Data Min Knowl Disc (2015) 29:1094–1112
DOI 10.1007/s10618-014-0389-3

Constrained elastic net based knowledge transfer for healthcare information exchange

Yan Li · Bhanukiran Vinzamuri · Chandan K. Reddy

Received: 3 April 2014 / Accepted: 22 September 2014 / Published online: 23 December 2014

Abstract Transfer learning methods have been successfully applied in solving a wide range of real-world problems. However, there is almost no attempt of effectively using these methods in healthcare applications. In the healthcare domain, it becomes extremely critical to solve the "when to transfer" issue of transfer learning. In highly divergent source and target domains, transfer learning can lead to negative transfer. Most of the existing works in transfer learning are primarily focused on selecting useful information from the source to improve the performance of the target task, but whether the transfer learning can help and when the transfer learning should be applied in the target task are still some of the impending challenges. In this paper, we address this issue of "when to transfer" by proposing a sparse feature selection model based on the constrained elastic net penalty. As a case study of the proposed model, we demonstrate the performance using the diabetes electronic health records (EHRs) which contain patient records from all fifty states in the United States. Our approach can choose relevant features to transfer knowledge from the source to the target tasks. The proposed model can measure the differences between multivariate data distributions conditional on the predicted model, and based on this measurement we can avoid unsuccessful transfer. We successfully transfer the knowledge across different states to improve the diagnosis of diabetes in a certain state with insufficient records to build an individualized predictive model with the aid of information from other states.

Keywords Transfer learning · Regularization · Electronic health records

Responsible editors: Fei Wang, Gregor Stiglic, Ian Davidson and Zoran Obradovic.

Y. Li · B. Vinzamuri · C. K. Reddy (✉)
Department of Computer Science, Wayne State University, Detroit, MI, USA
e-mail: reddy@cs.wayne.edu

Y. Li
e-mail: rock_liyan@wayne.edu

1 Introduction

Due to the rapid growth in the amount of healthcare data, the concept of information exchange is gaining a lot of attention recently. The basic idea here is to share the knowledge acquired about a particular disease from one healthcare facility and apply it at another facility. While the naive solution to this problem is to merely integrate the data into a consolidated manner and build predictive models on such integrated data, such approaches often show poor performance since the target task (where the prediction needs to be done) will be overwhelmed by many instances from the data acquired at other locations. The key in such problems is to identify the "useful" knowledge that can potentially improve the performance at the target facility and transfer knowledge into the model to be learned on the target data.

To motivate our work, let us consider the following scenario. Firstly, let us assume that there are many different hospitals and each hospital contains records of patients suffering from a particular disease. Now, if we want to predict the disease status of a patient at a specific hospital X, one can potentially build a predictive model on this data collected at hospital X (using the medical records) and estimate the disease status. However, if the hospital contains only a few records, it may not be sufficient to build robust model that can yield accurate predictions for the future data. One way to overcome this problem is to use the data collected from several other hospitals and build an integrated prediction model. In this case, the prediction model might perform well on the integrated data, but might not yield good results on the data collected from X because the data might have a slightly different data distribution (of population or medical resources). Hence, in order to build a robust prediction model for hospital X, we need to transfer knowledge acquired from similar hospitals (with different data distributions) and incorporate that knowledge into the prediction model being built on data from hospital X.

The main objective of this paper is to transfer knowledge acquired from health records at different locations to improve the disease prediction ability of models built for a particular location. It should be noted that due to the lack of hospital-wide data from many different hospitals, we show the performance of our models on state-wide health records and transfer knowledge from all other states to the states with fewer diabetes patients in order to improve the accuracy of prediction.

In the field of machine learning, transfer learning deals with transferring knowledge from source to target domain when sufficient instances are not available to build a robust model for a given target task. One of the primary challenges associated with transfer learning is that it does not guarantee an improvement in performance of the target task, since an improper source domain can induce negative effects on learning the model and potentially degrade the performance of the classifier. This is also known as "negative transfer". In Rosenstein et al. (2005), the authors empirically demonstrated that negative transfer may happen if the distribution divergence between the target and source domains is too large. In order to avoid negative transfer, we will provide a more systematic framework that allows us to decide "when to transfer" based on suitable transfer-ability measures, and then select common features which can be transferred across domains.

In this paper, we address the problem of "when to transfer" for improving the clinical diagnosis of a specific geographical location or a local hospital where sufficient patient records are not available. To solve this problem, we aim to find common (hidden) features which can be used to transfer knowledge across the source and target domains. These latent features are then selected in a unique way by using sparse regularization.

This latent feature subset selection for effective transfer is done using the transfer learning based on constrained elastic net (*TR-CEN*) framework which is built based on the constrained elastic net (CEN) penalty. The CEN penalty modifies the standard elastic net penalty by enforcing the prediction models on the source domain and target domain to be similar to each other. This promotes learning a sparse model on the common relevant features shared by the source domain and target domain. The CEN penalty also uses the notion of supervised distribution difference (SDD). The SDD estimates the divergence between the source and target distributions using the divergence between equivalent regression models built on the source and target. To the best of our knowledge, transfer learning on sparse features using novel regularizers has not been investigated in the literature so far. We build this model in the context of healthcare applications because in such noisy data, it becomes extremely important to build models that are robust and can intelligently decide upon the "when to transfer" issue. Our experimental results over the medical records of diabetes patients suggest that *TR-CEN* outperforms other competing methods such as multi-task-LASSO.

The main contributions of our work are summarized as follows:

- Propose a novel transfer learning framework which can deal with the "when to transfer" issue in the transfer learning and successfully apply it in mining healthcare data.
- Develop a measure of the distance between the source and target using the divergence between their corresponding prediction models.
- Develop a constrained version of elastic net algorithm that can capture the differences in data distributions and discover the common (hidden) features which can be used to transfer the knowledge from the source task to target task.
- Demonstrate the performance of the proposed transfer learning method using diabetes healthcare records, and compare with the existing state-of-the-art methods on the problem of disease diagnosis in the healthcare domain.

This paper is organized as follows: Sect. 2 provides some relevant background regarding various transfer learning and multi-task learning methods, and highlights the main contribution of our work. Our novel transfer learning framework *TR-CEN* is explained in detail in Sect. 3 and several additional details of the proposed CEN algorithm are described in Sect. 4. In Sect. 5, the diabetes health records are used to demonstrate the performance of *TR-CEN* method. It also gives the details of the clinical feature transformation for summarizing multiple attribute healthcare record values. The empirical results demonstrate that the *TR-CEN* method can efficiently prevent negative transfer and outperform several baseline and other state-of-the-art methods available for knowledge transfer. Finally, Sect. 6 concludes our discussion and gives some future research directions for the proposed work.

2 Related work

Transfer learning methods have been successfully applied to many real-world applications such as web-document classification, sentiment classification (Blitzer et al. 2007), WiFi localization (Pan et al. 2008), and sign language recognition (Farhadi et al. 2007); however, transfer learning approaches do not adequately address the question of "when to transfer". In this section, we introduce some of the relevant topics and highlight the primary contributions of our work.

In transfer learning, the primary goal is to adapt a model built on source domain D_S (or distribution) for prediction on the target domain D_T. Accordingly, scenarios for transfer between the source and target domains can be categorized into three different types, namely, *inductive*, *transductive* and *unsupervised* transfer learning (Pan and Yang 2010). In the inductive transfer learning, labeled data are available in the target domain; in addition, based on whether there exists labeled data in the source domain, these approaches can be grouped into two sub-categories: the first type is similar to *multi-task learning* (Caruana 1997), where the labeled source data are available, and the second type is *Self-taught learning* (Raina et al. 2007), where the labeled source data are unavailable. In the transductive transfer learning (Arnold et al. 2007), source domain labels are available while target domain labels are unavailable; finally, in the unsupervised transfer learning, there are no labeled data available in both source domain and target domain (Dai et al. 2008).

The three main research issues in transfer learning are: "what to transfer", "how to transfer", and "when to transfer" (Pan and Yang 2010). "What to transfer" deals with the problem of what knowledge can be transferred from the source to the target domain in order to improve the performance of the prediction model for the target task. Based on "what to transfer" the existing transfer learning approaches can be grouped into four cases: *instance-based*, *feature-based*, parameter-based, and *relational knowledge-based*. The first category is *instance-based*, where the assumption is that certain parts of the source data can be used to learn the target task; TrAdaboost (Dai et al. 2007) is one of the most popular instance based transfer learning algorithm. In *feature-based* transfer learning, multiple methods are used to learn a good feature representation (Rückert and Kramer 2008; Pan et al. 2008) or select subset of joint features (Evgeniou and Pontil 2007) for the target domain. The *parameter-based* transfer learning approach assumes that there is parameter sharing between the source and target task (Evgeniou and Pontil 2004; Pan and Yang 2010). Finally, *relational knowledge-based* transfer learning (Mihalkova and Mooney 2008) assumes that some relation among the data in the source and target domains is similar (Pan and Yang 2010). "How to transfer" has a strong relationship with "what to transfer"; based on what type of knowledge is used for transfer, the corresponding techniques are involved in the transfer learning approaches.

In this paper, we perform knowledge transfer in healthcare applications by analyzing the EHRs collected at different geographic locations, and we have the labeled data in both target domain and source domain; thus, more specifically, the model we propose in this paper belongs to the first type of inductive transfer learning approach. Our procedure selects a subset of joint features to transfer the knowledge from the source to the target domain. This is different from the inductive transfer learning and

multi-task learning. In multi-task learning (Caruana 1997), different tasks are learned simultaneously and perfectly, while transfer learning only aims to improve the performance of the target task by taking advantage of the knowledge acquired from the source data. Thus, in multi-task learning, different tasks are equally weighted, but in transfer learning one is more interested in the target domain and target task; furthermore, it is very convenient to change the multi-task learning algorithm to transfer learning algorithm just by enhancing the importance (weight) of the target task (Pan 2010). In multi-task feature learning (Evgeniou and Pontil 2007), the representation of the features for all the tasks are learned simultaneously, and the sum of the loss functions of each individual tasks is penalized by the (r, p)-norm of the regression parameter matrix. In contrast, in our framework, we propose a constrained version of elastic net penalty where the regression parameters of the target and source tasks are learned separately, and the regression models of these two tasks are tuned to be as similar as possible; thus, we can select as many joint features as possible, which can be used to transfer the knowledge from the source to the target domain.

In this work, using the parameters of the constrained model, we propose a model-based approach to compute the distance between the target data distribution and source data distribution conditional on the learning task. The experimental results indicate that there exists a relationship between this supervised distribution distance (SDD) and the performance of target task; thus, we can use this distance as a measurement to decide whether it is appropriate to transfer knowledge from the source domain to the target domain.

3 Proposed framework

In this section, we explain the overall framework of the *TR-CEN* method. This framework uses the constrained elastic net in a transfer learning setting in order to effectively evaluate which features to consider for knowledge transfer from the source domain to the target domain. We now introduce the notations used in this paper in Table 1.

3.1 Why constrained elastic net?

The transfer learning framework we propose in this paper uses a sparse regularizer to learn a relevant subset of features to transfer knowledge from the source to the target domain. We constrain this regularizer to promote similarity of the regression coefficient vectors on the source and target domains. Lasso is effective at giving sparse solutions (Tibshirani 1996) but when variables are correlated, Lasso does not include all of them in it's solution. Many other correlated variables are neglected by Lasso. This makes the elastic net an efficient choice as it promotes sparsity and it can handle correlation due to the L_2 ridge term. Further details on the formulation of the constrained elastic net are provided in Sect. 4.

3.2 Transfer learning using constrained elastic net

TR-CEN is a sparse transfer feature learning method which aims to learn a low-dimensional subset of features that can be used to transfer knowledge from source

Table 1 Notations used in this paper

Notation	Description
D_T	Target dataset
D_S	Source dataset
C	Index set of selected features
D_C	Features selected for transfer
n	Number of instances
p	Number of features
X	Data matrix, $X \in \mathbb{R}^{n \times p}$
Y	Corresponding labels$= \{0,\ 1\}^n$
β	Coefficient vector, $\beta \in \mathbb{R}^p$
Ω	Coefficient vector for combined dataset
ε	Threshold for knowledge transfer
τ	Threshold for accuracy loss
$\mathscr{L}(\ )$	Objective function
$L(\ )$	Loss function
$P(\ )$	Penalty term

to target domain, and simultaneously reduce the prediction error of target task. *TR-CEN* employs the constrained elastic net regularizer while selecting features. This regularizer modifies the L_2 penalty in the elastic net by replacing it with a modified vector which penalizes the difference between the current model (β) on the source (target) and the base model(Ω) learned on the combined dataset ($D_S \cup D_T$).

TR-CEN starts by learning an elastic net model for the unified dataset ($D_S \cup D_T$), and the coefficient vector obtained from this unified dataset is denoted by Ω. After Ω is learned, we apply the constrained elastic net method on D_S and D_T to learn $\beta(D_S)$ and $\beta(D_T)$ respectively. *Tr-CEN* measures the data distribution distance between the source dataset and target dataset using the absolute value of the difference of the regression coefficient vectors learned on the source and target datasets. This is also known as the supervised Distribution Difference (SDD). It measures the change in the classification criteria in terms of measuring the deviation in classification boundary while classifying as accurately as possible. The SDD between the source and target data can be calculated as:

$$SDD(D_S, D_T) = \|\beta(D_S) - \beta(D_T)\|_1 \tag{1}$$

$SDD(D_S, D_T)$ is a quantitative measurement of the divergence between the source and target domains, and using this difference we will decide whether it is appropriate to transfer knowledge across domains.

If the $SDD(D_S, D_T)$ is greater than ε, it means that the distance between the target data distribution and source data distribution is too large, and we can not transfer knowledge from the source domain to the target domain. Thus, choosing a proper ε is critical in the proposed *TR-CEN* framework. We optimize the parameter ε through an exhaustive grid-based search in a cross-validated setting. Since the optimal choice

of the ε value varies by the problem domain, we empirically chose the ε to be $\frac{\|\Omega\|_1}{10}$ that is the 10 percent of the L_1 norm of the base model in our implementation of the *TR-CEN* framework. This reduces the heuristic parameter setting for the proposed work. $|\beta(D_S) - \beta(D_T)|$ is a column vector where each element is the absolute value of the difference. $|\beta(D_S)|$ and $|\beta(D_T)|$ represents a non-negative vector, where each element is the corresponding absolute value of the coefficient in the source and target, respectively. Using $|\beta(D_S)|$, $|\beta(D_T)|$, and $|\beta(D_S) - \beta(D_T)|$ we can select the common (hidden) features which meet the following two criteria:

- Conditional on the learning task, the selected features which have large absolute value of coefficients both in $\beta(D_S)$ and $\beta(D_T)$ are important.
- The supervised distribution difference of the selected features between D_S and D_T is relatively small.

Thus, using the selected features, the knowledge can be transferred from the source domain to the target domain in order to improve the prediction performance on the target task. The flowchart of the proposed framework is shown in Fig. 1.

3.3 TR-CEN algorithm

Algorithm 1 outlines the *TR-CEN* method for generating a transfer learning model based on the constrained elastic net. In lines 1–2, we build the elastic net model Ω

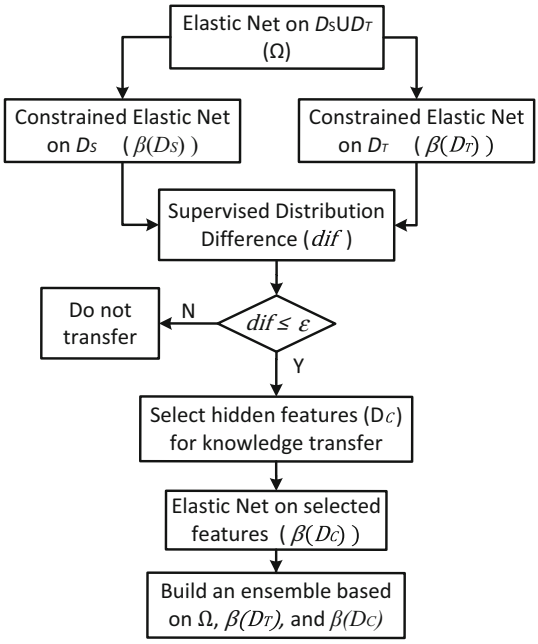

Fig. 1 Flowchart for the *TR-CEN* method

Algorithm 1 *TR-CEN*

Require: Source data (D_S), Target data (D_T), Threshold for knowledge transfer (ε)
1: Learn EN model Ω from $D_S \cup D_T$
2: Learn CEN models $\beta(D_S)$ from D_S, $\beta(D_T)$ from D_T
3: $dif \leftarrow SDD(D_S, D_T)$
4: **if** $dif > \varepsilon$ **then**
5: **return** do not transfer
6: **else**
7: Set the lower boundary of the feature significance and the upper boundary of the feature difference.
8: Select features from D_S and D_T to create a constrained features dataset D_C.
9: Learn EN model $\beta(D_C)$ from D_C.
10: Construct a linear combination of the predicted ensemble outputs from $\beta(D_C)$, Ω and $\beta(D_T)$.
11: Learn the weights in the ensemble using exhaustive search to optimize for the best AUC.
12: **end if**

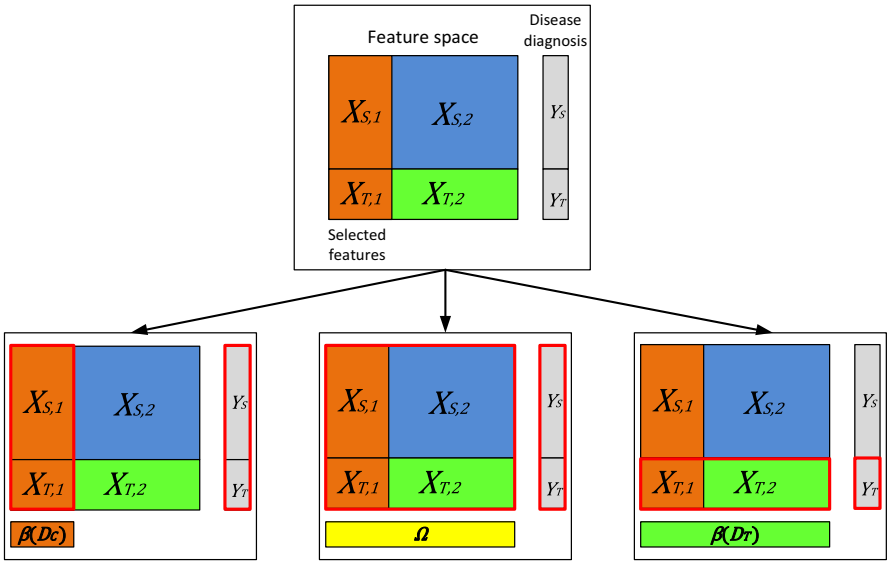

Fig. 2 Ensemble creation in *TR-CEN*

for the unified dataset ($D_S \cup D_T$) and CEN models $\beta(D_S)$ and $\beta(D_T)$ separately from D_S and D_T. Based on the SDD between D_S and D_T we decide whether it is appropriate to transfer knowledge across domains (lines 3–6). In lines 7–8, we set the lower boundary of the feature significance and the upper boundary of the feature difference, and the subset of features which are shared across the domains can be selected based on this constraint. The final output of this algorithm is a model which is a linear combination of three different models (lines 10–11). Two of these models are elastic net models on the combined dataset and the selected common feature space; the other one is CEN model on the target dataset. The relationship of the components of the combined model is clearly illustrated in Fig. 2. As shown in this figure, we learn the models represented by Ω, $\beta(D_T)$, and $\beta(D_C)$ from different parts of the entire dataset (shown using the red boxes).

4 Constrained elastic net

4.1 Preliminaries

In most of the real-world healthcare applications, the number of features (p) is almost equivalent to or even larger than the number of objects (n); it is unnecessary or even incorrect to fit the prediction model with all the features because of the overfitting issue. The primary motivation of using sparsity inducing norms is that in high dimensions, it is appropriate to proceed with the assumption that most of the attributes are not considered to be important, and hence only the vital features can be used for building the predictive models (Hastie et al. 2001; Ye and Liu 2012). In general, the classification and regression models can be built using an optimization-based problem formulation, given as follows:

$$arg \min_{\beta} L(\beta) + P(\beta) \tag{2}$$

where β is the parameter that will be learned from the training dataset, $L(\beta)$ is the empirical loss function, and $P(\beta)$ is the penalty term. Consider the L_P-norm penalty; the smaller the P that is chosen, the sparser the solution, but when $0 \leq P < 1$, the penalty is not convex, and the solution is difficult to obtain. In addition, the penalized methods have also been used to do feature selection in the $n > p$ scenario.

Among all the standard L_P-norm penalties, the Lasso (Tibshirani 1996) and elastic net (Zou and Hastie 2005) are the two most popular penalties which can induce sparsity in the regression coefficients. Lasso or the L_1-norm penalty can be formulated as:

$$P(\beta)_{\text{Lasso}} = \lambda \sum_{k=1}^{p} |\beta_k| \tag{3}$$

where λ is the regularization parameter to control the influence of the penalty. Elastic net is the combination of the L_1-norm and squared L_2-norm penalties which can obtain both sparsity and handle correlated feature spaces simultaneously. It is mathematically defined as follows:

$$P(\beta)_{\text{elastic net}} = \lambda \left(\alpha \sum_{k=1}^{p} |\beta_k| + \frac{1}{2}(1-\alpha) \sum_{k=1}^{p} \beta_k^2 \right) \tag{4}$$

where the $\lambda \geq 0$ is the Lagrange scalar, and $0 \leq \alpha \leq 1$ is used to adjust the weights of the L_1 and L_2 norm penalties.

4.2 Constrained elastic net (CEN)

We now develop the constrained elastic net, which can be used to select the common features to minimize data distribution divergence between source domain and target

domain. In this section, we will also introduce an efficient method to estimate the coefficient parameters of the regression problem. We define the CEN penalty as:

$$P(\beta)_{\text{CEN}} = \lambda_1\|\beta\|_1 + \lambda_2\|\Omega - \beta\|_2^2 \tag{5}$$

where Ω is the coefficient value learned by applying the standard elastic net on the combination of the source data and target data ($D_S \cup D_T$). Using the residual sum of squares (RSS) as the loss function, the penalized loss function will be shown as follows:

$$\mathscr{L}(\lambda_1, \lambda_2, \beta) = \|Y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\Omega - \beta\|_2^2 \tag{6}$$

and the regression coefficient $\hat{\beta}$ can be estimated by minimizing the following objective function

$$\hat{\beta} = arg\min_{\beta}\{\mathscr{L}(\lambda_1, \lambda_2, \beta)\} \tag{7}$$

Obviously, the only difference between the proposed penalty and the standard elastic net is introducing the base model Ω; by doing so, we ensure that in a certain range the estimated parameter (model) β will be as similar as possible to the original unconstrained model Ω.

In the case of obtaining an orthogonal solution $X^T X = \mathbf{I}$, it is straightforward to show that with parameters λ_1, λ_2 the solution of CEN is

$$\hat{\beta}_i(\text{CEN}) = \frac{S(\hat{\beta}_i(OLS),\ \lambda_1/2)}{1+\lambda_2} + \frac{\lambda_2 \cdot \Omega}{1+\lambda_2} \tag{8}$$

where $\hat{\beta}_i(OLS) = X^T Y$ (OLS stands for Ordinary Least Squares). $S(Z,\ \gamma)$ is the soft-thresholding function which can be calculated based on $sign(Z) \cdot (|Z| - \gamma)_+$, where $sign\{\cdot\}$ is the signum function, and $(|Z| - \gamma)_+$ refers to the positive part, which equals to $|Z| - \gamma$ if $(|Z| - \gamma) > 0$ and 0 otherwise.

It should be noted that, in this work, Eq. (8) is the only one which makes the orthogonality assumption. The reason being that it will provide the mathematical formulation of the simplest possible solution for the naive constrained elastic net. Making this assumption aids in comparing the difference between the naive constrained elastic net solution and the naive elastic net solution.

When we build the constrained model for D_S (or D_T), Ω is being introduced in the objective function; thus, compared with the standard elastic net for D_S (or D_T), we learn a relatively more generalized model, but specific to D_S (or D_T) the learned parameter might do worse than the standard model. Here, we choose AUC rather than any other evaluation metrics because AUC can provide a comprehensive measure for the model performance, and it is independent of the cut-off theshold that needs to be chosen for other metrics such as accuracy. For implementation, we use the standard glmnet package (Friedman et al. 2010) to estimate the standard elastic net model. The CEN solves the optimization problem which is given as follows:

$$arg\min_{\beta} \frac{1}{2n}\sum_{i=1}^{n}(y_i - X_i\beta)^2 + \lambda\left[\alpha\sum_{k=1}^{p}|\beta_k| + \frac{1}{2}(1-\alpha)\sum_{k=1}^{p}(\Omega_k - \beta_k)^2\right] \quad (9)$$

We utilize the coordinate descent for solving the optimization problem presented in Eq. (9). Coordinate descent (Donoho and Johnstone 1994) is based on the idea that the minimization of a multi-variable function can be achieved by minimizing it along one direction at a time. Authors in Tseng (2001) analyzed the convergence of the coordinate descent, and proved that we can use coordinate descent to find a minimum of the functional form $f(x) = \sum_{i=1}^{n} h_i(x_i) + g(x)$ if $g(x)$ is convex and differentiable and each $h_i(x_i)$ is convex. In Friedman et al. (2010), coordinate descent has been successfully used to estimate the elastic net coefficient vector. Let us suppose, except β_k, all other $\tilde{\beta}_l$ ($l = 1, 2, 3, \ldots, p$ and $l \neq k$) have already been estimated, and we would like to partially optimize with respect to β_k. Denote the objective function of the optimization problem by $\mathscr{L}(\beta)$; if $\tilde{\beta}_k > 0$, the partial derivative of $\mathscr{L}(\beta)$ with respect to the variable β_k can be calculated as:

$$\frac{\partial\mathscr{L}(\beta)}{\partial\beta_k} = -\frac{1}{n}\sum_{i=1}^{n} x_{ik}(y_i - X_i\tilde{\beta}) + \lambda(1-\alpha)\beta_k - \lambda(1-\alpha)\Omega + \lambda\alpha \quad (10)$$

For both the $\tilde{\beta} < 0$ and $\tilde{\beta} = 0$ cases, a similar expression can be calculated. For simplicity, we can standardize the input data X, and the coordinate-wise will be updated as the following form:

$$\tilde{\beta}_k \leftarrow \frac{S\left(\frac{1}{n}\sum_{i=1}^{n} x_{ik}\left(y_i - \tilde{y}_i^{(k)}\right),\ \lambda\alpha\right)}{1+\lambda(1-\alpha)} + \frac{\lambda(1-\alpha)\cdot\Omega_k}{1+\lambda(1-\alpha)} \quad (11)$$

where $\tilde{y}_i^{(k)} = \sum_{l\neq k} x_{il}\tilde{\beta}_l$ is the fitted value excluding the contribution from x_{ik}. This is simply the univariate regression coefficient of the partial residual sum of squares $y_i - \tilde{y}_i^{(k)}$ on the k^{th} variable. In each iteration, all of the p coefficient variables are repeatedly updated until convergence.

Algorithm 2 outlines our approach for generating the CEN model. First we use the *glmnet* package to learn a standard elastic model, and evaluate its performance using the AUC metric, which is denoted by auc_{EN} (lines 1-2). In line 3, we initialize the estimator of the parameter $\hat{\beta}$ to that of the parameter from Ω and by doing so, we can ensure that the estimator would be close to the base model Ω. In lines 5–7, each element of the coefficient vector is updated using the coordinate-wise update as shown in Eq. (11). After all the p coefficients are updated, we evaluate the current model performance using AUC (auc_{CEN}); if the current model $\hat{\beta}$ meets the requirement of the threshold for the AUC loss, we stop the learning process and output this CEN model β_{CEN} (lines 8–12).

5 Experimental results

In this section, we demonstrate the performance of the proposed transfer learning approach using real-world electronic health records of diabetes patients. We first

Algorithm 2 Constrained elastic net

Require: Feature space(X), Label space (Y), Threshold for AUC loss ($\tau = 0.02$), Elastic net model on combined dataset (Ω)

Learn a standard elastic net model β_{EN} based on X and Y
$\text{auc}_{EN} \leftarrow AUC(Y, X\beta_{EN})$
Initialize $\hat{\beta} \leftarrow \Omega$
repeat
  for $k = 1\ to\ p$ **do**
    $\tilde{\beta_k} \leftarrow \frac{S(\frac{1}{n}\sum_{i=1}^{n} x_{ik}(y_i - \tilde{y}_i^{(k)}),\ \lambda\alpha)}{1+\lambda(1-\alpha)} + \frac{\lambda(1-\alpha)\cdot\Omega_k}{1+\lambda(1-\alpha)}$
  end for
  $\hat{\beta} \leftarrow \tilde{\beta}$
  $\text{auc}_{CEN} \leftarrow AUC(Y, X\hat{\beta})$
until $\text{auc}_{EN} - \text{auc}_{CEN} \leq \tau$
$\beta_{CEN} \leftarrow \hat{\beta}$
Output: parameter of CEN β_{CEN}

present the clinical feature transformation performed on the EHR data. We will then compare the performance of our proposed model against the standard elastic net and multi-task learning algorithms (Zhou et al. 2012). In addition, we also show that there is a strong relationship between the SDD measure and the performance improvement on the target task.

5.1 Experimental setup

The type 2 diabetes dataset we used in our experiment was collected by the Practice Fusion (Practice Fusion Diabetes Classification 2012). There are a total of 9,948 patients in the training set, and among them 1,904 patients were diagnosed with diabetes. These patients are from all the 50 states, the District of Columbia in the U.S. and the Commonwealth of Puerto Rico; for each patient, the EHRs are comprehensively collected from 17 different resources and can be categorized into the following sources of information:

- *Demographic information* such as year of birth, gender, weight, and location of each patient.
- *Diagnosis information* consists of the ICD9 Codes collected for each patient during the Practice Fusion's program.
- *Allergy and immunization* consists of a list of allergies and vaccination records for each patient after they joined the Practice Fusion's program.
- *Laboratory information* consists of lab test observations for lab panels, and the lab test results received from lab facilities.
- *Medication and Prescription* consists of the medication history and prescription records for each patient after he joined the Practice Fusion's program.
- *Patient smoking status* is a binary status variable maintained for each patient on a yearly basis.
- *Transcript* consists of visit document records for a patient including the allergy, medication, and diagnosis information provided by the patients when they joined the Practice Fusion's program.

PtID	Lab Name	Lab Value	Height	SBP	ICD9	Medication	Smoking Status	Visiting Year
884	TSH	1.98	64.5	122	300	LS	Few	2009
884	TSH	1.37	64.5	106	477.9	LS	Few	2010
884	TSH	1.99	64.5	124	307.81	LS	Few	2011
884	TSH	1.18	64.5	90	724.5	BDS	Few	2012
884	TSH	2.4	64.5	108	625.4	BDS	Few	2012

Feature Representation

PtID	TSH max	TSH min	TSH avg	TSH count	SBP max	SBP min	SBP avg	height	anxiety	rhinitis	syndrome	backache	LS	BDS	Smoking status
884	2.4	1.18	1.78	5	124	90	110	64.5	2	1	1	1	3	2	1

Fig. 3 Feature representation for a single patient

We integrated these 17 different files and generated 536 features for each patient. Among these features, only the gender and location are categorical attributes; lab test results and personal information are real-valued attributes; some attributes such as smoking status, and emergency status are ordinal. A significant majority of the remaining features are count variables.

5.2 Clinical feature representation

We now explain the clinical feature transformation which exploits the semantics of EHRs. The original data contains EHRs for 9,948 patients. In Fig. 3, we explain the feature creation procedure by considering a simple example. In this example, we use a set of 5 records for a particular patient (with patientID 884). We extracted all the features for all the distinct lab variables present in the data. In this example, only the TSH lab variable is considered. To tackle the problem of multiple lab values for the same patient, we represent each lab using a set of summary statistics (maximum, minimum, and average). We compute values for these statistics over the 5 records (2.4, 1.18 and 1.78). In addition to these statistics for each distinct lab, we also create a variable which counts the number of times the lab was conducted for the patient (represented using the variable Tcount). Several anthropometric features such as systolic blood pressure (SBP) are generated by using a similar method as done in lab feature representation. Height is an example variable being considered in the demographics information. ICD-9 codes, is a list code for International Statistical Classification of Diseases, and each code presents a disease description. With the ICD-9 codes, a binary variable can be created to reflect if the patient has a special disease or not. The code of 300 and 307.81 can be combined since they can be considered to be anxiety. The code of 477.9, 625.4, 724.5 can represent the disease rhinitis, syndrome and backache

Table 2 Demographic statistics of the top 14 states based on patient population

State	Total	Diabetic	Nondiabetic	Prevalence rate
CA	1,917	258	1,659	0.1346
TX	897	243	654	0.2709
FL	804	158	646	0.1965
MO	702	104	598	0.1481
NJ	575	119	456	0.2070
NY	557	167	390	0.2998
OH	515	111	404	0.2155
NV	495	135	360	0.2727
VA	484	79	405	0.1632
IL	412	75	337	0.1820
MI	344	61	283	0.1773
SD	325	40	285	0.1231
AZ	295	56	239	0.1898
PA	250	67	183	0.2680

separately. For the procedures, we create variables for each distinct medications given to the patient. This feature represents the number of times the individual procedures were conducted for the patient. In Fig. 3, two new variables for each of the medication, namely, Levothyroxine Sodium (LS) and Bactrim DS oral tablet (BDS) are created. There are several stages of smoking status which can be represented using ordinal numbers as shown in Fig. 3. The status of "few (1–3) cigarettes per day" (Few) is denoted by 1. In summary, it can be seen that following this procedure immensely helps not only reduce the dimensionality and the complexity of the problem, but also summarize the complex EHRs into a succinct representation which is then used for disease diagnosis.

5.3 Goodness of fit

In this experiment, our aim is to improve the diabetes diagnosis for a particular state. Hence, in the transfer learning setting, this specific state would be our target domain and the data from remaining states is considered to be the source domain. Table 2 shows the demographic statistics of the top 14 states based on the total patient population. In our experiment, for these 14 states, each of them will be considered as the target domain and the rest of the population will be considered as the source domain.

We compared our proposed transfer learning model, *Tr-CEN*, with the standard elastic net, multi-task Lasso (*Multi-LASSO*) and multi-task feature learning method (*Multi-*$L_{2,1}$) proposed in Liu et al. (2009). We would first provide a brief description of the *Multi-LASSO* and *Multi-*$L_{2,1}$ optimization formulation. *Multi-LASSO* (Zhou et al. 2012) is a multitask extension of the elastic net[1], and with least squares loss it can be formulated as:

[1] Although in Zhou et al. (2012) it has been named as multi-task Lasso, both L_1-norm and L_2-norm penalties are used in the optimization formulation.

Table 3 Comparison of AUC values of *Local-EN*, *All-EN*, *Multi-LASSO*, *Multi-*$L_{2,1}$, and *Tr-CEN*

State	*Local-EN*	*All-EN*	*Multi-LASSO*	*Multi-*$L_{2,1}$	*Tr-CEN*
CA	0.7675	0.8679	0.8617	0.868	0.8908
TX	0.7096	0.8442	0.8296	0.8343	0.8586
FL	0.6551	0.8504	0.8694	0.8749	0.894
MO	0.6499	0.8797	0.8824	0.8921	0.9183
NJ	0.7096	0.7826	0.7595	0.7854	0.8177
NY	0.7672	0.8177	0.825	0.8329	0.8512
OH	0.8137	0.8688	0.8464	0.8565	0.9184
NV	0.7967	0.8324	0.8466	0.8553	0.876
VA	0.6734	0.8078	0.7667	0.7806	0.8488
IL	0.8482	0.8575	0.8606	0.8675	0.9128
MI	0.8265	0.8993	0.888	0.9048	0.9545
SD	0.7652	0.9132	0.749	0.8122	0.9744
AZ	0.6702	0.8057	0.7722	0.7889	0.8862
PA	0.6424	0.7811	0.7185	0.7589	0.8122

$$arg \min_{B} \sum_{i=1}^{t} \|Y_i - XB\|_2^2 + \lambda_1 \|B\|_1 + \lambda_2 \|B\|_2^2 \tag{12}$$

where B is a $p \times t$ coefficient matrix for t tasks. With the least squares loss, the *Multi-*$L_{2,1}$ has the following form (Evgeniou and Pontil 2007):

$$arg \min_{B} \sum_{i=1}^{t} \|Y_i - XB\|_2^2 + \lambda \|B\|_{2,1} \tag{13}$$

where the $L_{2,1}$-norm of B is defined as $\|B\|_{2,1} = \sum_{k=1}^{p} \|B^k\|_2$.

Each of the algorithms is validated using 10-fold cross validation. In our experiments, the standard elastic net is applied both on the target dataset (the specified state) and the entire dataset. For simplicity, they are referred to as *Local-EN* and *All-EN*, respectively. It should be noted that, in *All-EN*, although the model is built on the entire dataset, the performance is measured only on the target dataset; in other words, the *All-EN* reflects the performance of the base model Ω in each specific state. In Table 3, we provide the AUC values to assess the goodness of fit. In Table 4, we present the sensitivity (True Positive Rate) values. In addition, in Table 5, we present the comparison of *F_measure* values of *Local-EN*, *All-EN*, *Multi-LASSO*, $Multi - L_{2,1}$, and *Tr-CEN*. The *F_measure* can be calculated as follows:

$$F\_measure = \frac{2 \times Precision \times Sensitivity}{Sensitivity + Precision} \tag{14}$$

where $Sensitivity = \frac{TP}{TP+FN}$, and $Precision = \frac{TP}{TP+FP}$; therefore, a high value of *F_measure* indicates that both precision and sensitivity are reasonably high.

Table 4 Comparison of sensitivity values of *Local-EN*, *All-EN*, *Multi-LASSO*, *Multi-$L_{2,1}$*, and *Tr-CEN*

State	*Local-EN*	*All-EN*	*Multi-LASSO*	*Multi-$L_{2,1}$*	*Tr-CEN*
CA	0.686	0.7636	0.7636	0.7907	0.8538
TX	0.6914	0.8313	0.7737	0.7613	0.8378
FL	0.6962	0.7595	0.8165	0.7975	0.8396
MO	0.625	0.8462	0.8173	0.8077	0.8674
NJ	0.6975	0.7647	0.7647	0.7731	0.8333
NY	0.6707	0.7605	0.7605	0.7844	0.7981
OH	0.7297	0.8198	0.7748	0.8468	0.8681
NV	0.6963	0.8	0.8	0.8222	0.7903
VA	0.6076	0.7595	0.6329	0.6709	0.8092
IL	0.8	0.8	0.7867	0.8133	0.8191
MI	0.8	0.8197	0.7869	0.8525	0.9128
SD	0.7	0.85	0.6	0.65	1
AZ	0.625	0.75	0.7143	0.7143	0.875
PA	0.6567	0.6866	0.625	0.7612	0.7403

Table 5 Comparison of *F_measure* values of *Local-EN*, *All-EN*, *Multi-LASSO*, *Multi-$L_{2,1}$*, and *Tr-CEN*

State	*Local-EN*	*All-EN*	*Multi-LASSO*	*Multi-$L_{2,1}$*	*Tr-CEN*
CA	0.4009	0.49	0.4817	0.4798	0.5624
TX	0.5239	0.6413	0.646	0.6238	0.6924
FL	0.4313	0.5854	0.5917	0.6327	0.6891
MO	0.4013	0.5483	0.5414	0.5524	0.6722
NJ	0.4641	0.517	0.4946	0.5014	0.6383
NY	0.6074	0.6632	0.648	0.6313	0.7024
OH	0.5744	0.589	0.5695	0.5938	0.7242
NV	0.5943	0.6429	0.6526	0.6667	0.6986
VA	0.3852	0.4878	0.4386	0.4545	0.626
IL	0.5588	0.5286	0.554	0.5488	0.6974
MI	0.5535	0.5952	0.5926	0.6012	0.7974
SD	0.4528	0.5271	0.5106	0.4425	0.8591
AZ	0.4457	0.549	0.5132	0.5298	0.7144
PA	0.4868	0.6093	0.5325	0.5548	0.6353

We observe that for all the 14 states our proposed algorithm provides a better fit compared to the other algorithms. This demonstrates the effectiveness of our approach in real-world clinical setting.

5.4 Relationship between SDD and model performance

Compared with other existing transfer learning methods, another advantage of our proposed *Tr-CEN* is that, even before doing the actual knowledge transfer we can

Table 6 Relationship between the SDD and performance improvement compared to the base models

State	AUC of *All-EN*	AUC of *Tr-CEN*	AUC improvement	$SDD(D_S, D_T)$
TX	0.8442	0.8586	0.0144	63.2673
FL	0.8504	0.894	0.0436	29.4815
MO	0.8797	0.9183	0.0386	39.6149
NJ	0.7826	0.8177	0.0351	40.9582
NY	0.8177	0.8512	0.0335	38.8797
OH	0.8688	0.9184	0.0496	14.2148
NV	0.8324	0.876	0.0436	26.4018
VA	0.8078	0.8488	0.041	24.0069
IL	0.8575	0.9128	0.0553	17.0574

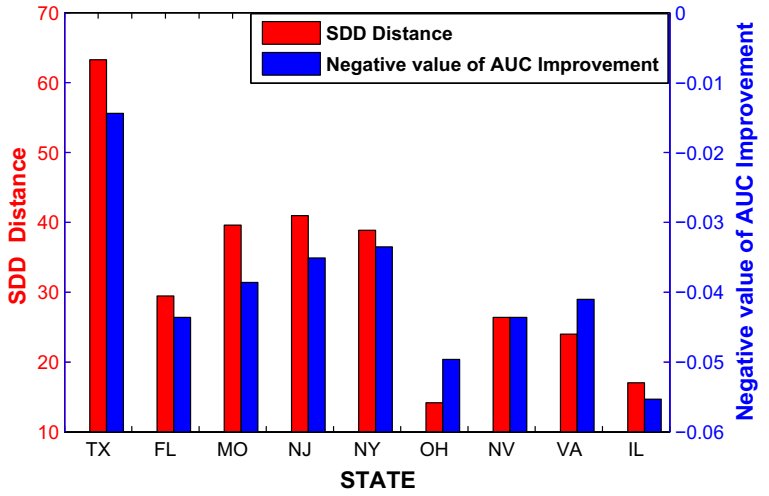

Fig. 4 Illustration of the relationship between the SDD and performance improvement

predict the improvement of performance in the target task based on the SDD measure between the source data and target data. In Table 3, we can see that, amongst the two base models *Local-EN* and *All-EN*, *All-EN* performs better. In Table 6, we select top 10 states (excluding CA) to present the AUC improvement between the *Tr-CEN* and *All-EN*, and the $SDD(D_S, D_T)$ for each state. We excluded CA because the patient population in California is relatively too large compared with the remaining 9 states. The results in this table show that there is a strong negative correlation between the quantity of AUC improvement and the SDD distance. We also illustrate this relationship in Fig. 4. Note that, for convenience, we take the negative of the AUC improvement value as one of the *Y*-axis. Thus, in Fig. 4, we can see the height of the red bar (SDD distance) and the blue bar (Opposite value of AUC improvement) are positively correlated to each other.

6 Conclusions and future work

Healthcare in the United State is currently undergoing a revolutionary transformation. One of the important components of this transformation is the healthcare information exchange wherein the primary objective is to share knowledge in an appropriate and adequate manner. Such sharing of knowledge between different institutions should potentially provide better predictive models which can then help in accurate diagnosis of diseases even in the presence of limited data from the local institution. Transfer learning, which is a subfield within machine learning, has not been studied in the context of healthcare applications. In this transfer learning paradigm, an important aspect of "when to transfer" has not been thoroughly investigated. This aspect becomes critical in healthcare applications due to the presence of abundant noisy information datasets which can potentially be used as sources to transfer knowledge. In this paper, we develop a novel transfer learning framework based on constrained sparse predictive model which can select a low-dimensional subset of common features to transfer knowledge from source domain to target domain and simultaneously measure the data distribution difference between source and target dataset. This CEN model is built by enforcing additional constraints on the standard elastic net. We demonstrate the performance of the proposed algorithms using real-world diabetes EHRs data. We showed that the distance between the source dataset and target dataset obtained from the proposed models can be used to predict the improvement of performance in the target task.

In the future, we will extend the proposed constrained modification on other predictive models. We plan to develop more accurate methods to measure the "transferability" between source domain and target domain to prevent negative transfer. We also plan to study the issue that if an entire domain leads to a negative transfer, whether we can potentially select a partial component of the source domain to improve the prediction performance on the target task. Most importantly, we will also apply this model on other healthcare problems.

Acknowledgments This work was supported in part by NSF Grants IIS-1242304, IIS-1231742 and NIH Grant R21CA175974.

References

Arnold A, Nallapati R, Cohen WW (2007) A comparative study of methods for transductive transfer learning. In: Seventh IEEE international conference on data mining workshops, 2007. ICDM Workshops 2007, p 77–82

Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. ACL 7:440–447

Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75

Dai W, Yang Q, Xue G, Yu Y (2007) Boosting for transfer learning. In: ICML'07: Proceedings of the 24th international conference on Machine learning, p 193–200

Dai W, Yang Q, Xue GR, Yu Y (2008) Self-taught clustering. In: Proceedings of the 25th international conference on machine learning, ACM, p 200–207

Donoho DL, Johnstone JM (1994) Ideal spatial adaptation by wavelet shrinkage. Biometrika 81(3):425–455

Evgeniou A, Pontil M (2007) Multi-task feature learning. In: Proceedings of the 2006 conference on advances in neural information processing systems, vol. 19. The MIT Press, Cambridge, p 41

Evgeniou T, Pontil M (2004) Regularized multi-task learning. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, p 109–117
Farhadi A, Forsyth D, White R (2007) Transfer learning in sign language. In: IEEE Conference on computer vision and pattern recognition, CVPR'07, IEEE, p 1–8
Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–22
Fung GPC, Yu JX, Lu H, Yu PS (2006) Text classification without negative examples revisit. IEEE Trans Knowl Data Eng 18(1):6–20
Hastie T, Tibshirani R, Friedman JJH (2001) The elements of statistical learning. Springer, New York
Liu J, Ji S, Ye J (2009) Multi-task feature learning via efficient l 2, 1-norm minimization. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. AUAI Press, Corvallis, p 339–348
Mihalkova L, Mooney RJ (2008) Transfer learning by mapping with minimal target data. In: Proceedings of the AAAI-08 workshop on transfer learning for complex tasks
Pan J (2010) Feature-based transfer learning with real-world applications. Ph.D. thesis, The Hong Kong University of Science and Technology
Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
Pan SJ, Zheng VW, Yang Q, Hu DH (2008) Transfer learning for wifi-based indoor localization. In: Association for the advancement of artificial intelligence (AAAI) workshop, p 6
Practice Fusion Diabetes Classification: Identify patients diagnosed with Type 2 Diabetes (2012). https://www.kaggle.com/c/pf2012-diabetes
Raina R, Battle A, Lee H, Packer B, Ng AY (2007) Self-taught learning: transfer learning from unlabeled data. In: Proceedings of the 24th international conference on Machine learning, ACM, p 759–766
Rosenstein MT, Marx Z, Kaelbling LP, Dietterich TG (2005) To transfer or not to transfer. In: NIPS 2005 workshop on inductive transfer: 10 years later, vol. 2, p 7
Rückert U, Kramer S (2008) Machine learning and knowledge discovery in databases., Kernel-based inductive transferSpringer, Heidelberg, pp 220–233
Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc B 58(1):267–288
Tseng P (2001) Convergence of a block coordinate descent method for nondifferentiable minimization. J Optim Theory Appl 109(3):475–494
Ye J, Liu J (2012) Sparse methods for biomedical data. ACM SIGKDD Explor Newslett 14(1):4–15
Zhou J, Chen J, Ye J (2012) Malsar: multi-task learning via structural regularization. Arizona State University, Phoenix
Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc B 67(2):301–320