Largest Source Subset Selection for Instance Transfer

Shuang Zhou SHUANG.ZHOU@MAASTRICHTUNIVERSITY.NL
Gijs Schoenmakers GM.SCHOENMAKERS@MAASTRICHTUNIVERSITY.NL
Evgueni Smirnov SMIRNOV@MAASTRICHTUNIVERSITY.NL
Ralf Peeters RALF.PEETERS@MAASTRICHTUNIVERSITY.NL
Kurt Driessens KURT.DRIESSENS@MAASTRICHTUNIVERSITY.NL
Department of Knowledge Engineering,
Maastricht University,
P.O. Box 616, 6200 MD, Maastricht, The Netherlands

Siqi Chen SIQI.CHEN09@GMAIL.COM
School of Computer and Information Science
Southwest University, Chongqing, China

Editor: Geoffrey Holmes and Tie-Yan Liu

Abstract

Instance-transfer learning has emerged as a promising learning framework to boost performance of prediction models on newly-arrived tasks. The success of the framework depends on the relevance of the source data to the target data. This paper proposes a new approach to source data selection for instance-transfer learning. The approach is capable of selecting the largest subset S^* of the source data which relevance to the target data is statistically guaranteed to be the highest among any superset of S^*. The approach is formally described and theoretically justified. Experimental results on real-world data sets demonstrate that the approach outperforms existing instance selection methods.

Keywords: instance transfer, subset selection, conformity test

1. Introduction

Instance-transfer learning has gained increasing attention (Pan and Yang, 2010) with goal of improving the prediction models for a *target* domain by exploiting data from (closely) related *source* domains. A thorough analysis of instance-transfer learning (Torrey and Shavlik, 2009) shows that its effectiveness depends on the relatedness of the source domain to the target domain. Therefore, a critical problem we may encounter in practice is how to properly select instances from the source data when training models on the target data. Adequately solving this problem is critical for the overall success of instance transfer.

Typically instance-transfer algorithms select source instances (explicitly/implcitly) based on their individual relevance to the target domain (Dai et al., 2007; Kamishima et al., 2009). However, the relevance of the selected source instances is usually not estimated as a set and thus this information is not used by the instance-transfer algorithms. In this paper we address the problem of estimating the set relevance of source instances. We propose a new source-set selection approach that we call **l**argest **s**ource **s**ubset **s**election (denoted as LSSS). The approach selects the largest subset S^* of the source instances which relevance to the target domain is guaranteed to be the highest among any superset of S^*.

The LSSS approach is based on the conformity framework (Vovk, 2014). It consists of two phases. In the first phase we decide whether the source domain is related to the target domain. For that purpose we statistically test the null hypothesis "the target instances and the source instances have been generated from the target distribution under the exchangeability assumption" (Aldous, 1985). To implement the test we propose a p-value function which validity is proven in the paper. The function computes p-values for the null hypothesis (i.e., high p-values indicate that source data are relevant to the target one). If the null hypothesis is not rejected, this an indication that the source domain is related to the target domain, and, thus, there are source instances that could be generated by the target distribution. Therefore, in this case we proceed with the second phase: we select the largest subset S^* of the source data which relevance to the target data is statistically guaranteed to be the highest among any superset of S^*. The selection process is described and theoretically justified. Experimental results on real-world data sets demonstrate that the LSSS approach outperforms existing instance-selection approaches.

The remainder of this article is as follows. Section 2 provides an overview of related work. The instance-transfer task is formalized in Section 3. Section 4 presents the LSSS approach. An experimental analysis is given in Section 5. Section 6 concludes the article.

2. Related Work

There exist several instance-transfer algorithms performing a selection of source instances. Most of them are boosting-based algorithms; e.g., TrAdaBoost (Dai et al., 2007) and Dynamic-TrAdaBoost (Al-Stouhi and Reddy, 2011). Those algorithms apply different update schemes on the weighted source and target instances. For the target instances, they increase the weights of misclassified ones using a reweighing factor based on the training error. For the source instances, they decrease the weight of misclassified ones by a constant factor set in accordance with the weighted majority algorithm. The average weighted training loss of boosting-based algorithms on the source data is guaranteed to converge to 0 as the number of performed iterations approaches infinity (Dai et al., 2007). However, when most of the source instances are irrelevant and the source-data size is much bigger than that of the target data, those algorithms are likely to stop at very first iterations due to the error on the target data exceeding 0.5. In this case, the selected set of source instances contains plenty of irrelevant source data and a negative transfer may occur.

Beyond boosting-based algorithms, TrBagg (Kamishima et al., 2009) employs an indirect selection of source instances as well. TraBagg includes two phases. In the training phase, first a set of bootstrap samples are generated from the combined target and source data, and, then several base prediction models are trained on those samples. In the filtering phase, a subset of the base prediction models are selected by minimizing the empirical error on the target data (i.e., source-instance selection is indirect through selecting the base models). Hence, if the bootstrap samples are not very relevant to the target data, instance transfer may become ineffective.

A double-bootstrapping instance-transfer algorithm was proposed by Lin et al. (2013). It first constructs an ensemble of prediction models trained on bootstrap samples from the target data. Then the ensemble classifies the source instances. A source instance is selected if it is correctly classified by majority voting of the ensemble prediction models. Hence, this approach is vulnerable to imbalanced-class distributions. In this case the ensemble simply labels all the instances by the majority class. Thus, only the source instances from the majority class are selected, and, the instance transfer becomes suboptimal.

Analyzing the instance-transfer algorithms considered we conclude that the relevance of the selected source instances as a set is not estimated and this information is not used by the algorithms. To overcome the aforementioned problem, in the next sections we propose an approach to robust source-subset selection.

3. Notations and Task Formalization

Let X be a feature space and Y be a class set. A domain is defined as a 2-tuple consisting of a labeled space $(X \times Y)$ and a probability distribution P over $(X \times Y)$. We consider a target domain $\langle (X \times Y), P_T \rangle$. The target data set T is a set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_{m_T}, y_{m_T})\}$ of m_T independently and identically distributed (i.i.d) instances drawn from the target distribution P_T. Given a test instance $x_{m_T+1} \in X$, the target classification task is to find an estimate $\hat{y} \in Y$ of the true class of x_{m_T+1} according to P_T.

Now consider a source domain $\langle (X \times Y), P_S \rangle$. Under the i.i.d assumption we generate a source data set S as a set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_{m_S}, y_{m_S})\}$ of m_S instances drawn from the source distribution P_S. Assuming that the target domain and the source domain are related, we define *the instance-transfer classification task* as a classification task with an auxiliary source data set S in addition to the target data set T. We note that the class of a new test instance is estimated according to the target distribution P_T. This implies that the source data has just to be used as auxiliary training data for the target task.

Instance-transfer learning is sensitive to the relevance of the source data to the target domain. Thus, the problem to select relevant source instances to the target task is important for the overall success of instance transfer.

4. Approach to Largest Source Subset Selection

In this section, we introduce our approach to the largest source subset selection (LSSS). The approach starts by deciding whether the source domain is relevant to the target domain by using a new conformity-based test presented in Subsection 4.1. If the test is positive, the approach proceeds with selecting the largest subset S^* of the source instances described in Subsection 4.2. If the test is negative, it stops to avoid negative transfer. The LSSS approach in its entirety is described in subsection 4.3.

4.1. Target-Domain Relevance

This subsection proposes a new non-parametric test to evaluate the relevance of the source data to the target data. The key idea is to test whether the target instances and the source instances are generated by the target distribution. Our test is a multi-instance extension of the Vovk's conformity test from (Shafer and Vovk, 2008; Vovk, 2014). Below we first describe the Vovk's test and then introduce our test in detail [1].

[1] We note that the tests are introduced for the case when the data are treated as sequences. At the end of subsection 4.1.2 we show how to extend the test for the case when the data are treated as sets.

4.1.1. Conformity Test for Prediction

The conformity test has been originally proposed by Shafer and Vovk (2008) for conformal prediction. Conformal prediction uses past experience to determine precise levels of confidence in new predictions in one domain (let say the target domain) (Vovk, 2014). Let us consider the target data T as a sequence of labeled training instances and let x_{m_T+1} be a new test instance. A conformal predictor provides an estimate $\hat{y}_{m_T+1}$ of the class for x_{m_T+1} by utilizing a conformity test for the null hypothesis "the sequence $T' = T \cup \{(x_{m_T+1}, \hat{y}_{m_T+1})\}$ is generated by the target distribution P_T under the exchangeability assumption" [2].

The test is based on nonconformity scores α_i of instances $(x_i, y_i) \in T'$. The nonconformity score α_i is a value indicating how unusual is instance (x_i, y_i) in the sequence T'. To compute a nonconformity score for an instance, we need an instance nonconformity function A. If $(X \times Y)^{(*)}$ denotes the set of all sequences defined over $(X \times Y)$, then the instance nonconformity function A is a mapping from $(X \times Y)^{(*)} \times (X \times Y)$ to $\mathbb{R}^+ \cup \{+\infty\}$ and it indicates how unusual is an instance (x_i, y_i) for the sequence $T' \setminus \{(x_i, y_i)\}$. We note that any instance nonconformity function has to produce the same result for an instance independently on the permutations of the sequence T' (otherwise, the instance will have $|T'|!$ possible nonconformity scores).

The nonconformity score α_{m_T+1} of the test instance $(x_{m_T+1}, \hat{y}_{m_T+1})$ is used as a test statistic. Under the null hypothesis, the p-value of the test is calculated as the fraction of the instances in T' that are associated with nonconformity scores that are as extreme as or more than α_{m_T+1}. The larger the p-value, the more likely is to observe this value of the test statistic under the null hypothesis, and the more confidence, therefore, we have in prediction $\hat{y}_{m_T+1}$.

4.1.2. Extended Conformity Test for Target-Domain Relevance

In this subsection we extend the Vovk's conformity test to multi-instance case; i.e., to the case of instance transfer. Given the target set T and source set S considered as sequences, we test the null hypothesis "the combined data sequence $TS = T \cdot S$ is generated by the target distribution P_T under the exchangeability assumption". Since in our setting the size of S is bigger than 1, we need to provide a nonconformity function A^* for any sequence.

Definition 1 (Sum Sequence Nonconformity Function) *Given an instance nonconformity function A, the combined data sequence TS, and a data sequence $U \subseteq TS$, the sum sequence nonconformity function A^* is a function from $(X \times Y)^{(*)} \times (X \times Y)^{(*)}$ to $\mathbb{R}^+ \cup \{+\infty\}$ defined equal to $\sum_{(x_i, y_i) \in U} \alpha_i$, where $\alpha_i = A(T, (x_i, y_i))$.*

Given the combined sequence TS and any sequence $U \subseteq TS$, the sum sequence nonconformity function returns a value $\alpha_U \in \mathbb{R}^+ \cup \{+\infty\}$ estimating how unusual U is with respect to all the elements from the set $\mathcal{P}(TS, |U|)$ of all the permutations of the combined sequence TS of size $|U|$. We note that this estimation employs information from the target data only. Hence, the sequence nonconformity score α_U can be used as a test statistic for the null hypothesis "the data sequence TS is generated by the target distribution P_T under the exchangeability assumption". To design the test below we propose a p-value function.

[2] The exchangeability assumption states that the joint probability distributions of a sequence of random variables and any of its permutations coincide. It is weaker than the i.i.d assumption.

Definition 2 **(p-value Function)** *Given a data sequence $U \in (X \times Y)^{(*)}$ and an integer $n \leq |U|$, the p-value function t is a function of type $t : (X \times Y)^{(*)} \times \mathbb{N} \to [0,1]$ equal to:*

$$t(U,n) = \frac{|\{V \in \mathcal{P}(U,n) | \alpha_V \geq \alpha_{L(U,n)}\}|}{|\mathcal{P}(U,n)|}.$$

where $L(U,n)$ is the sequence of the last n elements of U.

Given the combined data sequence TS and $n = m_S$, our function t returns a p-value equal to the proportion of permutations of m_S elements out of the sequence TS which nonconformity scores are greater than or equal to the nonconformity score of source data sequence S. Below in Theorem 3 we prove that function t is a p-value function.

Theorem 3 *If the sequence TS is exchangeable, then*

$$P\{t(TS, m_S) \leq r\} \leq r$$

Proof (Adapted from Saunder (2000)) Let TS be an exchangeable sequence and let $r \in [0,1]$. Since t can only take on values $\frac{j}{|\mathcal{P}(TS,m_S)|}$, where $j \in \{1, 2, \ldots, |\mathcal{P}(TS, m_S)|\}$, we assume w.l.o.g. the same for r, i.e. $r = \frac{j}{|\mathcal{P}(TS,m_S)|}$ for the appropriate value of j. Then:

$$\begin{aligned}
P\{t(TS, m_S) \leq r\} &= \frac{|\{U \in \mathcal{P}(TS, m_T + m_S)\ :\ t(TS, m_S) \leq r\}|}{|\mathcal{P}(TS, m_T + m_S)|} \\
&= \frac{\left|\left\{U \in \mathcal{P}(TS, m_T + m_S)\ :\ t(TS, m_S) \leq \frac{j}{|\mathcal{P}(TS,m_S)|}\right\}\right|}{(m_T + m_S)!} \\
&= \frac{\left|\left\{U \in \mathcal{P}(TS, m_T + m_S)\ :\ \left|\left\{V_{m_S} \in \mathcal{P}(TS, m_S) \mid \alpha_{V_{m_S}} \geq \alpha_{L(TS,m_S)}\right\}\right| \leq j\right\}\right|}{(m_T + m_S)!}
\end{aligned}$$

Now let $S_j(TS)$ be the following subset of $\mathcal{P}(TS, m_T + m_S)$: $U \in S_j(TS)$ if and only if there are at most j (sub-)sequences $V_{m_S} \in \mathcal{P}(TS, m_S)$ that have a nonconformity score $\alpha_{V_{m_S}} \geq \alpha_{L(U,m_S)}$. Say that there are $k(\leq j)$ such subsequences. For each of those k subsequences there are $m_T!$ ways to extend them to a sequence of length $m_T + m_S$ (by 'prefixing' them with a sequence of the length m_T). This means that $|S_j(TS)| = k \cdot m_T! \leq j \cdot m_T!$ with a possible strict inequality if there are multiple sequences V_{m_S} that have identical nonconformity scores. We have:

$$\begin{aligned}
P\{t(TS, m_S) \leq r\} &= \frac{|S_j(TS)|}{(m_T + m_S)!} \\
&\leq \frac{j \cdot m_T!}{(m_T + m_S)!} \\
&= \frac{j}{(m_T + m_S) \cdot (m_T + m_S - 1) \cdot \ldots \cdot (m_T + 1)} \\
&= \frac{j}{|\mathcal{P}(TS, m_S)|} = r
\end{aligned}$$

which completes the proof. ■

The p-value function t provides an indication of the evidence against the null hypothesis "the data sequence TS is generated by the target distribution under the exchangeability assumption", since it is the probability of observing a value of the nonconformity score of the source sequence as extreme as or more than the observed value under the null hypothesis. The higher the p-value is, the weaker is the evidence against the null hypothesis, and thus the more confident we feel to transfer the source data. Thus, *the p-value can be viewed as a measure of relevance of the source data to the target data.*

The p-value function t has been defined for data sequences. Below, we re-write the function definition for the case of data sets.

$$\begin{aligned} t(U,n) &= \frac{|\{V \in \mathcal{P}(U,n) | \alpha_V \geq \alpha_{L(U,n)}\}|}{|\mathcal{P}(U,n)|} \\ &= \frac{|\{V \in \mathcal{P}(U,n) | \alpha_V \geq \alpha_{L(U,n)}\}|/(|U|-n)!}{|\mathcal{P}(U,n)|/(|U|-n)!} \\ &= \frac{|\{V \in \mathcal{C}(U,n) | \alpha_V \geq \alpha_{L(U,n)}\}|}{|\mathcal{C}(U,n)|} \end{aligned} \quad (1)$$

where $\mathcal{C}(U,n)$ denotes the set of all combinations of n elements out of sequence U.

We note that the number of combinations is independent from the order of the sequence U, and the sum sequence nonconformity function A^* will compute the same nonconformity score for all permutations that have the same set of elements as the last n elements of U. Therefore, this equivalent definition of the p-value function t can be applied to data sets and it is used in the rest of the paper. We note that for large data sets our p-value function t can be approximated using the one-sided rank-sum test.

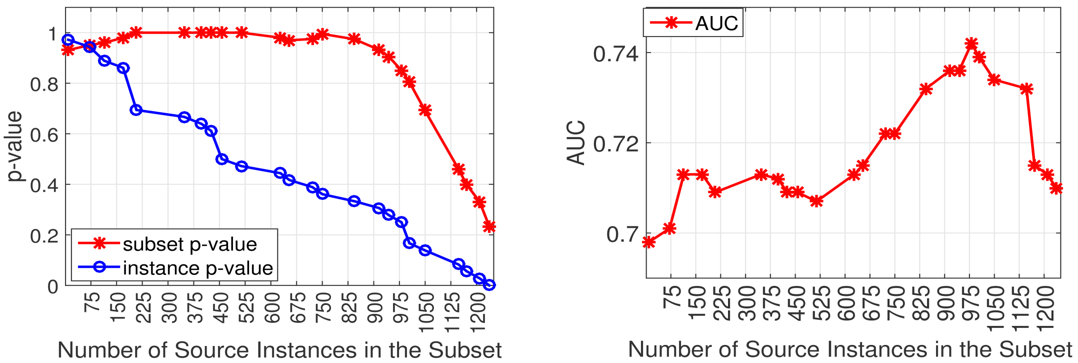
4.2. Largest Source Subset Selection

In this subsection we introduce largest source subset selection. For that purpose we first analyze the p-value function t. Then we define the largest source subset and analyze its properties. Finally, we propose a procedure how to compute that set.

4.2.1. Analyzing the p-Value Function t

Assume that we sort the instances from the source set S in increasing order of the nonconformity scores. Then, we add source instances with the lowest nonconformity score to a preliminary empty source subset and compute the subset p-value (using the function t). We repeat the last step till all the source instances from S have been added and we plot the obtained subset p-values against the size of the source subsets. An example for the "orgs vs people" task defined on Reuters-21578 (see Subsection 5.2) is given in Figure 1(a).

We repeat the process above for the same sorted order of the source instances. However, instead of computing subset p-values we compute individual p-value for each instance. This is done using the p-value function t for $n = 1$ (see definition 2). We plot the obtained instance p-values for the "orgs vs people" task in Figure 1(a). Comparing the instance



(a) (b)

Figure 1: (a): Instance p-values and subset p-values w.r.t. source instances ordered by the nonconformity scores. (b): AUCs of SVM trained on the target data and growing subsets of source instances ordered by the nonconformity scores.

p-value curve and subset p-value curve we observe that the subset p-values are much bigger than instance p-values. Why it can be a problem we can see in Figure 1(b). The figure shows the plot of the generalization performance of a prediction model (SVM) trained on the target data and growing source subsets based on the same sequence of the sorted source instances. The maximization of the model performance happens for instance p-value of 0.27 and subset p-value of 0.85; i.e., the subset p-value is a way higher than instance p-value. Therefore, if we follow a very confident policy of selecting source instances, we will be able to achieve successful instance transfer only with subset p-values. Using instance p-value we will stop much earlier with almost no benefit from instance transfer.

Another look on the subset p-value in Figure 1(a) shows that the p-value function t is non-monotonic with the size of the ordered source data. Therefore, if we need to select the biggest subset of source instances with the highest set p-value using our p-value function t, the selection procedure becomes brute-force. On each iteration the procedure adds the unvisited source instances with the lowest nonconformity score to the source subset and then computes the subset p-value (see Figure 1(a)). It stops before the first decrease of the subset p-value, and, thus, outputs the biggest source subset with the highest p-value. While linear in the number of source instances, this procedure requires $|S|$ runs of the p-value function t at the worst case. Thus, it is impractical especially for very large source data.

In the next subsections we introduce largest source subsets. These subsets are usually larger than the biggest source subsets with the highest p-value and have a lower p-value. However, the procedure for the largest source subsets is more efficient. In addition, these subsets provide a good balance between the subset size and p-value.

4.2.2. Largest Source Subsets

Assume that the instances in the target set T and source set S are ordered in increasing order of magnitude of the nonconformity scores, and $\alpha^t_{m_T}$ and $\alpha^s_{m_S}$ are the highest nonconformity scores in T and S, respectively. Then, we define the largest source subsets as follows:

Definition 4 (Largest Source Subset) *Given the p-value function t, a significance level ϵ, the ordered target data set T and the source data set S, assuming $t(T \cup S, m_S) \geq \epsilon$, a subset $S^* \subseteq S$ is the largest source subset if $S^* = \{(x_1^s, y_1^s), \dots (x_j^s, y_j^s), (x_{j+1}^s, y_{j+1}^s)\}$, where (x_j^s, y_j^s) is the source instance with the largest nonconformity score α_j^s such that $\alpha_j^s < \alpha_{m_T}^t$.*

Below we prove that for the largest source subset S^* we have the following property:

Theorem 5 *If $S^{'} \subseteq S$ such that $S^{'} \supset S^*$, then $t(TS^{'}, |S^{'}|) \leq t(TS^*, |S^*|)$ (where $TS^{'} = T \cup S^{'}$ and $TS^* = T \cup S^*$).*

According to Theorem 5, there does not exist any proper superset of S^* that corresponds to a bigger p-value. In this context *the set S^* is the largest in the sense that adding any new source instance which nonconformity score equal to or bigger than the largest target nonconformity score will decrease (probably strictly) the p-value.* Thus, the p-function t becomes monotonic after the largest source subset S^*.

Theorem 5 follows from the following two lemmas. Assuming that $\alpha_D = A^*(T, D)$ is a nonconformity score for a set D w.r.t T [3], we define S_i as a subset of i number of instances of ordered S, and D_i as a set of subsets D of $T \cup S_i$ which size is i and has a nonconformity score bigger than that of S_i. Formally,

- $S_i = \{(x_1^s, y_1^s), (x_2^s, y_2^s), \cdots, (x_i^s, y_i^s)\}$;
- $D_i = \{D \subset T \cup S_i : |D| = i, \alpha_D \geq \alpha_{S_i}\}$.

We will start with the following lemma that relates $|D_i|$ to $|D_{i+1}|$:

Lemma 6 *If $\alpha_i^s \geq \alpha_{m_T}^t$, then $|D_{i+1}| \leq (1 + \frac{m_T}{i+1}) \cdot |D_i|$.*

Proof Let D be a subset of $T \cup S_i$ with $|D| = i$. Now we consider the set $T \cup S_{i+1}$ and we will add one element of $(T \cup S_{i+1}) \backslash D$ to D to create a set G of size $i + 1$. We distinguish between two cases:

1. $D \notin D_i$ or $\alpha_D < \alpha_{S_i}$. There are $m_T + 1$ ways to create the set $G \subset T \cup S_{i+1}$. For all of these sets we have

   $$\alpha_G \leq \alpha_D + \alpha_{i+1}^s < \alpha_{S_i} + \alpha_{i+1}^s = \alpha_{S_{i+1}}$$

   and hence $G \notin D_{i+1}$.

2. $D \in D_i$ or $\alpha_D \geq \alpha_{S_i}$. Again, there are $m_T + 1$ ways to create the set $G \subset T \cup S_{i+1}$. One way to create G is to add (x_{i+1}^s, y_{i+1}^s) to D. This gives $G = D \cup \{(x_{i+1}^s, y_{i+1}^s)\}$, and in this case

   $$\alpha_G = \alpha_D + \alpha_{i+1}^s \geq \alpha_{S_i} + \alpha_{i+1}^s = \alpha_{S_{i+1}}$$

   and hence $G \in D_{i+1}$.

   The other m_T ways to create G are by adding one of the m_T elements of $(T \cup S_i)\backslash D$ to D, thus there are in total $m_T \cdot |D_i|$ number of G's. However, the resulting G will

[3] The sequence nonconformity function returns the same nonconformity score for all permutations of elements of a set. Thus, it can be also used to calculate nonconformity scores for sets.

be created as a superset of size $i+1$ of in total $i+1$ sets of size i. So, assuming that the newly created set G satisfies $\alpha_G \geq \alpha_{S_{i+1}}$, which is not necessarily the case, it will be created $i+1$ times and it should of course count only once towards $|D_{i+1}|$. In other words, there are at most $\frac{m_T}{i+1} \cdot |D_i|$ number of distinct G's such that $\alpha_G \geq \alpha_{S_{i+1}}$. For example, a set $G = \{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ could be resulting from $D = \{(x_1, y_1), (x_2, y_2)\}$ by adding (x_3, y_3), or $D = \{(x_1, y_1), (x_3, y_3)\}$ by adding (x_2, y_2), or $D = \{(x_2, y_2), (x_3, y_3)\}$ by adding (x_1, y_1). Since the order of elements does not matter, there is only one distinct G rather than three.

Combining the two results, we find: $|D_{i+1}| \leq (1 + \frac{m_T}{i+1}) \cdot |D_i|$, which completes the proof. ■

Let TS_i be the union of the target set T and the subset S_i as defined above, and TS_{i+1} be the union of the target set T and the subset S_{i+1}, we have:

Lemma 7 *If $\alpha^s_i \geq \alpha^t_{m_T}$, then $t(TS_i, i) \geq t(TS_{i+1}, i+1)$*

Proof Let D_i be the set of subsets D of $T \cup S_i$ which size is i and has a nonconformity score bigger than that of S_i, and assume that $\alpha_{s_i} \geq \alpha_{t_{m_T}}$. We have:

$$\begin{aligned}
t(TS_{i+1}, i+1) &= \frac{|\{D \in \mathcal{C}(TS_{i+1}, i+1) | \alpha_D \geq \alpha_{S_{i+1}}\}|}{|\mathcal{C}(TS_{i+1}, i+1)|} && \text{by Equation 1} \\
&= \frac{|D_{i+1}|}{\binom{m_T+i+1}{m_T}} \\
&\leq \frac{\left(1 + \frac{m_T}{i+1}\right)|D_i|}{\binom{m_T+i+1}{m_T}} && \text{by Lemma 6} \\
&= \frac{\left(1 + \frac{m_T}{i+1}\right)}{\binom{m_T+i+1}{m_T}} \cdot \binom{m_T+i}{m_T} \cdot t(TS_i, i) \\
&= \frac{\left(1 + \frac{m_T}{i+1}\right)(i+1)}{m_T + i + 1} \cdot t(TS_i, i) \\
&= t(TS_i, i)
\end{aligned}$$

■

4.2.3. Procedure for Selecting Largest Source Subsets

The procedure for selecting largest source subsets assumes the presence of an instance nonconformity function A, the target data T of size m_T and the source data S of size m_S. It executes the following steps. First, the procedure initializes the largest source subset S^* as an empty set. Then, it computes the nonconformity scores for all the target and source instances using the instance nonconformity function A. More precisely, the nonconformity score α^t_i for each *target* instance (x_i, y_i) is calculated w.r.t. $T \setminus \{(x_i, y_i)\}$, and the nonconformity score α^s_i for each *source* instance (x_i, y_i) is calculated w.r.t. T. After that the selection procedure determines the target instance with the highest nonconformity score $\alpha^t_{m_T}$ (see definition 4). Then the procedure adds to the largest source subset S^* all the source instances with non-conformity scores lower than $\alpha^t_{m_T}$ plus the source instance with the smallest nonconformity score equal to or greater than $\alpha^t_{m_T}$. Once the subset S^* has been set, the function t is called to compute the p-value of S^*.

The procedure for selecting largest source subsets is efficient. It is linear with the number of target and source instances, and it employs only one run of the p-value function t.

The largest source subset S^* can be directly added to the target data. In case the class distribution in S^* is imbalanced, we propose to balance it by manipulating instance weights using the information from the true positive rates of the classes of the models employed.

4.3. Approach

Once all the components of our approach to largest source subset selection (LSSS) have been introduced, we describe the approach itself. Given a significance level ϵ, LSSS first applies the extended conformity test to decide whether the source data is relevant to the target data. If the p-value of the test is smaller than ϵ, the whole source data set is discarded, since it may lead to a negative transfer. When the p-value is higher than ϵ, part of the source instances are likely to be generated by the target distribution. In this case, LSSS proceeds with selecting the largest source subset S^*. The set S^* is added to the target data T and then a base prediction model is trained on the combined data $T \cup S^*$.

5. Experiments

This section presents our experimental results and conclusions. We first provide the experiment setup in Subsection 5.1. The instance-transfer tasks under study are presented in Subsection 5.2. The experiments are partitioned into two parts. In *Part I* shown in Subsection 5.3, we study whether the p-value function t provides good estimates of the relevance of source data to target data. Then, in *Part II* given in Subsection 5.4, we compare the generalization performance of the LSSS approach with existing instance-transfer techniques.

5.1. Experiment Setup

To perform the experiments we needed to set up both components of the LSSS approach: the conformity test for target-domain relevance (Subsection 4.1.2) and the procedure for selecting largest source subsets (Subsection 4.2.3). To set up these components we needed to choose only the instance nonconformity function [4]. We chose that function to be the general instance nonconformity function defined in (Shafer and Vovk, 2008). The general instance nonconformity function A_G first trains a prediction model on a target data T and then outputs for an instance (x_i, y_i) a nonconformity score equal to $\sum_{y \in \mathbf{Y}, y \neq y_i} s_y$, where s_y is the score of class $y \in Y$ produced by the model for x_i. In our experiments we employed Random Forest for the prediction model of the function A_G.

The conformity test for target-domain relevance was done on a significance level $\epsilon = 0.2$. The base prediction model employed by the LSSS approach was SVM (default setting).

[4] The sum sequence nonconformity function is set once the instance nonconformity function is chosen.

The LSSS approach was compared experimentally with four instance-transfer algorithms (from Section 2): TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap. All those algorithms used SVM (default setting) as a base prediction model.

The evaluation method was a stratified holdout method on the target data. The holdout method was repeated 100 times. The performance of the instance-transfer classifiers was evaluated using the Area Under the ROC Curve (AUC).

5.2. Instance-Transfer Classification Tasks

Three real-world data sets were employed in our experiments. They are described below:

• the landmine detection (Al-Stouhi and Reddy, 2011) is a collection of 29 data sets related to detecting landmine in 29 different landmine fields. The 29 data sets have different distributions due to different geographic conditions. For example, data sets 1 to 15 correspond to foliated regions while sets 16 to 29 correspond to regions that have bare earth. In this context we derived target and source data sets as follows. Data sets 26 to 29 were combined together and used as the target data set. Data sets 16 to 20 and 21 to 25 were combined into two source data sets with a high similarity to the target one while data sets 1 to 5, 6 to 10, and 11 to 15 were combined into other three source data sets with a lower similarity. The target data set and a source data set defined together one instance-transfer classification task. For each task, 10% of instances were randomly sampled from the target for training and the remaining for testing. The p-values of the relevance of the source data to the target data (computed by function t) are given in the last column of Table 1.

Datasets		Description	Size	p-value
Landmine	T	instances from Mine 26 to 29	1799	1.0
	S1	instances from Mine 1 to 5	3086	0.174
	S2	instances from Mine 6 to 10	2547	0.274
	S3	instances from Mine 11 to 15	2902	0.237
	S4	instances from Mine 16 to 20	2240	0.465
	S5	instances from Mine 21 to 25	2246	0.446

Table 1: Landmine instance-transfer classification tasks.

• the 20-Newsgroups (Dai et al., 2007) is a data set of about 20,000 news documents organized in a two-level hierarchy. The hierarchy consists of 7 top categories and 20 subcategories. For example, 'comp' and 'sci' are two top categories such that 'comp' has two subcategories, 'comp1' and 'comp2', and 'sci' has two subcategories, 'sci1' and 'sci2'. Five instance-transfer classification tasks were defined as top-category tasks such that the target and source data were drawn from different subcategories. For each task 50 instances were randomly sampled from the target data for training and the remaining for testing. The p-values of the target relevance of the source data are given in the last column of Table 2.

• the Reuters-21578 (Dai et al., 2007) is a collection of data sets with text documents organized in hierarchical structures. Three instance-transfer classification tasks were defined in the same way as those of the 20-newsgroups task. For each task 50 instances were randomly sampled from the target data for training and the remaining for testing. The p-values of the target relevance of the source data are given in the last column of Table 3.

Datasets	Tasks	Sample Size		p-value
		$\|T\|$	$\|S\|$	
20-Newsgroups	comp vs sci	3930	4900	0.303
	comp vs talk	4482	3652	0.390
	rec vs sci	3961	3965	0.343
	rec vs talk	3669	3561	0.320
	sci vs talk	3374	3828	0.340

Table 2: 20-Newsgroups instance-transfer classification tasks.

Datasets	Tasks	Sample Size		p-value
		$\|T\|$	$\|S\|$	
Reuters	orgs vs people	1016	1046	0.372
	orgs vs places	1079	1080	0.272
	people vs places	1239	1210	0.146

Table 3: Reuters-21578 instance-transfer classification tasks.

5.3. Experiment Part I: Effectiveness of the p-Value Function t

In this subsection, the proposed p-value function t is evaluated with respect to its effectiveness in predicting the relevance of the source data to the target data. The generalization performance of the four classic instance-transfer algorithms in aforementioned tasks is used as an indicator of the effectiveness of the function t.

Figure 2 presents the results for the landmine-detection tasks. It shows the correspondence between the p-values of the target and source data and the AUC performance of the four aforementioned instance-transfer prediction models. More precisely, on the x-axis, $S_i(i = 1, ..., 5)$ represents source data sets from 1 to 5, sorted by associated p-value in increasing order of magnitude from left to right. The plots show the average AUC performance of the instance-transfer models on the corresponding classification tasks. The performance of the SVM classifier for the case of no instance transfer is given as baseline. The plots clearly show that the instance transfer achieves better results on the instance-transfer classification tasks associated with higher p-values produced by our p-value function.

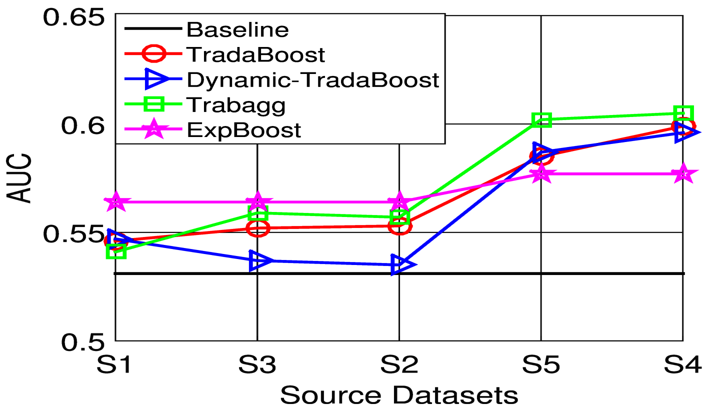
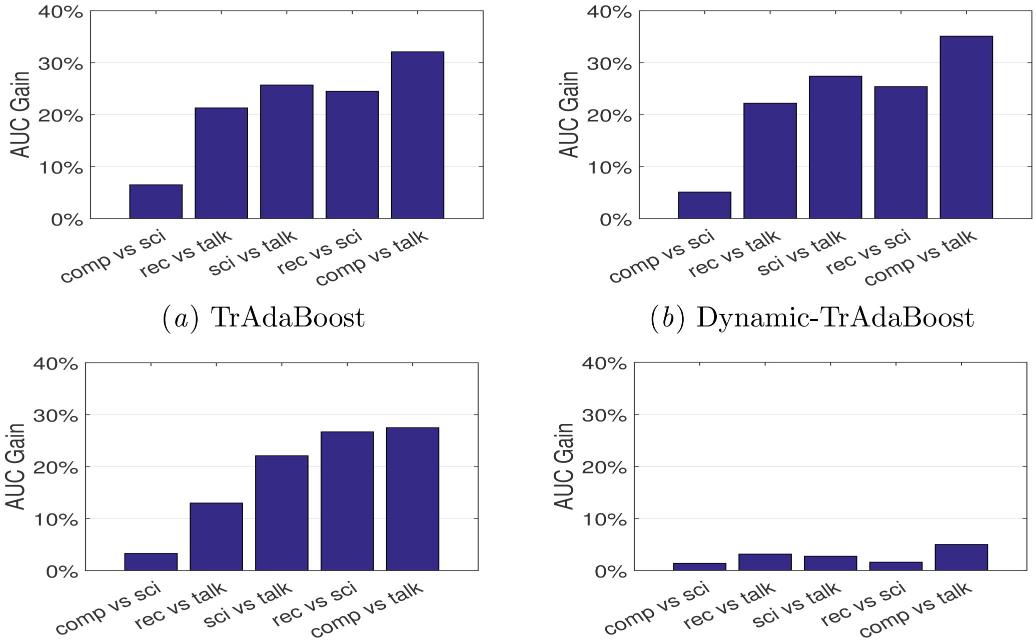


Figure 2: AUCs of the four instance-transfer algorithms on the Landmine tasks.

Figure 3 presents the results for the 20-newsgroups instance-transfer classification tasks. It shows indirectly the correspondence between the p-values of the target and source data and the AUC performance of the instance-transfer classifiers. More precisely, on the x-axis, the instance-transfer text classification tasks are sorted according to the associated p-values in increasing order of magnitude. On the y-axis is the gain in AUC of the instance-transfer classifiers w.r.t. SVM employed as a baseline classifier for the case of no transfer. Sub-figures 3(a), 3(b), 3(c) and 3(d) show that the AUC gain over SVM grows with the p-values.



(a) TrAdaBoost

(b) Dynamic-TrAdaBoost

(c) TraBagg

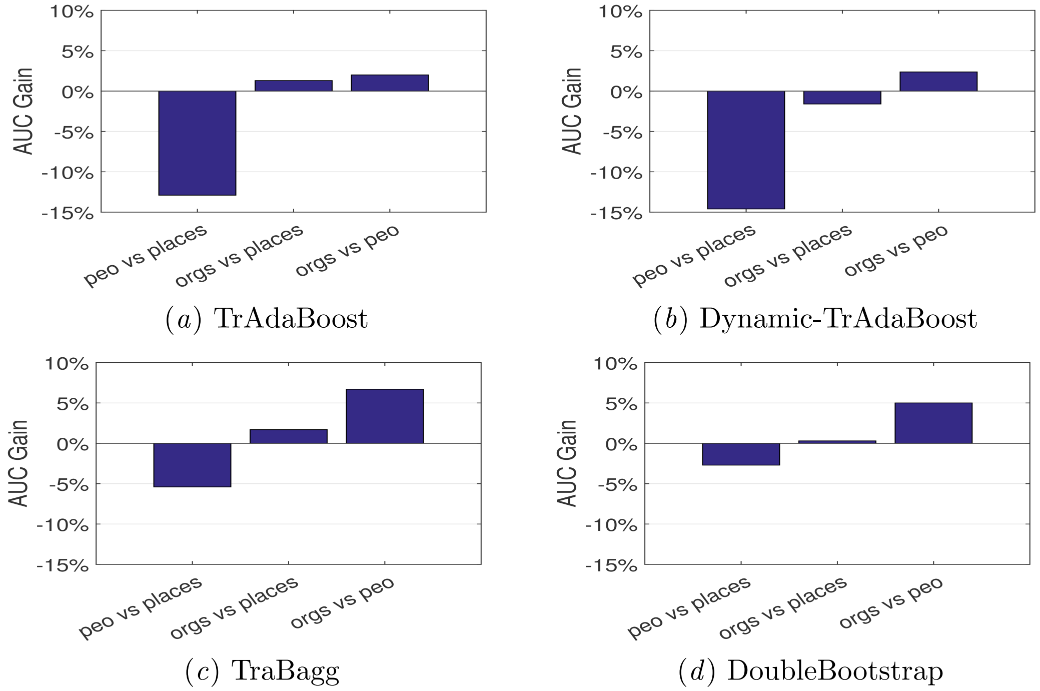
(d) DoubleBootstrap

Figure 3: AUC gain on the 20-Newsgroups instance-transfer classification tasks.

Figure 4 presents the results for the Reuters-21578 instance-transfer classification tasks. Analogously, it shows indirectly the correspondence between the p-values and the gain in AUC of the instance-transfer classifiers. As shown in those sub-figures, all those transfer learning algorithms result in negative transfer for the "people vs places" task which associated with a very lower p-value (0.146), and can hardly improve (even degrade) the performance for the "orgs vs places" task that with a relatively lower p-value. This result shows that our p-value provides a good prediction for negative transfer. The improvement achieved by instance-transfer algorithms for the "orgs vs people" is not significant, although it corresponds to a relatively high p-value. That is because the baseline classifier has already a good AUC (0.72) which limits the benefit of instance transfer.

5.4. Experiments II: Biggest Transferable Subsets

This subsection compares the generalization performance of largest transferable subset selection (LSSS) to four instance-transfer algorithms: TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap. Our approach and all the four algorithms employ SVM as



(a) TrAdaBoost

(b) Dynamic-TrAdaBoost

(c) TraBagg

(d) DoubleBootstrap

Figure 4: AUC gain on the Reuters-21578 instance-transfer classification tasks.

a base prediction model. Hence, the performance of SVM alone for the case of no instance transfer is given as baseline. The results for the landmine tasks are given in Table 4, for the 20-Newsgroup tasks in Table 5, and for the Reuters-21578 tasks in Table 6. In these tables significant negative transfer is marked with "-", while performance that is statistically better than others marked with "*".

Datasets	Source	p-value	Baseline	LSSS	TrAda-Boost	Dynamic-TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	Source 1	0.174	0.531	0.531	0.546	0.547	0.541	0.531
	Source 2	0.274	0.531	0.556	0.553	0.535	0.557	0.531
	Source 3	0.237	0.53	0.559	0.552	0.537	0.559	0.534
	Source 4	0.465	0.531	0.611*	0.599	0.596	0.605	0.592
	Source 5	0.446	0.531	0.614*	0.575	0.587	0.602	0.591

Table 4: Performance of instance-transfer transfer algorithms on the Landmine tasks.

Tables 4 - 6 show that LSSS outperforms all the four instance-transfer algorithms in most of the experiments. For the landmine tasks the performance of the algorithms and LSSS are comparable when the target relevance of the source data is low. However, when the relevance gets bigger, the LSSS performance gets slightly better than that of other algorithms. LSSS performs very well for the 20-Newsgroups tasks (see Table 5). The average performance of LSSS is significantly better than other algorithms with a margin of 10% over the mean AUCs of others. Most of the instance-transfer algorithms fail in the Reuters-21578 tasks (see Table 6). The reason is twofold. Firstly, the target training data is already sufficient to train a good prediction model (i.e., the average AUC of the baseline SVM model is 0.7). Secondly, the target and source data are poorly related. Comparing with other algorithms, LSSS is more robust in a such situation. LSSS has an improved performance in two of the tasks (one time significantly) and does not suffer from negative transfer compared with the other four algorithms.

Datasets	Task	*p*-value	Baseline	LSSS	TrAda-Boost	Dynamic-TrAda-Boost	TraBagg	Double-Bootstrap
20News -groups	comp vs sci	0.303	0.506	0.614*	0.539	0.532	0.523	0.513
	comp vs talk	0.390	0.501	0.778*	0.662	0.677	0.639	0.526
	rec vs sci	0.343	0.514	0.648	0.640	0.645	0.651	0.522
	rec vs talk	0.320	0.507	0.659*	0.615	0.62	0.573	0.523
	sci vs talk	0.340	0.510	0.691*	0.641	0.65	0.623	0.524

Table 5: Performance of instance-transfer transfer algorithms on the 20-Newsgroups tasks.

Datasets	Task	*p*-value	Baseline	LSSS	TrAda-Boost	Dynamic-TrAda-Boost	TraBagg	Double-Bootstrap
Reuters 21578	orgs vs people	0.372	0.72	0.753	0.734	0.737	0.768*	0.756
	orgs vs places	0.272	0.705	0.736*	0.714	0.694^{-}	0.717	0.707
	people vs places	0.146	0.704	0.704	0.613^{-}	0.601^{-}	0.666^{-}	0.690^{-}

Table 6: Performance of instance-transfer transfer algorithms on the Reuters-21578 tasks.

To conclude, the results show that in most of the experiments LSSS achieves a better improvement by using the source data than the remaining four algorithms. This is due to the following properties of LSSS:

Avoiding negative transfer: LSSS and DoubleBootstrap outperform other algorithms when the source and target domains are weakly related (see the results of the "people vs places" task in Table 6). The reason is that both algorithms adopt a prior selection process that filters out irrelevant sources before training. However, LSSS is a safer choice than DoubleBootstrap when the source could be totally unrelated. That because when the *p*-value is smaller than the significant level, LSSS completely disregards the source data.

Good performance with small target data: LSSS achieves good performance for small target data due to the use of prior source instance selection (see the results for the 20-Newsgroups tasks in Table 5). In contrast, the boosting-based algorithms (i.e., TrAdaBoost, Dynamic-TrAdaBoost) stop at very early iterations as the training error on the target data converges quickly to zero due to the small target-data size (e.g., for the "comp vs talk" task the algorithms stop after one or two iterations). In that case, some irrelevant instances cannot be filtered out, which limits the performance.

Robust against imbalanced class distribution: DoubleBootstarp is vulnerable to the imbalanced class distribution (see Section 2). For example, it only selected instances from the majority class for all the 20-Newsgroups tasks 5. In contrast, LSSS performs selection based on nonconformity scores so that instances from minority classes are still selected.

6. Conclusion

In this paper we showed that selecting a set of source instances can be done by estimating the relevance of that set to the target domain rather than estimating the individual relevance of the source instances in the set. For that purpose we proposed an approach to selecting the largest subset S^* of the source instances which relevance to the target domain is guaranteed to be the highest among any superset of S^*. The approach is formally described and theoretically justified. Experimental results on three real-world data sets demonstrated that the approach outperforms existing instance-transfer algorithms.

Acknowledgments

This work is financially supported by China Scholarship Council and Fundamental Research Funds for the Central Universities (Grant number: XDJK2016C042).

References

Samir Al-Stouhi and Chandan K Reddy. Adaptive boosting for transfer learning using dynamic updates. In *Machine Learning and Knowledge Discovery in Databases*, pages 60–75. Springer, 2011.

David J. Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, volume 1117, pages 1–198. Springer, 1985.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 193–200. ACM, 2007.

Toshihiro Kamishima, Masahiro Hamasaki, and Shotaro Akaho. Trbagg: a simple transfer learning method and application to personalization in collaborative tagging. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 219–228, 2009.

Di Lin, Xing An, and Jian Zhang. Double-bootstrapping source data selection for instance-based transfer learning. *Pattern Recognition Letters*, 34(11):1279–1285, 2013.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Craig Saunder. *Efficient implementation and experimental testing of transductive algorithms for predicting with confidence.* PhD thesis, Royal Holloway, University of London., 2000.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

Lisa Torrey and Jude Shavlik. Transfer learning. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1:242, 2009.

Vladimir Vovk. The basic conformal prediction framework. In *Conformal Prediction for Reliable Machine Learning Theory, Adaptations and Applications*, pages 1–20. 2014.