Neurocomputing 139 (2014) 97–106

Contents lists available at ScienceDirect

Neurocomputing

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

Multi-task and multi-view learning of user state

Melih Kandemir [a,*], Akos Vetek [b], Mehmet Gönen [c], Arto Klami [d], Samuel Kaski [d,e,**]

[a] Heidelberg University, HCI, Speyerer Str. 6, D-69115 Heidelberg, Germany
[b] Nokia Research Center Otaniemi, Espoo, Finland
[c] Sage Bionetworks, Seattle, WA, USA
[d] Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland
[e] Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, P.O. Box 15400, FI-00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 30 September 2013
Received in revised form
28 January 2014
Accepted 21 February 2014
Communicated by Christos Dimitrakakis
Available online 18 April 2014

Keywords:
Affect recognition
Machine learning
Multi-task learning
Multi-view learning

ABSTRACT

Several computational approaches have been proposed for inferring the affective state of the user, motivated for example by the goal of building improved interfaces that can adapt to the user's needs and internal state. While fairly good results have been obtained for inferring the user state under highly controlled conditions, a considerable amount of work remains to be done for learning high-quality estimates of subjective evaluations of the state in more natural conditions. In this work, we discuss how two recent machine learning concepts, multi-view learning and multi-task learning, can be adapted for user state recognition, and demonstrate them on two data collections of varying quality. Multi-view learning enables combining multiple measurement sensors in a justified way while automatically learning the importance of each sensor. Multi-task learning, in turn, tells how multiple learning tasks can be learned together to improve the accuracy. We demonstrate the use of two types of multi-task learning: learning both multiple state indicators and models for multiple users together. We also illustrate how the benefits of multi-task learning and multi-view learning can be effectively combined in a unified model by introducing a novel algorithm.

1. Introduction

Affective computing seeks to develop more efficient and pleasant user interfaces by taking into account the affective state of the user. For example, the information flow can be tailored by managing interruptions from e-mail alerts and phone calls when the user is in deep thought [7], and the affective state can be used to determine the most suitable time to intervene during a pedagogical game [8]. Apart from adapting the interface, information on the affective state can be used to gain a deeper understanding of how users and computers interact. A prerequisite of affective computing is the ability to recognize users' states of interest, either by observing the users' actions [26] or by analyzing physiological signals measured from the user [25,15,6]. In this work, we study the latter approach and discuss machine learning solutions for inferring the affective state of the user from physiological signals in unobtrusive and loosely controlled user setups.

During recent years, several databases of physiological measurements in affective computing tasks have been released [13,22,31], in an attempt to provide high-quality data for learning and benchmarking state inference models. The state of the art in the field is that the user's state can be inferred relatively accurately in highly controlled experiment setups where the stimuli evoke strong emotional responses [20,24,32]. For less controlled setups, where the ground truth labels come from user evaluations, some recent works have obtained positive results [22,2,9,11,33] but in many cases the prediction accuracies are not yet sufficiently high for practical use in adaptive interfaces.

We introduce two elements from machine learning literature to help improve the user state estimation: multi-view learning and multi-task learning. Both ideas can be incorporated into many of the current state estimation methods (for a recent review see [34]), to obtain better estimates of the user's affective states. We motivate these concepts for affective computing tasks and demonstrate their usefulness in learning user states especially when used in combination.

Multi-view learning studies how data sets having co-occurring observations can be combined. Most affective computing studies monitor the user with several sensors or sensor channels, which

* Corresponding author.
** Corresponding author at: Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, P.O. Box 15400, FI-00076 Aalto, Finland.
E-mail addresses: melih.kandemir@iwr.uni-heidelberg.de (M. Kandemir), akos.vetek@nokia.com (A. Vetek), mehmet.gonen@sagebase.org (M. Gönen), arto.klami@cs.helsinki.fi (A. Klami), samuel.kaski@aalto.fi (S. Kaski).

http://dx.doi.org/10.1016/j.neucom.2014.02.057
0925-2312/

can be considered as such co-occurring sets. Multi-view learning refers to various strategies for learning a joint model over all sensor data, to learn how the sources should be combined for building optimal models. In this paper, we work with a specific multi-view learning technique called *multiple-kernel learning* (MKL) [16], which allows using multiple sensors in any kernel-based learning algorithm while automatically revealing which sensors are useful for solving the task. Even though considerable effort has been put into finding out which physiological sensors are related to which affective dimensions, this is still useful for all practical applications with specific sensor hardware. Automatically learning the sensor importance is especially useful when developing practical systems for out-of-laboratory conditions.

The other concept, *multi-task learning* (MTL), studies learning of several prediction tasks together [5]. Within the scope of state inference, MTL takes advantage of the data of other users by learning from the cross-user similarities, without assuming that the users are identical. This helps particularly when the amount of labeled training data is limited. Alternatively, learning each output label, such as arousal and valence, could be considered as a task. Learning predictive models for all of the labels together is then useful assuming that all labels are one-dimensional summaries of a more complex unknown state of the user. The approach will be particularly useful if the dimensions are not independent.

We present a novel kernel-based model that combines both multi-view and multi-task aspects. It can be applied to both of the aforementioned MTL scenarios, and it uses the MKL formulation to make the approach multi-view. We then apply the model to two different data collections to study the accuracy of state recognition. The first collection, taken from Koelstra et al. [22], is an example of a laboratory-quality data. We have collected the other data set ourselves under less constrained conditions.

The main goal of the paper is to illustrate the benefits of the two aforementioned general purpose machine learning techniques in affective computing applications. To this end, we show how combining MTL and MKL within a unified model improves the prediction performance, and also highlight how MKL automatically learns the importance of individual sensors even when solving multiple inference tasks simultaneously. We demonstrate the models with generic features instead of carefully selecting the sensors and features to match the particular affective inference tasks. This highlights the main advantage of the proposed strategy: It allows working with a wide set of sensors and tasks, without requiring much manual labor in incorporating domain-specific knowledge into the solutions.

2. Inferring the user state

Given the input data from P sensors, the user state inference task consists of inferring for each data point a set of labels that jointly characterize the state of the user. We do not assume any particular emotional model, such as [28]. Instead, we simply require the states to be represented by a collection of numerical labels. The labels do not have to be independent; in fact, as will become more apparent later, the multi-task formulation we introduce is specifically tailored to capture correlations between the labels. In the experimental section we use Likert-scale evaluations of valence, arousal, liking, and mental workload as the labels, but the underlying machine learning techniques would apply to any other numerical characterizations of the state dimensions. Even though we resort to binarization of multi-category state labels to overcome data scarcity, extension of the presented techniques to multi-class setups is straightforward.

We study *user-specific* and *user-independent* setups for each learning model. The former is trained on data recorded from a single user and assumes this person to be the eventual user of the system, whereas the latter learns the models from M earlier users and assumes the eventual user to be a new one. User-specific models need to be separately customized to target users. On the other hand, user-independent models do not require any training data from the eventual user, and hence can be pre-trained on large data collections.

For both scenarios, each data sample $\boldsymbol{x}_i$ is represented as a collection of vectors $\boldsymbol{x}_i = \{\boldsymbol{x}_i^{(m)}\}_{m=1}^{P}$, one for each of the P views (here sensors), where $\boldsymbol{x}_i^{(m)} \in \mathbb{R}^{D_m}$ and D_m is the dimensionality of the feature representation for the sensor m. The output, characterization of the user's state, is given as (here binary) vector of labels $\boldsymbol{y}_i = [y_i(1), \dots, y_i(T)]$, where $y_i(j) \in \{\pm 1\}$ and T is the number of labels.

All learning setups considered in this paper are multi-view, due to the input data coming from P different sensors. MTL, in turn, can be applied in two different ways. When considering the different users as different but related tasks we can learn user-specific models for all users at the same time, separately for each label. In this case, each task takes as input the measurements taken from a different user $\boldsymbol{x}$, and predicts the corresponding label. Even though the models are learned together in the spirit of multi-task learning, the output will be a separate model for each user. Alternatively, we can learn a single user-independent model for all T labels at once, resulting in a MTL setup where the inputs $\boldsymbol{x}$ are the same for all tasks but the output labels are different.

In this paper, we formulate a novel kernel-based algorithm that performs multi-task and multi-view learning in a coupled and efficient manner. In Sections 2.1–2.3 we review the basics of kernel based learning and explain the earlier kernel-based multi-task and multi-view algorithms. Finally, in Section 2.4 we introduce our new model that combines both approaches.

2.1. Support vector machines (SVMs)

We take the standard support vector machine (SVM) [30] as a single-task and single-view building block on which we develop our novel multi-task multi-view learning algorithm. We denote by $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ a sample of N independent training instances, where $\boldsymbol{x}_i$ is a D-dimensional input vector with the target output y_i, and by $\Phi : \mathbb{R}^D \to \mathbb{R}^S$ a function that maps the input patterns to a preferably higher dimensional space. The support vector machine learns a linear discriminant that predicts the target output of an unseen test instance $\boldsymbol{x}$ as

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \Phi(\boldsymbol{x}) + b,$$

where $\boldsymbol{w}$ contains the hyperplane parameters and b is the bias parameter. Using the representer theorem, the discriminant in the dual form becomes

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i \underbrace{\Phi(\boldsymbol{x}_i)^\top \Phi(\boldsymbol{x})}_{k(\boldsymbol{x}_i, \boldsymbol{x})} + b$$

where N is the training set size, $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the kernel function that defines a similarity metric for pairs of data instances, and $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers defined in the domain

$$\mathcal{A} = \left\{ \boldsymbol{\alpha} : \sum_{i=1}^{N_r} \alpha_i = 0,\ \alpha_i \in \mathbb{R},\ \forall i \right\}. \tag{1}$$

For binary classification $y_i \in \{-1, +1\}$ and squared loss, the corresponding objective function is

$$J(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \left(k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{\delta_i^j}{2C} \right),$$