2.2 The Proposed Approach

As such, we have two observations. First, the estimation based on source-domain data need to be consistent with target-domain data. Second, the model should approximate the conditional distribution of target-domain, instead of the source-domain. Thus, we propose a new criterion by adding density ratio weighting and replacing the target conditional distribution as follows:

$$
\hat{f} = \arg\min_{f} \frac{1}{n} \sum_{x \in X_s} \frac{P_t(x)}{P_s(x)} \left| P_t(y|x) - P(y|x, f) \right| + \Theta_f
$$

(5)

We notice that it is a general criterion extending Eq.(1). Under the traditional setting that marginal and conditional distributions do not shift, it is the same as Eq.(1). With the analysis in Section 4, we prove that Eq.(5) approximates an unbiased estimation to ideal hypothesis \(f^*\). However, the model complexity term \(\Theta_f\) is usually hard to calculate in practice. Thus, following the same ideas, we propose a transfer cross validation (TrCV) method to solve the stated problems practically. It aims to select the model by minimizing the criterion:

$$
\hat{f} = \arg\min_{f} \frac{1}{k} \sum_{j=1}^{k} \sum_{(x, y) \in S_j} \frac{P_t(x)}{P_s(x)} \left| P_t(y|x) - P(y|x, f) \right|
$$

(6)

Thus, algorithm selection, parameter tuning and source-domain selection in transfer learning can be solved using TrCV. For algorithm selection, it is intuitive. For other two problems, it is equivalent to pick a set of parameters or a source-domain which can build a model minimizing the value in Eq.(6).

3 Transfer Cross Validation (TrCV)

We discuss two main issues of TrCV in this section. The first one is that the density ratio of two domains \( \frac{P_t(x)}{P_s(x)} \) needs to be calculated based on the observed finite set. We let \( \beta = \{ \beta(x_1), \ldots, \beta(x_n) \} \) be the density ratio vector, where \( \beta(x) = \frac{P_t(x)}{P_s(x)} \). Some methods have been exploited for this problem [9, 10]. We adopt an existing one KMM from [10] which aims to find suitable values of \( \beta \) to minimize the discrepancy between means of two domains. Formally, it tries to minimize the following object by calculating the optimal \( \beta \).

$$
\min_{\beta} \frac{1}{2} \beta^T K \beta - \kappa^T \beta
$$

s.t \( \beta_i \in [0, B] \), \( |\sum_{i=1}^{n} \beta_i - n| \leq n\epsilon \)

where \( K_{ij} = \phi(x_i, x_j) \), \( x_i, x_j \in X_s \), \( \kappa_i = \frac{n}{\ell+u} \sum_{j=1}^{\ell+u} \phi(x_i, x_j) \), \( x_i \in X_s, x_j \in X_\ell \cup X_u, \phi(\ast, \ast) \) is the kernel function, \( B \) is the upper bound for the ratio and \( \epsilon \) should be \( O(B/\sqrt{n}) \). In addition, \( \beta \) is restricted by two constraints: the first one limits the scope of discrepancy between \( p_t(x) \) and \( p_s(x) \) and the second one ensures that the measure \( \beta(x)p_s(x) \) is close to a probability distribution.
Cross Validation Framework to Choose amongst Models 551

Input: \( S_i, \overline{S}_i, T \), a learner \( F \)

Output: The estimation of \(|P_t(y|x) - P(y|x, f_i)|\)

1. Build a model \( f_i \) from \( \overline{S}_i \) using \( F \);
2. Predict the labels of \( X_u, \overline{Y}_u \);
3. Build another model \( \overline{f}_i \) from \( \{X_u, \overline{Y}_u\} \cup \{X_\ell, Y_\ell\} \) using \( F \);
4. Predict the labels of \( S_i, \overline{Y}_s \);
5. for each instance \( x_{ij} \) in \( S_i \) do
6. \( r(x_{ij}) = |y_{ij} - \overline{y}_{ij}| \), where \( \overline{y}_{ij} \in \overline{Y}_s \);
7. end
8. return \( r(x_{ij}), x_{ij} \in S_i \);

Fig. 1. Reverse Validation

Fig. 2. Flow chart of reverse validation

As follows, we focus on the second issue: how to calculate the difference between the conditional distribution estimated by model \( f \) and the true conditional distribution, \(|P_t(y|x) - P(y|x, f)|\). Due to the limited number of labeled examples in target-domain, it is impossible to estimate the conditional distribution \( P_t(y|x) \) reliably. To overcome this challenge, we propose a novel method “Reverse Validation” which estimates the approximation difference directly and avoids computing the conditional distribution \( P_t(y|x) \). To the best of our knowledge, this has not been well studied.

3.1 Reverse Validation (RV)

The main flow is presented in Figure 2 and the detail is stated in Figure 1. Let \( S_i \) be the source-domain data in \( i \)-th fold and \( \overline{S}_i \) be the remaining data. Firstly, for the given learner, we train a model \( f_i \) from \( \overline{S}_i \), and then use \( f_i \) to predict the labels of \( X_u \) and obtain \( \overline{Y}_u \). Next, we combine \( \{X_u, \overline{Y}_u\} \) and \( \{X_\ell, Y_\ell\} \) to form a new set. Afterwards, a new model \( \overline{f}_i \) is built from the new set using the same algorithm and used to classify the instances in \( S_i \). We denote the pseudo labels of \( S_i \) as \( Y_s \). Finally, for each instance \( \{x_{ij}, y_{ij}\} \in S_i \), we use the value of \( |y_{ij} - \overline{y}_{ij}| \) to estimate the difference, where \( \overline{y}_{ij} \) is the corresponding pseudo label of \( x_{ij} \). As analysed in Section 4.2, RV value \( r(x_{ij}) = |y_{ij} - \overline{y}_{ij}| \) is related to \(|P_t(y_{ij}|x_{ij}) - P(y_{ij}|x_{ij}, f_i)|\) and can be used as an indicator.

TrCV can now be introduced using KMM and RV as stated in Figure 3. Briefly, we calculate the density ratio qualitatively and apply reverse validation to estimate the loss of conditional distribution approximation in each fold.