Generalized Hidden-mapping Transductive Transfer Learning for Recognition of Epileptic Electroencephalogram Signals

Lixiao Xie[1], Zhaohong Deng [1, 2,*], *IEEE Senior Member*, Kup-Sze Choi[2], *IEEE Member*, Shitong Wang[1]

[1] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, P. R. China

[2] Centre for Smart Health, School of Nursing, Hong Kong Polytechnic University, Hong Kong

[*] Corresponding author

Abstract: Electroencephalogram (EEG) signal identification based on intelligent models is an important means in epilepsy detection. In the recognition of epileptic EEG signal, traditional intelligent recognition methods usually assume that the training and the testing dataset have the same distributions, and that adequate training data are available. However, these two conditions are not always met in practice, which degenerates the recognition abilities of the intelligent epilepsy detection model. In order to overcome this challenge, an effective strategy is the introduction of transfer learning for the construction of the intelligent models, where transfer learning is effectively used to learn from the knowledge of related scenes to enhance the performance of the model trained in the current scene. Although transfer learning has been used in EEG signal identification, many existing techniques are only designed for specific intelligent models and cannot be extensively applied to other models. To tackle this limitation, a more generalizable transductive transfer learning approach, namely, generalized hidden-mapping transductive learning method, is proposed to realize transfer learning for several classic intelligent models, including feedforward neural networks, fuzzy systems and kernelized linear models. A large number experiments on epileptic EEG recognition are carried out to demonstrate the effectiveness of the proposed method.

I Introduction

Epilepsy is a common brain functional disease caused by the abnormal brain neurons [1]. Electroencephalogram (EEG) is an important means to detect epilepsy. In recent years, intelligent modeling techniques have been studied widely in the detection and recognition of epileptic EEG signals because of their strong learning ability [2].

There have been a variety of intelligent identification methods that are applied to detect epileptic EEG signals, such as Naive Bayes method (NB), K nearest neighbor (KNN), linear discriminant analysis (LDA), decision tree algorithm (DT) and support vector machine (SVM) [3]. For the traditional intelligent epileptic EEG identification methods, it is usually to first train a prediction model with abundant training data, and then use the trained model to predict the classes that the testing data belong to. All these traditional intelligent methods assume that the source domain (the training dataset) and the target domain (the testing dataset) have the same distribution. However, the epileptic EEG data collected for training and testing do not necessarily satisfy this assumption and thus the performance of traditional identification methods is degenerated seriously

when there is a drifting in the data distribution between the source and the target. Transfer learning is a promising mechanism to deal with this challenge. It can transfer the relevant knowledge from the related tasks to facilitate the learning of the current one [4], without the need to assume that the training and testing data have the same distribution. [5].

Transfer learning is an effective adaptive learning strategy. The existing methods can be divided into three main categories: (1) inductive transfer learning, (2) transductive transfer learning and (3) unsupervised transfer learning [6]. In inductive transfer learning, both the source domain and target domain must contain labeled data for model training, while labeled data are only needed in the source domain for transductive learning. For unsupervised learning, data in both domains are unlabeled. For epileptic EEG recognition, inductive and transductive learning can be used to train a recognition model. Since the applicability of transductive transfer learning is superior to that of inductive transfer learning [7], we focus our work on the former in the paper for epileptic EEG recognition.

Transfer learning based methods have been proposed for epileptic EEG recognition, e.g. transfer learning based SVM [8] and transfer learning based fuzzy systems [9]. Although these existing methods have shown promising performance, they have a common shortcoming that they can only be used to a specific intelligent model. When a new intelligent model is developed and is to be trained using the transfer learning mechanism, the existing methods will not be applicable anymore. In this study, we investigate into transfer learning based methods that are more generally applicable for the training of different models.

The main contributions of this paper are as follows: (1) generalized hidden-mapping model is introduced to unify the representation of several classical intelligent models, including feedforward neural networks, fuzzy systems and kernel regression models; and (2) the generalized hidden-mapping transductive transfer learning method is proposed to realize the transfer learning of different classical intelligent models. Experimental studies on epileptic EEG recognition are conducted extensively to confirm the effectiveness of the proposed method.

The rest of the paper is organized as follows: Section II describes the generalized hidden-mapping model and the relationships between this model and the feedforward neural networks, fuzzy system and kernel methods respectively. Section III presents the transductive transfer learning based generalized hidden-mapping model construction method. Section IV presents the experiments conducted to evaluate the performance of the proposed method. Conclusions and future work are given in the final section.

II Generalized Hidden-mapping Model

A Hidden-mapping Linear Model

Most existing transductive transfer learning algorithms are only applicable to a specific intelligent model. For example, the large margin transductive transfer learning method (LMPROJ), developed based on SVM, cannot be used for the training of fuzzy rules based model when fuzzy systems are preferred for better interpretability. Therefore, it is significant to propose a generalized model based on which the general learning algorithm can be developed for different classical intelligent models. For this purpose, we introduced the generalized hidden-mapping model (GHMM) [10]. A variety of classical intelligent models such as fuzzy systems, neural networks and kernel methods, can be considered as the special cases of GHMM. Based on GHMM, the general

transductive transfer learning algorithm was proposed for several classical intelligent models and applied to epileptic EEG recognition. The mathematical expression of GHMM is as follows:

$$y = f(\mathbf{x}) = \mathbf{p}_{\mathrm{g}}^{\mathrm{T}} \mathbf{x}_{\mathrm{g}}, \tag{1}$$

where $\mathbf{p}_{\mathrm{g}}$ and $\mathbf{x}_{\mathrm{g}}$ denote the parameters of the linear model and the input vector in a hidden-mapping space respectively. This model is very general and encompasses many classical intelligent models. The relationships between this model and the classical intelligent models are briefly described below.

B Relationship with Feedforward Neural Networks

Feedforward neural networks (FNN) [11] are a class of neural network models that have been applied extensively in many areas, e.g. speech recognition [12], signal processing [13] and robotics control [14]. In general, an FNN contains an input layer, one or more hidden layers and an output layer.

According to the number of hidden layers, FNNs can be divided into single hidden layer FNNs and multi-hidden layer FNNs. Since the multiple hidden layers can be regarded as a hidden layer that is more complicated, multi-hidden layer FNN can be expressed as a special single hidden layer FNN [15]. For clarity and convenience, we only discuss single hidden layer FNN here. For a single hidden layer FNN, its output can be expressed as follows:

$$y = f(\mathbf{x}) = \sum_{i=1}^{N} g_i(\mathbf{x}, \theta_i) w_i, \tag{2}$$

where $g_i(\mathbf{x}, \theta_i)$ is the output of the i th node in the hidden layer and θ_i denotes the parameters of this node. According to the general approximation theorem, FNNs with a finite number of neurons can be trained to approximate an arbitrary continuous function [16]. In particular, according to the learning theory in [17], even if the hidden parameters are randomly generated for the activation function in hidden layer and the function is continuous, bounded, and non-constant, the FNNs can still approximate it with an arbitrary continuous function. Therefore, once the hidden layer is fixed, the outputs of the hidden layer can be expressed as the following vector

$$\mathbf{x}_{\mathbf{g}} = [g_1(\boldsymbol{x}, \theta_1), g_1(\boldsymbol{x}, \theta_2), \cdots, g_1(\boldsymbol{x}, \theta_{N_M})]^{\mathbf{T}} \in R^N. \tag{3}$$

$\mathbf{x}_{\mathbf{g}} \in R^N$ can be viewed as a vector in the new space which are mapped by the hidden layer of FNNs from the vector $\mathbf{x} \in R^d$ in the original feature space of samples. Thus, the output of a FNN can be written as

$$y = f(\mathbf{x}) = \mathbf{p}_{\mathrm{g}}^{\mathrm{T}} \mathbf{x}_{\mathrm{g}}, \tag{4}$$

where $\mathbf{p}_{\mathrm{g}} = [w_1, ..., w_{N_M}]^{\mathrm{T}} \in R^{N_M}$. Hence, it is obvious that FNNs can be considered as a special case of GHMM.

C Relationship with TSK Fuzzy system

Fuzzy systems are a kind of intelligent models based on fuzzy logic rules. They feature good

interpretability that are advantageous over the black-box of neural networks [18]. Besides, fuzzy systems are more robust as they exhibit strong abilities in modeling uncertainty [19]. Data-driven fuzzy systems are becoming popular modelling methods since the parameters of the systems can be easily optimized using the available training data, whereas TSK fuzzy systems (TSK FS) are also t widely used for fuzzy system modeling because of the strong learning ability [20]. The relationship between TSK FS and GHMM is discussed below. TSK FS consists of a group of "If-then" inference rules as follows [21],

$$\text{If} \quad x_1^k \text{ is } A_1^k \wedge x_2^k \text{ is } A_2^k \wedge \cdots \wedge x_{m_k}^k \text{ is } A_{m_k}^k \,, \tag{5}$$

$$\text{Then} \quad f^k(\mathbf{x}) = p_0^k + p_1^k x_1 + \cdots + p_d^k x_d \,, \quad k = 1, 2, \cdots, K \,.$$

In (5), A_i^k is the fuzzy set; $f^k(\mathbf{x})$ $(1 \le k \le K)$ represents the output of the *k*th rule of the TSK FS, where K is the number of rules in the rule base; and p_i^k is the parameters of the output function $f^k(\mathbf{x})$ [22]. For TSK FS, when multiplication is implemented by the conjunction and implication operators, addition by the combination operator [23], and the center of gravity by the defuzziness operation, the output of TSK FS can be expressed as

$$f(\mathbf{x}) = \sum_{k=1}^{K} f^k(\mathbf{x}).\mu^k(\mathbf{x}) \Big/ \sum_{k'=1}^{K} \mu^{k'}(\mathbf{x}) = \sum_{k=1}^{K} \tilde{\mu}^k(\mathbf{x}) f^k(\mathbf{x}) \,. \tag{6}$$

In (6), $\mu^k(\mathbf{x})$ and $\tilde{\mu}^k(\mathbf{x})$ are the fuzzy membership and the normalized fuzzy membership, which can be computed below

$$\mu^k(\mathbf{x}) = \prod_{i=1}^{d} \mu_{A_i^k}(x_i) \,, \tag{7}$$

$$\tilde{\mu}^k(\mathbf{x}) = \mu^k(x) \Big/ \sum_{k'=1}^{K} \mu^{k'}(x). \tag{8}$$

In (7), $\mu_{A_i^k}(\mathbf{x}_i)$ is the membership degree of the *i*th dimension x_i in input vector $\mathbf{x}$ to the fuzzy set A_i^k [24].

There are several ways to estimate the antecedent parameters of TSK FS. One of the classical method is to divide the input space by clustering techniques, e.g. classical fuzzy c-means (FCM) [25] algorithm, and then estimate the antecedent parameters according to the space partition results. For example, if the following Gaussian membership function is adopted as the fuzzy membership function in the fuzzy rules,

$$\mu_{A_i^k}(x_i) = \exp\left(\frac{-(x_i - c_i^k)^2}{2\delta_i^k}\right) \,, \tag{9}$$

then the parameter δ_i^k and c_i^k in the membership function can be estimated by the clustering results of FCM on the input data of the training dataset as follows,

$$c_i^k = \sum_{j=1}^{N} u_{jk} x_{ji} \Big/ \sum_{j=1}^{N} u_{jk} \,, \tag{10}$$

$$\delta_i^k = h \sum_{j=1}^{N} u_{jk} (x_{ji} - c_i^k)^2 \Big/ \sum_{j=1}^{N} u_{jk} \,. \tag{11}$$

Here, u_{jk} is the membership degree of the input vector $\mathbf{x}_j$ in *j*th sample to the *k*th cluster; h is adjustable parameters, which can be determined manually or by learning strategies such as cross-validation strategy [25].

Once the antecedent parameters of the TSK FS are evaluated, the output of a TSK FS can be expressed as follows,

$$y = f(\mathbf{x}) = \mathbf{p}_{\mathrm{g}}^{\mathrm{T}}\mathbf{x}_{\mathrm{g}}, \tag{12}$$

where

$$\mathbf{x}_{\mathbf{g}} = [(\tilde{\mathbf{x}}^1)^T, (\tilde{\mathbf{x}}^2)^T, \cdots, (\tilde{\mathbf{x}}^K)^T]^T \in R^{K(d+1)}, \tag{13a}$$

$$\tilde{\mathbf{x}}^K = \tilde{\mu}^k(\mathbf{x})\mathbf{x}_e, \tag{13b}$$

$$\mathbf{x}_e = (1, \mathbf{x}^T)^T, \tag{13c}$$

$$\mathbf{p}^k = [(\mathbf{p}^1)^T, (\mathbf{p}^2)^T, \cdots, (\mathbf{p}^K)^T]^T, \tag{13d}$$

$$\mathbf{p}_{\mathbf{g}} = (p_0^k, p_1^k, \cdots, p_d^k)^T. \tag{13e}$$

In (13), $\mathbf{x}_{\mathbf{g}}$ represents the data in the mapping feature space with the fuzzy rules from the input vector $\mathbf{x}$ in the original space maps; $\mathbf{p}_{\mathbf{g}}$ is the combined vector of the consequent parameters of all the fuzzy rules in the rule base of the trained TSK FS. It can be seen from (12) that TSK FS can be regarded as a special case of GHMM.

D Relationship with Kernelized Linear Regression

Kernel technique is very effective approach for solving nonlinear problems. In kernel based methods, a sample $\mathbf{x}$ is usually mapped into $\mathbf{x}_{\mathbf{g}}$ in an unknown feature space as follows [26],

$$\mathbf{x}_{\mathbf{g}} = \varphi(\mathbf{x}), \tag{14}$$

where $\varphi(\cdot)$ is the mapping function that satisfies the Mercer kernel conditions [27]. A representative kernel based methods is the kernelized linear regression model,

$$y = \mathbf{p}_{\mathbf{g}}^{\mathbf{T}}\varphi(\mathbf{x}), \tag{15}$$

which can be effectively used for classification and regression. Comparing (15) with (1), we can see that kernelized linear regression model is just a special case of GHMM.

III Transductive Transfer learning for GHMM

The relationships between GHMM and FNN, TSK FS and kernalized linear regression as discussed above show that the leaning algorithm of GHMM is a general algorithm that is applicable to many existing intelligent models. Therefore, a transductive transfer learning algorithm based on GHMM, called generalized hidden-mapping transductive transfer learning, is proposed in the paper

for the training of different intelligent models in order to overcome the issue due to distribution drifting between the source and the target domain.

A The Maximum Mean Distance and Projective Maximum Mean Distance

Maximum mean distance (MMD) and projected maximum mean distance (PMMD) have shown to be effective metrics for transductive transfer learning [28]. For the datasets in the source domain: $D_s = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and datasets in the target domain: $D_t = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$, the distance between the distributions of source domain and the target domain, i.e., p_{Ds} and p_{Dt}, can be approximated with MMD as follows,

$$d(p_{Ds}, p_{Dt})^2 = \mathrm{MMD}^2 = \left\| \frac{1}{N}\sum_{j=1}^{N}\phi(\boldsymbol{x}_j) - \frac{1}{M}\sum_{j=1}^{M}\phi(\boldsymbol{z}_j) \right\|^2, \tag{16}$$

where $\phi(\cdot)$ is a mapping function. Furthermore, given the projection vector $\mathbf{p_g}$, the projective maximum mean distance (PMMD) under $\mathbf{p_g}$ can be expressed as:

$$\begin{aligned} d(p_{Ds}, p_{Dt}\,|\mathbf{p_g})^2 = \mathbf{PMMD}^2 &= \left\| \frac{1}{N}\sum_{j=1}^{N}\mathbf{p_g^T}\phi(\mathbf{x}_j) - \frac{1}{M}\sum_{j=1}^{M}\mathbf{p_g^T}\phi(\mathbf{z}_j) \right\|^2 \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{p}_\mathbf{g}^T\phi(\mathbf{x}_i)\phi(\mathbf{x}_j)^T\mathbf{p_g} + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\mathbf{p}_\mathbf{g}^T\phi(\mathbf{z}_i)\phi(\mathbf{z}_j)^T\mathbf{p_g} \\ &\quad - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbf{p}_\mathbf{g}^T\phi(\mathbf{x}_i)\phi(\mathbf{z}_j)^T\mathbf{p_g}. \end{aligned} \tag{17}$$

In transfer learning algorithms, the above measure is often used to estimate the difference between two distributions, and have been effectively used in some transfer learning methods, e.g. LMPROJ [29].

B Generalized Hidden-mapping Transductive Transfer Learning

Based on the GHMM, the input data of the training set: $D_s = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and the input data of the testing set: $D_t = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_M\}$ can be mapped to the new feature space as follows,

$$\mathbf{x}_i \to \phi(\mathbf{x}_i) = \mathbf{x}_{gi}, \tag{18}$$

$$\mathbf{z}_i \to \phi(\mathbf{z}_i) = \mathbf{z}_{gi}. \tag{19}$$

Here, $\mathbf{x}_{gi}$ and $\mathbf{z}_{gi}$, are the data in the new feature space. For example, if GHMM is used for TSK FS and FNN, the new feature space is constructed by the fuzzy rules and hidden layers respectively.

For GHMM, the following objective function is defined by introducing the PMMD,

$$\begin{aligned} &\min \ \frac{1}{\tau N}\sum_{i=1}^{N}((\xi_i^+)^2 + (\xi_i^-)^2) + \frac{1}{2}\mathbf{p}_\mathbf{g}^T\mathbf{p_g} + \frac{2}{\tau}\varepsilon + d(p_{Ds}, p_{Dt}\,|\mathbf{p_g})^2, \\ &\mathbf{s.t} \ \begin{cases} y_i - \mathbf{p_g^T}\mathbf{x}_{gi} < \varepsilon + \xi_i^+ \\ \mathbf{p_g^T}\mathbf{x}_{gi} - y_i < \varepsilon + \xi_i^- \end{cases} \quad \forall i. \end{aligned} \tag{20}$$

Refer to (17), $d(p_{Ds}, p_{Dt}\,|\mathbf{p_g})^2$ can be calculated as follows,

$$
\begin{aligned}
d(p_{Ds}, p_{Dt}\,|\mathbf{p_g})^2 = \mathbf{PMMD}^2 &= \left\| \frac{1}{N}\sum_{j=1}^{N} \mathbf{p_g^T x}_{gj} - \frac{1}{M}\sum_{j=1}^{M} \mathbf{p_g^T z}_{gj} \right\|^2 \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{p_g^T x}_{gi}\mathbf{x}_{gj}^{\mathbf{T}}\mathbf{p_g} + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \mathbf{p_g^T z}_{gi}\mathbf{z}_{gj}^{\mathbf{T}}\mathbf{p_g} \\
&\quad - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \mathbf{p_g^T x}_{gi}\mathbf{z}_{gj}^{\mathbf{T}}\mathbf{p_g}.
\end{aligned} \tag{21}
$$

In (21), $\mathbf{x}_{gi}$ and $\mathbf{z}_{gi}$ can be explicitly expressed when the hidden-mapping is realized by using fuzzy system or neural network, and hence we can obtain the solution for $\mathbf{p_g}$ in an explicit form. However, the explicit expression is not known in some cases, e.g. when the mapping in the kernel methods is unknown, and the $\mathbf{p_g}$ cannot be solved directly. In order to solve for the objective function in (20) uniformity for different cases, $\mathbf{p_g}$ is expressed as follows according to the generated theory of Hilbert space,

$$
\mathbf{p_g} = \sum_{i=1}^{N+M} \beta_i \phi(\mathbf{s})_i = \mathbf{\Phi(s)}\boldsymbol{\beta} \quad , \tag{22}
$$

where $\mathbf{\Phi(s)} = [\phi(\mathbf{s}_1), \phi(\mathbf{s}_2), ...\phi(\mathbf{s}_{N+M})] = [\mathbf{x}_{g1}, ...\mathbf{x}_{gN}, \mathbf{z}_{g1}, ...\mathbf{z}_{gM}]$ and $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_{N+M}]^T$. By substituting (22) into (21), we have

$$
\begin{aligned}
d(p_{Ds}, p_{Dt}\,|\mathbf{p_g})^2 = \mathbf{PMMD}^2 &= \left\| \frac{1}{N}\sum_{j=1}^{N} \mathbf{p_g^T x}_{gj} - \frac{1}{M}\sum_{j=1}^{M} \mathbf{p_g^T z}_{gj} \right\|^2 \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{\beta}^{\mathbf{T}}\mathbf{K_{Train}}[1]^{N\times N}\mathbf{K}_{\mathbf{Train}}^{\mathbf{T}}\boldsymbol{\beta} + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} \boldsymbol{\beta}^{\mathbf{T}}\mathbf{K_{Test}}[1]^{M\times M}\mathbf{K}_{\mathbf{Test}}^{\mathbf{T}}\boldsymbol{\beta} \\
&\quad - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} \boldsymbol{\beta}^{\mathbf{T}}(\mathbf{K_{Train}}[1]^{N\times M}\mathbf{K}_{\mathbf{Test}}^{\mathbf{T}} + \mathbf{K_{Test}}[1]^{M\times N}\mathbf{K}_{\mathbf{Train}}^{\mathbf{T}})\boldsymbol{\beta} \\
&= \boldsymbol{\beta}^{\mathbf{T}}\mathbf{\Omega}\boldsymbol{\beta},
\end{aligned} \tag{23}
$$

Here, $\mathbf{\Omega}$ is a $(N+M)\times(N+M)$ matrix and can be expressed in compact form as follows,

$$
\begin{aligned}
\mathbf{\Omega} &= \frac{1}{N^2}\mathbf{K}_{Train}[1]^{N\times N}\mathbf{K}^{T}_{\ Train} + \frac{1}{M^2}\mathbf{K}_{Test}[1]^{M\times M}\mathbf{K}^{T}_{\ Test} \\
&\quad - \frac{1}{MN}(\mathbf{K}_{Train}[1]^{N\times M}\mathbf{K}^{T}_{\ Test} + \mathbf{K}_{Test}[1]^{M\times N}\mathbf{K}^{T}_{\ Train})
\end{aligned} . \tag{24}
$$

Let $\mathbf{X=[x}_1, \cdots, \mathbf{x}_N\mathbf{]}$, $\mathbf{Z=[z}, \cdots, \mathbf{z}_M\mathbf{]}$ and $\mathbf{S=[X, Z]}$. In (24), $\mathbf{K_{Train(i,j)}=K(S,X)}$ is a $(N+M)\times N$ kernel matrix of the training data and $\mathbf{K_{Test(i,j)}=K(S,Z)}$ is a $(N+M)\times M$ kernel matrix of the testing data. Furthermore, Eq.(20) can be expressed as follows,

$$
\begin{aligned}
&\min \ \frac{1}{\tau N}\sum_{i=1}^{N}((\xi_i^+)^2 + (\xi_i^-)^2) + \frac{1}{2}\mathbf{p_g^T p_g} + \frac{2}{\tau}\varepsilon + \lambda\mathbf{p_g^T}\mathbf{\Omega}\mathbf{p_g} \\
&\mathbf{s.t} \ \begin{cases} y_i - \mathbf{p_g^T x}_{gi} < \varepsilon + \xi_i^+ \\ \mathbf{p_g^T x}_{gi} - y_i < \varepsilon + \xi_i^- \end{cases} \quad \forall i.
\end{aligned} \tag{25}
$$

By substituting (22) into (25), the objective function in (25) can be written as

$$\begin{aligned}&\min\ \frac{1}{\tau N}\sum_{i=1}^{N}((\xi_i^+)^2+(\xi_i^-)^2)+\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Phi}(\mathbf{s})^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{s})\boldsymbol{\beta}+\frac{2}{\tau}\varepsilon+\lambda\boldsymbol{\beta}^T\boldsymbol{\Phi}(\mathbf{s})^{\mathbf{T}}\boldsymbol{\Omega}\boldsymbol{\Phi}(\mathbf{s})\boldsymbol{\beta}\\ &\mathbf{s.t}\ \begin{cases}y_i-\boldsymbol{\beta}^T\boldsymbol{\Phi}(\mathbf{s})^{\mathbf{T}}\mathbf{x}_{gi}<\varepsilon+\xi_i^+\\ \boldsymbol{\beta}^T\boldsymbol{\Phi}(\mathbf{s})^{\mathbf{T}}\mathbf{x}_{gi}-y_i<\varepsilon+\xi_i^-\end{cases}\quad\forall i\end{aligned}. \tag{26}$$

Based on optimization theory, the dual problem of (26) can is a quadratic programming problem, i.e.,

$$\begin{aligned}&\arg\max_{\tilde{\boldsymbol{\alpha}}}-\tilde{\boldsymbol{\alpha}}^{\mathbf{T}}\mathbf{H}\tilde{\boldsymbol{\alpha}}+\tilde{\boldsymbol{\alpha}}^{\mathbf{T}}\mathbf{f}\\ &\mathbf{s.t}\ \tilde{\boldsymbol{\alpha}}^{\mathbf{T}}1=1,\alpha_i\geq 0,\forall i.\end{aligned}, \tag{27}$$

where $\tilde{\boldsymbol{\alpha}}=\left(\lambda_1'^{+},\ldots\lambda_N'^{+},\lambda_1'^{-},\cdots,\lambda_N'^{-}\right)^{\mathrm{T}}$ is the vector of Lagraingian multipliers, $\mathbf{f}=\left(\frac{2}{\tau}\mathbf{y}^{\mathrm{T}},-\frac{2}{\tau}\mathbf{y}^{\mathrm{T}}\right)^{\mathrm{T}}$, $\mathbf{y}=\left(\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_N\right)^{\mathrm{T}}$ and $\mathbf{H}$ is given by Eq. (A17). The details of the derivation of (27) is given in Appendix A. Compared with (26), the corresponding dual problem (27) can be solved more easily. According to the dual theory and the results obtained in Appendix A, the optimal solution of $\boldsymbol{\beta}$ in the original optimization problem (26) can be expressed as follows

$$\boldsymbol{\beta}=\frac{2}{\tau}\left(\boldsymbol{\Phi}(\mathbf{s})^T\boldsymbol{\Phi}(\mathbf{s})+2\lambda\boldsymbol{\Omega}\right)^{-1}\sum_{i=1}^{N}\left(\lambda_i'^{+}-\lambda_i'^{-}\right)\boldsymbol{\Phi}(\mathbf{s})^T\mathbf{x}_{\rho i}. \tag{28}$$

Furthermore, the solution of $\mathbf{p_g}$ in GHMM or the decision function of GHMM can be obtained based on (22) and (28).

C Decision Function

When the hidden-mapping model is trained, the decision function can take different forms of expressions depending on whether the hidding-mapping is explicit or unknown.

(1) Case1: the hidden-mapping is explicit

The form of hidden-mapping is known in this case. The solution of $\mathbf{p_g}$ can be expressed explicitly based on (22) and (28). Accordingly, the decision function can be expression as

$$y=f(\mathbf{x})=\mathbf{p}_{\mathbf{g}}^{\mathbf{T}}\mathbf{x}_{\mathbf{g}}=\left(\boldsymbol{\Phi}(\mathbf{s})\boldsymbol{\beta}\right)^{\mathbf{T}}\mathbf{x}_{\mathbf{g}} \tag{29}$$

$$\boldsymbol{\beta}=\frac{2}{\tau}(\boldsymbol{\Phi}(\mathbf{s})^{\mathbf{T}}\boldsymbol{\Phi}(\mathbf{s})+2\lambda\boldsymbol{\Omega})^{-1}\sum_{i=1}^{N}\left(\lambda_i'^{+}-\lambda_i'^{-}\right)\boldsymbol{\Phi}(\mathbf{s})^{\mathbf{T}}\mathbf{x}_{gi}. \tag{30}$$

(2) Case2: the hidden-mapping is unknown

As the exact form of the hidden-mapping is unknown, the explicit form of $\mathbf{p_g}$ cannot be determined. However, by introducing kernel trick, the decision function can be written as follows.

$$y=f(\mathbf{x})=\mathbf{p}_{\mathbf{g}}^{\mathbf{T}}\mathbf{x}_{\mathbf{g}}=\left(\boldsymbol{\Phi}(\mathbf{s})\boldsymbol{\beta}\right)^{\mathbf{T}}\mathbf{x}_{\mathbf{g}}=\boldsymbol{\beta}^{\mathbf{T}}\mathbf{K}(\boldsymbol{\Phi}(\mathbf{s}),\mathbf{x}_g) \tag{31}$$

$$\boldsymbol{\beta}=\frac{2}{\tau}\left(\mathbf{K}(\Phi(\mathbf{s}),\Phi(\mathbf{s}))+2\lambda_1\boldsymbol{\Omega}\right)^{-1}\sum_{i=1}^{N}\left(\lambda_i'^{+}-\lambda_i'^{-}\right)\mathbf{K}(\Phi(\mathbf{s}),\mathbf{x}_g) \tag{32}$$

D Algorithm of the Generalized Hidden-mapping Transductive Transfer Learning

Based on the discussion above, the algorithm of the proposed Generalized Hidden-mapping Transductive Transfer Learning, called GHM-TTL, is detailed in Table I.

Table I Algorithm of GHM-TTL

Algorithm of GHM-TTL	
Stage1：Construction of data in the hidden-mapping feature space	
Step1：	Construct the datasets of the source and the target domain domains in the hidden-mapping feature space, i.e., $\tilde{D}_s=\{\mathbf{x}_{g1},\mathbf{x}_{g2},\cdots,\mathbf{x}_{gN}\}$ and $\tilde{D}_t=\{\mathbf{z}_{g1},\mathbf{z}_{g2},\cdots,\mathbf{z}_{gM}\}$ based on the source dataset $D_s=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N\}$, the target dataset $D_t=\{\mathbf{z}_1,\mathbf{z}_2,\cdots,\mathbf{z}_M\}$ in the original feature space and the intelligent model adopted.
Stage 2：Transfer learning of model parameters	
Step2：	Set the parameter τ and λ in the objective function for transfer learning
Step3：	Compute $\boldsymbol{\beta}$ using (28) and compute $\mathbf{p_g}$ using (22) if the hidden-mapping is explicit.
Step4:	Obtain the decision function using (29) or (31)

E Strategy for Multi-class Classification using Regression Model

Although the generalized hidden-mapping model is developed based on regression model, it is also suitable for multi-class classification problems, which can be achieved using simple strategies. One strategy is to use the output of the regression model to approximate the class labels in the corresponding classification task [30]. Once the regression model is trained, testing is performed such that the label that is closest to the model output is taken as the label of the test sample. A common strategy is to transform the multi-classification problem as a multiple-output regression problem. The procedure is described in brief as follows. Given a dataset with *m* classes: $\{\mathbf{x}_i,\mathbf{y}_i\}$, $\mathbf{y}_i\in\{1,2,\cdots,m\},i=1,2,\cdots,N$， an *m*-output regression dataset $\{\boldsymbol{x}_i,\tilde{\mathbf{y}}_i\}$ is firstly constructed. If the class label of the *i*th training sample in $\{\mathbf{x}_i,\mathbf{y}_i\}$ is $p(1\le p\le m)$, the corresponding output vector in the constructed *m*-output regression dataset is defined as $\tilde{\mathbf{y}}_i=[0,\cdots,\overset{p}{1},0,\cdots,0]^{\mathbf{T}}$, where only the *p*th element of $\tilde{\mathbf{y}}_i$ is one and the rest of the elements are set to zero. An *m*-output regression model can be regarded as *m* single-output regression models. Once the *m* regression models are obtained, the output vector can be expressed as $\tilde{\mathbf{y}}_i^{\mathbf{model}}=[\tilde{y}_{i1}^{model},\cdots,\tilde{y}_{im}^{model}]^{\mathbf{T}}$ for a given testing sample. Then the predicted class label of the testing sample is the index of the element containing the maximum value in the output vector. For example, if the value of $\tilde{y}_{il}^{model}$ is the largest element in the vector

$\tilde{\mathbf{y}}_i^{\text{model}} (i = 1,...,m)$, then the final predicted class label of the test sample is $\boldsymbol{l}$.

IV Experiments

A Dataset

The epileptic EEG data used in this study are publicly available on the web from the University of Bonn, Germany, (http://www.meb.uni-bonn.de/epileptologie/science/physik/eegdata.html). EEG signal measurement and complete description can be found in [31]. The epileptic EEG database contains 5 groups of data (denoted by groups A to E), each containing 100 samples. Groups A and B consist of segments acquired from surface EEG recording performed on five healthy volunteers using standardized electrode placement scheme. Recording was made when the subjects were relaxed in awaken state with eyes open (group A) and eyes closed (group B) respectively. Groups C, D and E are data obtained from volunteers with epilepsy. EEG signals in group C were recorded from the hippocampal formation of the opposite hemisphere of brain, while those in group D were measured from within the epileptogenic zone. Group E contains EEG signals recorded during seizure activity. Table II gives a brief description of the five groups of the EEG database.

Table II Description of EEG data base

Subject	Group	description of datasets
Healthy people	A	EEG signals measured from healthy people with eyes open
	B	EEG signals measured from healthy people with eyes closed
Patients with epilepsy	C	EEG signals obtained in hippocampal formation of the opposite hemisphere of brain during seizure free intervals
	D	EEG signals obtained from within epileptogenic zone during seizure free intervals
	E	EEG signals measured during seizure activity

B Experiment Setup

Two datasets with no drifting between the distribution of the training and testing data were constructed. In addition, six datasets with certain differences in distribution between the of training and the testing data were also constructed. The details of these eight datasets are shown in Table III. Datasets 1 and 2 have the same distribution, whereas difference in distribution exists in datasets 3 to 8. Among them, datasets 1, 3-6 are all binary classification datasets, i.e., healthy subjects versus epileptic subjects. Datasets 2, 7 and 8 are multi-class datasets where the classes are healthy and epileptic subjects in different states.

Table III Datasets used in the experiments

Distribution	Dataset	Composition of the training dataset	Composition of the testing dataset	Number of classes
Identical	1	B, E (75)	B, E (25)	2
	2	B, D, E (75)	B, D, E (25)	3
Different	3	A, E (50)	A, C (50)	2
	4	A, E (50)	A, D (50)	2
	5	B, E (50)	B, C (50)	2
	6	B, E (50)	B, D (50)	2
	7	A, C, E (50)	B, CE (50)	3
	8	A, D, E (50)	B, D, E (50)	3

The number in bracket gives the number of samples taken from the groups of data involved.

D Algorithms, Parameter Setting and Evaluation Index

Listed in Table IV are the ten algorithms involved the experiments. The grids for searching the optimal hyper-parameters based on five-fold cross-validation strategy are provided in detail. In particular, the proposed GHM-TTL is applied to TSK FS, single hidden-layer neural network (Sig-NN) and radial bias function with kernelized linear regression using radial basis function (RBF). Correspondingly, the three algorithms are denoted in the paper as TGHM(TSK FS), TGHM(Sig-NN) and TGHM(RBF-Ker). (TGHM is not defined in the texts, please define it?? I try to relate GHM-TTL with the three TGHM methods as highlighted in green above) For each of the ten algorithms, the experiments are repeated 10 times with the data sampled randomly from the five groups of data, and in proportion based on the composition given in Table III. The mean and standard deviation of classification accuracies of 10 runs are then evaluated for performance evaluation.

Table IV Parameter setup for the algorithms used in this study

Algorithm and description	Parameter setting
SVM [32]: a classical method based on kernel trick and margin maximization	Penalty factor: $c \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$; Kernel parameter: $\gamma \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$.
TSK FS [33]: a classical fuzzy modeling method with good interpretability and strong learning abilities	Fuzzy rules: $M \in \{10,20,\ldots 200\}$; Regularization parameters: $\tau \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$, $\lambda \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$
Sig-NN [34]: a popular single hidden-layer neural network	Parameters in Sigmoid function: $\gamma \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$ $\kappa \in \{-2^{-12}, -2^{-11}, -2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$
LMPROJ [29]: large margin transductive SVM based on the maximum mean difference	Regularization parameters: $\tau \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$, $\lambda \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$ Kernel parameter: $\delta \in \{-2^{-10}, -2^{-9}, -2^{-8}, ..., 2^{8}, 2^{9}, 2^{10}\}$.
TSVM [35]: An SVM based transfer learning algorithm	Upper bound for Lagrange multiplier: $c \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$; Kernel parameter: $\gamma \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$. Regularization parameter in transfer learning: $\lambda \in \{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^{4}, 2^{5}, 2^{6}\}$
GTL2 and GTL3 [36]: two graph co-regularization transfer learning algorithms based on non-negative matrix factorization	Neighbor parameter: $p \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$ Regularization parameters: $\lambda \in \{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^{4}, 2^{5}, 2^{6}\}$, $\gamma \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$ $\sigma \in \{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^{4}, 2^{5}, 2^{6}\}$.
TGHM (TSK FS): the proposed method for TSK FS transfer learning	Fuzzy rules: $M \in \{10,20,\ldots,200\}$; Adjustable parameters: $h \in \{0.1,0.2,\ldots 10\}$; Insensitive coefficient: $\varepsilon \in \{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^{4}, 2^{5}, 2^{6}\}$; Regularization parameter in transfer learning: $\lambda \in \{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^{4}, 2^{5}, 2^{6}\}$, (change the comma to full stop, I cannot edit)
TGHM (Sig-NN): the proposed method for Sig-NN transfer learning.	Parameters in Sigmoid function: $\gamma \in \{2^{-12}, 2^{-11}, 2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$; $\kappa \in \{-2^{-12}, -2^{-11}, -2^{-10}, ..., 2^{10}, 2^{11}, 2^{12}\}$.
TGHM (RBF-Ker): the proposed method for transfer learning of kernelized linear regression with RBF	Parameters in Gauss function: $\sigma \in \{2^{-6}, 2^{-5}, 2^{-4}, ..., 2^{4}, 2^{5}, 2^{6}\}$; Width in Gauss function: $\delta \in \{-2^{-10}, -2^{-9}, -2^{-8}, ..., 2^{8}, 2^{9}, 2^{10}\}$.

E Comparing with Classical Intelligent Modeling Methods

In this subsection, the performance of the proposed methods is compared with 3 classical non-transfer learning algorithms, i.e., SVM, TSK FS, and Sig-NN. The classification accuracies of are reported in Table V.

Table V Comparing of the classification accuracies between the proposed methods and several classical non-transfer methods

Dataset	Non-transfer learning methods			The proposed methods		
	SVM	TSK FS	Sig-NN	TGHM (TSK FS)	TGHM (Sig-NN)	TGHM (RBF-Ker)
1	0.9200 (0.0227)	**0.9800** (0.1065)	**0.9900** (0.0215)	0.9530 (0.0365)	**0.9700** (0.0354)	0.9670 (0.0183)
2	0.9200 (0.0213)	0.9530 (0.0305)	0.9500 (0.0187)	0.9800 (0.0183)	0.9400 (0.0380)	**0.9805** (0.0137)
3	0.8800 (0.0123)	0.8800 (0.0215)	0.8900 (0.0278)	**0.9670** (0.0430)	0.9500 (0.0447)	0.9330 (0.0279)
4	0.8000 (0.0275)	0.8900 (0.0165)	0.8500 (0.0370)	0.9400 (0.0245)	**0.9500** (0.0218)	0.9330 (0.0380)
5	0.8500 (0.0365)	0.9700 (0.0367)	0.9200 (0.0165)	0.9467 (0.0558)	**0.9500** (0.0418)	0.9467 (0.0471)
6	0.8200 (0.0360)	0.9600 (0.0195)	0.9500 (0.0395)	**0.9670** (0.0333)	0.9600 (0.0354)	0.9330 (0.0548)
7	0.8130 (0.1005)	0.8500 (0.0065)	0.8500 (0.1270)	0.9200 (0.0135)	0.9333 (0.0380)	**0.9600** (0.0224)
8	0.8130 (0.0165)	0.8500 (0.0167)	0.8200 (0.1300)	0.9200 (0.0162)	0.9133 (0.0435)	**0.9600** (0.0224)
Average	0.8520 (0.0341)	0.9166 (0.0318)	0.9025 (0.0523)	0.9492 (0.0301)	0.9458 (0.0373)	**0.9517** (0.0306)

Based on the results in Table V, it can be seen that while the proposed methods have comparable performance with the three classical non-transfer learning methods on datasets 1 and 2, their performance is obviously better on datasets 3-7. This indicates that when drifting exits in the distribution between the training and the testing data, transfer learning is necessary to improve the classification performance.

F Comparing with related transfer learning methods

To further evaluate the performance of the proposed methods, four existing transfer learning methods, i.e., LMPROJ, TSVM, GTL2 and GTL3, are used for performance comparison. The results are presented in Table VI.

Table VI Comparing with transfer learning methods

Dataset	Transfer learning methods						
	Existing transfer learning methods				The proposed methods		
	LMPROJ	TSVM	GTL2	GTL3	TGHM (TSK FS)	TGHM (SigNN)	TGHM (RBF-Ker)
1	0.9480 (0.0235)	0.9267 (0.0346)	0.6787 (0.0831)	0.5907 (0.1006)	0.9530 (0.0365)	**0.9700** (0.0354)	0.9670 (0.0183)
2	0.9370 (0.0258)	0.9500 (0.0593)	0.6424 (0.0713)	0.5536 (0.0623)	0.9800 (0.0183)	0.9400 (0.0380)	**0.9805** (0.0137)
3	0.9410 (0.0717)	0.8700 (0.0715)	0.5787 (0.1167)	0.5320 (0.2413)	**0.9670** (0.0430)	0.9500 (0.0447)	0.9330 (0.0279)
4	**0.9500** (0.0232)	0.9300 (0.0789)	0.5600 (0.1261)	0.5293 (0.2289)	0.9400 (0.0245)	**0.9500** (0.0218)	0.9330 (0.0380)
5	0.9380 (0.1740)	0.8900 (0.0994)	0.6013 (0.0687)	0.7213 (0.1547)	0.9467 (0.0558)	**0.9500** (0.0418)	0.9467 (0.0471)
6	0.9590 (0.0354)	0.9200 (0.0919)	0.6067 (0.1360)	0.5773 (0.1223)	**0.9670** (0.0333)	0.9600 (0.0354)	0.9330 (0.0548)
7	0.9470 (0.0532)	0.7344 (0.0752)	0.6856 (0.0522)	0.5024 (0.1104)	0.9200 (0.0135)	0.9333 (0.0380)	**0.9600** (0.0224)
8	0.9380 (0.0355)	0.7500 (0.0572)	0.6704 (0.0659)	0.5568 (0.1255)	0.9200 (0.0162)	0.9133 (0.0435)	**0.9600** (0.0224)
Average	0.9447 (0.0553)	0.8714 (0.0710)	0.62798 (0.0900)	0.57043 (0.1436)	0.9492 (0.0302)	0.9458 (0.0373)	**0.9517** (0.0306)

It can be seen from the results in Table VI that the proposed transfer learning based methods outperform, or are at least competitive with the four existing transfer learning methods. Moreover, the proposed methods have the following distinctive advantages: (1) the proposed methods are more general in that they can be used to train different classical intelligent models, e.g. feedforward neural networks, fuzzy systems and kernel methods; (2) the proposed transfer learning approach can be used in a more flexible way and provides more choices depending on the application scenario and requirements. For example, if a model having a good interpretability is desired, fuzzy system can be selected for modeling and trained using the proposed transfer learning method.

G Statistical analysis

In this subsection, the proposed methods are further evaluated by statistical analysis. The Friedman test [37], a nonparametric test method, is adopted to evaluate whether significant difference in performance exists among the different methods. The procedure is as follows. First, the original data which accepts K experiment process from the same object (this sentence, marked in green, is not clear to me, please rephrase) are ranked. After analyzing the optimal algorithm based

on the rankings, post-hoc test [37] is conducted to verify the algorithm. The rankings of all the algorithms adopted in this study based on the experiments are given in Table VII. The lower the ranking, the better the algorithm. (Maybe use other term, e.g. score, instead of ranking, since it is generally understood that the higher the raking the better. This is reversed in Table IV, for your consideration. For example, you may say the lower the score, the higher the ranking and the better the performance)

Table VII Friedman Test on all algorithms

Algorithms	Ranking (Score??)
SVM	7.5625
TSK FS	3.9375
Sig-NN	5
LMPROJ	3.6875
TSVM	6.8125
GTL2	9.125
GTL3	9.875
TGHM (TSK FS)	2.75
TGHM (Sig-NN)	3
TGHM (RBF-Ker)	3.25

It can be seen from Table VII that that the proposed three algorithms TGHM (TSK FS), TGHM (Sig-NN) and TGHM (RBF-Ker) are ranked top three, outperforming the other algorithms. Based on the results of Friedman test, post-hoc test is implemented to compare the optimal algorithm TGHM (TSK FS) with the other 9 algorithms to further verify the optimality of the TGHM (TSK FS) algorithm. The results are shown in Table VIII.

Table VIII Post-hoc Test on all algorithms

Algorithms	$z = (R_0 - R_i) / SE$	p	Holm=/i	Hypothesis
GTL3	4.665334	0.000003	0.005556	Rejected
GTL2	4.1699	0.000003	0.00625	Rejected
SVM	3.137747	0.001703	0.007143	Rejected
TSVM	2.642313	0.008234	0.008333	Rejected
Sig-NN	1.445015	0.148454	0.01	Not rejected
TSK FS	0.743151	0.457391	0.0125	Not rejected
LMPROJ	0.660578	0.508883	0.016667	Not rejected
TGHM(RBF-Ker)	0.165145	0.86883	0.025	Not rejected
TGHM (Sig-NN)	0.123858	0.901427	0.05	Not rejected

The p-values in the table are derived from the corresponding post-hoc test and the significance level is set to 0.05. If the p-value is smaller than the Holm value (the threshold for comparison), the null hypothesis is rejected and it indicates that TGHM (TSK FS) is superior to the corresponding algorithm. Otherwise, the null hypothesis is not rejected and no significant difference exist between TGHM (TSK FS) and the algorithm under comparison. It can be seen from Table VIII that TGHM (TSK FS) is highly competitive to TGHM (RBF-Ker), TGHM (Sig-NN) and LMPROJ for the epileptic EEG recognition in terms of accuracy, and that it is also advantageous over GTL3, GTL2,

SVM and TSVM. The results show that the group of transfer learning based algorithms – TGHM (TSK FS), TGHM (RBF-Ker), TGHM (Sig-NN) and LMPROJ – have demonstrated promising performance for epileptic EEG recognition, an appropriate algorithm can be selected depending on the nature of the model and the application requirements. For example, the fuzzy rules based TGHM (TSK FS) can be used if an interpretable recognition model is desired.

V Conclusions

In this study, a generalized transductive transfer learning method TGHM is proposed for epileptic EEG recognition. It can be used to implement transductive transfer learning for several classical intelligent models, e.g. feedforward neural networks, fuzzy systems and kernel methods. The experimental studies have proved that the proposed method is superior to or at least competitive with the existing transductive learning methods. The proposed approach has the distinctive advantage that it can be generally and flexibly applied to different types of epileptic EEG recognition models. To further exploit and enhance the transfer learning abilities, future work will be conducted to develop more effective learning mechanism.

Appendix A:

The Lagrange function corresponding to (26) can be expressed as follows,

$$\begin{aligned} L(\boldsymbol{\beta},\boldsymbol{\xi}^{+},\boldsymbol{\xi}^{-},\varepsilon,\boldsymbol{\lambda}^{+},\boldsymbol{\lambda}^{-}) = {} & \frac{1}{\tau N}\sum_{i=1}^{N}((\xi_i^{+})^2+(\xi_i^{-})^2)+\frac{1}{2}\boldsymbol{\beta}^{T}\Phi(\mathbf{s})^{\mathbf{T}}\Phi(\mathbf{s})\boldsymbol{\beta}+\frac{2}{\tau}\varepsilon \\ & +\lambda\boldsymbol{\beta}^{T}\Phi(\mathbf{s})^{\mathbf{T}}\boldsymbol{\Omega}\Phi(\mathbf{s})\boldsymbol{\beta}+\sum_{i=1}^{N}\lambda_i^{+}(y_i-\boldsymbol{\beta}^{T}\Phi(\mathbf{s})^{\mathbf{T}}\mathbf{x}_{gi}-\varepsilon-\xi_i^{+}). \\ & +\sum_{i=1}^{N}\lambda_i^{-}(\boldsymbol{\beta}^{T}\varphi(\mathbf{S})^{\mathbf{T}}\mathbf{x}_{gi}-y_i-\varepsilon-\xi_i^{-}) \end{aligned} \tag{A1}$$

According to the optimization theory, the conditions of the optimal solution are given by

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \Phi(\mathbf{s})^{\mathrm{T}}\Phi(\mathbf{s})\boldsymbol{\beta}+2\lambda\boldsymbol{\Omega}\boldsymbol{\beta}-\sum_{i=1}^{N}\left(\lambda_i^{+}-\lambda_i^{-}\right)\Phi(\mathbf{s})^{\mathrm{T}}\boldsymbol{x}_{gi}=0 \quad , \tag{A2}$$

$$\frac{\partial L}{\partial \xi^{+}} = \frac{2}{\tau N}\xi_i^{+}-\lambda_i^{+}=0 \quad , \tag{A3}$$

$$\frac{\partial L}{\partial \xi^{-}} = \frac{2}{\tau N}\xi_i^{-}-\lambda_i^{-}=0 \, , \tag{A4}$$

$$\frac{\partial L}{\partial \varepsilon} = \frac{2}{\tau}-\sum_{i=1}^{N}\left(\lambda_i^{+}+\lambda_i^{-}\right)=0 \, . \tag{A5}$$

Let

$$\lambda_i^{+} = \frac{2}{\tau}\lambda_i'^{+} \, , \tag{A6}$$

$$\lambda_i^{-} = \frac{2}{\tau}\lambda_i'^{-} \quad . \tag{A7}$$

We have

$$\xi_i^{+} = \frac{N\tau}{2}\lambda_i^{+} = N\lambda_i'^{+} \quad , \tag{A8}$$

$$\xi_i^- = \frac{N\tau}{2}\lambda_i^- = N\lambda_i'^- \quad , \tag{A9}$$

$$\sum_{i=1}^{N}\left(\lambda_i'^+ + \lambda_i'^-\right) = 1, \tag{A10}$$

$$\boldsymbol{\beta} = \left(\Phi(\mathbf{s})^T\,\Phi(\mathbf{s}) + 2\lambda\boldsymbol{\Omega}\right)^{-1}\sum_{i=1}^{N}\left(\lambda_i^+ - \lambda_i^-\right)\Phi(\mathbf{s})^{\mathrm{T}}\,\mathbf{x}_{gi}\,. \tag{A11}$$

Furthermore, let

$$\boldsymbol{\Psi} = \left(\Phi(\mathbf{s})^{\mathbf{T}}\,\Phi(\mathbf{s}) + 2\lambda\boldsymbol{\Omega}\right)^{-1}, \tag{A12}$$

$$\boldsymbol{\beta} = \boldsymbol{\Psi}\sum_{i=1}^{N}\left(\lambda_i^+ - \lambda_i^-\right)\Phi(\mathbf{s})^{\mathrm{T}}\,\mathbf{x}_{gi} = \frac{2}{\tau}\boldsymbol{\Psi}\sum_{i=1}^{N}\left(\lambda_i'^+ - \lambda_i'^-\right)\Phi(\mathbf{s})^{\mathrm{T}}\,\mathbf{x}_{gi}\ . \tag{A13}$$

By substituting (A8)-(A13), the dual problem of the primal problem is given by

$$\begin{aligned}\min\ L\left(\boldsymbol{\lambda}'^+,\boldsymbol{\lambda}'^-\right) = &-\sum_{i=1}^{N}\left(\frac{N}{\tau}\left(\lambda_i'^+\right)^2 + \frac{N}{\tau}\left(\lambda_i'^-\right)^2\right) + \sum_{i=1}^{N}\frac{2}{\tau}y_i\left(\lambda_i'^+ - \lambda_i'^-\right) + \\ &\frac{2}{\tau^2}\mathbf{x}_{gi}^T\Phi(\mathbf{s})\boldsymbol{\Psi}\Phi(\mathbf{s})^T\,\mathbf{x}_{gj}\sum_{i=1}^{N}\left(\lambda_i'^+ - \lambda_i'^-\right)\sum_{j=1}^{N}\left(\lambda_j'^+ - \lambda_j'^-\right) \quad ,\end{aligned} \tag{A14}$$

$$\mathbf{s.t}\begin{cases}\sum_{i=1}^{N}\left(\lambda_i'^+ + \lambda_i'^-\right) = 1 \quad \forall i \\ \lambda_i'^+ \ge 0,\ \ \lambda_i^- \ge 0\end{cases}$$

where

$$\tilde{\boldsymbol{\alpha}} = \left(\lambda_1'^+,\ldots\lambda_N'^+,\lambda_1'^-,\cdots,\lambda_N'^-\right)^{\mathrm{T}}, \tag{A15}$$

$$\mathbf{K} = \left[\tilde{k}_{ij}\right]_{N\times N},\ \tilde{k}_{ij} = \frac{2}{\tau^2}\mathbf{x}_{\rho i}^{\mathbf{T}}\Phi(\mathbf{s})\boldsymbol{\Psi}\Phi(\mathbf{s})^{\mathbf{T}}\,\mathbf{x}_{gi} + \frac{N}{\tau}\delta_{ij}, \tag{A16.a}$$

$$\delta_{ij} = \begin{cases}1, & i = j\\ 0, & i \ne j\end{cases}, \tag{A16.b}$$

$$\mathbf{H} = \begin{bmatrix}\mathbf{K} & \mathbf{-K}\\ \mathbf{-K} & \mathbf{K}\end{bmatrix}, \tag{A17}$$

$$\mathbf{f} = \left(\frac{2}{\tau}\mathbf{y}^T, -\frac{2}{\tau}\mathbf{y}^T\right)^{\mathrm{T}}, \mathbf{y} = \left(\mathbf{y}_1,\mathbf{y}_2,\cdots,\mathbf{y}_N\right)^T. \tag{A18}$$

References

[1]. C. Yang, Z. Deng, K. S. Choi, Y. Jiang, S. Wang, "Transductive domain adaptive learning for epileptic electroencephalogram recognition," *Artificial Intelligence in Medicine*, vol. 62, no. 3, pp. 165-177, 2014.

[2]. K. Polat, S. Güneş, "Classification of epileptic form EEG using a hybrid system based on decision tree classifier and fast Fourier transform," *Applied Mathematics and Computation*, pp. 1017–1026, 2007.

[3]. Z. Deng, K. S. Choi, F. L. Chung, et al. "Scalable TSK fuzzy modeling for very large datasets using minimal-enclosing-ball approximation," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 2, pp. 210-226, 2011.

[4]. M. Setnes, H. Roubos "GA-fuzzy modeling and classification: complexity and performance," *IEEE transactions on Fuzzy Systems*, vol. 8, no. 5, pp. 509-522, 2000.

[5]. A. Arnold, R. Nallapati, W. W. Cohen, "A comparative study of methods for transductive transfer learning," *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007),* Vol. 7, pp. 77-82, 2007.

[6]. S. J. Pan, Q. Yang, "A survey on transfer learning. Knowledge and Data Engineering," *IEEE Transactions on Knowledge and Data Engineering* , vol. 22, no. 10, pp. 1345-1359, 2010.

[7]. B. Quanz, J. Huan, "Large margin transductive transfer learning," *Proceedings of the 18th ACM conference on Information and knowledge management*. *ACM*, pp. 1327-1336, 2009.

[8]. J. M. Leski, "TSK-fuzzy modeling based on ϵ-insensitive learning," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 2, pp. 181-193, 2005.

[9]. C. Yang, Z. Deng Z, K. S. Choi, et al., "Takagi-Sugeno-Kang Transfer Learning Fuzzy Logic System for the Adaptive Recognition of Epileptic Electroencephalogram Signals*," IEEE Transactions on Fuzzy Systems,* DOI: 10.1109/TFUZZ.2015.2501438,2016.

[10]. Z. Deng, K. S. Choi, Y. Jiang Y, et al., "Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods," *IEEE Transactions on Cybernetics* ,vol. 44, no. 12, pp. 2585-2599, 2014.

[11]. G. B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE Transactions on Neural Network*s, vol. 14, no. 2, pp. 274-281, 2003.

[12]. R. J. Williams, D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no.2, pp. 270-280, 1989.

[13]. R. Harikumar, B. S. Narayanan, "Fuzzy Techniques for Classification of Epilepsy risk level from EEG Signals," *Conference on Convergent Technologies for the Asia-Pacific Region*, pp. 209-213, 2003.

[14]. A. Bahrammirzaee, A. R. Ghatari, P. Ahmadi, et al., "Hybrid credit ranking intelligent system using expert system and artificial neural networks," *Applied Intelligence*, vol. 34, no. 1, pp. 28-46, 2011.

[15]. G. B. Huang, Q. Y. Zhu, C. K. Siew. "Extreme learning machine: a new learning scheme of feedforward neural networks," *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on,* vol. 2, pp. 985-990, 2004.

[16]. K. Polat, S. Güneş, "Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast Fourier transform," *Applied Mathematics and Computation*, vol. 187, no. 2, pp. 1017-1026, 2007.

[17]. T. T. Teo, T. Logenthiran, W. L. Woo, "Forecasting of photovoltaic power using extreme learning machine", *Smart Grid Technologies-Asia (ISGT ASIA), 2015 IEEE Innovative. IEEE*, pp. 1-6, 2015.

[18]. S. Yun, J. Lee, Y. Chung, et al., "Centroid localization method in wireless sensor networks using TSK fuzzy modeling," *ISIS 2007 PROCEEDINGS OF THE 8TH SYMPOSIUM ON ADVANCED INTELLIGENT SYSTEMS.* pp. 971-974, 2007.

[19]. A. Esfahanipour, W. Aghamiri, "Adapted neuro-fuzzy inference system on indirect approach TSK fuzzy rule base for stock market analysis," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4742-4748, 2010.

[20]. C. F. Juang, C. Lo, "Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence algorithm," *Fuzzy Sets and Systems*, vol. 159, no. 21, pp. 2910-2926, 2008.

[21]. A. Aarabi, R. Fazel-Rezai, Y. Aghakhani, "A fuzzy rule-based system for epileptic seizure detection in intracranial EEG," *Clinical Neurophysiology*, vol. 120, no. 9, pp. 1648-1657, 2009.

[22]. C. Saunders, A. Gammerman, V. Vovk, "Ridge regression learning algorithm in dual variables," *Proceedings of the 15th International Conference on Machine Learning. Morgan Kaufmann*, pp. 515-521, 1998.

[23]. I. W. Tsang, A. Kocsor, and J. T. Kwok, "Large-scale maximum margin discriminant analysis using core vector machines," *IEEE Trans. Neural Networks.*, vol. 19, no. 4, pp. 610–624, Apr. 2008.

[24]. I. W. Tsang, A. Kocsor, and J. T. Kwok, “Large-scale maximum margin discriminant analysis using core vector machines,” *IEEE Trans. Neural. Networks*, vol. 19, no. 4, pp. 610–624, Apr. 2008.

[25]. Z. Deng, Y. Jiang, L. Cao, et al., “Enhanced knowledge-leveraged TSK fuzzy systems for inductive transfer learning,” *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 01, article 11.

[26]. H. Tang, W. S. Meddaugh, N. Toomey, “Using an artificial-neural-network method to predict carbonate well log facies successfully,” *Society of Petroleum Engineers* , vol. 14, no. 01, pp. 35-44, 2011.

[27]. A. E. Hoerl, R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55-67, 1970.

[28]. P. Wahlberg, G. Salomonsson, “Feature extraction and clustering of EEG epileptic spikes,” *Computers and biomedical research*, vol. 29, no. 5, pp. 382-394, 1996.

[29]. C. Yang, Z. Deng, K. S. Choi, et al., “Transductive domain adaptive learning for epileptic electroencephalogram recognition,” *IEEE Transactions on Fuzzy Systems* , vol. 62, no. 3, pp. 165-177, 2014.

[30]. G. B. Huang, P. Saratchandran, N. Sundararajan, “A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation,” *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57-67, 2005.

[31]. R. G. Andrzejak, K. Lehnertz, F. Mormann, et al., “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” *Physical Review E*, vol. 64, no. 6, pp. 1907-1914, 2001.

[32]. M. Shen, J. Chen, C. Lin, “Modeling of nonlinear medical signal based on local support vector machine,” In *IEEE Instrumentation and Measurement Technology Conference,* pp. 675-679, 2009.

[33]. Z. Deng, Y. Jiang, K. S. Choi, et al., “Knowledge-leverage-based TSK fuzzy system modeling,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1200-1212, 2013.

[34]. Z. Deng, Y. Jiang, F. L. Chung, et al., “Knowledge-leverage-based fuzzy system and its modeling,” *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 4, pp. 597-609, 2013.

[35]. C. X. Ren, D. Q. Dai, K. K. Huang, et al., “Transfer Learning of Structured Representation for Face Recognition”, *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5440 – 5454, Dec. 2014.

[36]. M. Long, J. Wang, G. Ding, et al., “Transfer learning with graph co-regularization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1805-1818, 2014.

[37]. T. Joachims, “Transductive inference for text classification using support vector machines,” *In Proc. ICML, Morgan Kaufmann Publishers*, San Francisco, pp. 200-209, 1999.