Inductive Transfer Learning for Handling Individual Differences in Affective Computing

Dongrui Wu[1] and Thomas D. Parsons[2]

[1] Machine Learning Laboratory, GE Global Research Center
One Research Circle, Niskayuna, NY 12309 USA
wud@ge.com

[2] Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094 USA
tparsons@ict.usc.edu

Abstract. Although psychophysiological and affective computing approaches may increase facility for development of the next generation of human-computer systems, the data resulting from research studies in affective computing include large individual differences. As a result, it is important that the data gleaned from an affective computing system be tailored for each individual user by re-tuning it using user-specific training examples. Given the often time-consuming and/or expensive nature of efforts to obtain such training examples, there is a need to either 1) minimize the number of user-specific training examples required; or 2) to maximize the learning performance through the incorporation of auxiliary training examples from other subjects. In [11] we have demonstrated an active class selection approach for the first purpose. Herein we use transfer learning to improve the learning performance by combining user-specific training examples with auxiliary training examples from other subjects, which are similar but not exactly the same as the user-specific training examples. We report results from an arousal classification application to demonstrate the effectiveness of transfer learning in a Virtual Reality Stroop Task designed to elicit varying levels of arousal.

Keywords: Affective computing, arousal classification, individual differences, nearest neighbors classification, transfer learning.

1 Introduction

The use of psychophysiological measures in studies of persons immersed in high-fidelity virtual environment scenarios offers the potential to develop current physiological computing approaches [1] into affective computing [7] scenarios. An important task in implementing an affective computing system is affect recognition, which recognizes the user's affect from various signals, e.g., speech [12], facial expressions [5], physiological signals [10], and multimodal combination [15]. In [10,6] we introduced an adaptive virtual environment for assessment and rehabilitation of neurocognitive and affective functioning. The Virtual Reality Stroop Task (VRST) [6], utilized also in this paper, involves the subject being immersed

S. D´Mello et al. (Eds.): ACII 2011, Part II, LNCS 6975, pp. 142–151, 2011.

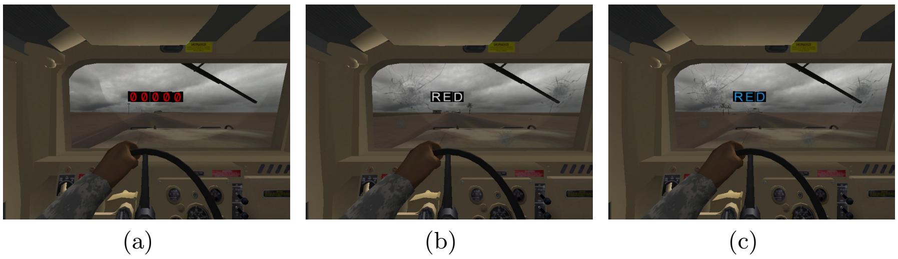

Fig. 1. The Humvee Stroop scenarios. (a) Color Naming; (b) Word Reading; and, (c) Interference.

into a virtual Humvee as it travels down the center of a road, during which Stroop stimuli [8] appear on the windshield, as shown in Fig. 1. The VRST stimuli are presented within both "safe" (low threat) and "ambush" (high threat) settings. Low threat zones consist of little activity aside from driving down a desert road, while the more stressful high threat zones include gunfire, explosions, and shouting amongst other stressors. Psychophysiological measures of skin conductance level, respiration, vertical electrooculograph, electrocardiographic activity, and electroencephalographic activity are recorded continuously throughout exposure to the virtual environment.

In [10] we used a support vector machine (SVM) to classify the three arousal levels in the VRST, and showed that when each subject is considered separately, an average classification rate of 96.5% can be obtained; however, the average classification rate was much lower (36.9%, close to random guess) when a subject's arousal level was predicted from other subjects' arousal levels.

We believe this reflects research into psychophysiology and individual differences. Results from a study conducted by Ito and Cacioppo [2] revealed that individuals respond with positivity offset and negativity bias. Positivity offset means that persons respond more strongly to mildly pleasant than to mildly unpleasant stimuli. Negativity bias means that a person responds more strongly to very unpleasant stimuli than to very pleasant stimuli. Individual differences were quite obvious in our previous experiments as we also performed feature selection in constructing the individual classifiers, and the features for different subjects were significantly different from each other.

Given the large individual differences, it is difficult to accurately classify a subject's arousal levels using a classifier trained from only responses from other subjects. However, the responses from other subjects still contain some useful information, as people exhibit similar (though usually not exactly the same) behaviors at the same affect state (otherwise we cannot recognize others' affects in social activities). So, increased classification accuracy may be obtained by combining the responses from other subjects and a small number of training examples from the subject. This is the idea of transfer learning [4], which is elaborated in the next section. If this hypothesis is veridical, then only a very small number of user-specific training examples are needed to individualize an affective computing system, which will greatly increase its usability and popularity. This paper presents some experimental results on transfer learning for handling individual differences in arousal classification and proposes several future research directions. To the best of the authors' knowledge, this is the first time that transfer learning has been introduced to the affective computing community.

The remainder of this paper is organized as follows: Section 2 introduces the concept of transfer learning. Section 3 presents some experimental results on transfer learning for handling individual differences in arousal classification. Section 4 draws conclusions and proposes some future research directions.

2 Transfer Learning

A major assumption in many classification and prediction algorithms is that the training and future (test) data are in the same feature space and have the same distribution [4]. However, it does not hold in many real-world applications. For example, in the arousal level classification experiment introduced above, a subject's physiological responses at a certain arousal level are generally quite different from another's, and their perceptions of arousal are also different. In such cases, knowledge transfer, if done successfully, would greatly improve the learning performance by eliminating much training example acquisition efforts. Transfer learning [4, 13] is a framework for addressing this problem.

Definition 1 ***(Transfer Learning).*** *[4] Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

In the above definition, a domain is a pair $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where $\mathcal{X}$ is a feature space and $P(X)$ is a marginal probability distribution, in which $X = \{x_1, ..., x_n\} \in \mathcal{X}$. $\mathcal{D}_s \neq \mathcal{D}_T$ means that either $\mathcal{X}_s \neq \mathcal{X}_T$ or $P_S(X) \neq P_T(X)$, i.e., either the features in the source domain and the target domain are different, or their marginal probability distributions are different. Similarly, a task is a pair $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$, where $\mathcal{Y}$ is a label space and $P(Y|X)$ is a conditional probability distribution. $\mathcal{T}_S \neq \mathcal{T}_T$ means that either $\mathcal{Y}_S \neq \mathcal{Y}_T$ or $P(Y_S|X_S) \neq P(Y_T|X_T)$, i.e., either the label spaces between the source and target domains are different, or the conditional probability distributions between the source and target domains are different.

Particularly, in this paper we are interested in *inductive transfer learning* [4], where $\mathcal{T}_S \neq \mathcal{T}_T$ but there is no requirement on whether $\mathcal{D}_S$ should be the same as $\mathcal{D}_T$ or not. As a result, a few labeled data in the target domain are required as training data to induce the target predictive function.

In the transfer learning literature the data in the source domain is usually called *auxiliary data* and the data in the target domain *primary data*. For the arousal level classification application introduced in the Introduction, primary data are user-specific training examples, and auxiliary data can be training examples from other subjects. In [13] it has been shown that when the primary training dataset is very small, training with auxiliary data can significantly improve classification accuracy, even when the auxiliary data is significantly different from the primary data. This result can be understood through a bias/variance analysis. When the number of primary training data is very small, a learned classifier will have large variance and hence large error. Incorporating auxiliary data, which increases the number of training examples, can effectively reduce this variance, but possibly increase the bias, because the auxiliary and primary training data have different distributions. This also suggests that as the amount of primary training data increases, the utility of auxiliary data should decrease [13].

3 Experiment

In this section we present some experimental results on transfer learning for handling individual differences in arousal classification. Data for this experiment was drawn from the VRST. Psychophysiological measures were used to predict levels of threat and cognitive workload. Herein primary data are user-specific training examples, and auxiliary data are training examples from other subjects. For simplicity, we use the k-nearest neighbors (kNN) classifier.

3.1 Method

Suppose there are N^p training examples $\{\mathbf{x}_i^p, y_i^p\}_{i=1,2,\ldots,N^p}$ for the primary supervised learning problem (user-specific training examples), where $\mathbf{x}_i^p$ is the feature vector of the ith training example and y_i^p is its corresponding class label. The superscript p indicates the *primary* learning task. Additionally, there are N^a auxiliary training examples (training examples from other subjects) $\{\mathbf{x}_i^a, y_i^a\}_{i=1,2,\ldots,N^a}$, whose distribution is somehow similar to the primary training examples but not exactly the same. So, the auxiliary training examples should be treated as weaker evidence in designing a classifier.

In kNN we need to optimize the number of NNs, k. This must be done through internal I-fold cross-validation [9, 13]. The most important parameter in determining the optimal k is the *internal I-fold cross-validation accuracy*, i.e., the number of the internal cross-validation examples that are correctly classified, n_v. However, because N^p is very small, different k may easily result in the same n_v. For example, when $N^p = 3$ and $I = 3$, there are a total of three examples in the internal 3-fold cross-validation (one in each fold, and the process is repeated three times); so, n_v can only be $\{0, 1, 2, 3\}$, and several different k may result in $n_v = 3$. The *classification margin in internal cross-validation*, m_v, is then used to break the ties: when two ks give the same number of n_v, the one with a larger m_v is preferred. This idea was motivated by the concept of error margin in [13]. In this paper m_v is computed as the number of votes for the correct class minus the number of votes for the predicted class, summed over all instances in the internal I-fold cross-validation. So, m_v is a non-positive number, and the larger the better. Once the optimal ks are identified for different kNN classifiers, their performances can be compared using the testing accuracy.

As pointed out in [13], in many learning algorithms, the training data play two separate roles. One is to help define the objective function, and the other is to help define the hypothesis. Particularly, in kNN one role of the auxiliary data is to help define the objective function and the other is to serve as potential neighbors. In this paper we investigate these two roles separately. The following three kNN classifiers were implemented and compared:

1. kNN$_1$, which is a baseline kNN classifier without using the auxiliary data, i.e., it uses only the N^p primary training examples in the internal I-fold cross-validation algorithm.
2. kNN$_2$, which also uses the N^a auxiliary examples in the training part of the internal I-fold cross-validation algorithm, i.e., in each iteration of the internal cross-validation it combines the $I-1$ folds of the N^p primary training examples with the N^a auxiliary examples in training and uses the rest fold of the N^p primary training examples in validation.
3. kNN$_3$, which also uses the N^a auxiliary examples in the validation part of the internal I-fold cross-validation algorithm, i.e., in each iteration it uses the $I-1$ folds of the N^p primary training examples in training and combines the rest fold of the N^p primary training examples with the N^a auxiliary examples in validation.

The pseudo-codes for training kNN$_2$ and kNN$_3$ are given below. The pseudo-code for training kNN$_1$ is the same as that for kNN$_2$, except that the N^a auxiliary data are not used at all. Note that in testing kNN$_1$ and kNN$_3$ use their optimal k and the N^p primary training examples, whereas kNN$_2$ uses its optimal k and the N^p primary training examples plus the N^a auxiliary training examples, to be consistent with how its k is trained.

3.2 Results

A total of 19 college-aged subjects participated in the study. Presentation of the VRST version of the Stroop was counterbalanced. While experiencing the VRST, participant psychophysiological responses were recorded using the Biopac MP150 system. The University of Southern California's Institutional Review Board approved the study.

We classify three arousal levels in this paper, which are the same as those in [10]. One of the 19 subjects did not respond at all in one of the three scenarios, and was excluded as an outlier. Only the remaining 18 subjects were studied. The features were the same as those 29 features in [10].

In the experiments $I = 3$ and $K = 5$. Because each subject had 150 responses (50 for each arousal level), $N^a = 150 \times 17 = 2550$. We performed experiments for each subject separately, and for each subject we increased N^p from 3 (one response at each arousal level) to 15 (5 responses at each arousal level) in internal cross-validation, and used the resulting optimal k for testing on the remaining $150 - N^p$ primary examples. We repeated the experiment 100 times (each time the N^p primary training examples were chosen randomly) for each subject and report the average performance of the three kNN algorithms, as shown in Fig. 2. Observe that:

Algorithm 1. The algorithm for training kNN_2

```
Input: $N^p$ primary training examples; $K$, the maximum number of NNs in
       $I$-fold internal cross-validation
Output: $k_o$, the optimal number of NNs in the kNN classifier
Initialize: The maximum number of correct classifications in internal
cross-validation $\overline{n}_v = 0$; The maximum classification margin in internal
cross-validation $\overline{m}_v = -10^{10}$;
Partition the $N^p$ primary training examples into $I$ folds;
foreach $k$ in $[1, K]$ do
    $n_v = 0$, $m_v = 0$;
    foreach $i$ in $[1, I]$ do
        Compute the kNN classification results using the $i$th fold of the primary
        training examples in validation, the rest $I - 1$ folds plus the $N^a$
        auxiliary data in training, and $k$ as the number of NNs;
        $n_v = n_v + n_v^i$ and $m_v = m_v + m_v^i$, where $n_v^i$ is the number of correct
        classifications in the $i$th validation, and $m_v^i$ is the total classification
        margin in the $i$th validation;
    end
    if $n_v > \overline{n}_v$ then
        $\overline{n}_v = n_v$, $\overline{m}_v = m_v$, $k_o = k$;
    else if $n_v = \overline{n}_v$ and $m_v > \overline{m}_v$ then
        $\overline{m}_v = m_v$, $k_o = k$;
    end
end
```

1. For all three kNN classifiers, generally as N^p increases, the testing performance also increases, which is intuitive.
2. Sometimes kNN_2 significantly outperforms kNN_1 and kNN_3, but its overall performance is the worst. Recall that in kNN_2 for each iteration of the internal I-fold cross-validation the $(I-1)N^p/I$ primary training examples were combined with the $N^a = 17N^p$ auxiliary examples in training. As the number of auxiliary examples is significantly larger than that of the primary examples, most of the neighbors are from the auxiliary data. So, the performance of kNN_2 is highly dependent on the amount of individual differences. We conjecture that for subjects that kNN_2 performs very well (e.g., Subject 14), at least one of the rest 17 subjects must have very similar profile. This also suggests that significant performance improvement may be obtained if we can identify the most similar subjects and only use them in kNN_2. This will be one of our future research directions.
3. kNN_3 always outperforms kNN_1, especially when N^p is very small. This is because when N^p is small, several different ks may give the same validation accuracy (i.e., a very small number of training examples do not have enough discriminative power), and hence the information in the N^a auxiliary training examples is very useful in helping determine the optimal k. To show that the difference between kNN_1 and kNN_3 is statistically significant, we performed paired t-tests, as shown in Table 1. Observe that except for Subject 6, all differences are significant.

Algorithm 2. The algorithm for training kNN$_3$

```
Input: $N^p$ primary training examples; $K$, the maximum number of NNs in
    $I$-fold internal cross-validation
Output: $k_o$, the optimal number of NNs in the kNN classifier
Initialize: The maximum number of correct classifications in internal
cross-validation $\overline{n}_v = 0$; The maximum classification margin in internal
cross-validation $\overline{m}_v = -10^{10}$;
Partition the $N^p$ primary training examples into $I$ folds;
foreach $k$ in $[1, K]$ do
    $n_v^p = 0$, $m_v^p = 0$, $n_v^a = 0$, $m_v^a = 0$;
    foreach $i$ in $[1, I]$ do
        Compute the kNN classification results using the $i$th fold of the primary
        training examples plus the $N^a$ auxiliary data in validation, the rest
        $I - 1$ folds in training, and $k$ as the number of NNs;
        $n_v^p = n_v^p + n_v^{i,p}$, and $m_v^p = m_v^p + m_v^{i,p}$, where $n_v^{i,p}$ is the number of
        correct classifications in the $i$th fold of the primary data, and $m_v^{i,p}$ is the
        total classification margin in the $i$th fold of the primary data;
        $n_v^a = n_v^a + n_v^{i,a}$, and $m_v^a = m_v^a + m_v^{i,a}$, where $n_v^{i,a}$ is the number of
        correct classifications in the auxiliary data, and $m_v^{i,p}$ is the total
        classification margin in the auxiliary data;
    end
    if $n_v^p > \overline{n}_v^p$ then
        $\overline{n}_v^p = n_v^p$, $\overline{m}_v^p = m_v^p$, $\overline{n}_v^a = n_v^a$, $\overline{m}_v^a = m_v^a$, $k_o = k$;
    else if $n_v^p = \overline{n}_v^p$ and $n_v^a > \overline{n}_v^a$ then
        $\overline{m}_v^p = m_v^p$, $\overline{n}_v^a = n_v^a$, $\overline{m}_v^a = m_v^a$, $k_o = k$;
    else if $n_v^p = \overline{n}_v^p$ and $n_v^a = \overline{n}_v^a$ and $m_v^p > \overline{m}_v^p$ then
        $\overline{n}_v^a = n_v^a$, $\overline{m}_v^a = m_v^a$, $k_o = k$;
    else if $n_v^p = \overline{n}_v^p$ and $n_v^a = \overline{n}_v^a$ and $m_v^p = \overline{m}_v^p$ and $m_v^a > \overline{m}_v^a$ then
        $\overline{m}_v^a = m_v^a$, $k_o = k$;
    end
end
```

4. When N^p increases, the performances of kNN$_3$ and kNN$_1$ converge. This is because for large N^p, the validation accuracy in the internal cross-validation on the primary training examples only is enough to distinguish among different ks, and hence the effect of the auxiliary data is reduced, i.e., the *"else if"* loops in Algorithm 2 are rarely used.

In summary, the accuracy of a kNN classifier can be improved with the help of auxiliary training examples from other subjects, especially when the number of primary training examples is very small. As a result, fewer user-specific training examples may be needed to tailor an affective computing system for individual use.

Finally, note that the purpose of the experiments is not to show how good a kNN classifier can be in arousal classification; instead, we aim to demonstrate how transfer learning can improve the performance of an existing classifier. Also, as we have shown in this section, not necessarily all transfer learning algorithms

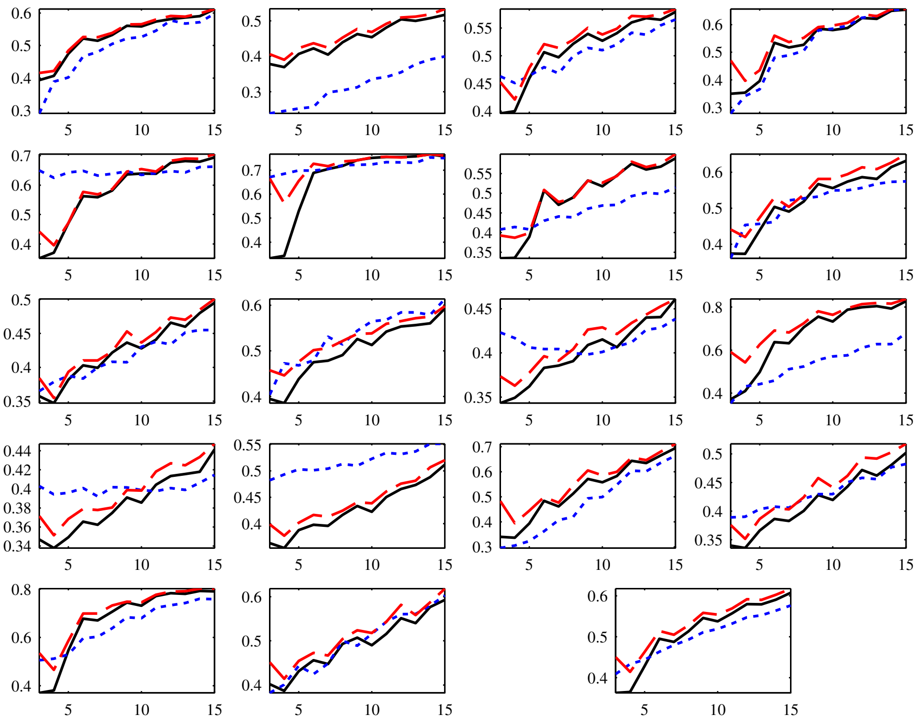

Fig. 2. Performance comparison of the three kNN classifiers on the 18 subjects. Each of the first 18 sub-figures represents a different subject. The last sub-figure shows the average performance of the three kNN classifiers over the 18 subjects. The horizontal axis shows N^p, and the vertical axis shows the testing accuracy on the $150 - N^p$ examples from the same subject. —: kNN$_1$; - - -: kNN$_2$; — — —: kNN$_3$.

Table 1. Paired t-test results on kNN$_1$ and kNN$_3$. $\alpha = 0.05$ and $df = 12$.

Subj.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
t	5.5	9.9	4.0	3.1	2.6	2.0	2.3	6.4	5.8	5.3	6.1	3.1	8.9	6.8	3.5	14.0	2.6	8.3
p	<.01	<.01	<.01	<.01	.02	**.07**	.04	<.01	<.01	<.01	<.01	<.01	<.01	<.01	<.01	<.01	.02	<.01

can always improve the classification performance. For each particular application, a small dataset may be needed to identify the best transfer learning approach.

4 Conclusions and Future Research Directions

In this paper we have introduced the concept of transfer learning and demonstrated how it can be used to handle individual differences in affective computing through an arousal level classification example. Experiments on 18 subjects showed that a kNN classifier which made use of the auxiliary data in cross-validation can always achieve better performance than the one using the primary data only. Especially, the performance improvement was significant when the number of primary training examples is very small. The results suggest that transfer learning is a very promising technique for handling individual differences in affective computing, and it can help tailor an affective computing system for individual users with very few user-specific training examples. To the best knowledge of the authors, this is the first time that transfer learning has been applied to affective computing problems.

This paper represents our first attempt on using transfer learning in affective computing. We plan to further pursue this interesting research in the following directions:

1. To integrate transfer learning with feature selection. As it has been shown in [10], many of the 29 features are not useful. However, the useful features are subject-dependent. As the features directly affect the NNs, it is necessary to integrate transfer learning with feature selection for further performance improvement.
2. We will consider removing outliers from the auxiliary data to make them more consistent, because we believe that the data for each subject in the auxiliary data should be consistent by themselves: If a subject's arousal levels cannot be classified reliably based on his/her own previous responses, how can that subject's profile be used to help classify another subject's arousal level? In other words, if a subject cannot reliably classify his/her own arousal level, then unlikely he/she can give good suggestions on another subject's arousal level. One possible approach is that for each subject in the auxiliary data, we remove a minimum number of outliers so that a 100% accurate kNN classifier can be obtained for him/her. The remaining data from all subjects can then be combined to form the auxiliary dataset.
3. It may be beneficial to not mix the responses from all subjects in the auxiliary data, e.g., we can treat each of the remaining 17 subjects' responses as a separate auxiliary dataset, perform transfer learning on each auxiliary dataset, and then fuse the outcome.
4. We will carefully design the primary training examples to further minimize the number of training examples needed to tailor an affective computing system for individual use. Active class selection [3] is one such approach. In [11] we have shown that it can achieve better performance than learning from random training examples by using feedback during learning to guide the generation of new training data.
5. We will apply transfer learning to affective computing problems beyond classification, e.g., regression [12], and preference learning [14], where the users' affects are expressed as preferences instead of classes or numbers.

References

1. Allanson, A., Fairclough, J.: A research agenda for physiological computing. Interacting With Computers 16, 858–878 (2004)

2. Ito, T., Cacioppo, J.: Variations on a human universal: Individual differences in positivity offset and negativity bias. Cognition & Emotion 19, 1–26 (2005)
3. Lomasky, R., Brodley, C.E., Aernecke, M., Walt, D., Friedl, M.: Active class selection. In: Proc. 18th European Conference on Machine Learning, Warsaw, Poland, pp. 640–647 (September 2007)
4. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Knowledge and Data Engineering 22(10), 1345–1359 (2010)
5. Pantic, M., Rothkrantz, L.: Automatic analysis of facial expressions: The state of the art. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(12), 1424–1445 (2000)
6. Parsons, T., Courtney, C., Arizmendi, B., Dawson, M.: Virtual reality Stroop task for neurocognitive assessment. Studies in Health Technology and Informatics 143, 433–439 (2011)
7. Picard, R.: Affective Computing. The MIT Press, Cambridge (1997)
8. Stroop, J.: Studies of interference in serial verbal reactions. Journal of Experimental Psychology 18, 643–661 (1935)
9. Varma, S., Simon, R.: Bias in error estimation when using cross-validation for model selection. BMC Bioinformatics 7(91) (2006)
10. Wu, D., Courtney, C.G., Lance, B.J., Narayanan, S.S., Dawson, M.E., Oie, K.S., Parsons, T.D.: Optimal arousal identification and classification for affective computing: Virtual Reality Stroop Task. IEEE Trans. on Affective Computing 1(2), 109–118 (2010)
11. Wu, D., Parsons, T.D.: Active class selection for arousal classification. In: D´Mello, S., et al. (eds.) Affective Computing and Intelligent Interaction, Part II, vol. 6975, pp. 132–141. Springer, Heidelberg (2011)
12. Wu, D., Parsons, T.D., Mower, E., Narayanan, S.S.: Speech emotion estimation in 3D space. In: Proc. IEEE Int'l. Conf. on Multimedia & Expo. (ICME), Singapore, pp. 737–742 (July 2010)
13. Wu, P., Dietterich, T.G.: Improving SVM accuracy by training on auxiliary data sources. In: Proc. Int'l. Conf. on Machine Learning, Banff, Alberta, Canada, pp. 871–878 (July 2004)
14. Yannakakis, G.N., Maragoudakis, M., Hallam, J.: Preference learning for cognitive modeling: A case study on entertainment preferences. IEEE Systems, Man and Cybernetics – A 39(6), 1165–1175 (2009)
15. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(1), 39–58 (2009)