An output-based knowledge transfer approach and its application in bladder cancer prediction

Guanjin Wang[1,2], Guangquan Zhang[1], Kup-Sze Choi[2], Kin-Man Lam[3], and Jie Lu[1]

[1]Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Broadway, NSW, 2007, Australia
Email: Guanjin.Wang@student.uts.edu.au, Jie.Lu, Guangquan.Zhang@uts.edu.au
[2]Centre for Smart Health, School of Nursing,
The Hong Kong Polytechnic University, Hong Kong, China
Email: thomasks.choi@polyu.edu.hk
[3]Department of Surgery, Tseung Kwan O Hospital, Hong Kong, China
Email: lkm154@ha.org.hk

***Abstract*—Many medical applications face a situation that the on-hand data cannot fully fit an existing predictive model or on-line tool, since these models or tools only use the most common predictors and the other valuable features collected in the current scenario are not considered altogether. On the other hand, the training data in the current scenario is not sufficient to learn a predictive model effectively yet. In order to overcome these problems and construct an efficient classifier, for these real situations in medical fields, in this work we present an approach based on the least squares support vector machine (LS-SVM), which utilizes a transfer learning framework to make maximum use of the data and guarantee its enhanced generalization capability. The proposed approach is capable of effectively learning a target domain with limited samples by relying on the probabilistic outputs from the other previously learned model using a heterogeneous method in the source domain. Moreover, it autonomously and quickly decides how much output knowledge to transfer from source domain to the target one using a fast leave-one-out cross validation strategy. This approach is applied on a real-world clinical dataset to predict 5-year mortality of bladder cancer patients after radical cystectomy, and the experimental results indicate that the proposed method can achieve better performances compared to traditional machine learning methods, consistently showing the potential of the proposed method under the circumstances with insufficient data.**

***Index Terms*—transfer learning, machine learning, support vector machine, cancer prediction**

I. Introduction

In medical fields, it is very common that clinical assessments from different hospitals or pathologists are inconsistent (up to 30% in many cases) [1]. Therefore, in some cases the existing predictive softwares or on-line tools are constructed based on a smaller set of features than those found in real world datasets. Even though some existing models use the most universal and highly reproducible features such as age, gender, tumour stage, etc., it is important to note that as time passes the importance or relevance of these clinical measures might change, and thus the predictive model needs to adapt to different feature sets accordingly.

Another big issue in many medical applications is a lack of data. If we want to construct a predictive model on a specific feature sets instead of using existing models with fixed features, the performance might deteriorate a lot due to insufficient training samples in the current domain. In particular, the process of collecting labelled data in the real world may perhaps be time consuming and/or expensive. For example, patients do not wish their medical records to be exposed to others, because this might cause them to become depressed, affect their employment and insurance coverages, etc. Additionally, because the training sample size is small, it is necessary to employ cross-validation as an unbiased estimation of the classification errors for the trained model. However, how to greatly reduce the high computational complexity of the cross-validation procedure triggers another issue worthy to be studied.

In order to solve the above issues, transfer learning is considered such that the knowledge from a related but different domain (source domain) can be leveraged to construct a predictive model on the current domain of interest (target domain) with few data.

There are two types of transfer learning when source and target domains are different but aim to achieve the same task. a). The feature space between domains are the same, but the data distributions of the inputs are different. b). The feature spaces between domains are different, which is also known as heterogeneous transfer learning. Apparently our problem fits in the latter situation. For example, suppose our task is to diagnose a specific disease into a 'yes' or 'no' class, given that we could only obtain a limited amount of patient records from a local hospital. Moreover, suppose that there is an on-line diagnosis tool of the same disease which is based on a subset of the feature space in the dataset obtained from the local hospital. Comparatively the on-line tool has been constructed by many more training examples with fewer features than these of the dataset obtained from the local hospital. In this case, we can model the classification task using few labelled data from the local hospital as the current domain, and the existing on-line predictive model as

the knowledge of the source domain. We ask: is it possible to use the auxiliary knowledge from the source domain model to help improve the classification performance using a heterogeneous method on the diagnosis of local patients in the target domain?

Ideally the subset of the robust features are shared in both source and target domains, and an extra subset of the unique features are contained in the target domain. In this way, the source and target data form an inverted pyramid dataset, as shown in Fig. 1. Due to the obvious commonality between source and target data, the outputs from predictive models in both domains should remain similar to a certain extent. Therefore the output-based transfer learning across domains is feasible to guide a better classification performance in the target domain. In this work, we propose a novel output-based transfer least square support vector machine (LS-SVM) [2] from the transfer learning perspective, which can effectively leverage the probabilistic output knowledge from the existing predictive model built using a heterogeneous method to the target domain for classification. Moreover, This approach also has the ability to autonomously and quickly determine how much output knowledge to transfer from source domain to target one using the proposed fast leave-one-out cross validation strategy. Our main contributions are:

(1) A novel output-based transfer LS-SVM classifier is proposed for classification on few labelled data, by the means of leveraging knowledge of the probabilistic output from the existing predictive model built by a heterogeneous method.

(2) The proposed approach can autonomously and quickly determine the influence level on the target domain model caused by the probabilistic outputs from the existing predictive model in the source domain, by using a fast leave-one-out cross validation strategy.

(3) The proposed approach under the framework of the LS-SVM can directly handle the probabilistic outputs from a heterogeneous method, which well matches the real situations in medical fields.

(4) Without knowing the details of the existing model in the source domain, we can still achieve transfer learning by leveraging output knowledge from the existing model to help improve generalization performance on the target domain. This is very helpful in real world scenarios where the data and its modelling details are private.

The paper is organized as follows. The related work is introduced in Section II. In Section III the proposed output-based transfer LS-SVM classifier is presented. In particular, a fast leave-one-out cross validation strategy for the choice of parameter is developed as well. In Section V, we give the experimental results on the bladder cancer dataset from the real world. Finally, the conclusions and future work are given in Section VI.

II. Related Work

In transfer learning, there is a very important problem to solve, which is what to transfer [3]. It focuses on which part of knowledge or how much knowledge is planned to leverage across domains. Based on this, transfer learning approaches in literature can be categorized into four types.

The first category is instance transfer, which assumes that certain amount of data in the source domain can be useful for learning in the target domain, via instance re-weighting and importance sampling techniques. For example, in [4], a nonparametric method was proposed to directly obtain resampling weights with no distribution estimation. Xia et al. presented another novel method to re-weight the training instance using in-target-domain probability by positive and unsupervised learning [5].

The second category is feature representation transfer, which aims to learn an appropriate common feature representation for the target domain such that the difference between the source and target domains and the classification error is decreased. The knowledge to transfer across domains is embedded into the common feature representation. For example, in [6], a framework is proposed for common feature and kernel selection in multiple SVMs trained on different but related datasets. Another feature representation method transfer component analysis (TCA) was discovered in [7] such that the distance between domains can be reduced in a latent space for domain adaptation. Duan et al. [8] proposed a method which is able to augment the heterogeneous features from the source and target domains through utilizing two novel feature mapping functions. After that, the SVMs can be used to incorporate with new generated feature representations for classification cross domains with different feature spaces. Zuo et al. developed a method in [9], which uses Stacked Denoising Autoencoder (SDA) to extract multiple feature spaces and the predictive model can be constructed based on every feature space. Also, two fuzzy sets are defined to analyse the variation of the accuracies in these feature spaces in the target task.

The third category is relational knowledge transfer, which assumes that the relationship among the data in the source and target domain is similar to a certain extent. The knowledge to transfer is the relationship among the data. In this context, statistical relational learning techniques are used to solve problems. For example, in [10], Mihalkova et al. proposed Markov logic networks (MLN) based transfer system which maps the predicates in the source MLN to the target domain, and then edits the mapping structure to improve its performance.

The last category is parameter transfer, which assumes that the source and target domains share some parameters or priors of the models. The knowledge to transfer is embedded into the shared parameters or priors. A novel model was proposed in [11] which aims to learn a shared covariance function on input dependent features and a "free-form" covariance matrix among tasks. Schwaighofer et al. [12] presented a method under a hierarchical Bayesian framework to learn a common prior for mean and covariance across domains. In [13], Gao et al. proposed a locally weighted ensemble framework to leverage knowledge from multiple models for transfer learning. The weights are dynamically determined based on every model's predictive ability on

each testing sample. In [14] [15], a knowledge-leverage-based Takagi-Sugeno-Kang fuzzy system (KL-TSK-FS) and an advanced version were proposed for parameter learning of the TSK-FS model on the target domain by utilizing existing knowledge in the source domain.

Among existing transfer learning approaches, the parameter-transfer learning is most related to the proposed output-based transfer LS-SVM classifier here. In general, most approaches aim to find the shared parameters/hyper-parameters of the models. In the proposed classifier, since the existing model in the source domain is unknown, we put our focus on the obtained outputs from it, and aim to learn a weighting parameter on the outputs of the target domain influenced by the predicted outputs of the existing model. The basic assumption behind is that the outputs from the models in both domains should be similar to some extent. This is because that in this study, we only consider the inverted pyramid dataset in which a subset of features are shared in the feature spaces of both source and target domains, such that there is a certain similarity between them. Furthermore we can assume that the outputs from the ideal classifier on the target domain should keep a certain degree of consistency with those from the existing model on the source domain. On the other hand, we also find that there is an extra subset of the unique features only in the target domain in the inverted pyramid dtaset. Therefore, the problem in this study also involves transferring knowledge cross different feature spaces, which is referred as heterogeneous transfer learning [3].

III. Output-based transfer LS-SVM classifier

A. Inverted pyramid dataset

In this work, we denote the data in the target domain as $D_T = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_N, y_N)\}$, where $\boldsymbol{x}_i = (x_1^i, x_2^i, ..., x_d^i) \in \boldsymbol{X_T} \subset \mathbf{R}^d$ and $y_i \in \boldsymbol{Y_T} = \{-1, 1\}$. $\boldsymbol{X}_T$ is the input dataset and $\boldsymbol{Y}_T$ is the corresponding class label set. Each sample $\boldsymbol{x}_l$ contains d features, i.e., f_1,f_2, ..., f_d. Since the existing model in the source domain only contains a subset of the feature space in the target domain, in order to fit the existing model, we project D_T to the data D_S only with the shared feature subset. $D_S = \{(\boldsymbol{x}'_1, y_1), ..., (\boldsymbol{x}'_N, y_N)\}$, where $\boldsymbol{x}'_i = (x_1^i, x_2^i, ..., x_{d'}^i) \in \boldsymbol{X_S} \subset \mathbf{R}^{d'}$ and $y_i \in \boldsymbol{Y_S} = [0, 1]$. $\boldsymbol{X}_S$ is the input dataset in which each sample $\boldsymbol{x}_l$ contains d' features, i.e., f_1,f_2,...,$f_{d'}$, ($d' < d$), and $\boldsymbol{Y}_S$ is the set of the probabilistic outputs obtained from the existing model.

We want to find a decision function $\boldsymbol{F} : \boldsymbol{X}_T \rightarrow \boldsymbol{Y}_T$, such that it can find the matching y for any new incoming sample $\boldsymbol{x}$. Using a diagram to present data, if we stack D_T onto D_S, it shapes like an inverted pyramid. Therefore, we call the adopted dataset in the proposed approach the inverted pyramid dataset and demonstrated them in Fig. 1.

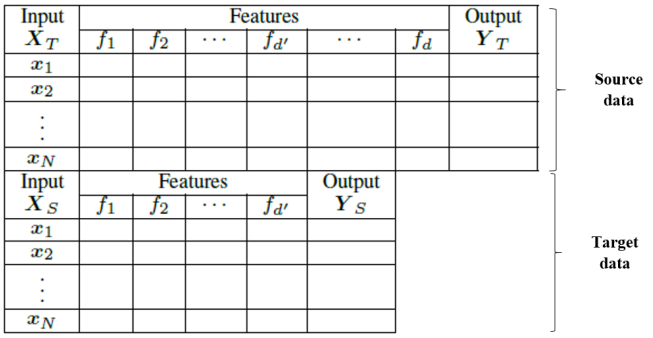

Fig. 1: The inverted pyramid dataset in which $d' < d$

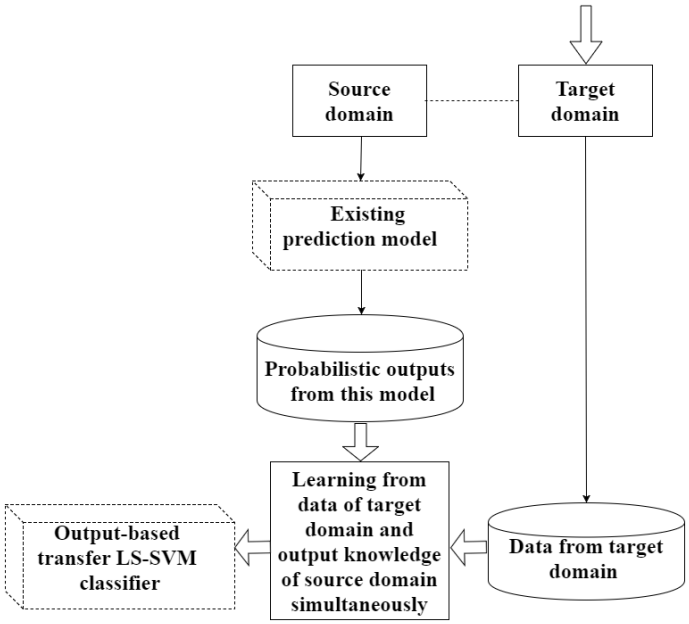

Fig. 2: Framework of the proposed output-based transfer LS-SVM

B. Framework of the proposed classifier

The framework of the proposed output-based transfer LS-SVM classifier is illustrated in Fig. 2. There is an existing predictive model on the source domain which has the probabilistic outputs for inputs. We first obtain the probabilistic outputs of D_S in the inverted pyramid dataset using the existing predictive model and then the proposed classifier not only makes full use of the data from D_T in the inverted pyramid dataset in the learning procedure, but also leverages the output knowledge from the existing model to help classification on the target domain.

C. Handle probabilistic outputs from the existing model

Most existing predictive models or on-line tools in medical fields produce probabilistic outputs by using statistical methods. Considering that the classification on the target domain is achieved using a heterogeneous method and that the representation of the output is different, the proposed classifier is designed to directly handle the probabilistic output learning from the existing model into the target model.

First we put the data from the source domain D_S into the existing predictive model and obtain the corresponding probabilistic outputs p_i ($0 \leq p_i \leq 1$, $i = 1, 2, ..., N$). Then we can regard p_i and $1 - p_i$ as the probability of $\boldsymbol{x}_i$ being classified into positive and negative class respectively. We set a threshold θ to 0.5 such that $\boldsymbol{x}_i$ is classified into

positive class if its output probability is greater than 0.5. For example, suppose that the sample $\boldsymbol{x}_i$ gets the probabilistic output 0.65 from an existing predictive model. This means the probability of $\boldsymbol{x}_i$ being classified into the positive class and the negative class is 0.65 and 0.35 ($1 - 0.65 = 0.35$), respectively. Based on the threshold θ, we classify $\boldsymbol{x}_i$ into the positive class ($0.65 - 0.5 = 0.15 > 0$), rather than negative class ($0.35 - 0.5 = -0.15 < 0$). In other words, since the sign of $(2p_i - 1)$ reflects which class $\boldsymbol{x}_i$ belongs to, it can be used to maintain the consistency between the outputs of the existing model and another heterogeneous method, i.e., LS-SVM. These processed probabilistic outputs are the knowledge we want to explore and effectively leverage onto the target domain for classification.

D. Output-based transfer LS-SVM classifier on the target domain

After handling the probabilistic outputs from the existing predictive model, we can construct a model on the target domain in which the signs of the classification outputs and those of the processed probabilistic outputs from the existing source model keep the same as much as possible.

According to LS-SVM framework, the input $\boldsymbol{x}_i$ can be classified in terms of the decision function:

$$\boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b \begin{cases} > 0 & \text{positive class} \\ < 0 & \text{negative class} \end{cases}$$

Let us recall LS-SVM uses the constraint, $y_i = \boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b + \xi_i$, therefore, in order to keep the signs of both y_i and $(2p_i - 1)$ $(i = 1, 2..., N)$ the same as much as possible, we should make $\sum_{i=1}^{N}(y_i - \xi_i)(2p_i - 1)$ as large as possible.

Moreover, we use a weighting parameter μ to reflect the influence level of the processed probabilistic outputs from the existing predictive model onto the predicted outputs from the target domain. μ is treated as the learning parameter, and selected by the fast leave-one-out cross-validation strategy which will be discussed in the following section. Therefore, the objective function based on the LS-SVM framework becomes

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\boldsymbol{w}^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \mu\sum_{i=1}^{N}(y_i - \xi_i)(2p_i - 1) \\ \text{s.t} \quad & y_i = \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{1}$$

After simple derivations, we can get the following equivalent formulation

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\boldsymbol{w}^2 + \frac{C}{2}\sum_{i=1}^{N}(\xi_i + \frac{\mu}{2C}(2p_i - 1))^2 \\ \text{s.t} \quad & y_i = \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b + \xi_i, i = 1, 2, ..., N \end{aligned} \tag{2}$$

where $\frac{\mu}{2C}$ represents the influence level of the probabilistic outputs from the existing predictive model onto the target domain. We can observe that if we set μ to 0 in Eq. (2), it is the standard LS-SVM objective function. Obviously, Eq. (2) is a QP problem [16].

The Lagrangian J of Eq. (2) is

$$J = \frac{1}{2}\boldsymbol{w}^2 + \frac{C}{2}\sum_{i=1}^{N}(\xi_i + \frac{\mu}{2C}(2p_i - 1))^2 + \sum_{i=1}^{N}\alpha_i(y_i - \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) - b - \xi_i) \tag{3}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N) \in \mathbf{R}^N$ is the vector of all Largrangian multipliers. With respect to $\boldsymbol{w}$, ξ_i, b, α_i, we have

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \quad \Rightarrow \boldsymbol{w} = \sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i) \tag{4}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \xi_i = \frac{1}{C}[\alpha_i - \frac{\mu}{2}(2p_i - 1)] \tag{5}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \sum_{i=1}^{N}\alpha_i = 0 \tag{6}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \quad \Rightarrow y_i = \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b + \xi_i \tag{7}$$

Combining the Eq. (4) and Eq. (5) with Eq. (7), We can get

$$\sum_{j=1}^{N}\alpha_i\varphi(\boldsymbol{x}_j)^T\varphi(\boldsymbol{x}_i) + b + \frac{\alpha_i}{C} = y_i + \frac{\mu}{2C}(2p_i - 1) \tag{8}$$

Using the kernel trick, we replace $\varphi(\boldsymbol{x}_j)\varphi(\boldsymbol{x}_i)$ by $K(\boldsymbol{x}_j, \boldsymbol{x}_i)$, and further write the linear equation in Eq. (8) in matrix form

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{\Lambda} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} + \frac{\mu}{2C}\boldsymbol{M} \\ 0 \end{bmatrix} \tag{9}$$

where $\mathbf{\Lambda}$ is a matrix in which each diagonal entry is one and all other entries are zero, $\boldsymbol{y}$ is the output vector of all the samples in the training dataset, and $\boldsymbol{M} = \big(2p_1 - 1, 2p_2 - 1, ..., 2p_N - 1\big)^T$.

Finally, the model parameters can be calculated simply by using matrix inversion:

$$\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \mathbf{P}\begin{bmatrix} \boldsymbol{y} + \frac{\mu}{2C}\boldsymbol{M} \\ 0 \end{bmatrix} \tag{10}$$

where $\mathbf{P} = \mathbf{V}^{-1}$ and $\mathbf{V}$ is the first matrix on the left in Eq. (9). Once we obtain μ, $\boldsymbol{\alpha}$ and b can be calculated accordingly from Eq. (10).

E. Decision Function

Combined with Eq. (4), the decision function for the new sample $\boldsymbol{x}_t$ becomes

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{x}_t) &= \boldsymbol{w}^T\varphi(\boldsymbol{x}_t) + b \\ &= \sum_{i=1}^{N}\alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_t) + b \end{aligned} \tag{11}$$

IV. Fast leave-one-out cross validation strategy for parameter tuning

From the last section, we know that the classification performance of the proposed classifier depends on the value of the parameter μ. Usually, the traditional leave-one-out cross-validation method has been widely recognized as an unbiased estimator to choose the values for the parameters in various models. However, the procedure is computationally expensive and time-consuming at the same time. In this section, we introduce a fast version of the leave-one-out cross-validation strategy to find the optimal value of μ in Eq. (10).

We decompose $\mathbf{V}$ into block presentation with the isolation of the first row and column as follows:

$$\mathbf{V} = \begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{\Lambda} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} = \begin{bmatrix} v_{11} & \mathbf{v}_1^T \\ \mathbf{v}_1 & \mathbf{V}_{(-1)} \end{bmatrix} \tag{12}$$

We denote $\boldsymbol{\alpha}_{(-i)}$ and $b_{(-i)}$ as the model parameters during the i-th iteration of the leave-one-out cross validation. In the first iteration, we have:

$$\begin{bmatrix} \boldsymbol{\alpha}_{(-1)} \\ b_{(-1)} \end{bmatrix} = \mathbf{P}_{(-1)} \left(\boldsymbol{y}_{(-1)} + \frac{\mu}{2C}\boldsymbol{M} \right) \tag{13}$$

where $\mathbf{P}_{(-1)} = \mathbf{V}_{(-1)}^{-1}$ and $y_{(-1)} = [y_2, y_3, ..., y_N, 0]^T$. We denote the predicted label of the i-th sample excluded from the training dataset by $\tilde{y}_i$, so that the predicted label of the first training sample becomes

$$\begin{aligned} \tilde{y}_1 &= \mathbf{v}_1^T \begin{bmatrix} \boldsymbol{\alpha}_{(-1)} \\ b_{(-1)} \end{bmatrix} + \frac{\mu}{2C}\boldsymbol{M}_{(-1)} \\ &= \mathbf{v}_1^T \mathbf{P}_{(-1)} \left(\boldsymbol{y}_{(-1)} + \frac{\mu}{2C}\boldsymbol{M}_{(-1)} \right) + \frac{\mu}{2C}\boldsymbol{M}_{(-1)} \end{aligned} \tag{14}$$

Considering the last N equations in the system of Eq. (9), we can get $\left[\mathbf{v}_1 \, \mathrm{V}_{(-1)}\right] \left[\boldsymbol{\alpha}^T, \, b\right]^T = \left(\boldsymbol{y}_{(-1)} + \frac{\mu}{2C}\boldsymbol{M}_{(-1)} \right)$, and Eq. (14) can be further written as

$$\begin{aligned} \tilde{y}_1 &= \mathbf{v}_1^T \mathbf{P}_{(-1)} \left[\mathbf{v}_1 \, \mathbf{V}_{(-1)}\right] \left[\alpha_1, \cdots, \alpha_N, b\right]^T - \frac{\mu}{2C}\boldsymbol{M}_{(-1)} \\ &= \mathbf{v}_1^T \mathbf{P}_{(-1)} \mathbf{v}_1 \alpha_1 + \mathbf{v}_1^T \left[\alpha_2, \cdots, \alpha_N, b\right]^T - \frac{\mu}{2C}\boldsymbol{M}_{(-1)} \end{aligned} \tag{15}$$

In Eq. (9), the first equation of the system is $y_1 + \frac{\mu}{2C}\boldsymbol{M}_{(-1)} = v_{11}\alpha_1 + \mathbf{v}_1^T \left[\alpha_2, \alpha_3, \cdots, \alpha_N, b\right]^T$. Combined with Eq. (15), we can get $\tilde{y}_1 = y_1 - \alpha_1(v_{11} - \mathbf{v}_1^T \mathbf{P}_{(-1)} \mathbf{v}_1)$. Finally, by using $\mathbf{P} = \mathbf{V}^{-1}$ and the block matrix inversion lemma, we can obtain

$$\mathbf{P} = \begin{bmatrix} u^{-1} & -u^{-1}\mathbf{v}_1\mathbf{P}_{-1} \\ \mathbf{P}_{(-1)} + u^{-1}\mathbf{P}_{(-1)}\mathbf{v}_1^T\mathbf{v}_1\mathbf{P}_{(-1)} & -u^{-1}\mathbf{P}_{(-1)}\mathbf{v}_1^T \end{bmatrix} \tag{16}$$

where $u = v_{11} - \mathbf{v}_1^T \mathbf{P}_{(-1)} \mathbf{v}_1$. Since the system of linear equations in Eq. (9) is not sensitive to permutations of the ordering of the equ-ations, we get

$$\tilde{y}_i = y_i - \alpha_i / \mathbf{P}_{ii} \tag{17}$$

By defining $\left[\boldsymbol{\alpha}^{'T}, b^{'}\right]^T = \mathbf{P}\left[\boldsymbol{y}^T, 0\right]$, $\left[\boldsymbol{\alpha}^{''T}, b^{''}\right] = \mathbf{P}\left[\boldsymbol{M}^T, 0\right]$, and $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{'} + \frac{\mu}{2C}\boldsymbol{\alpha}^{''}$, then we can get

$$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{\mathbf{P}_{ii}} - \frac{\frac{\mu}{2C}\alpha_i^{''}}{\mathbf{P}_{ii}} \tag{18}$$

Algorithm 1: Projected Sub-gradient Descent Algorithm
Input: $\boldsymbol{\alpha}^{'}$, $\boldsymbol{\alpha}^{''}$
Initialize: $\mu \leftarrow 0$ and $t \leftarrow 1$
Repeat
$\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{\boldsymbol{P}_{ii}} - \frac{\frac{\mu}{2C}\alpha_i^{''}}{\boldsymbol{P}_{ii}}$, $i = 1, 2, ..., N$
$d_i \leftarrow \mathbf{1}\{\tilde{y}_i y_i > 0\}$, $i = 1, 2, ..., N$
$\mu \leftarrow \mu - \frac{1}{\sqrt{t}} d_i y_i \frac{\alpha_i^{''}}{P_{ii}}$
If $\mu > D$ then $\mu \leftarrow \frac{\mu}{\|\mu\|_2} D$
End if
$\mu \leftarrow max(\mu, 0)$
$t \leftarrow t + 1$
Until convergence
Output: μ

It can be seen from Eq. (18) that $\boldsymbol{\alpha}$ and μ has a linear relationship, which indicates that we can obtain the learning model if μ is determined. The optimal μ is supposed to keep the same sign of $\tilde{y}_i$ and y_i for all the samples in the training dataset. However, this might bring many local minima issues due to its non-convex formulation. Thus, we adopt the following loss function, which is similar to the hinge loss:

$$(\tilde{y}_i, y_i) = |1 - \tilde{y}_i y_i|_+ = \left| y_i \frac{\alpha_i^{'} - \frac{\mu}{2C}\alpha_i^{''}}{\mathbf{P}_{ii}} \right|_+ \tag{19}$$

where $|x|_+ = \max\{0, x\}$. This is a convex upper bound to the leave-one-out misclassification loss, and it prefers the solutions in which $\tilde{y}_i$ has an absolute value equal or bigger than 1 and the same sign of y_i. Finally, the objective function is

$$\begin{aligned} &\sum_{i=i}^{N} l(\tilde{y}_i, y_i) \\ &\text{s.t} \quad 0 \leq \mu \leq D \end{aligned} \tag{20}$$

where D is a constant. This optimization process can be implemented by a projected sub-gradient descent algorithm and the pseudo-code is given in Algorithm 1.

A. Computational complexity

Compared with the traditional cross validation, the proposed fast leave-one-out cross validation strategy features in its fast computational ability. Its computational cost contains two parts, which can be represented as $O(N^3 + N)$. The first part calculates the matrix $\mathbf{P}$ by the inverse related to the training dataset on the target domain, therefore the corresponding computational complexity becomes $O(N^3)$. Another part includes the computational complexity of each iteration in the Algorithm 1 for optimizing Eq. (20), which can be represented as $O(N)$.

In terms of the traditional cross-validation, if we use the grid search strategy, from $[\mu_1, \mu_2, ..., \mu_T]$ for $\boldsymbol{\mu}$ in the proposed classifier in Eq. (10), the whole time complexity would become $TO(N^3 * N) = TO(N^4)$, which is extremely computationally expensive than $O(N^3 + N)$ occupied by the proposed fast cross-validation strategy.

V. Experimental results

A. The clinical dataset and the existing predictive model

In the experiments, a real clinical dataset was adopted in this study. The data was collected in a urology unit in Hong Kong from 2003 to 2011, which contain clinical records of 117 bladder cancer patients after radical cystectomy [17], [18]. 99 of the patients were male. The mean age of patients were 68 years old (SD=10 years). There is no case losing follow-up. The mean follow-up time was 2 years and 7 months (SD=29 months). The 30-day mortality, 5-year cancer-specific mortality, other-cause mortality, and the overall mortality rates were 3%, 33%, 22% and 55% respectively. In this study, we only focus on 5-year mortality of patients. 71 cases were undertaken open radical cystectory. 96 Patient had ileal conduit diversion. Other data include tumour stage and grade, preoperative serum albumin level and lymph node stage. The original dataset has eight records with missing values, which were removed during data processing. More details about the adopted dataset can be found in Table I.

The existing predictive model for predicting 5-year overall mortality on bladder cancer patients can be found in CancerNomograms.com [19]. It was created on 11,260 bladder cancer patients treated with radical cystectomy between 1988 and 2006 within 17 Surveillance, Epidemiology, and End Results registries in the United Statesby [20]. Patients were stratified into 20 strata based on the patient age and tumour stage when undertaking radical cystectomy. The smoothed Poison regression model was applied to gain the probability of mortality rate at 5 years after radical cystectomy. On the on-line user interface on the website, the user first provide the 'tumour stage', 'lymph node stage' and 'age at surgery'. After clicking 'show result' button, it will give the likelihood of overall survival after five years.

It can be observed that only a subset of the clinical dataset with the features 'age at operation', 'tumour stage' and 'lymph node stage' can be fit into the on-line predictive tool and therefore the requirements of the inverted pyramid dataset adopted in this work are satisfied.

B. Experimental Design

In this study, the main purpose of the experiment is to evaluate the performance of the proposed output-based transfer LS-SVM classifier to predict 5-year mortality of bladder cancer patients after radical cystectomy using the inverted pyramid dataset, compared with those obtained by using the traditional machine learning methods on the original dataset. The comparative methods include standard LS-SVM [2], standard SVM [21], back-propagation neural network (BPNN) [22], K nearest neighbouring (KNN) algorithm [23].

For the proposed classifier, firstly the subset of the clinical dataset with the features 'age at operation', 'tumour stage' and 'lymph node stage' were fed into the existing on-line tool and the corresponding probabilistic outputs of patient records were obtained. After that, the proposed classifier was applied on the whole clinical dataset as well as the probabilistic outputs obtained from the last step. The comparative methods were only applied on the whole dataset for classification. In order to make our comparison fair, we use the grid search in terms of classification accuracy to find out the optimal parameters during the training process. We select the polynomial kernel for kernel based methods in the experiments [17]. We set up the trade-off parameter C and the degree parameter γ by searching $C \in \{150, 200, 250\}$ and $\gamma \in \{2e-5, 2e-4, 2e-3, 2e-2, 2e-1, 1\}$ for the proposed classifier, LS-SVM and SVM. Finally, $C = 150$ and $\gamma = 2e-2$ were chosen for the proposed classifier and the LS-SVM due to the outstanding performance. For the standard SVM, $C = 200$ was chosen. For BPNN, the number of hidden neurons, the momentum and the learning rate were selected from $\{3, 5, 7, 9, 1, 13, 15, 17, 19, 21, 23, 25, 27, 29\}$, $\{0, 0.2, 0.5, 0.9\}$ and {0.01,0.05,0.09} respectively. Hidden neurons = 15, learning rate = 0.05 and momentum = 0.9 were chosen in the end. For KNN algorithm the value of the neighbouring parameter k was selected from {10, 12, 15, 18, 20} experimentally and $K = 18$ was finally determined with the best performance. The parameter settings are summarised in the Table. II. All the experiments are implemented using 64 bit MATLAB on a computer with Intel Core i5-6300 2.40 GHz CPU and 8.00GB RAM.

TABLE I: The clinical dataset adopted in this work

Features	Values
Gender	1 (female) 2 (male)
Age at operation	Normalized to [0, 1]
Surgery Type	1 (open surgery) 2 (laparoscopic surgery) 3 (robotic surgery)
Preoperative serum albumin level	Normalized to [0,1]
Tumor stage	1 (T1) 2 (T2) 3 (T3) 4 (T4)
Lymph node stage	0 (N0) 1 (N1) 2 (N2) 3 (N3)
Overall cancer stage	1 (Stage I) 2 (Stage II) 3 (Stage III) 4 (Stage IV)
Follow up period	Normalized to [0,1]
Grade	1 (Grade 1) 2 (Grade 2) 3 (Grade 3)
Type of diversion	1 (ideal conduit) 2 (neo bladder)
5-year overall mortality	1 (dead) 0 (alive)

TABLE II: Parameter settings of the proposed and comparative methods

Models	Proposed classifier	LS-SVM	SVM	BPNN	KNN
Parameter settings	C=150 $\gamma = 2e-2$	C=150 $\gamma = 2e-2$	C=200 $\gamma = 2e-2$	number of hidden neurons=15 learning rate=0.05 momentum=0.9	K=18

C. Performance evaluation

10-fold cross validation strategy was used in the experiments for performance evaluation, which ensures that every sample from the dataset has a chance to be in the training and testing sets. Here, the dataset was randomly divided into ten subsets. The model was built using nine subsets and tested on the remaining one. This process was repeated 10 times, and the mean and standard deviation of accuracy in the 10-fold cross validation procedure were calculated.

D. Classification performances

In this experiment, we compare the performance of the proposed approach and other five traditional machine learning methods on the prediction of 5-year mortality of bladder cancer patients after radical cystectomy. From the experimental results presented in Table III, we can observe that the predictive models using BPNN and KNN obtain comparatively low performances, with a mean classification accuracy of 0.6758 and 0.7061 respectively. The standard LS-SVM and SVM exhibit better performances than BPNN and KNN. Their mean accuracies are 0.7424 and 0.7485 respectively. Our proposed approach achieved the highest classification performance accuracy of 0.7697. Also, its performance stood out at sensitivity (0.7848), specificity (0.7579) and precision (0.7805) compared with the remaining methods. Thus it can be seen that by leveraging knowledge from the additional domain we can readily improve the traditional LS-LVM algorithm. The experiments indicate that the explored knowledge from the probabilistic outputs using the existing on-line model can benefit the model construction on the on-hand clinical dataset. Therefore, by using the proposed classifier, we are able to construct the classifier by utilizing the probabilistic output knowledge from the existing on-line tool, and even achieve comparatively better classification performance than those using other traditional methods without leveraging knowledge from the other domain. It is shown that the classification improvement through using the knowledge from the other domain has a great potential for improving traditional methods from different measurement indices.

TABLE III: Performance of the proposed classifier and comparative methods

Performance		Proposed classifier	LS-SVM	SVM	BPNN	KNN
Accuracy	Mean	**0.7697**	0.7424	0.7485	0.6758	0.7061
	SD	0.0456	0.0860	0.0655	0.0516	0.0516
	Max	0.8788	0.8182	0.7879	0.7879	0.8182
	Min	0.6970	0.6364	0.6364	0.6061	0.6061
Sensitivity	Mean	0.7848	0.7805	0.7551	0.6542	0.6940
	SD	0.0678	0.1365	0.0806	0.0900	0.0876
Specificity	Mean	0.7579	0.7216	0.7507	0.7119	0.7287
	SD	0.0625	0.1315	0.1105	0.1368	0.0688
Precision	Mean	0.7805	0.7462	0.7672	0.7485	0.7497
	SD	0.0612	0.1261	0.1154	0.1124	0.0767

VI. Conclusions and future work

Nowadays, medical data in many real-world applications cannot fit an existing predictive model or on-line tool, since a set of valuable features from the current scenario are not entirely involved in those models or tools. Yet the on-hand medical data is not sufficient to learn predictive models. Therefore, in order to overcome these issues and construct a classifier on this kind of data, we propose a novel LS-SVM classifier which leverages probabilistic outputs from the existing model from a perspective of transfer learning to make maximum use of the data and guarantee the enhanced generalization capability. The experimental results show that our proposed approach has a better result than the existing solutions, with an accuracy of 0.7697 compared to the reported accuracies of comparative machine learning methods, such as LS-SVM, SVM, BPNN and KNN. Moreover, because of our proposed approach's fast leave-one-out cross validation strategy, the weighting parameter μ can be determined autonomously and quickly and accordingly the classifier can be achieved within a reasonable time such that it has the potential for practical applications. Due to the improved accuracy and ability to work readily with the existing statistical predictive softwares or on-line tools currently in use within medical fields via exploring their probabilistic outputs, it is clear to see that this work has a great potential for improving prediction and prognosis within the medical industry. Most importantly, this study has shown to work with a real world dataset, proving its potential feasibility to be implemented into a real world clinical setting.

In near future, more investigations are required to ensure the robustness of the proposed approach. Currently we only apply the proposed method on the bladder caner prognosis. In future, we plan to apply the proposed method on different real world applications. Moreover, because we only consider one source domain in this study, we have actually imposed a limitation on its potential. In future work it would be worth investigating how the inclusion of multiple existing models could improve the accuracy of the target domain prediction.

VII. Acknowledgement

The work was supported by the Australian Research Council (ARC) under Discovery Grant DP140101366, and was also supported in part by the Research Grants Council of the Hong Kong SAR (PolyU5134/12E), the Hong Kong Polytechnic University (G-UC93, G-YBKX) and a scholarship donated by Nelson Y.C Yu.

References

[1] J. A. Cruz and D. S. Wishart, "Applications of machine learning in cancer prediction and prognosis," Cancer Informatics, vol. 2, 2006.

[2] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, Least Squares Support Vector Machine Classifiers. Singapore: World Scientific, 2002.

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

[4] J. Huang, A. Gretton, K. M. Borgwardt, B. Schölkopf, and A. J. Smola, "Correcting sample selection bias by unlabeled data," in Advances in Neural Information Processing Systems, 2006, pp. 601–608.

[5] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in 19th International Conference on Machine Learning, vol. 2, July 2002, pp. 387–394.

[6] T. Jebara, "Multi-task feature and kernel selection for svms," in Proceedings of the 21st International Conference on Machine Learning. ACM, 2004, p. 55.

[7] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199–210, 2011.

[8] L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.

[9] H. Zuo, G. Zhang, V. Behbood, and J. Lu, "Feature spaces-based transfer learning," in 16th World Congress of the International Fuzzy Systems Association, and 9th Conference of the European Society for Fuzzy Logic and Technology, 2015.

[10] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in 22nd Conference on Artificial Intelligence, vol. 7, July 2007, pp. 608–614.

[11] E. V. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in Advances in Neural Information Processing Systems, 2007, pp. 153–160.

[12] A. Schwaighofer, V. Tresp, and K. Yu, "Learning gaussian process kernels via hierarchical bayes," in Advances in Neural Information Processing Systems, 2004, pp. 1209–1216.

[13] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2008, pp. 283–291.

[14] Z. Deng, Y. Jiang, K.-S. Choi, F.-L. Chung, and S. Wang, "Knowledge-leverage-based tsk fuzzy system modeling," IEEE transactions on neural networks and learning systems, vol. 24, no. 8, pp. 1200–1212, 2013.

[15] Z. Deng, Y. Jiang, H. Ishibuchi, K.-S. Choi, and S. Wang, "Enhanced knowledge-leverage-based tsk fuzzy system modeling for inductive transfer learning," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8, no. 1, p. 11, 2016.

[16] M. Frank and P. Wolfe, "An algorithm for quadratic programming," Naval Research Logistics Quarterly, vol. 3, no. 1-2, pp. 95–110, 1956.

[17] G. Wang, K.-M. Lam, Z. Deng, and K.-S. Choi, "Prediction of mortality after radical cystectomy for bladder cancer by machine learning techniques," Computers in Biology and Medicine, vol. 63, pp. 124–132, 2015.

[18] E. Chan, S. Yip, S. Hou, H. Cheung, W. Lee, and C. Ng, "Age, tumour stage, and preoperative serum albumin level are independent predictors of mortality after radical cystectomy for treatment of bladder cancer in hong kong chinese," Hong Kong Medical Journal, vol. 19, no. 5, pp. 400–406, 2013.

[19] Nomograms. Nomogram predicting the probability of mortality due to bladder cancer versus other causes. [Online]. Available: http://labs.fccc.edu/nomograms/nomogram.php?id=48&audience=1.

[20] G. Lughezzani, M. Sun, S. F. Shariat, L. Budäus, R. Thuret, C. Jeldres, D. Liberman, F. Montorsi, P. Perrotte, and P. I. Karakiewicz, "A population-based competing-risks analysis of the survival of patients treated with radical cystectomy for bladder cancer," Cancer, vol. 117, no. 1, pp. 103–109, 2011.

[21] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 2, no. 3, p. 27, 2011.

[22] M. Buscema, "Back propagation neural networks," Substance Use & Misuse, vol. 33, no. 2, pp. 233–270, 1998.

[23] L. E. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, no. 2, p. 1883, 2009.