this to the number of parameters needed if instead the tasks are flattened as a Cartesian product, \( H \cdot \prod_{k=1}^{L} Y_k \). Our factored representation of the multiple tasks uses only linearly many parameters instead of the exponentially many parameters needed for the flattened representation.

### 6.3. Quantitative Results

We first define baselines and variants of the model, followed by the average classification accuracy results on the two datasets.

**Baselines and Variants:** Since we compare our approach against the results presented in [8] we decided to use the same baselines they used. They used SVM classifiers on a combination of features. **SVM+RAW:** The first set of features consisted of first order static and dynamic handcrafted skeleton features. The static features are computed per frame. The features consist of relationships between all pairs of joints of a single actor, and the relationships between all pairs of joints of both the actors. The dynamic features are extracted per window (a set of 300 frames). In each window, they compute first and second order dynamics of each joint, as well as relative velocities and accelerations of pairs of joints per actor, and across actors. The dimensionality of their static and dynamic features is \((257400 \text{ D})\). **SVM+BoW100** and **SVM+BoW300:** To reduce their dimensionality they used, Bag-of-Words (BoW) \((100 \text{ and } 300 \text{ D})\) [54, 52]. We also evaluate our approach using **HCRF** [55]. We define our own model’s variants, **D-CRBMs** which is our single-task model presented in Section 3.2, **MT-CRBMs** which is our multi-task model presented in Section 3.3, **MTM-CRBMs** the multi-modal multi-task model presented in Section 3.4 and **DM-CRBMs** an extension to the D-CRBMs to be multimodal similar to MTM-CRBMs. We also add two new variants M**T-CRBMs-Deep** and **MTM-CRBMs-Deep** shown in Fig.4, which are a deeper version of the original models, by adding a task specific representation layer.

**Classification:** For the Body Affect dataset, Table 1 shows the results of the baselines as well as our model and its variants. For the Tower Game dataset, Table 2 shows our average classification accuracy using different features and baselines combinations as well as the results from our models. We can see that the **MT-CRBMs-Deep** model outperforms all the other models for both cases, thereby demonstrating its effectiveness on predicting multi-task labels.
Table 1. Average Classification Accuracy on The Body Affect Dataset.

<table>
<thead>
<tr>
<th>Classifier (labels)</th>
<th>AC(4)</th>
<th>AF(4)</th>
<th>G(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guess</td>
<td>25.0</td>
<td>25.0</td>
<td>50.0</td>
</tr>
<tr>
<td>SVM+Raw</td>
<td>35.6</td>
<td>32.2</td>
<td>65.1</td>
</tr>
<tr>
<td>SVM+BoW100</td>
<td>41.3</td>
<td>34.1</td>
<td>71.4</td>
</tr>
<tr>
<td>SVM+BoW300</td>
<td>39.9</td>
<td>32.8</td>
<td>69.5</td>
</tr>
<tr>
<td>HCRF[55]</td>
<td>44.8</td>
<td>34.7</td>
<td>74.1</td>
</tr>
<tr>
<td>D-CRBMs</td>
<td>52.6</td>
<td>30.7</td>
<td>78.4</td>
</tr>
<tr>
<td>MT-CRBMs</td>
<td>53.5</td>
<td>31.2</td>
<td>78.2</td>
</tr>
<tr>
<td>MT-CRBMs-Deep</td>
<td>54.5</td>
<td>32.7</td>
<td>78.4</td>
</tr>
</tbody>
</table>

Table 2. Average Classification Accuracy on The Tower Game Dataset.

<table>
<thead>
<tr>
<th>Classifier (labels)</th>
<th>TS (3)</th>
<th>C (3)</th>
<th>SM (3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Guess</td>
<td>33.3</td>
<td>33.3</td>
<td>33.3</td>
</tr>
<tr>
<td>SVM+Raw</td>
<td>59.3</td>
<td>52.2</td>
<td>39.5</td>
</tr>
<tr>
<td>SVM+BoW100</td>
<td>65.6</td>
<td>55.8</td>
<td>44.3</td>
</tr>
<tr>
<td>SVM+BoW300</td>
<td>54.4</td>
<td>47.5</td>
<td>42.8</td>
</tr>
<tr>
<td>HCRF[55]</td>
<td>67.2</td>
<td>58.8</td>
<td>44.5</td>
</tr>
<tr>
<td>DM-CRBMs</td>
<td>76.5</td>
<td>62.0</td>
<td>49.2</td>
</tr>
<tr>
<td>MT-CRBMs</td>
<td>86.2</td>
<td>70.0</td>
<td>63.5</td>
</tr>
<tr>
<td>MT-CRBMs-Deep</td>
<td>87.2</td>
<td>70.0</td>
<td>72.8</td>
</tr>
</tbody>
</table>

The generative approaches make adding additional tasks a trivial matter.

The generative capabilities of our approach enable new and interesting applications. A future direction of work is to further explore and improve these generative applications of the models.

Acknowledgments

This research was partially developed with funding from the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References


