INTRODUCING SHARED-HIDDEN-LAYER AUTOENCODERS FOR TRANSFER LEARNING AND THEIR APPLICATION IN ACOUSTIC EMOTION RECOGNITION

Jun Deng1, Rui Xia2, Zixing Zhang1, Yang Liu2, Björn Schuller3,1

1 Machine Intelligence & Signal Processing Group, MMK, Technische Universität München, Germany
2 Computer Science Department, University of Texas at Dallas, USA
3 Department of Computing, Imperial College London, UK

ABSTRACT

This study addresses a situation in practice where training and test samples come from different corpora – here in acoustic emotion recognition. In this situation, a model is trained on one database while tested on another disjoint one. The typical inherent mismatch between the corpora and by that between test and training set usually leads to significant performance degradation. To cope with this problem when no training data from the target domain exists, we propose a ‘shared-hidden-layer autoencoder’ (SHLA) approach for learning common feature representations shared across the training and test set in order to reduce the discrepancy in them. To exemplify effectiveness of our approach, we select the Interspeech Emotion Challenge’s FAU Aibo Emotion Corpus as test database and two other publicly available databases as training set for extensive evaluation. The experimental results show that our SHLA method significantly improves over the baseline performance and outperforms today’s state-of-the-art domain adaptation methods.

Index Terms— Transfer Learning, Cross-Corpus, Shared-Hidden-Layer Autoencoder, Emotion Recognition

1. INTRODUCTION

Representation learning, i.e., learning transformations of the data that make it easier to extract useful information when building classifiers or other predictors, is recently gathering a lot of attention [1,2]. In this paper, we propose a ‘shared-hidden-layer autoencoder’ (SHLA) method which can learn common feature representations shared across training and test set in order to reduce the discrepancy in them. Our basic idea is to feed training and test examples in the SHLA, and then minimize the reconstruction error on the training examples as well as on the test examples at the same time, such that the SHLA captures a common structure of the data-generating distribution induced by the training and test examples in an unsupervised way. Afterwards, we use features learned from the SHLA to carry out normal supervised algorithms for classification.

We will exemplify SHLA’s efficiency by application to automatic emotion recognition in speech. Over the last decade, research in this application field has increasingly drawn attention [3–6]. Most of these works present promising performance based on training and test set coming from the same session or corpus. However, such results may not be obtained if training and test data have different characteristics. The mismatch between training and testing can be due to different speakers, different acoustic conditions, and/or type of emotion such as acted, elicited, or naturalistic [7]. It may even be that the spoken languages or the emotion annotation schemes are different. These differences are known to produce a detrimental effect on the real-world performance of acoustic emotion recognition systems, since in training they will not have prepared for data subsequently encountered in use.

The influence of such differences can be partly alleviated by building a feature representation that incorporates domain knowledge into the data [8,9]. However, such feature engineering can be very application-specific and labor-intensive. Therefore, directly learning the underlying explanatory factors hidden in the corpora seems more promising, and more importantly is able to expand the scope of applicability to novel target tasks. The few previous works in this specific application field include [10] where feature transfer learning was proposed based on a sparse autoencoder method for discovering knowledge in acoustic features from small target data to improve performance of speech emotion recognition when applying the knowledge to source data.

2. RELATED WORK

Transfer learning has been proposed to deal with the problem of how to reuse the knowledge learned previously from ‘other’ data or features [11]. Among the various ways of transfer learning, domain adaptation of statistical classifiers has been shown to be well suited for a problem where the data distribution in the test domain is different from the one in the training domain. One general approach to address the domain adaptation problem is to assign more weight to those training examples that are most similar to the test data, and less weight to those that poorly reflect the distribution of the target (test) data. This idea of weighting the input data based on the test data is known as importance weighting. The goal is to estimate importance weights, denoted \( \beta \), from training examples and test examples by taking the ratio of their densities \( \beta(x) = \frac{p_{te}(x)}{p_{tr}(x)} \) where \( p_{te}(x) \) and \( p_{tr}(x) \) are test and training input densities. Kanamori et al. proposed unconstrained least-squares importance fitting (uLSIF) to estimate the importance weights by a linear model [12]. Tsuboi et al. modeled the importance function by a linear (or kernel) model, which resulted in a convex optimization problem with a sparse solution, called KLIEP [13].

Kernel mean matching (KMM) has recently been shown to lead to significant improvement in acoustic emotion recognition when Hassan et al. first considered to explicitly compensate for acoustic
and speaker differences between training and test databases [14]. KMM was proposed to deal with sampling bias in various learning problems [15], which allows to directly estimate the resampling weights by matching training and test distribution feature means in a reproducing kernel Hilbert space. The objective function is given by the discrepancy term between the two empirical means

$$
\left\| \frac{1}{n_{tr}} \sum_{i=1}^{n_{tr}} \beta_{i} \Phi(x_{i}^{tr}) - \frac{1}{n_{te}} \sum_{i=1}^{n_{te}} \beta_{i} \Phi(x_{i}^{te}) \right\|^2,
$$

(1)

where $\Phi$ are the mapping functions.

Using $K_{ij} := k(x_{i}^{tr}, x_{j}^{tr})$ and $\kappa_{i} := \frac{1}{n_{tr}} \sum_{j=1}^{n_{tr}} k(x_{i}^{tr}, x_{j}^{te})$, Eq. (1) above becomes the quadratic problem for finding suitable $\beta$:

$$
\text{arg min}_{\beta} \frac{1}{2} \beta^{T} K \beta - \kappa^{T} \beta
$$

s. t. $\beta_{i} \in [0, B]$ and $\left| \sum_{i=1}^{m} \beta_{i} - n_{te} \right| \leq n_{te} \epsilon$, (2)

where the upper limit of importance weight $B > 0$ and $\epsilon > 0$ are tuning parameters, and $k$ is the kernel function. Since KMM optimization is formulated as a convex quadratic programming problem, it leads to a unique global solution.

3. PROPOSED METHODOLOGY

3.1. Basic Autoencoder

A basic autoencoder – a kind of neural network typically consisting of only one hidden layer –, sets the target values to be equal to the input. Deep neural networks use it, as an element, to find common data representation from the input [16, 17]. Formally, in response to an input example $x \in \mathbb{R}^n$, the hidden representation $h(x) \in \mathbb{R}^m$ is

$$
h(x) = f(W_{1}x + b_{1}),
$$

(3)

where $f(z)$ is a non-linear activation function, typically a logistic sigmoid function $f(z) = 1/(1 + \exp(-z))$ applied component-wise, $W_{1} \in \mathbb{R}^{m \times n}$ is a weight matrix, and $b_{1} \in \mathbb{R}^{m}$ is a bias vector.

The network output maps the hidden representation $h$ back to a reconstruction $\tilde{x} \in \mathbb{R}^{n}$:

$$
\tilde{x} = f(W_{2}h(x) + b_{2}),
$$

(4)

where $W_{2} \in \mathbb{R}^{n \times m}$ is a weight matrix, and $b_{2} \in \mathbb{R}^{n}$ is a bias vector.

Given an input set of examples $X$, autoencoder training consists in finding parameters $\theta = \{W_{1}, W_{2}, b_{1}, b_{2}\}$ that minimize the reconstruction error, which corresponds to minimizing the following objective function:

$$
\mathcal{J}(\theta) = \sum_{x \in X} \| x - \tilde{x} \|^2.
$$

(5)

The minimization is usually realized by stochastic gradient descent as in the training of neural networks. In this paper, we also add a weight-decay regularization term into the objective function which favors small weights. Note that, if the number of hidden units $m$ is less than the number of input units $n$, then the network is forced to learn a compressed representation of the input.

3.2. Shared-hidden-layer Autoencoder (SHLA)

The idea behind transfer learning is to exploit commonalities between different learning tasks in order to share statistical strength, and transfer knowledge across tasks [1, 18, 19]. Based on the motivation of the ‘sharing idea’ in transfer learning, we propose an alternative structure of autoencoder that attempts to minimize the reconstruction error on both training set and test set. The ‘shared-hidden-layer autoencoder’ (SHLA for short) shares the same parameters for the mapping from the input layer to the hidden layer, but uses independent parameters for the corresponding reconstructions $\tilde{X}_{t}$ and $\tilde{X}_{e}$.

![Fig. 1: Structure of the shared-hidden-layer autoencoder (SHLA) on the training set $X^{tr}$ and test set $X^{te}$. The SHLA shares same parameters for the mapping from the input layer to the hidden layer, but uses independent parameters for the corresponding reconstructions $\tilde{X}_{t}$ and $\tilde{X}_{e}$.](image)

where $\tilde{X}_{t}$ and $\tilde{X}_{e}$ share the same parameters $\theta_{tr} = \{W_{1}, W_{2}^{tr}, b_{1}, b_{2}^{tr}\}$ and $\theta_{te} = \{W_{1}, W_{2}^{te}, b_{1}, b_{2}^{te}\}$ share the same parameters $\{W_{1}, b_{1}\}$.

Besides, we optimize the joined distance for the two sets, which leads to the following overall objective function:

$$
\mathcal{J}_{SA}(\theta_{SA}) = \mathcal{J}_{te}(\theta_{te}) + \gamma \mathcal{J}_{te}(\theta_{te})
$$

(8)

where $\theta_{SA} = \{W_{1}, W_{2}^{te}, b_{1}, b_{2}^{te}\}$ are the parameters to be optimized during training, the hyper-parameter $\gamma$ controls the strength of the regularization. Training the SHLA is equivalent to training a basic autoencoder, and the standard back-propagation algorithm can be applied.

By adding the regularization term from the target (test) set, the SHLA is equipped with extensive flexibilities to directly incorporate the knowledge from the target (test) set. Hence, to minimize the objective function, the shared hidden layer is biased to make the distribution induced by the training set as similar as possible to the distribution induced by the target set. This helps to regularize the functional behavior of the autoencoder. It further turns out to lessen the effects of the difference in training and target set.
10–13 years interacting with the pet robot Aibo in German speech. In this work, we use the representation learned from the hidden layer in the trained SHLA (see Eq. (3)), which will be taken to build a standard supervised classifier for acoustic emotion recognition in the following.

4. EXPERIMENTS

Let us now investigate the efficacy of the proposed SHLA on a standard Machine Learning task. This example stems from the field of (cross-corpus) acoustic emotion recognition. Most previous approaches do not consider the difference between corpora before building emotion recognition models, and have demonstrated the difficulty in cross-corpus processing [7, 22, 23]. Recently, [10] uses sparse autoencoders to transfer useful knowledge from other corpora to the target one using the labels of the target set. [14] considers important weights to shift the separating hyperplane of Support Vector Machines (SVMs) in such a way as to take into consideration the more important training data, but without considering a cross-corpus scenario. In the following, we provide experimental results for a challenging real-life task by using other disjoint corpora as training set based on the proposed SHLA representation learning.

4.1. Selected Task and Data

To investigate the performance of the proposed method, we consider the INTERSPEECH 2009 Emotion Challenge (EC) two-class task [24]. It is based on the spontaneous FAU Aibo Emotion Corpus (FAU AEC), which contains recordings of 51 children at the age of 10–13 years interacting with the pet robot Aibo in German speech. The children were made believe that the Aibo was responding to their commands, whereas the robot was actually remote-controlled in a Wizard-of-Oz manner and did not respond to their commands. Hence, the database contains induced emotionally-colored speech. The details of the two-class task are given in Table 1. For the experiments to follow, we always evaluate the emotion recognition model on the test set of the FAU AEC.

Additionally, for the training set we chose two publicly available and popular databases, namely the Airplane Behavior Corpus (ABC) [25], and the Speech Under Simulated and Actual Stress (SUSAS) set [26]. These are highly different from the target set FAU AEC in terms of speaker age (adults vs. children), partially spoken language, type of emotion, degree of spontaneity, phrase length, type of recording situation, and annotators and subjects. For comparability with FAU AEC, we have to map the diverse emotion classes onto the valence axis in the dimensional emotion model. The mapping defined for the cross-corpus experiments is used to generate labels for binary valence from the emotion categories in order to generate a unified set of labels. This mapping is given in Table 2. In addition, Table 3 summarizes the three chosen databases and shows the existing differences in them.

3.3. Recognition with SHLA-based Representation Learning

It has been observed widely that autoencoders can automatically capture useful features hidden in data. Such features are often used in building a deep hierarchy of features, within the contexts of supervised, semi-supervised, or unsupervised modeling [17, 20, 21]. In this work, we use the representation learned from the hidden layer in the trained SHLA (see Eq. (3)), which will be taken to build a standard supervised classifier for acoustic emotion recognition in the following.

### Table 1: Number of instances for the two-class task of the FAU AEC.

<table>
<thead>
<tr>
<th></th>
<th>Negative</th>
<th>Positive</th>
<th>( \sum )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>3 358</td>
<td>6 601</td>
<td>9 959</td>
</tr>
<tr>
<td>Test</td>
<td>2 465</td>
<td>5 792</td>
<td>8 257</td>
</tr>
</tbody>
</table>

### Table 2: Emotion categories mapping onto negative and positive valence classes for the three chosen databases.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Negative</th>
<th>Positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAU AEC</td>
<td>angry, emphatic, repri-manding,</td>
<td>joyful, motherese, neutral, rest</td>
</tr>
<tr>
<td></td>
<td>touchy</td>
<td></td>
</tr>
<tr>
<td>ABC</td>
<td>aggressive, intoxicated, nervous,</td>
<td>cheerful, neutral, rest</td>
</tr>
<tr>
<td>SUSAS</td>
<td>high stress, screaming, fear</td>
<td>medium stress, neutral</td>
</tr>
</tbody>
</table>

### Table 4: Overview of the standard feature set provided by the INTERSPEECH 2009 EC.

<table>
<thead>
<tr>
<th>LLDs (16 × 2)</th>
<th>Functionals (12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(( \Delta )) ZCR</td>
<td>mean</td>
</tr>
<tr>
<td>(( \Delta )) RMS Energy</td>
<td>standard deviation</td>
</tr>
<tr>
<td>(( \Delta )) F0</td>
<td>kurtosis, skewness</td>
</tr>
<tr>
<td>(( \Delta )) HNR</td>
<td>extremes: value, rel, position, range</td>
</tr>
<tr>
<td>(( \Delta )) MFCC 1–12</td>
<td>linear regression: offset, slope, MSE</td>
</tr>
</tbody>
</table>

4.2. Acoustic Features

To keep in line with the INTERSPEECH 2009 EC [24], we decided to use its standard feature set of 12 functionals applied to \( 2 \times 16 \) acoustic Low-Level Descriptors (LLDs) including their first order delta regression coefficients as shown in Table 4. In detail, the 16 LLDs are zero-crossing-rate (ZCR) from the time signal, root mean square (RMS) frame energy, pitch frequency (normalized to 500 Hz), harmonics-to-noise ratio (HNR) by autocorrelation function, and Mel-frequency cepstral coefficient (MFCC) 1–12. Then, 12 functionals — mean, standard deviation, kurtosis, skewness, minimum and maximum value, relative position, and ranges as well as two linear regression coefficients with their mean square error (MSE) — are applied on the chunk level. Thus, the total feature vector per chunk contains \( 16 \times 2 \times 12 = 384 \) attributes. To ensure reproducibility as well, the open source toolkit openEAR toolkit [27] was used with the pre-defined EC configuration.

4.3. Experimental Setup and Evaluation Metrics

As classifier, we use linear SVMs as were used in the baseline of the EC with a fixed penalty factor \( C = 0.5 \) as the basic supervised learner. The toolkit LIBLINEAR [28] is applied in the experiments.

In the SHLA learning process, the number of hidden units \( m \) was fixed to 200, and attempted hyper-parameter \( \gamma \) and weight decay values \( \lambda \) were the following : \( \gamma \in \{0.1, 0.3, 0.5, 1, 2, 3\} \), \( \lambda \in \{0.0001, 0.001, 0.01, 0.1\} \).

We evaluate the performance of the baselines and SHLA systems using unweighted average recall (UAR) as was the competition measure in the EC. It equals the sum of the recalls per class divided by the number of classes, and better reflects overall accuracy in the presence of class imbalance. Besides, we validate statistical significance of the results according to a one-sided z-test.

4.4. Models for Comparison

We compare the following methods:
Table 5: Cross-corpus average UAR over ten trials for the training sets ABC and SUSAS.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Age</th>
<th>Language</th>
<th>Speech</th>
<th>Emotion</th>
<th># Valence</th>
<th># All</th>
<th>h:mm</th>
<th>#m</th>
<th>#f</th>
<th>Rec</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAU AEC</td>
<td>children</td>
<td>German</td>
<td>variable</td>
<td>natural</td>
<td>58.23</td>
<td>12.39</td>
<td>18.216</td>
<td>9.20</td>
<td>21</td>
<td>30</td>
<td>normal</td>
</tr>
<tr>
<td>ABC</td>
<td>adults</td>
<td>German</td>
<td>fixed</td>
<td>acted</td>
<td>213</td>
<td>217</td>
<td>430</td>
<td>1:15</td>
<td>4</td>
<td>4</td>
<td>studio</td>
</tr>
<tr>
<td>SUSAS</td>
<td>adults</td>
<td>English</td>
<td>fixed</td>
<td>natural</td>
<td>1,616</td>
<td>1,977</td>
<td>3,593</td>
<td>1:01</td>
<td>4</td>
<td>3</td>
<td>noisy</td>
</tr>
</tbody>
</table>

- **Matched Training (MT):** randomly (repeated ten times) picks a number of instances from the FAU AEC training set to train a SVM, i.e., without the need of transferring in intra-corpus scenario. For comparison, this number is given by the number of learning instances as in the ABC or SUSAS sets, respectively.
- **Cross Training (CT):** uses ABC or SUSAS to train the standard (SVM) classifier, i.e., without using SHLA-based representation learning.
- **KMM:** utilizes the KMM (see Section 2) on the ABC and SUSAS database for covariate shift adaptation. We choose the ‘tuning parameters’ in KMM following [14, 15].
- **DAE:** employs denoising autoencoders for representation learning in order to match training examples to test examples, which was successfully applied to the transfer learning challenge and domain adaptation [18, 29].
- **SHLA:** uses the proposed SHLA to extract common features on the training and target test set, then trains standard SVMs using the learned features and labels in the training set.

### 4.5. Results

First we evaluate the cross-corpus scenario, where we train acoustic emotion recognition models on ABC or SUSAS while testing on the FAU AEC test set (except the MT condition that uses FAU data for training). We run the experiments ten times for **MT**, **DAE**, and **SHLA** methods that involve random sampling. The averaged UAR over the ten trials is visualized in Figure 2, including the error bar, and given quantitatively in Table 5. As can be seen, the **SHLA** method outperforms all the other approaches.

More specifically, for the small database ABC, one can easily see that the two standard methods (**CT** and **MT**) only obtain average UAR around the chance level (55.28% and 58.32%). When the accuracy obtained by the **DAE** method reaches 56.20%, the covariate shift adaptation **KMM** can boost the accuracy to 62.52%. However, with **SHLA** one reaches 63.36%. This improvement has a statistical significance at the 0.001 level compared with the baselines **CT** and **MT**.

In comparison with ABC, although SUSAS’s average UAR from the **CT** method is still close to chance level, it is worth noting that the average UAR achieved by the **MT** method increases sharply to 62.41% because of the larger size of SUSAS leading to more instances chosen from the FAU AEC training set. But SUSAS cannot obtain a great benefit from the covariate shift adaptation **KMM**, like ABC. Nevertheless, the **SHLA** method still gives an average UAR of 62.72%, which is slightly larger than the maximum average UAR obtained by the **MT**. Compared with the four methods in use, the proposed **SHLA** method passes the significant test at the 0.01 and 0.02 level against the **CT** and **KMM** methods.

Finally, we consider the intra-corpus scenario, which means that we conduct the representation learning between the FAU AEC training set and its test set by the **SHLA** method. In this case, the **SHLA** obtains an average UAR of 68.29% compared to the baseline (the standard SVM) UAR of 67.04%. The improvement is significant at the 0.05 level.

Overall, **SHLA**-based representation learning could be shown as useful in reducing the difference for cross-corpus recognition.

### 5. CONCLUSIONS AND OUTLOOK

We proposed a ‘shared-hidden-layer autoencoder’ (**SHLA**) for representation learning shared across training and target corpora. In this method, we use the **SHLA** to explore the common feature representation in order to compensate for the differences in corpora caused by language, speaker, acoustic conditions. Such learned representations were successfully applied to a standard machine learning task: acoustic emotion recognition. Experimental results on three publicly available corpora demonstrate that the proposed method effectively and significantly enhances the emotion classification accuracy and competes well with other domain adaptation methods. In future work, we plan to build a deep architecture based on **SHLAs** and modify the **SHLA** method for on-line applications.
6. REFERENCES


