Cross Validation Framework to Choose amongst Models and Datasets for Transfer Learning

Erheng Zhong[1], Wei Fan[2], Qiang Yang[3],
Olivier Verscheure[2], and Jiangtao Ren[1]

[1] Sun Yat-Sen University, Guangzhou, China
{sw04zheh@mail2,issrjt@mail}.sysu.edu.cn
[2] IBM T.J Watson Research, USA
{weifan,ov1}@us.ibm.com
[3] Department of Computer Science, Hong Kong University of Science and Technology
qyang@cse.ust.hk

Abstract. One solution to the lack of label problem is to exploit transfer learning, whereby one acquires knowledge from source-domains to improve the learning performance in the target-domain. The main challenge is that the source and target domains may have different distributions. An open problem is how to select the available models (including algorithms and parameters) and importantly, abundance of source-domain data, through statistically reliable methods, thus making transfer learning practical and easy-to-use for real-world applications. To address this challenge, one needs to take into account the difference in both marginal and conditional distributions in the same time, but not just one of them. In this paper, we formulate a new criterion to overcome "double" distribution shift and present a practical approach "Transfer Cross Validation" (TrCV) to select both models and data in a cross validation framework, optimized for transfer learning. The idea is to use density ratio weighting to overcome the difference in marginal distributions and propose a "reverse validation" procedure to quantify how well a model approximates the true conditional distribution of target-domain. The usefulness of TrCV is demonstrated on different cross-domain tasks, including wine quality evaluation, web-user ranking and text categorization. The experiment results show that the proposed method outperforms both traditional cross-validation and one state-of-the-art method which only considers marginal distribution shift. The software and datasets are available from the authors.

1 Introduction

Transfer learning works in the context that the number of labeled examples in target-domain is limited. It assumes that source-domain and target-domain are under different marginal and conditional distributions. Recently, a number of algorithms have been proposed to overcome the distribution shift, such as those reviewed in but not limited to [1]. Moreover, for a given target-domain in transfer learning, a likely large number of source-domains are available. For example, if we

J.L. Balcázar et al. (Eds.): ECML PKDD 2010, Part III, LNAI 6323, pp. 547–562, 2010.

Table 1. Definition of notations

Notation	Description	Notation	Description
S	Source-domain, $S = \{X_s, Y_s\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$	k	Number of folds in cross validation
S_i	Data in i-th fold	$r(\mathbf{x})$	Value of $\mathbf{x}$ got by reverse validation
$\overline{Y}_s^i$	Pseudo labels of S_i	$\ell^*(f)$	Expected loss of model f
$\overline{S}_i$	Remaining data in i-th fold	$\ell(f)$	Empirical loss of model f in T
T	Target-domain, $T = \{X_\ell, Y_\ell, X_u\}$	$\ell_w(f)$	Weighted empirical loss of model f in S
L	Labeled data in T, $L = \{X_\ell, Y_\ell\}$	$\varepsilon_u(f)$	Estimated accuracy of model f by TrCV
U	Unlabeled data in T, $U = X_u$	Θ_f	Model complexity of f
n	Number of instances in S	$P(\mathbf{x})$	Marginal distribution of $\mathbf{x}$
ℓ	Number of instances in L	$P(y\|\mathbf{x})$	Conditional distribution of $(\mathbf{x}, y)$
u	Number of instances in U	β	Density ratio vector of X_s

aim to classify the documents of 20-Newsgroup [2], RCV1 [3] and Reuters-21578 [2] or other text collections can be treated as the candidates of source-domain. Thus, for a transfer learning task, it is crucial to solve three problems effectively: (1) How to select the right transfer learning algorithms? (2) How to tune the optimal parameters? (3) How to choose the most helpful source-domain from a large pool of datasets? However, to the best of our knowledge, neither any analytical criterion nor efficient practical procedures have been proposed and reported.

Although some analytical techniques such as AIC (Akaike Information Criterion) [4], BIC (Bayesian Information Criterion) [5] and SRM (Structural Risk Minimization) principle [6] or sample re-use method (such as Cross Validation (CV)) to selecting the suitable model or training data (source-domain in transfer learning) have been studied, as reviewed later, they can not guarantee their performances in transfer learning for two reasons. First, due to the "double" distribution shift, including marginal and conditional distributions, the unbiasedness which guarantees the accuracy of these techniques does not hold anymore. Second, due to the very small number of labeled data in target-domain, it is unreliable to estimate the conditional distribution of target-domain directly.

To cope with these challenges, we first formulate a general criterion for model selection in transfer learning scenario, followed by a novel variant of CV method "Transfer Cross Validation" (TrCV) to solving the above three problems practically. Briefly, we introduce density ratio weighting to reduce the difference of marginal distributions between two domains. As proved in Section 4.1, it makes the estimation of TrCV unbiased. In addition, we exploit a method "Reverse Validation" (RV) to approximate the difference between the estimated and true conditional distribution of target-domain directly. As stated in Section 4.2, the value of RV is reliable to indicate the true difference. In summary, by eliminating the difference between two domains, the model selected by TrCV has a confidence bound on accuracy as shown in Section 4.3. In other words, the model or source-domain selected by TrCV is highly likely the best one among candidates as evaluated in Section 5.

2 Problem Statement

We review the limitation of traditional validation methods and then introduce a general criterion with transfer cross validation. The notations are summarized

in Table 1. Let $S = \{X_s, Y_s\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ denote the source-domain and $T = \{X_\ell, Y_\ell, X_u\} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell} \cup \{(\mathbf{x}_j)\}_{j=1}^{u}$ denote the target-domain, where n is the number of instances in source-domain, ℓ and u are the number of labeled and unlabeled instances in target-domain respectively. Then, let $P_s(\mathbf{x})$ and $P_s(y|\mathbf{x})$ denote the marginal and conditional distribution of source-domain, $P_t(\mathbf{x})$ and $P_t(y|\mathbf{x})$ for target-domain. We use $\hat{f}$ to represent the model expected to obtain.

2.1 Limitations of Existing Approaches

The model selected by analytical techniques is as follows:

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

where the first term represents the empirical loss and Θ_f is model complexity: the number of model parameters in AIC and BIC or the VC-Dimension in SRM. On the other hand, k-fold cross validation aims to select the model as:

$$\hat{f} = \arg\min_{f} \frac{1}{k} \sum_{j=1}^{k} \sum_{(\mathbf{x},y) \in S_j} \Big| P_s(y|\mathbf{x}) - P(y|\mathbf{x}, f_j) \Big| \tag{2}$$

where k is the number of folds, S_j are the data in j-th fold and f_j is the model trained from the remaining data. However, these methods do not work as one would desire, for the following two reasons. First, because $P_s(\mathbf{x}) \neq P_t(\mathbf{x})$, Eq.(1) no longer provides consistent estimation [7]. In other words, $\lim_{n\to\infty}(\hat{f}) \neq f^*$, where f^* is the ideal hypothesis which achieves the minimal expected loss to approximate $P_t(y|\mathbf{x})$, regulated by model complexity:

$$f^* = \arg\min_{f} \mathbf{E}_{\mathbf{x} \sim P_t(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f) \Big| + \Theta_f \tag{3}$$

To cope with similar problem in sample selection bias, previous work Weighted CV (WCV) [8] proposes to use density ratio to eliminate the difference in marginal distributions when performing cross-validation. It selects the model that minimizes the following objective.

$$\hat{f} = \arg\min_{f} \frac{1}{k} \sum_{j=1}^{k} \sum_{(\mathbf{x},y) \in S_j} \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \Big| P_s(y|\mathbf{x}) - P(y|\mathbf{x}, f_j) \Big| \tag{4}$$

However, neither of these explicitly considers the effect of conditional distribution shift between two domains, which is essential for most transfer learning problems. Because $P_s(y|\mathbf{x}) \neq P_t(y|\mathbf{x})$ under transfer learning context, a model approximating $P_s(y|\mathbf{x})$ is not necessarily close to $P_t(y|\mathbf{x})$. Thus, the model selected by Eq.(2) and Eq.(4) based on source-domain can not guarantee its performance in target-domain, as demonstrated experimentally in Section 5.

On the other hand, one may consider to perform CV on the labeled target-domain data L or to select the model trained using source-domain data S and has a high accuracy on L. But these methods fail to perform well on the whole target-domain, because the number of labeled data is so limited that they cannot reliably describe the true conditional distribution of target-domain, $P_t(y|\mathbf{x})$.

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:

$$\hat{f} = \arg\min_f \frac{1}{n} \sum_{\mathbf{x} \in X_s} \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f) \Big| + \Theta_f \tag{5}$$

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 Θ_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:

$$\hat{f} = \arg\min_f \frac{1}{k} \sum_{j=1}^{k} \sum_{(\mathbf{x},y) \in S_j} \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f) \Big| \tag{6}$$

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 $\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}$ needs to be calculated based on the observed finite set. We let $\beta = \{\beta(\mathbf{x}_1), \ldots, \beta(\mathbf{x}_n)\}$ be the density ratio vector, where $\beta(\mathbf{x}) = \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}$. 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 β to minimize the discrepancy between means of two domains. Formally, it tries to minimize the following object by calculating the optimal β.

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2}\beta^T K \beta - \kappa^T \beta \\ s.t \quad & \beta_i \in [0, B], \qquad |\sum_{i=1}^{n} \beta_i - n| \leq n\epsilon \end{aligned}$$

where $K_{ij} = \phi(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{x}_i, \mathbf{x}_j \in X_s$, $\kappa_i = \frac{n}{\ell+u} \sum_{j=1}^{\ell+u} \phi(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{x}_i \in X_s, \mathbf{x}_j \in X_\ell \cup X_u$, $\phi(*,*)$ is the kernel function, B is the upper bound for the ratio and ϵ should be $O(B/\sqrt{n})$. In addition, β is restricted by two constraints: the first one limits the scope of discrepancy between $p_t(\mathbf{x})$ and $p_s(\mathbf{x})$ and the second one ensures that the measure $\beta(\mathbf{x})p_s(\mathbf{x})$ is close to a probability distribution.

Input: S_i, $\overline{S}_i$, T, a learner $\mathcal{F}$
Output: The estimation of $|P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f_i)|$
1 Build a model f_i from $\overline{S}_i$ using $\mathcal{F}$;
2 Predict the labels of X_u, $\overline{Y}^i_u$;
3 Build another model $\overline{f}_i$ from $\{X_u, \overline{Y}^i_u\} \cup \{X_\ell, Y_\ell\}$ using $\mathcal{F}$;
4 Predict the labels of S_i, $\overline{Y}^i_s$;
5 **for** *each instance* $\mathbf{x_{ij}}$ *in* S_i **do**
6 $r(\mathbf{x_{ij}}) = |y_{ij} - \overline{y}_{ij}|$, where $\overline{y}_{ij} \in \overline{Y}^i_s$;
7 **end**
8 **return** $r(\mathbf{x_{ij}})$, $\mathbf{x_{ij}} \in S_i$;

Fig. 1. Reverse Validation

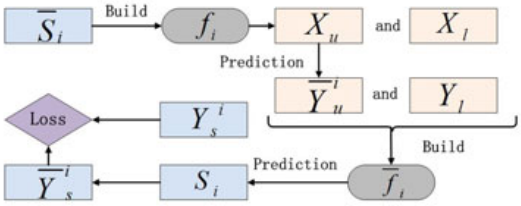


Fig. 2. Flow chart of reverse validation

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, $|P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f)|$. Due to the limited number of labeled examples in target-domain, it is impossible to estimate the conditional distribution $P_t(y|\mathbf{x})$ reliably. To overcome this challenge, we propose a novel method "Reverse Validation" which estimates the approximation difference directly and avoids computing the conditional distribution $P_t(y|\mathbf{x})$. 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 we use f_i to predict the labels of X_u and obtain $\overline{Y}^i_u$. Next, we combine $\{X_u, \overline{Y}^i_u\}$ 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 $\overline{Y}^i_s$. Finally, for each instance $\{\mathbf{x}_{ij}, y_{ij}\} \in S_i$, we use the value of $|y_{ij} - \overline{y}_{ij}|$ to estimate the difference, where $\overline{y}_{ij}$ is the corresponding pseudo label of $\mathbf{x}_{ij}$. As analysed in Section 4.2, RV value $r(\mathbf{x}_{ij}) = |y_{ij} - \overline{y}_{ij}|$ is related to $|P_t(y_{ij}|\mathbf{x}_{ij}) - P(y_{ij}|\mathbf{x}_{ij}, f_i)|$ 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.

Input: S, T, a learner $\mathcal{F}$, number of fold k
Output: The measure value of TrCV
Calculate β using KMM;
for $i = 1$ ***to*** k **do**
    Perform reverse validation, $V_i = RV(S_i, \overline{S}_i, T, \mathcal{F})$;
    $\ell = \ell + \sum_j v_{ij} \cdot \beta(\mathbf{x_{ij}})$, $v_{ij} \in V_i$;
end
return ℓ/n;

Fig. 3. Transfer Cross Validation

4 Formal Analysis

We analyse three issues. First, does the general principle bound the risk in transfer learning? Second, is the loss calculated by reverse validation related to the true difference $|P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f)|$? Third, how is the confidence of the transfer cross validation?

4.1 Generalization Bound

We first demonstrate that the model selected by Eq.(5), $\hat{f}$, provides an unbiased estimator to f^* defined in Eq.(3). Let the expected loss of a model f be $\ell^*(f)$, the weighted empirical loss in source-domain be $\ell_w(f)$ and n be the number of examples in S, then we get the lemma.

Lemma 1. *$\ell_w(\hat{f}) + \Theta_{\hat{f}} = \ell^*(f^*) + \Theta_{f^*}$, when $n \to \infty$ and f^* and $\hat{f}$ belong to the same hypothesis class.*

Proof

$$\begin{aligned}
\ell_w(\hat{f}) &= \frac{1}{n} \sum_{\mathbf{x} \in X_s} \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, \hat{f}) \Big| \\
&= E_{\mathbf{x} \in X_s} \Big[\int_{\mathbf{x}} \frac{P_t(\mathbf{x})}{P_s(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, \hat{f}) \Big| P_s(\mathbf{x}) d\mathbf{x} \Big] \\
&= E_{\mathbf{x} \in X_s} \Big[\int_{\mathbf{x}} P_t(\mathbf{x}) \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, \hat{f}) \Big| d\mathbf{x} \Big] \\
&= \frac{1}{n} \sum_{\mathbf{x} \in X_s, X_s \sim P_t(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, \hat{f}) \Big| \\
&= E_{\mathbf{x} \in X_s, X_s \sim P_t(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, \hat{f}) \Big|
\end{aligned}$$

This means that, as n approaches infinity, the model minimizing the value of weighted empirical loss in source-domain also minimizes the expected loss in target-domain, $\ell_w(\hat{f}) = \ell^*(f^*)$. In addition, if f^* and $\hat{f}$ belong to the same hypothesis class, it leads to $\Theta_{f^*} = \Theta_{\hat{f}}$. □

In addition, we conclude that the model minimizing the value of general principal in Eq.(5) is equal to the model minimizing the empirical error of target-domain data. In other words,

$$\begin{aligned}\ell(\hat{f}) &= \frac{1}{n}\sum_{\mathbf{x}\in X_s, X_s\sim P_t(\mathbf{x})}\Big|P_t(y|\mathbf{x}) - P(y|\mathbf{x},\hat{f})\Big| \\ &= \frac{1}{n}\sum_{\mathbf{x}\in X_s}\frac{P_t(\mathbf{x})}{P_s(\mathbf{x})}\Big|P_t(y|\mathbf{x}) - P(y|\mathbf{x},\hat{f})\Big|\end{aligned} \tag{7}$$

Next we demonstrate that if the estimator of $\Theta_{\hat{f}}$ is related to VC-dimension, $\hat{f}$ constructed from source-domain data has a generalization bound over target-domain data.

Theorem 1. *Let $G(\hat{f})$ denote the generalization error of $\hat{f}$ in the target-domain, n is the number of data in S and d_{vc} is the VC-dimension of the hypothesis class which $\hat{f}$ belongs to, then with the probability at least $1-\delta$*

$$G(\hat{f}) \le \ell_w(\hat{f}) + \sqrt{\Big(\frac{d_{vc}(\log(2n/d_{vc})+1) - \log(\delta/4)}{n}\Big)} \tag{8}$$

Proof As a conclusion from [6], for a given model f, it has a generalization bound:

$$G(f) \le \ell(f) + \sqrt{\Big(\frac{d_{vc}(\log(2n/d_{vc})+1) - \log(\delta/4)}{n}\Big)} \tag{9}$$

In addition, let us recall Eq.(7), thus we obtain Eq.(8). □

4.2 Estimation by Reverse Validation

Due to the limited number of labeled examples in target-domain, we use reverse validation (RV) to estimate the difference between $P_t(y|\mathbf{x})$ and $P(y|\mathbf{x},f)$ instead of estimating the conditional probability $P_t(y|\mathbf{x})$ directly. As follows we provide some insights in RV. Let f_i be the model trained from $\overline{S}_i$, $\{X_u, \overline{Y}_u\}$ be the unlabeled data and corresponding pseudo labels predicted by f_i in the target-domain, $\overline{f}_i$ be the model built from $\{X_u, \overline{Y}^i_u\} \cup \{X_\ell, Y_\ell\}$ and $\epsilon(f)$ be the approximation error of a model f. Thus, for a given instance $\mathbf{x}$ from S_i, RV returns a value

$$r(\mathbf{x}) = \Big|P_s(y|\mathbf{x}) - P(y|\mathbf{x},\overline{f}_i)\Big| \tag{10}$$

As an approximation to $P_s(y|\mathbf{x})$, $P(y|\mathbf{x},f_i)$ can be rewritten as

$$P(y|\mathbf{x},f_i) = P_s(y|\mathbf{x}) + \epsilon(f_i) \tag{11}$$

where ϵ is the approximation error. In addition, because $\overline{f}_i$ is trained from the label information $\overline{Y}^i_u$ and Y_ℓ, $P(y|\mathbf{x},\overline{f}_i)$ can be treated as an approximation to the nuisance between $P(y|\mathbf{x},f_i)$ and $P_t(y|\mathbf{x})$.

$$P(y|\mathbf{x},\overline{f}_i) = \alpha\cdot P(y|\mathbf{x},f_i) + (1-\alpha)\cdot P_t(y|\mathbf{x}) + \epsilon(\overline{f}_i) \tag{12}$$

where α is the nuisance parameter related to the ratio between the size of X_u and X_l. Thus, by combining Eq.(10), (11) and (12), $r(\mathbf{x})$ can be rewritten as

$$\begin{aligned} & r(\mathbf{x}) \\ &= \Big| P_s(y|\mathbf{x}) - P(y|x, \overline{f}_i) \Big| \\ &= \Big| P_s(y|\mathbf{x}) - \alpha P(y|\mathbf{x}, f_i) + (1-\alpha) P_t(y|\mathbf{x}) - \epsilon(\overline{f}_i) \Big| \\ &= \Big| (1-\alpha) \Big(P(y|\mathbf{x}, f_i) - P_t(y|\mathbf{x}) \Big) - \epsilon(f_i) - \epsilon(\overline{f}_i) \Big| \end{aligned} \tag{13}$$

This demonstrates that $r(\mathbf{x})$ is related to $|P(y|x, f_i) - P_t(y|\mathbf{x})|$ reliably. Thus, when the number of training data is large enough such that the model can approximate the true conditional probability reliably. In other words, when $\epsilon(f_i)$ and $\epsilon(\overline{f}_i)$ are small, RV can approach a confident estimation. In addition, when more labeled data obtained in target-domain, α tends to be smaller. This implies that $r(\mathbf{x})$ estimates $|P(y|x, f_i) - P_t(y|\mathbf{x})|$ more precisely. On the other hand, if no labeled data in target-domain but $P_t(y|\mathbf{x}) = P_s(y|\mathbf{x})$, $r(\mathbf{x})$ becomes $|\epsilon(f_i) + \epsilon(\overline{f}_i)|$ instead, which approximates as much as twice the error in traditional cross validation.

4.3 Confidence by TrCV

The discussion is based on the assumption that the classifiers are consistent: the classifiers built in each folds have the same predictability. Following Eq.(7), minimizing the weighted empirical loss of source-domain data in TrCV is equal to minimizing the empirical loss of target-domain data. In addition, combining Eq.(6) and Eq.(13), when model can approximate the true distribution well if obtaining enough labeled data, we rewrite the accuracy estimated by TrCV, $\varepsilon_u(f)$, as

$$\begin{aligned} \varepsilon_u(f) &= 1 - \frac{1}{k} \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} \beta(\mathbf{x}) \Big| r(\mathbf{x})/(1-\alpha) \Big| \\ &= 1 - \frac{1}{k} \sum_{j=1}^{k} \sum_{\mathbf{x} \in X_s, X_s \sim P_t(\mathbf{x})} \Big| P_t(y|\mathbf{x}) - P(y|\mathbf{x}, f) \Big| \end{aligned} \tag{14}$$

where $r(\mathbf{x})$ is the value of reverse validation on data $\mathbf{x}$ and $\beta(\mathbf{x})$ is the density ratio of $\mathbf{x}$. Let $\varepsilon(f)$ be the true accuracy of f, based on the statement in [11], when the size of validation set is reasonably large, the distribution of $\varepsilon_u(f)$ is approximately normal with mean $\varepsilon(f)$ and a variance of $\varepsilon(f) \cdot (1 - \varepsilon(f))/n$. By De Moivre-Laplace Limit theorem, we have

$$Pr\Big\{ -z < \frac{\varepsilon_u(f) - \varepsilon(f)}{\sqrt{\varepsilon(f) \cdot (1 - \varepsilon(f))/n}} < z \Big\} \approx \lambda \tag{15}$$

where z is the $(1+\lambda)/2$-th quantile point of the standard normal distribution. The low and high confidence points of $\varepsilon(f)$ is calculated by inverting Eq.(15) as

$$\frac{2n \cdot \varepsilon_u(f) + z^2 \pm z \cdot \sqrt{4n \cdot \varepsilon_u(f) + z^2 - 4n \cdot \varepsilon_u^2(f)}}{2(n + z^2)}$$

In addition, if the accuracy of f obtains the normal distribution in this interval with mean $\mu = \frac{2n\cdot\varepsilon_u(f)+z^2}{2(n+z^2)}$ and variance $\sigma = \frac{z\cdot\sqrt{4n\cdot\varepsilon_u(f)+z^2-4n\cdot\varepsilon_u^2(f)}}{2(n+z^2)}$, the probability between two models, f_1 and f_2, $P\Big(\varepsilon(f_1) > \varepsilon(f_2)\Big)$ can be calculated.

$$\begin{aligned} & P\Big(\varepsilon(f_1) > \varepsilon(f_2)\Big) \\ & = P\Big(\varepsilon(f_1) - \varepsilon(f_2) > 0\Big) \\ & = P(x > 0), \quad x \sim N(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2) \\ & = 1 - \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} \int_{-\infty}^{\frac{\mu_2-\mu_1}{\sqrt{\sigma_1^2+\sigma_2^2}}} e^{-t^2/2} dt \end{aligned} \tag{16}$$

where μ_1 and μ_2 are the means of accuracy distributions obtained by f_1 and f_2 with TrCV and σ_1 and σ_2 are the corresponding variances. By calculating the means and variances based on the loss value of TrCV, the confidence of TrCV can be obtained by Eq.(16).

5 Experiment

TrCV criterion is evaluated to show if it can select the best algorithm for one task, can tune suitable parameters for one model and can choose the most useful source-domain over different candidates. For each task, several data collections have been utilized.

5.1 Experimental Setup

The proposal approach TrCV is compared against several other cross validation methods. The first two are the standard k-fold CV formulated by Eq.(2). One is on source-domain (SCV), another is on labeled data from target-domain (TCV). The third one is to build a model on the source-domain data and validate on labeled target-domain data (STV). Most importantly, we compare with Weighted CV (WCV) [8]. As discussed earlier, WCV is proposed for sample selection bias problems. It uses density ratio weighting to reduce the difference of marginal distribution between two domains, but ignores the difference in conditional probability, as shown in Eq.(6).

To test different criteria, we introduce five traditional classifiers, including Naive Bayes(NB), SVM, C4.5, K-NN and NNge(Ng), and three state-of-the-art transfer learning methods: TrAdaBoost(TA) [12], LatentMap(LM) [13] and LWE [14]. Among them, TrAdaBoost is based on instances weighting, LatentMap is through feature transform and LWE uses model weighting ensemble. As a comparison, the number of folds in SCV, TCV and TrCV is set to be the same: 10, and the number of labeled data in target-domain is fixed as the larger one between $0.1 \times |T|$ and 20. As follows, we use "correlation" between the best classifiers and the selected classifiers by the criteria as the measure of evaluation.

Table 2. Dataset for Algorithm and Parameters Selection

Data Set	$\|S\|$	$\|T\|$	Description
Red-White(RW)	1599	4998	physicochemical
White-Red(WR)	4998	1599	variables
orgs vs. people(ope)	1016	1046	Documents
orgs vs. places(opl)	1079	1080	from different
people vs. places(pp)	1239	1210	subcategories
Sheep(Sp)	61	65	Web pages
Biomedical(Bl)	61	131	with different
Goats(Gs)	61	70	contents

Table 3. Dataset for Source-domain Selection

Data Set	S	T	$\|S\|$	$\|T\|$
comp	windows vs. motorcycles	graphics	1596	
vs.	pc.hardware vs. baseball	vs.	1969	1957
rec	mac.hardware vs. hockey	autos	1954	
sci	crypt vs. guns	electronics	1895	
vs.	med vs. misc	vs.	1761	1924
talk	space vs. religion	mideast	1612	

Let f and g denote any two models, and $\varepsilon(\cdot)$ and $v(\cdot)$ are the accuracy and value of criteria (e.g. TrCV, standard CV, etc) on each model, respectively. Then the measure is

$$corr = C^2_{|\mathcal{H}|} - \sum_{f,g \in \mathcal{H}} \left[\Big(\varepsilon(f) - \varepsilon(g) \Big) \times \Big(v(f) - v(g) \Big) < 0 \right]$$

where $\Big[x \Big]$ is 1 when x is true and 0 otherwise, and $\mathcal{H}$ is the set of models. The first term $C^2_{|\mathcal{H}|}$ is the number of comparisons where $|\mathcal{H}|$ is the number of models and the second term indicates how many times the criterion selects the worse one among two models. This measure means that if one criterion can select the better model in the comparison, it gains a higher measure value. The main results can be found in Table 4 and 5.

Three data collections from three different domains are employed to evaluate the algorithm selection and parameter tuning by TrCV. Among them, Wine Quality dataset [2] contains two subsets related to red and white variants of the Portuguese "Vinho Verde" wine. The task is to classify wine's quality according to their physicochemical variables. In the experiment, red-wine set and white-wine set are treated as source-domain and target-domain alternately. Reuters-21578 [2] is the primary benchmark of text categorization formed by different news with a hierarchial structure. It contains five top categories of news wire articles, and each main category contains several subcategories. Three top categories, "orgs", "people" and "places" are selected in the study. All of the subcategories from each category are divided into two parts, one source-domain and one target-domain. They have different distributions and are approximately equal in size. The learning objective aims to classify articles into top categories. SyskillWebert [2] is the standard dataset used to test web page ratings, generated by the HTML source of web pages plus the user rating ("hot" or "not hot") on those web pages. It contains four separate subjects belonging to

Table 4. Algorithm Selection and Parameters Tuning

Method	RW	WR	ope	opl	pp	Sp	Bl	Gs	RW	WR	ope	opl	pp	Sp	Bl	Gs	RW	WR	ope	opl	pp	Sp	Bl	Gs
	Algorithm Selection								Parameter Tuning (LatentMap)								Parameter Tuning (SVM)							
SCV	18	17	13	17	13	19	16	17	4	5	5	5	**8**	4	**4**	6	4	7	5	4	3	7	7	**8**
TCV	17	18	14	17	10	15	10	11	3	3	3	5	5	4	1	2	5	4	3	4	4	4	5	5
STV	16	15	13	15	14	18	**17**	**20**	4	5	4	4	7	**8**	1	6	4	7	4	7	3	**8**	7	5
WCV	20	19	17	19	18	18	15	15	4	5	5	**8**	**8**	4	3	**7**	**8**	7	6	6	5	**8**	6	7
TrCV	**22**	**23**	**22**	**20**	**22**	**20**	15	18	**5**	**7**	**8**	**8**	**8**	5	3	**7**	7	**8**	**7**	**8**	**6**	**8**	**8**	**8**

Table 5. Source-domain Selection

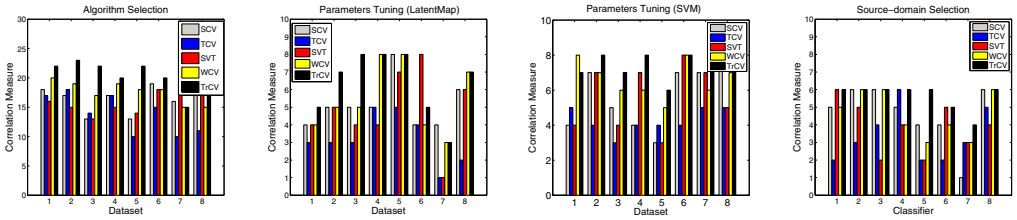
Method	NB	SVM	C45	KNN	Ng	TA	LM	LWE	*Pr*
SCV	5	**6**	**6**	5	4	4	1	**6**	436
STV	2	3	4	**6**	2	2	3	5	371
TCV	**6**	5	2	4	2	**5**	3	4	399
WCV	5	**6**	**6**	4	3	4	3	**6**	442
TrCV	**6**	**6**	**6**	**6**	**6**	**5**	**4**	**6**	**512**

different topics. The learning task is to predict the user's preferences for the given web pages. In the experiment, we randomly reserve "Bands-recording artists" as source-domain and the three others as target-domain data. The details of datasets are summarized in Table 2. These datasets are chosen because they are highly representative of the real world data we typically encounter. For example, some of them have few instances but have high dimensions, while others have the opposite. In addition, to evaluate the performance of source-domain selection with TrCV, 20-Newsgroup [2] is chosen. It is another primary benchmark of text categorization similar to Reuters-21578. In our study, 16 subcategories from 4 top subjects, including "comp", "rec", "sci" and "talk", are selected to form 8 different datasets of two tasks, "comp vs. rec" and "sci vs. talk". Data of source-domain and target-domain come from the same top categories but different sub-topics. As shown in Table 3, for "comp vs. rec" task, "graphics vs. autos" is chose as the target-domain and three others are treated as source-domains. Similarly, "electronics vs. mideast" is target-domain in "sci vs. talk" task, while others are source-domains. Moreover, for SyskillWebert, Reuters-21578 and 20-Newsgroup, only 500 features with highest information gains are selected.

5.2 Experiment Procedure

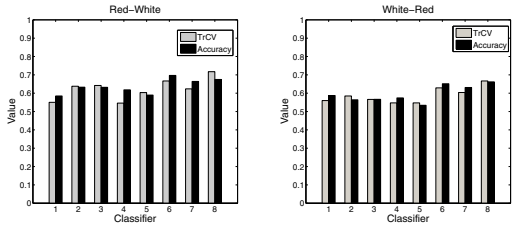
Selection among Different Algorithms. As a comparison, the parameters of traditional classifiers are set as the default values in Weka[1] and those of transfer learning approaches are chosen as the values which are suggested in the corresponding papers. In addition, for TrAdaBoost, SVM with polynomial kernel is set as base model; for LWE, five traditional classifiers stated above with default parameters are the base models. There are 8 approaches, thus the number of comparison is $C_8^2 = 28$. Table 4 and Figure 4(a) present correlation measure values for each domain transfer datasets, given by five competitive approaches:

[1] `www.cs.waikato.ac.nz/ml/weka/`



(a) (b) (c) (d)

Fig. 4. The comparison of TrCV with other validation methods



(a) (b)

Method	NB	SVM	C45	KNN	Ng	TA	LM	LWE
Red-White								
TrCV	0.550	0.638	0.642	0.546	0.603	0.667	0.624	0.717
Accuracy	0.585	0.633	0.632	0.618	0.590	0.697	0.664	0.675
White-Red								
TrCV	0.560	0.585	0.566	0.547	0.547	0.629	0.604	0.667
Accuracy	0.588	0.564	0.567	0.574	0.534	0.651	0.631	0.662

Fig. 5. The comparison between TrCV's accuracy and the true accuracy

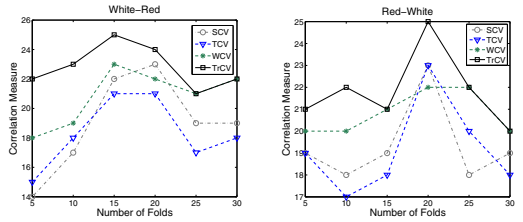
SCV, TCV, STV, WCV and TrCV. Dataset $1 \sim 8$ correspond to those in Table 4. It is evident that TrCV achieves the best performance in 6 out of 8 runs. Due to "distribution gap" between source and target domains, SCV fails to select the better model among the comparisons most of the time. To be specific, the correlation value got by SCV is just 18 on the Red-White set and no more than 17 on the Reuters collection. In addition, TCV and STV also fail to select a better model. This can be ascribed to the limited number of labeled data in the target-domain. One classifier performing well in this small subset can not guarantees its generalizability over the whole target-domain collection. However, the proposed approach, TrCV, which considers the difference on marginal distribution and conditional possibility between source and target domains, has a much better performance. Specifically, we notice that TrCV performs better than SCV by at least 4 in correlation value on Wine Quality collection, and as high as 9 on the Reuters-21578 collection. Moreover, the performance of TrCV is better than WCV consistently. The main reason is that although WCV reduces the difference of marginal distribution, it still selects those models which approach conditional distribution of source-domain in stead of target-domain. Thus, as analysed in the section 2, these models can not guarantee their performances in

target-domain. This also affords an evidence that reverse validation is necessary under the context of transfer learning.

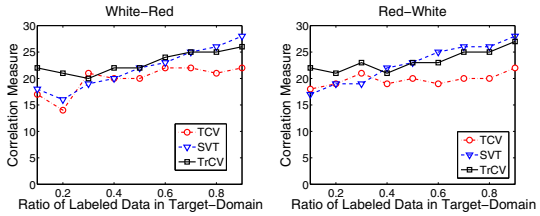
In addition, the advantage of TrCV over STV is that TrCV explores the power of unlabeled data and multiple validations, thus reducing the variance of the testing. As analyzed, these characteristics of TrCV reduce the distribution gap between two domains during algorithm selection. This, from the empirical perspective, provides justification to the analysis in Section 4. In addition, Figure 5 plots TrCV values and accuracies of each classifiers in Wine collection. It is intuitive that when one classifier achieves a higher TrCV value, it gets a higher accuracy with high confidence. In other words, accuracy obtained by TrCV is highly correlated to the true accuracy. Moreover, three transfer learning algorithms beat those traditional classifiers because they accommodate the distribution gap between two domains. Classifier $1 \sim 8$ in the figure correspond to those list in the table.

Parameter Tuning. We select SVM and LatentMap as the learning models and generate two tasks. The first one is to select a suitable margin parameter C for SVM (from 10^{-2} to 10^{2}) and the second one is to tune a good number of nearest neighbors for LatentMap (from 5 to 45). The size of these two parameter set is 5, so we get $C_5^2 = 10$ comparisons. Table 4 and Figure 5(b) and (c) summarize the correlation values of baselines: SCV, TCV, STV and WCV and the proposed criteria TrCV on 8 datasets. Clearly, TrCV achieves higher correlation value (from 1 to 4 higher in 6 out of 8 datasets) than the corresponding baseline approaches on tuning the parameters of LatentMap and performs best in 7 out of 8 cases when we adjust the margin parameter in SVM. For example, on the Red-White dataset, the correlation value has been improved from 5 achieved by SCV and WCV to 7 by the proposed TrCV. More importantly, in total 16 comparisons, TrCV beats WCV consistently with only one exception in RW dataset when tuning margin parameter of SVM. On the other hand, two exceptions happened on the SyskillWebert collection. We observe that TrCV fails to tune the best parameters for LatentMap and does not have significant improvements to tune SVM. This can be ascribed to the limited number of data in both domains that makes the density ratio estimation imprecise and the reverse validation can not reflect the approximation error to the true conditional distribution significantly as shown in Eq.(13).

Source-domains selection. We aim to select a best source-domain among multiple candidates. Two comparisons are involved. One is to evaluate the ability of TrCV to select among source-domains when the model is fixed, another is to test whether TrCV can select the best pair of source-domain and classifier given a set of classifiers and a set of source-domains. The result is presented in Table 5 and Figure 5(d). For the first evaluation, both datasets have 3 candidate source-domains, thus the number of comparison is $2 \times C_3^2 = 6$. Among them, TrCV achieves the best performance over all 8 tasks in the correlation measure. In particular, TrCV beats SCV by as much as 5 times while it defeat WCV by 7 times. Table 5 also presents the second evaluation results over 2 data collections,



(a) Different number of folds



(b) Different number of labeled data in T

Fig. 6. Parameter Analysis

where $2\times C^2_{(8\times 3)} = 552$ comparisons are obtained. We denote the result of this comparison as "Pr". Obviously, under this setting, TrCV still performs better than SCV, STV, WCV and TCV over two datasets, implying the TrCV can still select the best pairs of source-domains and algorithms. The performance improvement is due to density ratio weighting and reverse validation that effectively accommodate the difference between two domains. For WCV, although it boost the ability of SCV with density ratio weighting, it does not perform well due to the ignoring the conditional distribution shift.

Parameter Analysis. Two extended experiments were conducted on the Wine Quality collection to test the parameter sensitivity and the relationship between the number of labeled target-domain data and correlation value, *corr*. As shown in Section 3, the number of folds need be set before running TrCV. In addition, those labeled target-domain data affect the accuracy of TrCV to selecting a good model or a source-domain as shown in Eq.(13).

For sensitivity testing, we vary the value of folds from 5 to 30 with step size 5 to perform algorithm selection over 8 candidate approaches. As a comparison, we also attach the results obtained by SCV, TCV and WCV. The results are presented in Figure 5(a). Obviously, TrCV achieves the highest correlation value under all settings. This clearly demonstrates TrCV's advantage over SCV, TCV and WCV. In addition, we test TrCV when the number of labeled data ℓ increases from $0.1\times|T|$ to $0.9\times|T|$ by comparing with TCV, SVT. $|T|$ is the number of data in target-domain. The results are presented in Figure 5(b). Overall, three criteria achieve a higher value with more labeled data and SVT performs better than TrCV when the number of labeled data is significantly large. With more labeled data in target-domain, SVT can obtain more precise estimate to the prediction accuracies of remaining target-domain data. However, when only a few labeled data($< 0.4 \times |T|$) can be obtained in the target-domain, the performance of TrCV is much better than both SVT and TCV.

6 Related Work

Many solutions for transfer learning have been proposed previously, such as but not limited to [12–14], while few approach has been studied to select models or source-domains. Though several existing standard techniques [4–6] can be applied for model selection, they fail to work in transfer learning due to the distribution shift between source and target domains. Two recent approaches [8, 15] have been proposed for model selection in covariant shift or sample selection bias. The method in [8] "WCV" adapts the density ratio into cross validation to handle unbias estimation under covariant shift. The technique described in [15] performs "Reverse Testing" to select model under sample selection bias. We notice that "Reverse Testing" evaluates or rather "orders" the ability of one model based on another model and does not apply density ratio weighting that returns an estimated value, that is different from the method proposed in this paper. In addition, both of them do not consider the conditional distribution shift which may make them fail under transfer learning context as demonstrated in Section 2.1. Beside these, some techniques have been proposed to estimate the density ratio directly, including Kullback-Leibler importance estimation procedure [9] and nonparametric kernel mean matching (KMM) method [10]. The former one finds the density ratio to minimize the KL-divergence between two domains while the latter estimates by making the discrepancy between means of two domains small. On the other hand, works in [16] solved the similar problems under the context of meta-learning, including algorithm selection, parameter tuning and dataset selection.

7 Conclusion

Several challenges need to be resolved in order to make transfer learning methods practical: algorithm selection, parameter tuning and source-domain data selection. Traditional approach fails to solve these problems well due to the distribution gap between two domains. This paper firstly formulates a general criterion followed by proposing a transfer cross validation (TrCV) method. It works by applying density weighting to reduce the difference between marginal distributions of two domains, as well as utilizing reverse validation to measure how well a model approximates the true conditional distribution of target-domain. Formal analysis demonstrates that the newly proposed general criterion has a generalization bound on target-domain, and the confidence of transfer cross validation can also be bounded. Empirical studies under different tasks demonstrate that TrCV has higher chance to select the best models, parameters or source-domains than traditional approaches. In summary, it achieves the best in 28 out of 33 cases comparing with all baselines. Importantly, by considering both marginal and conditional distribution shift, the proposed TrCV approach outperforms in 23 out of 33 cases than WCV [8], a recently proposed method that only considers marginal distribution but ignores difference in conditional distribution.

References

1. Pan, S.J., Yang, Q.: A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China (November 2008)
2. Asuncion, A., Newman, D.J.: uci machine learning repository (2007), `http://www.ics.uci.edu/mlearn/MLRepository.html`
3. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research 5, 361–397 (2004)
4. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (2003)
5. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics 6(2), 461–464 (1978)
6. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
7. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. Journal of Statistical Planning and Inference 90(2), 227–244 (2000)
8. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. Journal of Machine Learning Research 8, 985–1005 (2007)
9. Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P.V., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: NIPS '07: Proceedings of the 2007 Conference on Advances in Neural Information Processing Systems, vol. 20, pp. 1433–1440. MIT Press, Cambridge (2008)
10. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: NIPS '06: Proceedings of the 2006 Conference on Advances in Neural Information Processing Systems, vol. 19, pp. 601–608. MIT Press, Cambridge (2007)
11. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 1137–1143. Morgan Kaufmann Publishers Inc, San Francisco (1995)
12. Dai, W., Yang, Q., Xue, G.R., Yu, Y.: Boosting for transfer learning. In: ICML '07: Proceedings of the 24th International Conference on Machine Learning, pp. 193–200. ACM, New York (2007)
13. Xie, S., Fan, W., Peng, J., Verscheure, O., Ren, J.: Latent space domain transfer between high dimensional overlapping distributions. In: WWW '09: Proceedings of the 18th International Conference on World Wide Web, pp. 91–100. ACM, New York (2009)
14. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 283–291. ACM, New York (2008)
15. Fan, W., Davidson, I.: Reverse testing: an efficient framework to select amongst classifiers under sample selection bias. In: KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 147–156. ACM, New York (2006)
16. Brazdil, P., Giraud-Carrier, C., Soares, C., Vilalta, R.: Metalearning: Applications to Data Mining. In: Cognitive Technologies. Springer, Heidelberg (2009)