

Affect Estimation in 3D Space Using Multi-Task Active Learning for Regression

Dongrui Wu, *Senior Member, IEEE*, and Jian Huang, *Senior Member, IEEE*

Abstract—Acquisition of labeled training samples for affective computing is usually costly and time-consuming, as affects are intrinsically subjective, subtle and uncertain, and hence multiple human assessors are needed to evaluate each affective sample. Particularly, for affect estimation in the 3D space of valence, arousal and dominance, each assessor has to perform the evaluations in three dimensions, which makes the labeling problem even more challenging. Many sophisticated machine learning approaches have been proposed to reduce the data labeling requirement in various other domains, but so far few have considered affective computing. This paper proposes two multi-task active learning for regression approaches, which select the most beneficial samples to label, by considering the three affect primitives simultaneously. Experimental results on the VAM corpus demonstrated that our optimal sample selection approaches can result in better estimation performance than random selection and several traditional single-task active learning approaches. Thus, they can help alleviate the data labeling problem in affective computing, i.e., better estimation performance can be obtained from fewer labeling queries.

Index Terms—Active learning, affective computing, emotion estimation, multi-task learning, regression, greedy sampling

1 INTRODUCTION

THE amount of labeled training samples is critical to the performance of machine learning models. However, in many real-world applications it is easy to obtain unlabeled data, but labeling them may be very costly or time-consuming. This is particularly true for affective computing [15]. Affects are very subjective, subtle, and uncertain. So, usually multiple human assessors are needed to obtain the groundtruth affect label for each affective sample (video, audio, image, text, etc.). For example, 14-16 assessors were used to evaluate each video clip in the DEAP dataset [10], and six to 17 assessors were used for each utterance in the VAM corpus [7].

Many machine learning approaches have been proposed to alleviate the data labeling effort, including [8]:

- 1) *Semi-supervised learning* [3], which uses typically a small amount of labeled data and a large amount of unlabeled data simultaneously in model training.
- 2) *Transfer learning* [14], which makes use of data or knowledge from similar or relevant tasks to help the learning in a new task, which typical has a small number of labeled samples.
- 3) *Multi-task learning* (MTL) [32], in which multiple learning tasks are solved simultaneously, while exploiting commonalities and differences across them.
- 4) *Active learning* [12], [20], [23], which optimally selects the most informative unlabeled samples to label, so that a good learning model could be built from a small number of labeled samples.

The above four approaches are independent and complementary, so they could be combined for even better performance. For example, we have developed a collaborative filtering approach, which integrates transfer learning and active class selection, a

variant of active learning, for reducing the calibration data requirement in brain-computer interface (BCI). We have also developed active weighted adaptation regularization [24], which integrates active learning and domain adaptation, a specific form of transfer learning, for reducing the subject-specific calibration effort in BCI. Most recently, we have also developed an active semi-supervised transfer learning approach [22], which integrates semi-supervised learning, transfer learning and active learning, for offline BCI calibration.

The focus of this paper is MTL, which has been successfully used in many real-world applications, including affective computing. For example, Jiang et al. [9] proposed a multi-task fuzzy system that uses simultaneously independent sample information from each task and the inter-task common hidden structure among multiple tasks to enhance the generalization performance. It demonstrated promising performance in real-world applications including glutamic acid fermentation process modeling, polymer test plant modeling, wine preferences modeling, concrete slump modeling, etc. Su et al. [21] proposed MTL with low rank attribute embedding to perform person re-identification on multi-cameras, and demonstrated that it significantly outperformed existing single-task and multi-task approaches. Abadi et al. [1] proposed MTL-based regression models to simultaneously learn the relationship between low-level audio-visual features and high-level valence/arousal ratings from a collection of movie scenes. They can better predict valence and arousal ratings than scene-specific models. Xia and Liu [28] integrated MTL and deep belief network to leverage activation and valence information for acoustic emotion recognition. Zhang et al. [30] treated corpus, domain, and gender as different tasks in cross-corpus MTL, and showed that it outperformed approaches that treat the tasks as either identical or independent.

However, there have been very few approaches on integrating MTL and active learning. Reichart et al. [17] proposed multi-task active learning for linguistic annotations, which considers two annotation tasks (named entity and syntactic parse tree) and

D. Wu and J. Huang are with the Key Laboratory of the Ministry of Education for Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China. Email: drwu@hust.edu.cn, huang_jan@hust.edu.cn.

demonstrated promising performance. Zhang [31] studied multi-task active learning with output constraints, and demonstrated the effectiveness of the proposed framework in web information extraction and document classification. Li et al. [11] proposed a multi-domain active learning approach for text classification, which jointly selects samples from multiple domains with duplicate information considered. Experiments on three real-world applications (sentiment classification, newsgroup classification and email spam filtering) showed that it outperformed several state-of-the-art single-task active learning approaches. Harpale [8] gave so far the most comprehensive study on multi-task active learning in his PhD Dissertation, which proposed approaches for homogeneous tasks, heterogeneous tasks, hierarchical classification, and collaborative filtering, and verified their performances in text classification, movie genre classification, image annotation, etc.

The above review shows that among the small number of studies on multi-task active learning, only one was related to affective computing (text sentiment classification), and none had considered regression problems¹. However, affect estimation is a very pertinent application because affects intrinsically have multiple dimensions, e.g., affects can be represented in the 2D space of arousal and valence [18], or in the 3D space of arousal, valence, and dominance [13]. This paper fills the gap by proposing two multi-task active learning for regression (ALR) approaches, which extend two ALR approaches we proposed earlier [25] from single-task learning to MTL. Experimental results on the VAM corpus [7] demonstrate the effectiveness of our proposed approaches. Moreover, our proposed multi-task ALR approaches are generic, and they can also be used in other application domains beyond affective computing.

The remainder of this paper is organized as follows: Section 2 introduces two single-task ALR approaches based on greedy sampling, and their multi-task extensions. Section 3 compares the performances of multi-task ALR with several state-of-the-art single-task ALR approaches on the VAM corpus. Section 4 discusses why MTL should be preferred over single-task learning in affective computing. Section 5 draws conclusions and points out some future research directions.

2 MT-ALR USING GREEDY SAMPLING

This section extends two single-task ALR approaches we proposed recently [25] to MTL.

2.1 Single-task GSy

The GSy ALR approach, proposed in our recent work [25] for single-task regression, was inspired by the greedy sampling (GS) ALR approach proposed in [29]. GS tries to select the most diverse samples in the *input* space to label, whereas GSy aims to achieve diversity in the *output* space.

The basic idea of GSy is as follows. Given a pool of unlabeled samples, GSy first selects a few samples using GS in the *input* space to build an initial regression model, and then in each subsequent iteration selects a new sample located furthest away from all previously selected samples in the *output* space to achieve diversity among the selected samples. Implementation details are given next.

1. Chapter 6 of [8] presented an aspect model with Bayesian active learning algorithm for regression problems and applied it to two movie rating applications, but it is single-task active learning instead of multi-task active learning.

Assume the pool consists of N samples $\{\mathbf{x}_n\}_{n=1}^N$, initially none of which is labeled. Our goal is to select K of them to label, and then construct an accurate regression model from them to estimate the outputs for the remaining $N - K$ samples. GSy selects the first sample as the one closest to the centroid of all N samples (i.e., the one with the shortest average distance to the remaining $N - 1$ samples), and the remaining $K - 1$ samples incrementally.

To achieve diversity in the output space, GSy needs to know first the outputs (labels) of all samples, either true or estimated. Let K_0 be the minimum number of labeled samples required to build a regression model (in this paper K_0 is set as the number of features in the input space). GSy first uses GS in the input space to select the first K_0 samples to label. Without loss of generality, assume the first k ($k < K_0$) samples have already been selected. For each of the remaining $N - k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, GSy computes first its distance to each of the k labeled samples:

$$d_{nm}^{\mathbf{x}} = \|\mathbf{x}_n - \mathbf{x}_m\|, \quad m = 1, \dots, k; n = k + 1, \dots, N \quad (1)$$

then $d_n^{\mathbf{x}}$, the shortest distance from \mathbf{x}_n to all k labeled samples:

$$d_n^{\mathbf{x}} = \min_m d_{nm}^{\mathbf{x}}, \quad n = k + 1, \dots, N \quad (2)$$

and finally selects the sample with the maximum $d_n^{\mathbf{x}}$ to label.

Once K_0 samples have been selected and labeled, a regression model can be constructed, and then GSy can select the remaining $K - K_0$ samples to achieve diversity in the output space. Without loss of generality, assume the first k ($k \geq K_0$) samples have already been labeled with outputs $\{y_m\}_{m=1}^k$, and a regression model $f(\mathbf{x})$ has been constructed. For each of the remaining $N - k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, GSy computes first its distance to each of the k outputs:

$$d_{nm}^y = \|f(\mathbf{x}_n) - y_m\|, \quad m = 1, \dots, k; n = k + 1, \dots, N \quad (3)$$

and d_n^y , the shortest distance from $f(\mathbf{x}_n)$ to $\{y_m\}_{m=1}^k$:

$$d_n^y = \min_m d_{nm}^y, \quad n = k + 1, \dots, N \quad (4)$$

and then selects the sample with the maximum d_n^y to label.

For the completeness of this paper, the pseudo-code of single-task GSy, initially proposed in [25], is shown in Algorithm 1.

2.2 Multi-Task GSy (MT-GSy)

The original GSy was proposed for single-task learning, i.e., each sample in the input space has only one output (task). This subsection extends it to MTL.

Let P be the number of tasks (dimensionality of the output space), i.e., $\mathbf{y}_n = (y_{n,1}, \dots, y_{n,P})^T$. Multi-task GSy (MT-GSy) tries to select samples can benefit all P tasks simultaneously.

MT-GSy first uses GSy to select and label K_0 samples. It then builds P regression models $\{f_p(\mathbf{x})\}_{p=1}^P$, for the P tasks, and next selects the remaining $K - K_0$ samples to achieve diversity in the P output spaces simultaneously. Without loss of generality, assume the first k ($k \geq K_0$) samples have already been labeled with outputs $\{\mathbf{y}_m\}_{m=1}^k$, and P regression models $f_p(\mathbf{x})$ ($p = 1, \dots, P$) have been built. For each of the remaining $N - k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, MT-GSy computes first its distance to each of the k outputs, for each of the P tasks:

$$d_{nm,p}^y = \|f_p(\mathbf{x}_n) - y_m\| \quad (5)$$

Algorithm 1: GSy for single-task ALR [25].

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
 K , the maximum number of labels to query.
Output: The regression model $f(\mathbf{x})$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify \mathbf{x}' , the sample closest to the centroid of Z ;
Move \mathbf{x}' from Z to S ;
Re-index the sample in S as \mathbf{x}_1 , and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K_0 - 1$ more samples incrementally using GS in the input space
Identify K_0 , the minimum number of labeled samples required to construct $f(\mathbf{x})$;
for $k = 1, \dots, K_0 - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute $d_n^{\mathbf{x}}$ in (2);
 end
 Identify the \mathbf{x}' that has the largest $d_n^{\mathbf{x}}$;
 Move \mathbf{x}' from Z to S ;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label the K_0 samples in S ;
Construct the regression model $f(\mathbf{x})$ from S ;
// Select $K - K_0$ more samples incrementally
for $k = K_0, \dots, K - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute d_n^y in (4);
 end
 Identify the \mathbf{x}' that has the largest d_n^y ;
 Move \mathbf{x}' from Z to S ;
 Query to label \mathbf{x}' in S ;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
 Update the regression model $f(\mathbf{x})$ using S .
end

Algorithm 2: MT-GSy for multi-task ALR.

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
 K , the maximum number of labels to query.
Output: P regression models $f_p(\mathbf{x})$, $p = 1, \dots, P$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify \mathbf{x}' , the sample closest to the centroid of Z ;
Move \mathbf{x}' from Z to S ;
Re-index the sample in S as \mathbf{x}_1 , and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K_0 - 1$ more samples incrementally using GS in the input space
Identify K_0 , the minimum number of labeled samples required to construct $f(\mathbf{x})$;
for $k = 1, \dots, K_0 - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute $d_n^{\mathbf{x}}$ in (2);
 end
 Identify the \mathbf{x}' that has the largest $d_n^{\mathbf{x}}$;
 Move \mathbf{x}' from Z to S ;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label the K_0 samples in S , for all P tasks;
Construct P regression models $f_p(\mathbf{x})$ from S , one for each task;
// Select $K - K_0$ more samples incrementally
for $k = K_0, \dots, K - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute d_n^y in (6);
 end
 Identify the \mathbf{x}' that has the largest d_n^y ;
 Move \mathbf{x}' from Z to S ;
 Query to label \mathbf{x}' in S , for all P tasks;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
 Update the P regression models $f_p(\mathbf{x})$ using S .
end

where $m = 1, \dots, k$, $n = k + 1, \dots, N$, and $p = 1, \dots, P$. MT-GSy then computes d_n^y , the product of the shortest distances from $f_p(\mathbf{x}_n)$ to $\{y_{m,p}\}_{m=1}^k$, $p = 1, \dots, P$:

$$d_n^y = \min_m \prod_{p=1}^P d_{nm,p}^y, \quad n = k + 1, \dots, N \quad (6)$$

and selects the sample with the maximum d_n^y to label. The pseudo-code of MT-GSy is given in Algorithm 2.

It's interesting to note that:

- 1) In (6) we combine the P $d_{nm,p}^y$ using product instead of summation, to avoid the problem that different task outputs may have different scales, and hence a task with large outputs may dominate other tasks.
- 2) MT-GSy degrades to the single-task GSy when $P = 1$.
- 3) We considered a simple multi-task setting that all tasks share the same inputs. More general settings that allow different tasks to have different inputs will be considered in our future research.

2.3 Single-Task iGS

The improved greedy sampling (iGS) approach, proposed in our recent work [25] for single-task learning, considers the diversity in both the input and output spaces.

Like the single-task GSy, initially the pool consists of N unlabeled samples and zero labeled sample. In iGS we again set K_0 to be the number of features in the input space, and use GS in the input space to select the first K_0 samples to label. Assume the first k samples have already been labeled with labels $\{y_n\}_{n=1}^k$. For each of the remaining $N - k$ unlabeled sample $\{\mathbf{x}_n\}_{n=k+1}^N$, iGS computes first its distance to each of the k labeled samples in the input space:

$$d_{nm}^{\mathbf{x}} = \|\mathbf{x}_n - \mathbf{x}_m\|, \quad m = 1, \dots, k; n = k + 1, \dots, N \quad (7)$$

and d_{nm}^y in (3), and then d_n^y :

$$d_n^y = \min_m d_{nm}^{\mathbf{x}} d_{nm}^y, \quad n = k + 1, \dots, N \quad (8)$$

Next, iGS selects the sample with the maximum d_n^{xy} to label.

In summary, iGS uses the same procedure as GSy to select the first K_0 samples to build an initial regression model, and then in each subsequent iteration a new sample located furthest away from all previously selected samples in both the input and output spaces is selected to achieve balanced diversity among the selected samples. Its pseudo-code, originally proposed in [25], is given in Algorithm 3, for the completeness of this paper.

Algorithm 3: iGS for single-task ALR [25].

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
 K , the maximum number of labels to query.
Output: The regression model $f(\mathbf{x})$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify \mathbf{x}' , the sample closest to the centroid of Z ;
Move \mathbf{x}' from Z to S ;
Re-index the sample in S as \mathbf{x}_1 , and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K_0 - 1$ more samples
incrementally using GS in the input space
Identify K_0 , the minimum number of labeled samples required to construct $f(\mathbf{x})$;
for $k = 1, \dots, K_0 - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute d_n^x in (2);
 end
 Identify the \mathbf{x}' that has the largest d_n^x ;
 Move \mathbf{x}' from Z to S ;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label the K_0 samples in S ;
Construct the regression model $f(\mathbf{x})$ from S ;
// Select $K - K_0$ more samples
incrementally
for $k = K_0, \dots, K - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute d_n^{xy} in (8);
 end
 Identify the \mathbf{x}' that has the largest d_n^{xy} ;
 Move \mathbf{x}' from Z to S ;
 Query to label \mathbf{x}' in S ;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
 Update the regression model $f(\mathbf{x})$ using S .
end

2.4 Multi-Task iGS (MT-iGS)

This subsection extends the single-task iGS to multi-task iGS (MT-iGS). Similar to MT-GSy, here we again consider a simple multi-task setting that all P tasks share the same inputs.

MT-iGS first uses iGS to select and label the K_0 samples. It then builds P regression models $\{f_p(\mathbf{x})\}_{p=1}^P$ for the P tasks, and next selects the remaining $K - K_0$ samples to achieve diversity in both the input and output spaces. Without loss of generality, assume the first k ($k \geq K_0$) samples have already

been labeled with outputs $\{\mathbf{y}_m\}_{m=1}^k$, and P regression models $f_p(\mathbf{x})$ ($p = 1, \dots, P$) have been built. For each of the remaining $N - k$ unlabeled samples $\{\mathbf{x}_n\}_{n=k+1}^N$, MT-iGS computes d_{nm}^x in (7) and $d_{nm,p}^y$ in (5), and then d_n^{xy} :

$$d_n^{xy} = \min_m d_{nm}^x \prod_{p=1}^P d_{nm,p}^y, \quad n = k + 1, \dots, N \quad (9)$$

and selects the sample with the maximum d_n^{xy} to label. The pseudo-code of MT-iGS is given in Algorithm 4.

Algorithm 4: MT-iGS for multi-task ALR.

Input: N unlabeled samples, $\{\mathbf{x}_n\}_{n=1}^N$;
 K , the maximum number of labels to query.
Output: P regression models $f_p(\mathbf{x})$, $p = 1, \dots, P$.
// Initialize the first selection
Set $Z = \{\mathbf{x}_n\}_{n=1}^N$, and $S = \emptyset$;
Identify \mathbf{x}' , the sample closest to the centroid of Z ;
Move \mathbf{x}' from Z to S ;
Re-index the sample in S as \mathbf{x}_1 , and the samples in Z as $\{\mathbf{x}_n\}_{n=2}^N$;
// Select $K_0 - 1$ more samples
incrementally using GS in the input space
Identify K_0 , the minimum number of labeled samples required to construct $f(\mathbf{x})$;
for $k = 1, \dots, K_0 - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute d_n^x in (2);
 end
 Identify the \mathbf{x}' that has the largest d_n^x ;
 Move \mathbf{x}' from Z to S ;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
end
Query to label the K_0 samples in S , for all P tasks;
Construct P regression models $f_p(\mathbf{x})$ from S , one for each task;
// Select $K - K_0$ more samples
incrementally
for $k = K_0, \dots, K - 1$ **do**
 for $n = k, \dots, N$ **do**
 Compute d_n^{xy} in (9);
 end
 Identify the \mathbf{x}' that has the largest d_n^{xy} ;
 Move \mathbf{x}' from Z to S ;
 Query to label \mathbf{x}' in S , for all P tasks;
 Re-index the samples in S as $\{\mathbf{x}_m\}_{m=1}^{k+1}$, and the samples in Z as $\{\mathbf{x}_n\}_{n=k+2}^N$;
 Update the P regression models $f_p(\mathbf{x})$ using S .
end

Similar to MT-GSy, we can also note that:

- 1) In (9) we combine d_{nm}^x and the P $d_{nm,p}^y$ using product instead of summation, to avoid the problem that the inputs and different task outputs may have different scales, and hence one distance may dominate others.
- 2) MT-iGS degrades to the single-task iGS when $P = 1$.

3 EXPERIMENT

The VAM corpus [7] is used in this section to demonstrate the performances of MT-GSy and MT-iGS.

3.1 Dataset and Feature Extraction

The VAM corpus was released in ICME2008 [7] and has been used in many studies [5], [6], [26], [27]. It contains spontaneous speech with authentic emotions recorded from guests in a German TV talk-show *Vera am Mittag* (*Vera at Noon* in English). There are 947 emotional utterances from 47 speakers (11m/36f). Each sentence was evaluated by 6-17 listeners in the 3D space of valence, arousal and dominance, and the evaluations were merged by a weighted average to obtain the groundtruth emotion primitives in $[-1, 1]$ [4].

The same 46 acoustic features extracted in our previous research [26], [27] were used again in this paper. They included nine pitch features, five duration features, six energy features, and 26 Mel Frequency Cepstral Coefficient (MFCC) features. Each feature was then normalized to mean 0 and standard deviation 1.

3.2 Sample Selection Algorithms

We compared the performances of nine sample selection algorithms:

- 1) Baseline 1 (BL1), which randomly selects all K samples.
- 2) Baseline 2 (BL2), which assumes all samples in the training pool are labeled, and uses them to build a regression model. BL2 represents the upper bound of the performance we could get given a specific training pool.
- 3) Expected model change maximization (EMCM) [2], which selects the sample with the maximum expected model change to label. EMCM is for single-task learning.
- 4) Query-by-Committee (QBC) [16], which selects the sample with the maximum variance (computed from a committee of regression models) to label. QBC is for single-task learning.
- 5) GSx, which was introduced in our recent research [25]. It is almost identical to the GS approach in [29], except that the first sample is selected as the one closest to the centroid of all N unlabeled samples. Since GSx considers only the diversity in the input space, it can be used in both single-task and MTL settings, without any modification.
- 6) GSy, which has been introduced in Section 2.1.
- 7) MT-GSy, which has been introduced in Section 2.2.
- 8) iGS, which has been introduced in Section 2.3.
- 9) MT-iGS, which has been introduced in Section 2.4.

3.3 Performance Evaluation Process

For the 947 samples in the VAM corpus, we first randomly selected 30% as the training pool and the remaining 70% as the test dataset, initialized the first K_0 labeled samples (K_0 is the dimensionality of the input space) either randomly (for BL, QBC and EMCM) or by GSy (for GSx, GSy, iGS, MT-GSy, and MT-iGS), identified one sample to label in each iteration by different algorithms, built linear regression models, and computed the root mean squared error (RMSE) and correlation coefficient (CC) as the performance measures on the test dataset. The iteration terminated when all samples in the training pool were selected.

To obtain statistically meaningful results, we ran this evaluation process 100 times for each algorithm, each time with a randomly chosen training pool containing 30% unlabeled samples.

3.4 Experimental Results

First, ridge regression (RR) was used as the linear regression model, and $\lambda = 10/K$ was used in its objective function $\min_{\beta} (\|y - X\beta\|^2 + \lambda\|\beta\|^2)$. Given a fixed training pool, GSx will select a fixed sequence of samples to label because it only considers the diversity in the input space, regardless of how many tasks are there. MT-GSy (MT-iGS) also generates a fixed sequence of samples to label because it always considers all tasks simultaneously. However, each single-task ALR approach (EMCM, QBC, GSy and iGS) will give a different sequence of samples when a different task is considered. So, we compare the performances of the sample selection algorithms under three scenarios: 1) *Valence* estimation is considered in the single-task ALR approaches; 2) *Arousal* estimation is considered in the single-task ALR approaches; and, 3) *Dominance* estimation is considered in the single-task ALR approaches.

The results are shown in Figs. 1-3, respectively, where the RMSEs and CCs have been averaged over 100 runs. In each subfigure, the first column shows the results when the single-task ALR approaches focused on Valence estimation, the second column on Arousal estimation, and the third column on Dominance estimation. The last column shows the average across the first three columns.

Observe from Fig. 1 that:

- 1) Generally as K increased, all eight sample selection algorithms (excluding BL2, which did not change with K) achieved better performance (smaller RMSE and larger CC), which is intuitive, because more labeled training samples generally result in a more reliable RR model.
- 2) When $K = K_0 = 46$ (the first point in each subfigure of Fig. 1), the five GS based ALR approaches (GSx, GSy, iGS, MT-GSy and MT-iGS), which initialized the K_0 samples by considering the diversity in the input space, all had better performances than the other three approaches (BL1, EMCM and QBC), which initialized the K_0 samples randomly.
- 3) For Valence estimation, which was the task that all single-task ALR approaches focused on, GSy and iGS achieved comparable performance as MT-GSy and MT-iGS; however, for the other two tasks (Arousal and Dominance estimations), MT-GSy and MT-iGS achieved better performances.
- 4) When all three tasks are considered together (the last column of Fig. 1), on average all ALR approaches outperformed BL1, all GS based approaches outperformed EMCM and QBC, and both MT-iGS and MT-GSy outperformed their single-task counterparts. For a given K , the average performances were generally in the order of MT-iGS > MT-GSy > iGS > GSy > GSx > EMCM > QBC > BL1.

Similar observations can also be made from Figs. 2 and 3, except that in Fig. 2 MT-GSy and GSy (MT-iGS and iGS) had comparable performances on Arousal estimation, and in Fig. 3 MT-GSy and GSy (MT-iGS and iGS) had comparable performances on Dominance estimation.

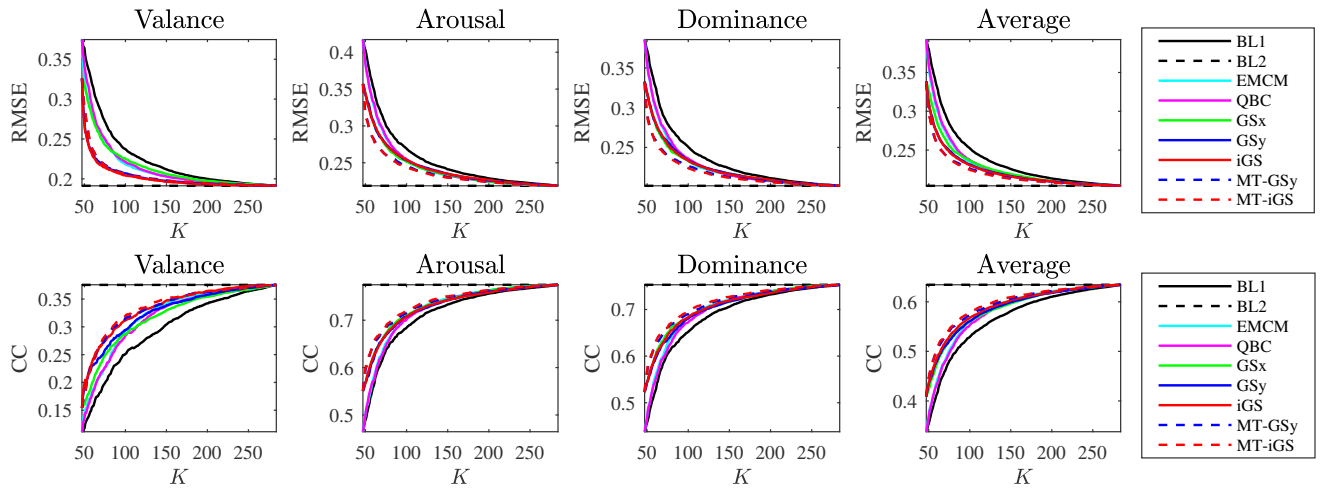


Fig. 1. Performances of the sample selection algorithms, averaged over 100 runs, when the single-task ALR approaches focused on *Valence* estimation. The last column shows the average RMSE and CC across the three tasks. RR was used as the regression model.

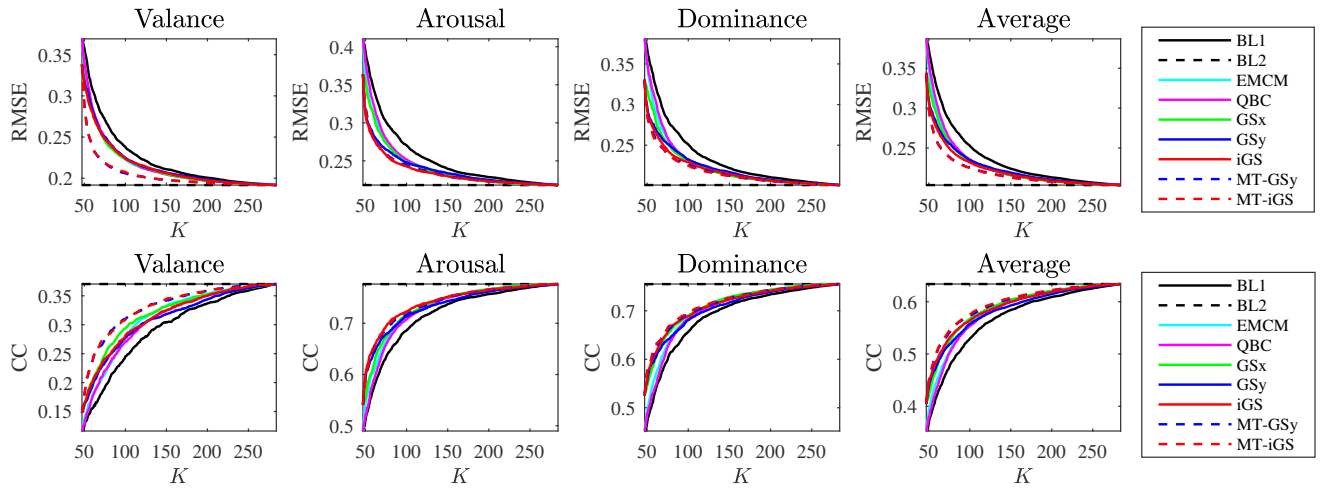


Fig. 2. Performances of the sample selection algorithms, averaged over 100 runs, when the single-task ALR approaches focused on *Arousal* estimation. The last column shows the average RMSE and CC across the three tasks. RR was used as the regression model.

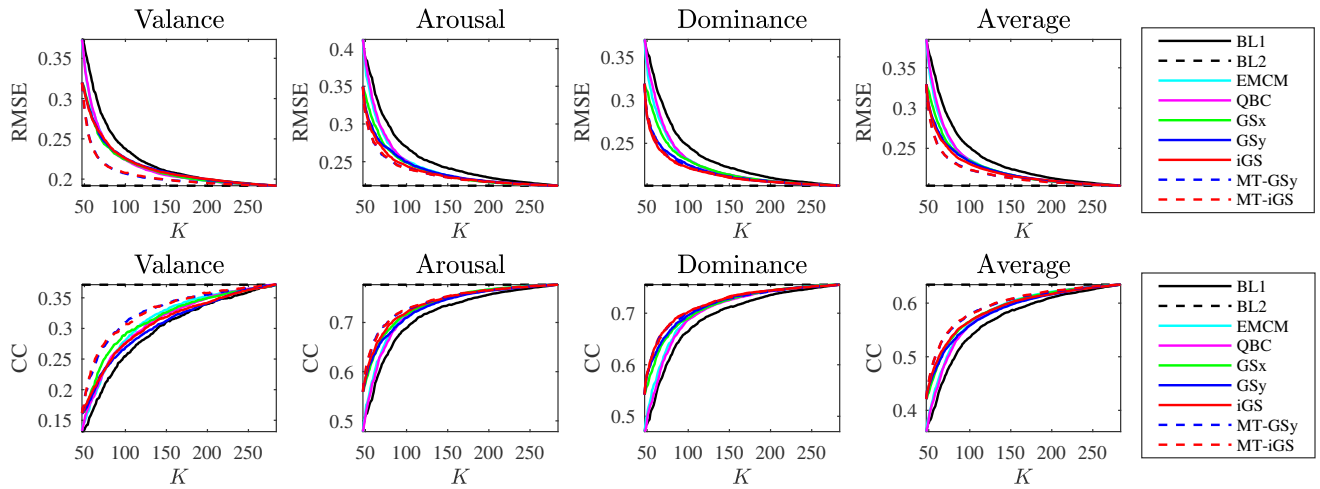


Fig. 3. Performances of the sample selection algorithms, averaged over 100 runs, when the single-task ALR approaches focused on *Dominance* estimation. The last column shows the average RMSE and CC across the three tasks. RR was used as the regression model.

To quantify the performance improvements of different ALR approaches over the random sampling approach (BL1), we picked $K = \{50, 100, 150, 200, 250\}$ and recorded the performances (RMSE and CC) of BL1, shown in the fourth column of Table 1. Then, we also show the performances of the seven ALR approaches in Columns 5-11 of Table 1, where the numbers in the parentheses representing the performance improvements over the corresponding BL1 performance, and the best two are marked in bold. For example, the first row shows that for Valence, using 50 samples, BL1 achieved an RMSE of 0.380, whereas EMCM achieved an RMSE of 0.356, representing a $(0.380 - 0.356)/0.380 = 6\%$ improvement.

Table 1 shows that:

- 1) Given a specific K , all ALR approaches outperformed BL1 in both RMSE and CC. Among them, MT-GSy and MT-iGS almost always achieved the best performance.
- 2) As K increased, the performance improvement of ALR approaches decreased. This is because when K becomes larger, the samples selected by different approaches, no matter ALR or BL1, overlap more, and hence the performance differences among them become smaller.

These observations are consistent with those made from Figs. 1-3.

3.5 ALR Saved the Number of Queries over BL1

The improved performances of ALR can also be verified by quantifying the numbers of saved queries over BL1, when a desired regression performance is needed.

To do this, we first count the number of labeled samples required for BL1 to achieve $(100 + \alpha)\%$ RMSE of BL2 ($\alpha = \{1, 2, 3, 5, 10\}$), and $(100 - \alpha)\%$ CC of BL2, as shown in the fourth column of Table 2. We then also count the number of samples required by different ALR approaches, and the corresponding saving over BL1, as shown in the remaining columns of Table 2. For example, the first row shows that for Valence, to achieve 101% ($\alpha = 1$) RMSE of BL2, BL1 needed 261 labeled samples, whereas EMCM only needed 242 samples, representing a $(261 - 242)/261 = 8\%$ saving.

Table 2 shows that:

- 1) All ALR approaches can save the number of queries over BL1. Among them, MT-GSy and MT-iGS, particularly MT-iGS, almost always saved the most number of queries.
- 2) As α increased, the percentage of saving also increased for almost all ALR approaches, especially MT-GSy and MT-iGS.

These observations are consistent with those made in the previous subsection.

3.6 Model Parameters from ALR Converged Faster

As BL2 used all samples in the training pool, the regression coefficients obtained from BL2 represented the global optimum. It's interesting to study how fast the model parameters from different approaches converged to the solution given by BL2. The mean absolute errors (MAEs) between the coefficients of BL2 and the other eight approaches for different K are shown in Fig. 4. To save space, we only show the results when single-task ALR approaches focused on *Valence*.

Fig. 4 shows that:

- 1) Generally as K increased, the model parameters from all eight sample selection algorithms converged to the solution of BL2.
- 2) When $K = K_0 = 46$ (the first point in each subfigure of Fig. 4), the five GS based ALR approaches (GSx, GSy, iGS, MT-GSy and MT-iGS), which initialized the K_0 samples by considering the diversity in the input space, all had smaller MAEs than the other three approaches (BL, EMCM and QBC), which initialized the K_0 samples randomly.
- 3) GSy and iGS achieved comparable MAEs with MT-GSy and MT-iGS on Valence estimation, because this was the task that GSy and iGS focused on; however, for the other two tasks (Arousal and Dominance estimations), MT-GSy and MT-iGS achieved smaller MAEs.
- 4) When all three tasks are considered together (the last column of Fig. 4), generally all ALR approaches converged faster than BL1, all GS based approaches converged faster than EMCM and QBC, and both MT-iGS and MT-GSy converged faster than their single-task counterparts.

3.7 Sample Selection Results: Impact of Gender

The VAM dataset consists of 947 utterances, 196 of which are from males (20.7%), and 751 from females (79.3%). Male and female utterances have different feature standard deviations, as shown in Fig. 5. For 27 of the 46 features, male utterances have larger standard deviations than female utterances.

As the initial 46 samples from the GS based ALR approaches are selected based on the diversity in the feature space, and the male utterances have larger feature variations than the female utterances, we expect that more male utterances would be selected by the GS based ALR approaches in initialization than other approaches. Fig. 6, which shows the percentage of male utterances selected by different algorithms, confirms this. When $K = 46$, BL1, EMCM and QBC selected about 20.7% male utterances, which is the average percentage of male utterances in the dataset. However, all five GS based ALR approaches selected over 33% male utterances, much larger than the average percentage. As K increased, this percentage gradually decreased. Interestingly, the percentage of male utterances selected by EMCM and QBC first increased with K , and then gradually decreased.

3.8 Sample Selection Results: Impact of Emotion Primitive Values

To study how the values of the emotion primitives impacted the sample selection algorithms, we computed the standard deviation of the primitives of the selected samples, and show the results in Fig. 7, when the single-task ALR approaches focused on Valence. The samples selected by the five GS based ALR approaches generally had larger standard deviations than those by other approaches, especially when K was small. This is intuitive, as GS tends to select the most diverse samples. The first K_0 samples selected by EMCM and QBC had the same standard deviation as those by BL1, as all of them were random. However, as K increased, the standard deviation of the samples selected by EMCM and QBC increased rapidly and were much larger than that by BL1. The standard deviation of the samples selected by EMCM was larger than that by QBC, and the performance of EMCM was also slightly better than QBC (Table 1).

TABLE 1

Performances and percentages of improvement (in the parentheses) when different ALR approaches are compared with BL1. The best two in each row are marked in bold.

Emotion Primitive	Performance Measure	K	BL1	Performance and percentage improvement over BL1						
				EMCM	QBC	GSx	GSy	iGS	MT-GSy	MT-iGS
Valence	RMSE	50	0.380	0.356 (6%)	0.361 (5%)	0.326 (14%)	0.311 (18%)	0.310 (18%)	0.300 (21%)	0.299 (21%)
		100	0.252	0.235 (7%)	0.237 (6%)	0.237 (6%)	0.232 (8%)	0.230 (9%)	0.226 (10%)	0.225 (11%)
		150	0.226	0.217 (4%)	0.217 (4%)	0.219 (3%)	0.216 (4%)	0.216 (4%)	0.214 (5%)	0.213 (6%)
		200	0.213	0.210 (2%)	0.210 (2%)	0.210 (1%)	0.210 (2%)	0.210 (2%)	0.209 (2%)	0.208 (2%)
		250	0.207	0.206 (1%)	0.206 (1%)	0.206 (1%)	0.206 (1%)	0.206 (1%)	0.206 (1%)	0.205 (1%)
	CC	50	0.354	0.371 (5%)	0.367 (4%)	0.424 (20%)	0.434 (23%)	0.437 (23%)	0.446 (26%)	0.448 (26%)
		100	0.529	0.560 (6%)	0.553 (5%)	0.560 (6%)	0.561 (6%)	0.568 (7%)	0.574 (8%)	0.579 (9%)
		150	0.581	0.604 (4%)	0.600 (3%)	0.597 (3%)	0.599 (3%)	0.603 (4%)	0.606 (4%)	0.609 (5%)
		200	0.610	0.621 (2%)	0.619 (1%)	0.618 (1%)	0.618 (1%)	0.619 (1%)	0.622 (2%)	0.623 (2%)
		250	0.626	0.630 (1%)	0.630 (1%)	0.629 (1%)	0.630 (1%)	0.631 (1%)	0.630 (1%)	0.631 (1%)
Arousal	RMSE	50	0.374	0.350 (6%)	0.357 (4%)	0.330 (12%)	0.311 (17%)	0.308 (18%)	0.300 (20%)	0.298 (20%)
		100	0.253	0.235 (7%)	0.236 (7%)	0.234 (7%)	0.235 (7%)	0.232 (8%)	0.226 (11%)	0.225 (11%)
		150	0.224	0.217 (3%)	0.216 (4%)	0.216 (4%)	0.219 (2%)	0.217 (3%)	0.213 (5%)	0.213 (5%)
		200	0.213	0.209 (2%)	0.209 (2%)	0.209 (2%)	0.210 (1%)	0.209 (2%)	0.208 (3%)	0.208 (3%)
		250	0.207	0.205 (1%)	0.205 (1%)	0.205 (1%)	0.206 (1%)	0.205 (1%)	0.205 (1%)	0.205 (1%)
	CC	50	0.368	0.393 (7%)	0.379 (3%)	0.419 (14%)	0.436 (18%)	0.442 (20%)	0.447 (21%)	0.449 (22%)
		100	0.529	0.559 (6%)	0.554 (5%)	0.567 (7%)	0.557 (5%)	0.564 (7%)	0.573 (8%)	0.576 (9%)
		150	0.584	0.603 (3%)	0.599 (3%)	0.604 (3%)	0.593 (1%)	0.600 (3%)	0.606 (4%)	0.608 (4%)
		200	0.609	0.620 (2%)	0.620 (2%)	0.621 (2%)	0.615 (1%)	0.619 (2%)	0.622 (2%)	0.622 (2%)
		250	0.626	0.630 (1%)	0.630 (1%)	0.630 (1%)	0.628 (0%)	0.630 (1%)	0.631 (1%)	0.631 (1%)
Dominance	RMSE	50	0.370	0.354 (4%)	0.359 (3%)	0.321 (13%)	0.304 (18%)	0.303 (18%)	0.296 (20%)	0.296 (20%)
		100	0.251	0.236 (6%)	0.235 (6%)	0.235 (7%)	0.233 (7%)	0.231 (8%)	0.224 (11%)	0.224 (11%)
		150	0.224	0.217 (3%)	0.217 (3%)	0.217 (3%)	0.217 (3%)	0.216 (4%)	0.213 (5%)	0.213 (5%)
		200	0.213	0.209 (2%)	0.209 (2%)	0.209 (2%)	0.210 (2%)	0.210 (1%)	0.208 (2%)	0.208 (2%)
		250	0.207	0.205 (1%)	0.205 (1%)	0.205 (1%)	0.205 (1%)	0.206 (1%)	0.205 (1%)	0.205 (1%)
	CC	50	0.377	0.388 (3%)	0.384 (2%)	0.433 (15%)	0.445 (18%)	0.446 (18%)	0.453 (20%)	0.454 (21%)
		100	0.536	0.560 (5%)	0.559 (4%)	0.566 (6%)	0.558 (4%)	0.566 (6%)	0.579 (8%)	0.579 (8%)
		150	0.586	0.602 (3%)	0.600 (2%)	0.601 (3%)	0.597 (2%)	0.601 (3%)	0.607 (4%)	0.608 (4%)
		200	0.611	0.621 (2%)	0.619 (1%)	0.620 (2%)	0.616 (1%)	0.617 (1%)	0.621 (2%)	0.623 (2%)
		250	0.626	0.630 (1%)	0.630 (1%)	0.630 (1%)	0.629 (1%)	0.629 (1%)	0.630 (1%)	0.631 (1%)

TABLE 2

Number of samples and percentages of saved queries (in the parentheses) when different ALR approaches are compared with BL1. The best two in each row are marked in bold.

Emotion Primitive	Performance Measure	$\alpha\%$	No. BL1 Samples	Number of samples and percentage saving over BL1						
				EMCM	QBC	GSx	GSy	iGS	MT-GSy	MT-iGS
Valence	RMSE	1%	261	242 (8%)	248 (5%)	247 (6%)	246 (6%)	242 (8%)	243 (7%)	233 (12%)
		2%	241	218 (11%)	217 (11%)	221 (9%)	218 (11%)	216 (12%)	207 (16%)	202 (19%)
		3%	222	197 (13%)	197 (13%)	201 (10%)	194 (14%)	197 (13%)	183 (21%)	179 (24%)
		5%	197	168 (17%)	168 (17%)	175 (13%)	164 (20%)	162 (22%)	148 (33%)	144 (37%)
		10%	154	123 (25%)	126 (22%)	129 (19%)	118 (31%)	116 (33%)	106 (45%)	101 (52%)
	CC	1%	258	236 (9%)	242 (7%)	242 (7%)	244 (6%)	238 (8%)	242 (7%)	230 (12%)
		2%	235	202 (16%)	211 (11%)	211 (11%)	215 (9%)	208 (13%)	201 (17%)	194 (21%)
		3%	215	181 (19%)	187 (15%)	190 (13%)	189 (14%)	185 (16%)	175 (23%)	172 (25%)
		5%	184	149 (23%)	154 (19%)	162 (14%)	157 (17%)	150 (23%)	144 (28%)	135 (36%)
		10%	138	109 (27%)	115 (20%)	112 (23%)	110 (25%)	104 (33%)	98 (41%)	93 (48%)
Arousal	RMSE	1%	261	239 (9%)	237 (10%)	245 (7%)	252 (4%)	247 (6%)	231 (13%)	238 (10%)
		2%	242	213 (14%)	210 (15%)	216 (12%)	227 (7%)	220 (10%)	199 (22%)	196 (23%)
		3%	225	193 (17%)	192 (17%)	196 (15%)	208 (8%)	197 (14%)	176 (28%)	174 (29%)
		5%	196	165 (19%)	165 (19%)	167 (17%)	175 (12%)	161 (22%)	143 (37%)	139 (41%)
		10%	152	125 (22%)	125 (22%)	126 (21%)	126 (21%)	116 (31%)	98 (55%)	97 (57%)
	CC	1%	261	235 (11%)	235 (11%)	242 (8%)	247 (6%)	241 (8%)	231 (13%)	228 (14%)
		2%	241	202 (19%)	203 (19%)	208 (16%)	219 (10%)	210 (15%)	198 (22%)	188 (28%)
		3%	223	181 (23%)	184 (21%)	186 (20%)	200 (12%)	188 (19%)	174 (28%)	165 (35%)
		5%	185	147 (26%)	155 (19%)	155 (19%)	168 (10%)	152 (22%)	135 (37%)	132 (40%)
		10%	136	110 (24%)	113 (20%)	105 (30%)	115 (18%)	105 (30%)	92 (48%)	91 (49%)
Dominance	RMSE	1%	261	237 (10%)	238 (10%)	237 (10%)	249 (5%)	242 (8%)	232 (13%)	235 (11%)
		2%	242	210 (15%)	212 (14%)	213 (14%)	221 (10%)	216 (12%)	204 (19%)	201 (20%)
		3%	225	192 (17%)	194 (16%)	191 (18%)	204 (10%)	193 (17%)	182 (24%)	180 (25%)
		5%	197	165 (19%)	166 (19%)	162 (22%)	174 (13%)	164 (20%)	148 (33%)	146 (35%)
		10%	151	123 (23%)	127 (19%)	124 (22%)	126 (20%)	121 (25%)	106 (42%)	103 (47%)
	CC	1%	258	236 (9%)	239 (8%)	231 (12%)	248 (4%)	238 (8%)	226 (14%)	229 (13%)
		2%	234	200 (17%)	210 (11%)	203 (15%)	219 (7%)	211 (11%)	196 (19%)	196 (19%)
		3%	217	181 (20%)	190 (14%)	181 (20%)	197 (10%)	186 (17%)	175 (24%)	173 (25%)
		5%	184	148 (24%)	159 (16%)	147 (25%)	167 (10%)	155 (19%)	143 (29%)	139 (32%)
		10%	133	110 (21%)	116 (15%)	104 (28%)	113 (18%)	107 (24%)	96 (39%)	95 (40%)

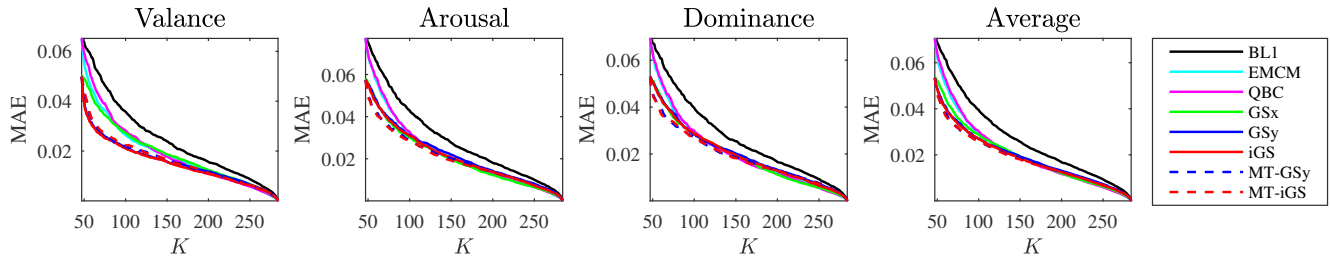


Fig. 4. MAE between the regression coefficients of BL2 and the other eight approaches, when the single-task ALR approaches focused on *Valence*. RR was used as the regression model.

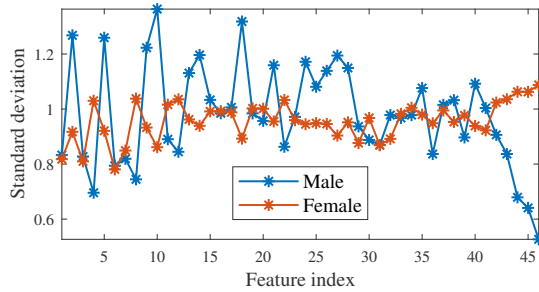


Fig. 5. Feature standard deviations of male and female utterances.

3.9 Impact of Features on Algorithm Performance

This subsection evaluates the performances of different ALR approaches w.r.t. different feature sets. We repeated the experiments in Section 3.4 using only the 26 MFCC features instead of all 46 features. To save space, here we only show the average results across the three tasks in Fig. 8, when the single-task ALR approaches focused on Valence. The average performances of the algorithms are in the order of MT-iGS > MT-GSy > iGS > GSy > GSx > EMCM > QBC > BL1. The RMSEs in Fig. 8 are similar to those in Fig. 1, but the CCs were generally smaller, i.e., reducing the number of features did not change the rank of the algorithms, but resulted in overall worse performances for all approaches.

3.10 Impact of Regression Models on Algorithm Performance

This subsection studies the stability of the proposed multi-task ALR approaches w.r.t. different regression models. To save space, we only present the results when the single-task ALR approaches focused on Valence.

First, ordinary least square regression was used as the linear regression model. We ran the experiments as previously for RR, where the single-task ALR approaches focused on different tasks. The results are shown in Fig. 9(a). Again we can make similar observations as those for RR, although initially ordinary least square regression performed much worse than RR, as there were not enough training samples to adequately determine the coefficients of the features, and no regularization was done either on the coefficients.

Next, LASSO was used as the linear regression model, where $\lambda = 0.001$ in its objective function $\min_{\beta} (\|y - X\beta\|^2 + \lambda \|\beta\|_1)$. The average results are shown in Fig. 9(b). Finally, elastic net was used as the linear regression model, where $\lambda_1 = \lambda_2 = 0.0005$ in its objective function $\min_{\beta} (\|y - X\beta\|^2 + \lambda_1 \|\beta\|_1) + \lambda_2 \|\beta\|^2$.

The average results are shown in Fig. 9(c). For a given K , the performances were in the order of MT-iGS \approx MT-GSy > iGS > GSy > GSx > EMCM > QBC > BL, consistent with those when RR was used as the regression model.

4 DISCUSSION: WHY MTL IN AFFECTIVE COMPUTING

The previous section has presented extensive experiments and comprehensive analyses, showing that our proposed multi-task ALR approaches, MT-GSy and MT-iGS, outperformed their single-task counterparts, GSy and iGS, and also three other state-of-the-art single-task ALR approaches in the literature. However, a natural question that a user may ask is: Why should MTL be used in affective computing at the first place, instead of viewing each emotion primitive estimation as a separate problem and acquiring the labeled samples completely independently?

Our argument is that usually it takes time to evaluate (label) an affective signal, whether it is text, image, utterance, video, or others. So, a multi-task labeling approach, i.e., evaluating a single affective signal and then assigning valence, arousal and dominance simultaneously to it, is much more efficient than the combination of three separate single-task approaches, i.e., first evaluating Signal 1 and assigning Valence to it, then evaluating a different Signal 2 and assigning Arousal to it, and finally evaluating another different Signal 3 and assigning Dominance to it. This is particularly true when the affective signals are long and time-consuming to evaluate, e.g., movies. To obtain a label each for valence, arousal and dominance, the multi-task approach requires the assessor to watch only one movie and then assign three primitive values, whereas the single-task approach needs the assessor to watch three different movies. The former is usually much faster, easier, and more user-friendly. A well-designed MTL algorithm, like MT-GSy or MT-iGS, lets one select a small number of most informative affective signals to label, and can achieve comparable performance as the combination of three optimal single-task algorithms, while saving a significant amount of evaluation time.

As an example, we performed an experiment using RR and 26 MFCC features on the VAM dataset. A random 30% pool was reserved for training, and the remaining 70% for testing. K increased from 26 to 150. The first 26 samples for GSy, iGS, MT-GSy and MT-iGS were selected using GSy and were the same for all four approaches. Then, MT-GSy and MT-iGS proceeded just as before, but we used GSy (iGS) to separately select the optimal samples for Valence, Arousal and Dominance. For each K , we counted the number of unique utterances selected by the multi-task and single-task approaches. For example, when $K = 50$, MT-GSy

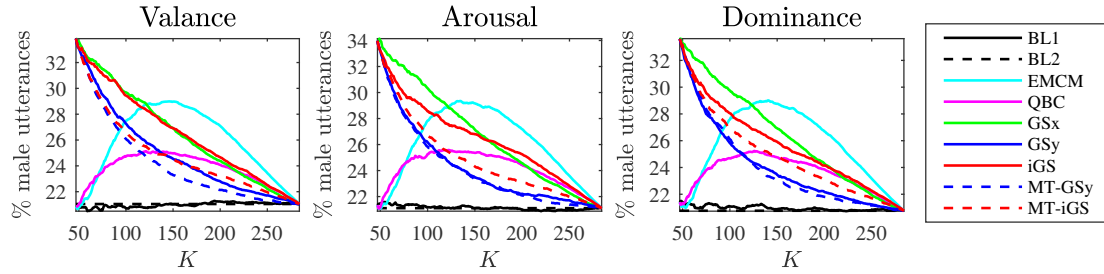


Fig. 6. Percentage of male utterances selected by different algorithms, when the single-task ALR approaches focused on *Valence*, *Arousal*, and *Dominance*, respectively. RR was used as the regression model.

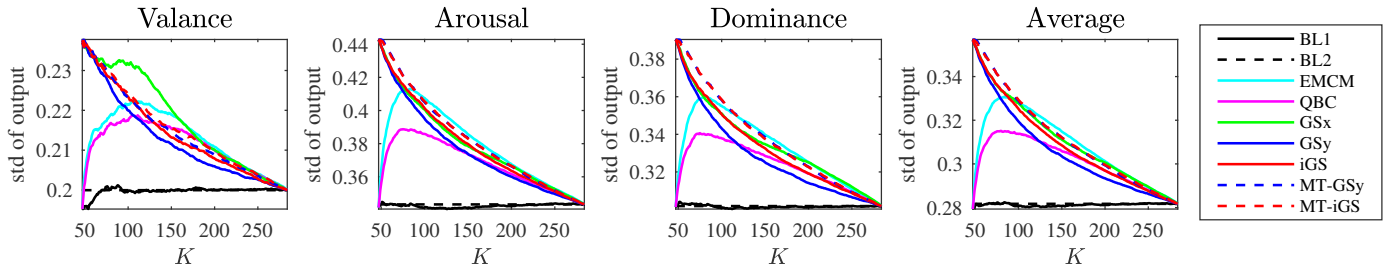


Fig. 7. Standard deviation (std) of the primitives of the selected samples, when the single-task ALR approaches focused on *Valence*. RR was used as the regression model.

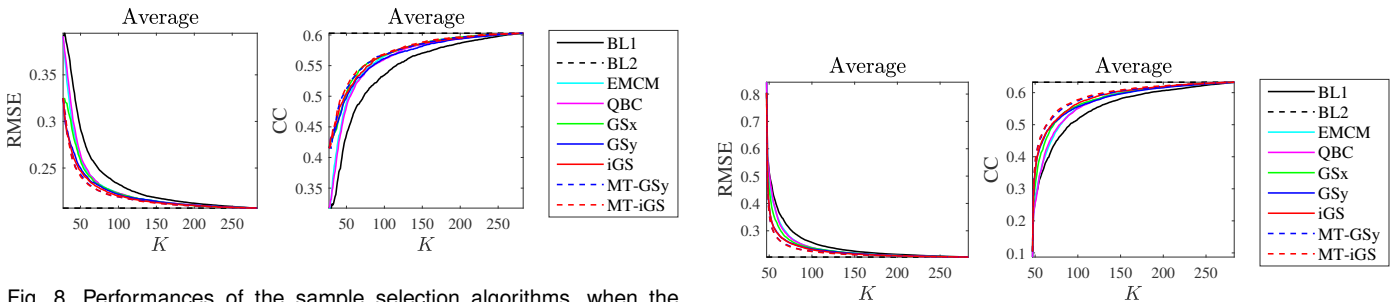


Fig. 8. Performances of the sample selection algorithms, when the single-task ALR approaches focused on *Valence*. RR was used as the regression model, and only the 26 MFCC features were used.

and MT-iGS each selected 50 unique utterances, but separately running GSy on Valence, Arousal and Dominance estimations selected 150 utterances, of which 87 were unique. The results are shown in Fig. 10. Clearly, the single-task ALR approaches used much more unique utterances than the multi-task ones.

Fig. 11 shows the estimation performances of the multi-task and single-task approaches, based on the samples selected in Fig. 10. MT-iGS and MT-GSy achieved similar performances as three independent single-task GSy or iGS combined.

In summary, MTL can achieve similar estimation performances as optimizing multiple single-tasks separately, but can significantly reduce the number of unique utterances that an assessor needs to evaluate. So, MTL is advantageous and effective in affective computing.

5 CONCLUSIONS AND FUTURE RESEARCH

Acquisition of labeled training samples for affective computing is usually costly and time-consuming, as multiple human assessors are needed to evaluate each affective sample. Particularly, for affect estimation in the 3D space of valence, arousal and dominance, each assessor has to perform the evaluations in three dimensions,

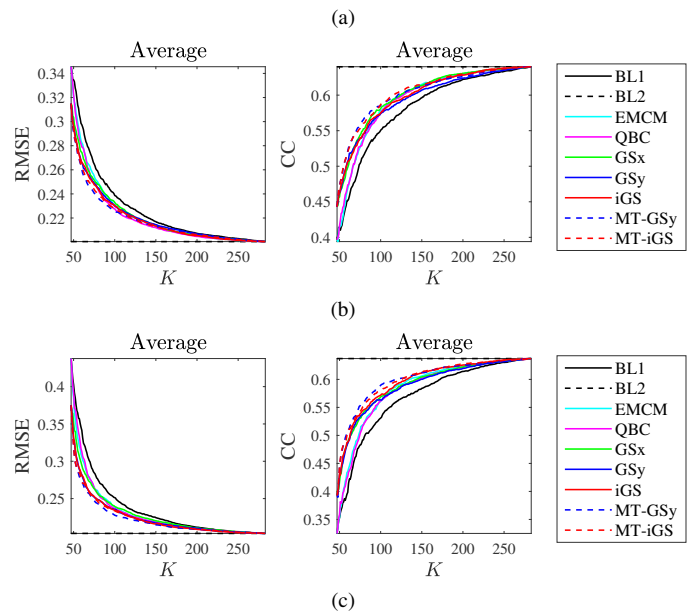


Fig. 9. Performances of the sample selection algorithms, when the single-task ALR approaches focused on *Valence*. Different regression models were used. (a) Ordinary least square regression; (b) LASSO; (c) Elastic net.

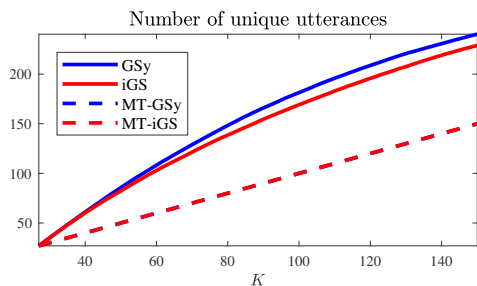


Fig. 10. Number of unique utterances when each single-task ALR approach (GSy and iGS) was independently optimized for the three emotion primitives. RR was used as the regression model, and only the 26 MFCC features were used. The curves for MT-GSy and MT-iGS are identical.

which makes the labeling problem even more challenging. This paper has proposed two MT-ALR approaches, MT-GSy and MT-iGS, which select the most informative samples to label, by considering the three affect primitives simultaneously. Experimental results on the VAM corpus demonstrated that MT-GSy and MT-iGS outperformed random selection and several traditional single-task ALR approaches, i.e., better affect estimation performance can be achieved when MT-GSy or MT-iGS is used to select the affective samples to label. Or, in other words, when a desired performance (RMSE or CC) is needed, using MT-GSy or MT-iGS can save the number of labelling queries.

Our future research directions include:

- 1) Extend MT-GSy and MT-iGS from regression to classification, as affects can also be classified simultaneously in multiple dimensions, e.g., paralinguistics in speech and language [19].
- 2) Extend our MT-ALR approaches from offline pool-based regression to online streaming regression [20].
- 3) Develop new MT-ALR approaches for nonlinear regression models, e.g., deep neural networks, as [25] shows that GSy and iGS do not perform well when nonlinear regression models are used.
- 4) Consider the more general case that different tasks use different inputs, in contrast to the case in this paper that all tasks share the same inputs.

REFERENCES

- [1] M. K. Abadi, A. Abad, R. Subramanian, N. Rostamzadeh, E. Ricci, J. Varadarajan, and N. Sebe, "A multi-task learning framework for time-continuous emotion estimation from crowd annotations," in *Proc. Int'l ACM Workshop on Crowdsourcing for Multimedia*, Orlando, FL: ACM, Nov. 2014, pp. 17–23.
- [2] W. Cai, Y. Zhang, and J. Zhou, "Maximizing expected model change for active learning in regression," in *Proc. IEEE 13th Int'l. Conf. on Data Mining*, Dallas, TX, December 2013.
- [3] O. Chapelle, B. Scholkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006.
- [4] M. Grimm and K. Kroschel, "Evaluation of natural emotions using self assessment manikins," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, San Juan, Puerto Rico, November 2005, pp. 381–385.
- [5] M. Grimm and K. Kroschel, "Emotion estimation in speech using a 3D emotion space concept," in *Robust Speech Recognition and Understanding*, M. Grimm and K. Kroschel, Eds. Vienna, Austria: I-Tech, 2007, pp. 281–300.
- [6] M. Grimm, K. Kroschel, E. Mower, and S. S. Narayanan, "Primitives-based evaluation and estimation of emotions in speech," *Speech Communication*, vol. 49, pp. 787–800, 2007.
- [7] M. Grimm, K. Kroschel, and S. S. Narayanan, "The Vera Am Mittag German audio-visual emotional speech database," in *Proc. Int'l Conf. on Multimedia & Expo (ICME)*, Hannover, German, June 2008, pp. 865–868.
- [8] A. Harpale, "Multi-task active learning," Ph.D. dissertation, Carnegie Mellon University, 2012.
- [9] Y. Jiang, F. L. Chung, H. Ishibuchi, Z. Deng, and S. Wang, "Multitask TSK fuzzy system modeling by mining intertask common hidden structure," *IEEE Trans. on Cybernetics*, vol. 45, no. 3, pp. 534–547, 2015.
- [10] S. Koelstra, C. Muhl, M. Soleymani, J. S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras, "DEAP: A database for emotion analysis using physiological signals," *IEEE Trans. on Affective Computing*, vol. 3, no. 1, pp. 18–31, 2012.
- [11] L. Li, X. Jin, S. J. Pan, and J.-T. Sun, "Multi-domain active learning for text classification," in *Proc. 18th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, Beijing, China, August 2012, pp. 1086–1094.
- [12] A. Marathe, V. Lawhern, D. Wu, D. Slayback, and B. Lance, "Improved neural signal classification in a rapid serial visual presentation task using active learning," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 3, pp. 333–343, 2016.
- [13] A. Mehrabian, *Basic Dimensions for a General Psychological Theory: Implications for Personality, Social, Environmental, and Developmental Studies*. Oelgeschlager, Gunn & Hain, 1980.
- [14] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [15] R. Picard, *Affective Computing*. Cambridge, MA: The MIT Press, 1997.
- [16] T. Raychaudhuri and L. Hamey, "Minimisation of data collection by active learning," in *Proc. IEEE Int'l. Conf. on Neural Networks*, vol. 3, Perth, Australia, November 1995, pp. 1338–1341.
- [17] R. Reichart, K. Tomanek, U. Hahn, and A. Rappoport, "Multi-task active learning for linguistic annotations," in *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, Columbus, OH, June 2008, pp. 861–869.
- [18] J. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.
- [19] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. Narayanan, "Paralinguistics in speech and language – state-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, 2013.
- [20] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.
- [21] C. Su, F. Yang, S. Zhang, Q. Tian, L. S. Davis, and W. Gao, "Multi-task learning with low rank attribute embedding for multi-camera person re-identification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1167–1181, 2018.
- [22] D. Wu, "Active semi-supervised transfer learning (ASTL) for offline BCI calibration," in *Proc. IEEE Int'l. Conf. on Systems, Man and Cybernetics*, Banff, Canada, October 2017.
- [23] D. Wu, "Pool-based sequential active learning for regression," *IEEE Trans. on Neural Networks and Learning Systems*, 2018, accepted.
- [24] D. Wu, V. J. Lawhern, W. D. Hairston, and B. J. Lance, "Switching EEG headsets made easy: Reducing offline calibration effort using active weighted adaptation regularization," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 11, pp. 1125–1137, 2016.
- [25] D. Wu, C.-T. Lin, and J. Huang, "Active learning for regression using greedy sampling," *Information Sciences*, 2018, submitted.
- [26] D. Wu, T. D. Parsons, E. Mower, and S. S. Narayanan, "Speech emotion estimation in 3D space," in *Proc. IEEE Int'l Conf. on Multimedia & Expo (ICME)*, Singapore, July 2010, pp. 737–742.
- [27] D. Wu, T. D. Parsons, and S. S. Narayanan, "Acoustic feature analysis in speech emotion primitives estimation," in *Proc. InterSpeech*, Makuhari, Japan, September 2010.
- [28] R. Xia and Y. Liu, "A multi-task learning framework for emotion recognition using 2D continuous space," *IEEE Trans. on Affective Computing*, vol. 8, no. 1, pp. 3–14, 2017.
- [29] H. Yu and S. Kim, "Passive sampling for regression," in *IEEE Int'l. Conf. on Data Mining*, Sydney, Australia, December 2010, pp. 1151–1156.
- [30] B. Zhang, E. M. Provost, and G. Essl, "Cross-corpus acoustic emotion recognition with multi-task learning: Seeking common ground while preserving differences," *IEEE Trans. on Affective Computing*, 2017, in press.
- [31] Y. Zhang, "Multi-task active learning with output constraints," in *Proc. 24th AAAI Conf. on Artificial Intelligence*, Atlanta, GA, July 2010.
- [32] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

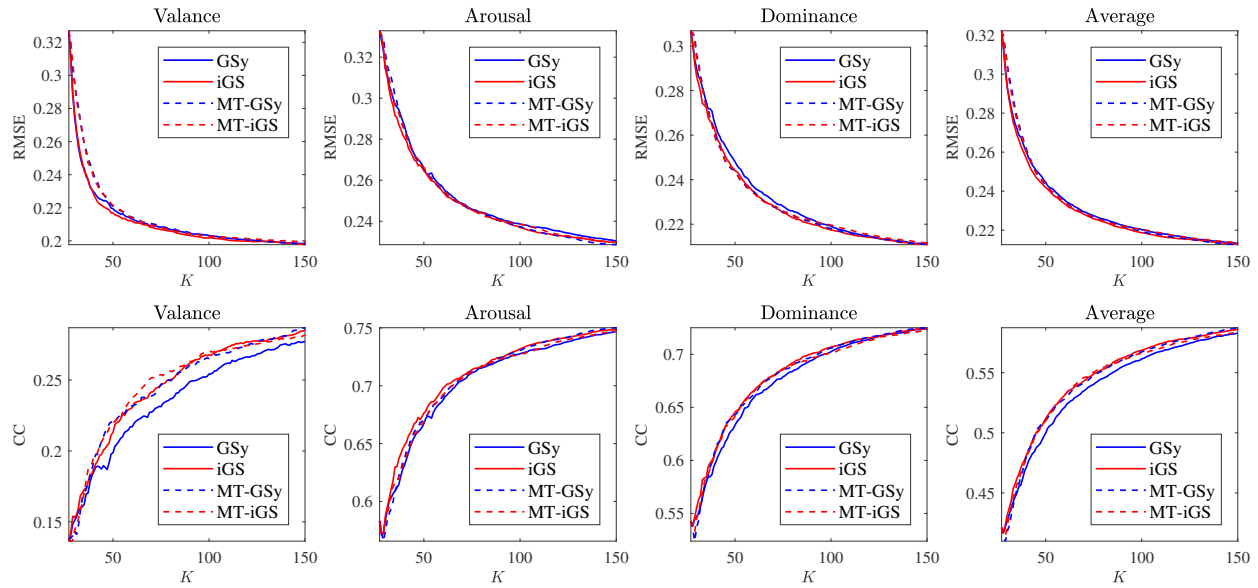


Fig. 11. Estimation performances when each single-task ALR approach (GSy and iGS) was independently optimized for the three emotion primitives. RR was used as the regression model, and only the 26 MFCC features were used.



Dongrui Wu (S'05-M'09-SM'14) received the BE degree in automatic control from the University of Science and Technology of China in 2003, the ME degree in electrical engineering from the National University of Singapore in 2005, and the PhD degree in electrical engineering from the University of Southern California in 2009. He is now Professor in the School of Automation, Huazhong University of Science and Technology, Wuhan, China. His research interests include affective computing, brain-computer interface, computational intelligence, and machine learning. He has more than 100 publications, including a book entitled *Perceptual Computing* (Wiley-IEEE, 2010).

Dr. Wu received the IEEE International Conference on Fuzzy Systems Best Student Paper Award in 2005, the IEEE CIS Outstanding PhD Dissertation Award in 2012, the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award in 2014, the NAFIPS Early Career Award in 2014, and the IEEE SMC Society Early Career Award in 2017. He was also a finalist of another three Best Paper Awards. He is an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS, the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, and the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE.



Jian Huang (M'07-SM'17) received the B.E., M.Eng., and Ph.D. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1997, 2000, and 2005, respectively. From 2006 to 2008, he was a Postdoctoral Researcher with the Department of Micro-Nano System Engineering and the Department of Mechano-Informatics and Systems, Nagoya University, Japan. He is currently a Full Professor with the School of Automation, HUST. His main research interests include rehabilitation robot, robotic assembly, networked control systems, and bioinformatics.

robot, robotic assembly, networked control systems, and bioinformatics.