Testing Exchangeability for Transfer Decision

Pattern Recognition Letters 88 (2017) 64–71

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Testing exchangeability for transfer decision

Shuang Zhou*, Evgueni Smirnov, Gijs Schoenmakers, Kurt Driessens, Ralf Peeters

Department of Data Science and Knowledge Engineering, Maastricht University, P.O.BOX 616, Maastricht, 6200 MD, The Netherlands

ARTICLE INFO

Article history:
Received 23 March 2016
Available online 16 January 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Instance-transfer learning
Conformity prediction framework
Exchangeability test

ABSTRACT

This paper introduces a non-parametric test to decide whether to transfer data from a source domain to a target domain to improve the generalization performance of predictive models on the target domain. The test is based on the conformal prediction framework: it statistically tests whether the target and source data are generated from the same distribution under the exchangeability assumption. The experiments show that the test is capable of outperforming existing methods when it decides on instance transfer.

1. Introduction

Instance transfer has received significant attention in the last decade [19]. The goal is to improve the predictive models for a *target* domain by exploiting data from a (closely) related *source* domain. A thorough analysis of instance transfer [22] shows that its effectiveness depends on the relevance of the source domain to the target domain. As a result, a critical problem we have in practice is to decide whether we can transfer the source data while training predictive models for the target one. Adequately addressing this problem guarantees that we avoid negative transfer when adopting source data degrades the performance of the final models [19] .

The standard approach to the problem of deciding whether to transfer source data has been proposed in a number of works all based on the same principle [6,7,21,26]. This approach considers the distance between the target and source probability distributions as the difference between the target and source domains. Thus, the source data is transferred iff the distance between the probability distributions is small enough. There are however two major drawbacks of this approach. First, it is sensitive to the accuracy of estimating target and source probability distributions, since the real distributions are usually unknown. When this accuracy degrades (for example when the data is limited in size or high dimensional), the approach can be misleading. Second, the bound used to select or disregard source data is set by the user in an ad-hoc fashion and the approach does not lend itself to making statistically sound decisions; e.g., to transfer only if the target and source probability distributions are similar with high probability. This is because the distance values provided by the approach are very difficult to relate to the null hypothesis that "the target and source probability distributions are similar" in a statistical sense. This can easily result in negative transfer.

In this paper, we propose to avoid the aforementioned drawbacks by using a non-parametric test to decide on transfer using source domain based on the conformal prediction framework [20]. It tests whether the target data and the source data have been generated from the same distribution under the exchangeability assumption [2] and makes it possible to decide on transfer using an interpretable significance level.

The essential part of the new test is a p-value function that can be used to estimate the relevance of the target and source data in a statistically sound way. For any target and source data the function returns a p-value related to the null hypothesis "the target and source data have been generated from the target distribution under the exchangeability assumption". The function can be instantiated dependent/independent on/of the predictive models used. The validity of the function is proven.

The rest of the paper is organized as follows. Section 2 formalizes the classification task and instance transfer task, and provides an overview of related work. Our conformity-based test to decide on transfer from source data is introduced in Section 3. Section 4 provides an experimental comparison with existing approaches. Finally, Section 5 concludes the paper.

* Corresponding author.
E-mail address: shuang.zhou@maastrichtuniversity.nl (S. Zhou).

http://dx.doi.org/10.1016/j.patrec.2016.12.021
0167-8655/

2. Classification and instance transfer

In this section we present classification tasks in the context of instance transfer learning and discuss past work.

2.1. Task definition

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_{XY} over the labeled space[1]. We consider first a domain $\langle (X \times Y), P^t_{XY} \rangle$ that we call a target domain. The target data set T is a finite multi set of m_T instances $(x_t, y_t) \in X \times Y$ drawn from the target distribution P^t_{XY} under the randomness assumption. Given a test instance $x_{m_T+1} \in X$, the target classification task is to find an estimate $\hat{y}_{m_T+1} \in Y$ for the true class of x_{m_T+1} according to P^t_{XY}.

Let us consider a second domain $\langle (X \times Y), P^s_{XY} \rangle$ that we call a source domain. The source data S is a finite multi set of m_S instances $(x_s, y_s) \in X \times Y$ drawn from the source distribution P^s_{XY} under the randomness assumption. Knowing that the target domain and the source domain are relevant, 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_{XY}.

From the definition above it follows that instance transfer is sensitive to the relevance of the source data S to the target data T. Hence, the problem of deciding whether we can transfer the source data in order to improve class estimation in the target domain is important for its overall success.

2.2. Related work

As it is mentioned in Section 1, the standard approach of deciding whether to transfer source data consists of two stages. First, the target and source probability distributions are estimated, and then the distance between the distributions is computed. Based on the distance, the transfer decision is made. Below the main methods within this approach are described.

By assuming that the underlying distributions are normal, the distance can be estimated using the Mahalanobis distance, which is a common statistical distance between an instance and a distribution [8]. It measures how many standard deviations the instance is away from the mean of the distribution. Given target and source data, the distance between these two sets can be estimated by averaging the Mahalanobis distance of each source instance to the target distribution (estimated from the target data). Although the measure is widely used, it assumes a normal distribution of the target instances. In practice, this assumption does not often hold.

Kullback–Leibler divergence (KL-divergence) is one of the most widely used measures to compare probability distributions [14]. Given target and source data, KL-divergence from the P^s_{XY} to P^t_{XY} is defined as the expectation of the logarithmic quotient of the estimated target and source densities, where the expectation is taken w.r.t. the target density. Some applications of KL-divergence can be found in [1,6,7,27]. In [6], by assuming the independence between features, the KL-divergence from P^s_{XY} to P^t_{XY} was computed as the sum of feature-level KL-divergences. In [27], P^s_{XY} and P^t_{XY} were assumed to be mixtures of Gaussians. The KL-divergence was then calculated based on two Gaussian mixtures estimated from T and S. KL-divergence assumes that the probability densities can be estimated precisely from the data. When the number of the measurements is small and/or the data-space is highly dimensional the approximations can result in an inaccurate KL-divergence estimation.

The $\mathcal{A}$-distance was introduced by [12]. Given target and source probability distributions P^t_{XY} and P^s_{XY}, and a collection $\mathcal{A}$ of data sets, the $\mathcal{A}$-distance between P^t_{XY} and P^s_{XY} is defined as the upper bound of the absolute difference of the probabilities of generating sets $A \in \mathcal{A}$ w.r.t. P^t_{XY} and P^s_{XY}. The $\mathcal{A}$-distance depends on the choice of the sets collection $\mathcal{A}$, and determining a good collection is an open problem.

The discrepancy distance, proposed by Mansour et al. [17], estimates the difference between the target and source conditional distributions $P^t_{Y|X}$ and $P^s_{Y|X}$ from the perspective of a hypothesis space H. The key idea is that the target (source) classifier $h^t \in H$ ($h^s \in H$) based on the target (source) data sets T (S) can be used to approximate the conditional target (source) distribution $P^t_{Y|X}$ ($P^s_{Y|X}$). Therefore, the discrepancy distance is computed as the disagreement between the target and source classifier h^t and h^s by labeling instances from the union of target and source data. One drawback of using the discrepancy distance is that the difference between the target and source domains is estimated only in terms of the difference in conditional distributions without taking into account the difference between the marginal distributions.

Since any joint distribution P_{XY} can be given as the product of marginal distribution P_X and the conditional distribution $P_{Y|X}$, the transfer cross validation framework (TrCV) [26] measures the distance between marginal distributions and conditional distributions and then combines them to indicate the joint distribution discrepancy. More precisely, applying TrCV is a two-step process. First, a density ratio weighting approach is used to assess the difference in marginal distributions P^t_X and P^s_X. Second, a reverse validation framework is employed to quantify the discrepancy between conditional probabilities $P^t_{Y|X}$ and $P^s_{Y|X}$. The distance between target and source joint distributions is then calculated as the product of marginal discrepancy and conditional discrepancy.

As it is mentioned in Section 1, all these methods are sensitive to the accuracy of distribution estimation, and do not support instance transfer decision in a statistical sense. To avoid these problems, in the next section we propose our solution.

3. A conformity-based test for transfer decisions

We propose a non-parametric statistical test to decide on instance transfer from given source data. In the original problem formulation, target and source data are generated under the randomness assumption. This assumption leads to a null hypothesis that the joint data set $T \cup S$ was generated from the target probability distribution P^t_{XY} under the randomness assumption. Thus, one could employ some of the randomness tests from the algorithmic theory of randomness [5,18,25]. However, it is a well-known fact that those tests are incomputable [24].

To go around the computability problem of the randomness tests we propose to employ the conformal prediction framework [20] instead. In this context we introduce a conformity-based test to decide on transfer from the source data. The key idea stays the same but under the exchangeability assumption of data generation [2] that treats the data-sets as finite sequences sampled from the probability distributions. The null hypothesis then becomes that the data sequence TS consisting first of target data sequence T and followed by source data sequence S was generated from the target probability distribution P^t_{XY} under the exchangeability assumption. If the null hypothesis is accepted at some significance level, it implies that at that level the target and source data sequences T and S are relevant, and the source data sequence S can be transferred. Otherwise, the target data sequence T should be used on its own. This way we avoid probability-distribution estimations and provide

[1] For the sake of completeness the marginal distribution over X is denoted by P_X, and the conditional distribution over Y given X by $P_{Y|X}$.

a way to make the transfer decision in a statistical sense; i.e., we overcome both drawbacks of the existing work.

Below we describe the test in detail. We start with the exchangeability assumption, then introduce our p-value function and the test, and, finally, provide properties and a computationally efficient approximation of the p-value function.

3.1. Test derivation

The exchangeability assumption is a weaker assumption than the randomness assumption [2]. It holds for a finite sequence of random variables iff the joint probability distribution of those variables is invariant under any variables' permutation [10]. Applying the exchangeability assumption to decide to transfer instances from S to T means that we need to decide whether the combined data sequence TS has been generated by the target distribution P^t_{XY} under the exchangeability assumption. This is equivalent to testing the hypothesis "the probability distribution of all the permutations of the data sequence TS is uniform".

When $m_S = 1$, Shafer and Vovk [20] proposed such a test in the context of conformal prediction. The test is based on instance nonconformity scores as statistics for the null hypothesis "the distribution of all the permutations of the data sequence TS is uniform". The nonconformity score $\alpha_{(x,\,y)}$ of an instance $(x,\,y) \in TS$ is defined as a score (result of a function) indicating how unusual that instance is in the data sequence $TS\backslash\{(x,\,y)\}$. Let $(X \times Y)^{(*)}$ represents the set of all sequences over $(X \times Y)$, an instance nonconformity function A is formally a mapping from $(X \times Y)^{(*)} \times (X \times Y)$ to $\mathbb{R}^+ \cup \{+\infty\}$, indicating how unusual the instance $(x,\,y)$ is for the instances in the data sequence $TS\backslash\{(x,\,y)\}$. We note that any instance nonconformity function has to produce the same result for an instance independently of the ordering of TS. Otherwise, the instance will have $|TS|!$ number of possible nonconformity values.

There are several instance nonconformity functions [3] available. They are divided into those that depend on the predictive models used and those that do not. Among the former there exist functions for Decision Trees, SVMs, k-NN predictors, AdaBoost etc.

Since in the transfer setting the source data S usually consists of more than one element, we need to generalize the work of Shafer and Vovk [20] and define a nonconformity function for data sequences of any length. Given the combined sequence TS and any sequence U of some elements of $T \cup S$, the nonconformity function should return a value $\alpha_U \in \mathbb{R}^+ \cup \{+\infty\}$ indicating how unusual the data sequence U is with respect to all subsequnces with size $|U|$ of the data sequence TS.

Definition 1 (Sum sequence nonconformity function)**.** Given an instance nonconformity function A, data sequence T and a data sequence U of some elements of $T \cup S$, the sum sequence nonconformity function $A^* : (X \times Y)^{(*)} \times (X \times Y)^{(*)} \to \mathbb{R}^+ \cup \{+\infty\}$ is defined as

$$A^*(T, U) = \sum_{(x,y)\in U} \alpha_{(x,y)},$$

$$\text{where } \alpha_{(x,y)} = \begin{cases} A(T \setminus \{(x,y)\}, (x,y)), & for\ (x,y) \in T \\ A(T, (x,y)), & otherwise \end{cases}$$

The sum sequence nonconformity function A^* returns the same nonconformity value for a data sequence U independently of the ordering of T if this property holds for the instance nonconformity function A. It is also independent of the ordering of U, which will become important for computations later on. Since the instance nonconformity function A can be model (in)dependent, the same holds for A^*.

Using an instance nonconformity function that estimates the unusualness of the instance w.r.t. the target data sequence T, we can employ the sequence nonconformity $\alpha_U = A^*(T, U)$ to test the null hypothesis: "the distribution of all the permutations of the data sequence TS is uniform". To design the test, we employ the p-value function defined next.

Definition 2 (p-value function)**.** Given the data sequence T, a data sequence U of some elements of $T \cup S$, and an integer $n \le |U|$, the p-value function $t : (X \times Y)^{(*)} \times \mathbb{N} \to [0, 1]$ equal to:

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

where $\mathcal{P}(U, n)$ is the set of all length n subsequences of U, $L(U, n)$ is the sequence of the last n elements of U, α_V and $\alpha_{L(U,\,n)}$ are sequence nonconformity scores returned by $A^*(T, V)$ and $A^*(T, L(U, n))$, respectively[2].

Given the combined data TS and $n = m_S$, the p-value function $t(TS, m_S)$ returns the proportion of length m_S subsequences of the sequence TS whose nonconformity values are greater than or equal to that of the source data sequence S.

Theorem 1. *If the sequence TS is exchangeable, then*

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

Proof. 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 $P\{t(TS,\ m_S) \le r\}$ equals:

$$\frac{|\{U \in \mathcal{P}(TS, m_T + m_S) :\ t(TS, m_S) \le r\}|}{|\mathcal{P}(TS, m_T + m_S)|}$$

$$= \frac{\left|\left\{U \in \mathcal{P}(TS, m_T + m_S) :\ t(TS, m_S) \le \frac{j}{|\mathcal{P}(TS,m_S)|}\right\}\right|}{(m_T + m_S)!}$$

$$= \frac{\left|\left\{U \in \mathcal{P}(TS, m_T + m_S) :\ \left|\left\{V \in \mathcal{P}(TS, m_S) |\ \alpha_V \ge \alpha_{L(TS,m_S)}\right\}\right| \le j\right\}\right|}{(m_T + m_S)!}$$

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 $V \in \mathcal{P}(TS, m_S)$ that have a nonconformity value $\alpha_V \ge \alpha_{L(U,m_S)}$. Say that there are $k(\le j)$ 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 an appropriate sequence of the length m_T). This means that $|S_j(TS)| = k \cdot m_T! \le j \cdot m_T!$ with a possible strict inequality if there are multiple sequences V that have identical nonconformity values. We have:

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

which completes the proof. □

Theorem 1 shows the validity of the p-value function t. We note that this theorem can be viewed as a corollary of Theorem 4.1 in the work of [23], which proves the validity of the p-value function t using an online protocol.

[2] T is a parameter of the p-value function t as well. However, for the sake of simplicity of the notation, it is skipped from the definition of t.

We can now employ our p-value function t to design a conformity-based statistical test for the null hypothesis that the combined data sequence TS has been generated by the target distribution P^t_{XY} under the exchangeability assumption. If the returned p-value is greater than or equal to a user defined significance level ϵ, the null hypothesis is accepted and the source data S is transferable. Otherwise, the null hypothesis is rejected and only the target data T should be used.

Note that our conformity-based test can be dependent or independent on/of the predictive models used subject to the instance nonconformity function used. Model dependency is important as this allows us to base the decision for instance transfer on the classifier that will be used in actual transfer (in analogy with feature-selection wrappers [13]).

3.2. Computing p-values using the wilcoxon rank-sum test

As mentioned in Section 3.1, the sum sequence nonconformity function $A^*(T, U)$ is independent of the ordering of the sequence U, so that the nonconformity score of a set can be defined equal to the nonconformity score of any sequence of elements of that set. Given that $\mathcal{C}(U, n)$ is the set of all combinations of n elements out of sequence U and the cardinality of $\mathcal{C}(U, n)$ is independent of the order of U ($|\mathcal{P}(U, n)| = |\mathcal{C}(U, n)| \times n!$), we re-write the p-value function t as follows:

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

We note that the p-value function defined in this way exhibits an analogy to the notion of Wilcoxon rank-sum test [15]. If we assign ranks from 1 to $m_T + m_S$ to all instances in TS according to their nonconformity scores in ascending order, the p-value function $P(TS, m_S)$ can be approximated using the Wilcoxon rank-sum test. In this setting, the nonconformity score α_V of any set V from TS with size m_S is replaced by the rank sum W that equals to $\Sigma_{(x,\,y)\,\in\,V} R_{(x,\,y)}$ where $R_{(x,\,y)}$ is the rank of nonconformity score $\alpha_{(x,\,y)}$ of instance $(x, y) \in V$. Accordingly, α_S is replaced by the sum of ranks of all instances in S denoted by W_S. In this way the probability $P(W \geq W_S)$ that the rank sum of any m_S instances is bigger than that of the source instances is approximately equal to $t(TS, m_S)$; i.e., the p-value function can be implemented using the rank-sum test.

When $|TS| > 30$, the rank sum W is approximately normally distributed according to the law of large number. The expectation of the rank sum is:

$$E(W) = \frac{1}{2} m_S (m_T + m_S + 1)$$

and variance is:

$$Var(W) = \frac{1}{12} m_T m_S (m_T + m_S + 1)$$

This implies that the probability $P(W \geq W_S)$ and, thus, the p-value $t(TS, m_S)$, can be easily approximated from this normal distribution.

3.3. Analyzing the p-value function t

Assume that all the instances in the source sequence S are sorted in increasing order of the nonconformity values and S_i is a subsequence consisting of the first i instances of the ordered S. For example, let us consider 3 target instances and 4 source instances with corresponding nonconformity scores are 1,5,6,2,3,4, and 7. In this case $t(TS_1, 1) = 1.0,\ t(TS_2, 2) = 0.7,\ t(TS_3, 3) = 0.8$ and $t(TS_4, 4) = 0.54$. Thus, *$t(TS_i, i)$ is not a monotonic function of index i.*

In addition, we note that the function t can be modified to test the null hypothesis that the data sequence ST was generated from the source probability distribution P^s_{XY} under the exchangeability assumption. To draw this test we redefine the sum sequence nonconformity function A^*. The new function computes the nonconformity values w.r.t. the source sequence S; i.e., the nonconformity value $\alpha_{(x,\,y)}$ of any instance (x, y) is computed by $A(S\backslash\{(x, y)\}, (x, y))$. In doing so, the value $t(ST, m_T)$ indicates the relevance of the target data to the source data and it helps in instance transfer by showing how well the source data "covers" the target one. In this context we note that the values $t(TS, m_S)$ and $t(ST, m_T)$ in general are different due to different nonconformity values used and different computations involved. Thus, *the p-value function t is an asymmetric measure of relevance between the target data and source data.*

3.4. Off-line testing and on-line testing

Our conformity-based test is designed for off-line testing when all the source instances are available in advance. Still, it can be extended for on-line testing when the source instances arrive one by one [3]. In this setting upon arrival of a new source instance we first compute its nonconformity score (using the target data only) and then depending on the data size apply either the direct implementation (see Definition 2) or the rank-sum implementation of our test. Hence, we test on-line whether the target data that stay intact and the source instances that have arrived so far have been generated from the target distribution under the exchangeability assumption. In this respect our test extension differs the on-line test proposed in [10] that tracks the deviation from the exchangeability assumption in one data stream.

4. Experiments

In this section we present our experiments with our conformity-based test and related methods presented in Section 2.2. First, we describe the experimental setup, then the instance-transfer classification tasks under study, and, finally, the experimental results.

4.1. Experimental setup

For our experiments we chose the Wilcoxon rank-sum test implementation of the p-value function t (see Section 3.2). The sum sequence nonconformity function A^* was implemented according to Definition 1. For the instance nonconformity function we chose the general instance nonconformity function A_G as defined in [20]. A_G maps target data T and an instance (x_i, y_i) to a nonconformity value $\sum_{y \in \mathbf{Y}, y \neq y_i} p_y$ where p_y is the score of class $y \in Y$ for that instance produced by a nonconformity classifier trained on target data T. In our experiments we employed Random Forest [4] as a nonconformity classifier.

For the instance-transfer classification tasks, we employed four instance-transfer classifiers: TrAdaBoost [7], Dynamic-TrAdaBoost [1], TraBagg [11], and DoubleBootstrap [16]. All the classifiers used Support Vector Machines (SVM) as base learners.

The method of evaluation was a stratified holdout method on the target data repeated 100 times. For the non-text data (text data), 10% (4%) of instances were randomly sampled from the target data for training and the remaining for testing. The smaller percentage for the text data was due to the fact that for bigger percentage instance transfer is no longer required. The generalization performance of the instance transfer classifiers was evaluated using the Area Under the ROC Curve (AUC) [9].

Table 1
Landmine detection instance-transfer classification tasks.

Datasets		Description	Size	p-value
Land mine	T	Mine 26 to 29	1799	
	S1	Mine 16 to 20	2240	0.465
	S2	Mine 21 to 25	2246	0.446
	S3	Mine 6 to 10	2547	0.274
	S4	Mine 11 to 15	2902	0.237
	S5	Mine 1 to 5	3086	0.174
	S6	Mine 1 to 5 by shifting the mean of feature 1 to 0	3086	0.086
	S7	Mine 1 to 5 by shifting the mean of feature 2 to 0	3086	0.024

Our conformity-based test was evaluated w.r.t. its effectiveness in deciding whether to transfer source data. Rather than performing the test on all possible significance levels, we directly evaluated the test p-value function t w.r.t. its capability of predicting successful instance transfer. The capacity was defined as the Pearson's correlation coefficient between the p-values the function outputs for source data sets and the AUCs of the instance-transfer classifiers that employed those sets. Thus, the higher the capacity (correlation) the better the function can indicate a successful transfer. In this context we note that the five methods from Section 2.2 that we use for comparison output values that indicate distances between probability distributions. That is why, for these methods the capacity was defined as the negative Pearson's correlation coefficient between the distances the methods output and the AUCs of the instance-transfer classifiers.

4.2. Instance-transfer classification tasks

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

- Landmine detection[3] is a collection of data sets related to detecting landmines in different geographical locations. It consists of 29 data sets from 29 landmine fields. The 29 data sets have different distributions due to various ground surface conditions. For example, data sets 1 to 15 correspond to regions that are relatively foliated 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 S_1 and S_2 with a high relevance to the target one, while data sets 1 to 5, 6 to 10, 11 to 15 were combined into other three source data sets S_3, S_4 and S_5 with a lower relevance. To emphasize the negative effect of irrelevant source data, we generated two additional source data sets S_6 and S_7 from source data set S_3. S_6 (S_7) was generated from S_3 by shifting the mean of the distribution of feature 1 (2) to 0.
  The target data set and a source data set defined together one instance-transfer classification task. For all the tasks, the p-values for the source data sets are given in the last column of Table 1. The tasks are sorted by the p-values in descending order.
- Wine quality[4] is a data set of in total 1599 red-wine and 4898 white-wine instances. Each instance is represented by 11 physiochemical features (e.g. pH values) and a grade given by experts. In the experiments, red-wine instances were used as the target data set and seven source data sets were sampled from white-wine instances based on different conditions. The target data set and a source data set defined together one instance-transfer classification task. The p-values for the source data sets of all the tasks are given in the last column of Table 2.
- 20-newsgroups[3] 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. The p-values for the source data sets of all the five tasks are given in the last column of Table 3.
- Reuters-21578[3] is a collection of data sets with text documents organized in a hierarchical structure. Three instance-transfer classification tasks were defined in the same way as those of the 20-newsgroups tasks. The p-values for the source data sets of all the three tasks are given in the last column of Table 4.

4.3. Experimental results

4.3.1. Non-Text data

Fig. 1 presents the results for the landmine-detection and wine-quality instance-transfer classification tasks. It shows the correspondence between the p-values of the source data sets and the AUC performance of the aforementioned instance-transfer classifiers. On the x-axis, S_i $(i = 1, 2, \ldots, 7)$ represents the source data sets from 1 to 7, sorted in descending order of p-values from left to right while the y-axis shows the average AUCs of the corresponding classification models trained on $T \cup S$. The performance of an SVM classifier trained on T alone 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. When the source data is irrelevant to the target data (i.e. p-value is smaller than 0.1), these classifiers may result in negative transfer.

We empirically compared our p-value function to the five methods discussed in Section 2.2 in terms of capability of predicting the success of instance transfer. The results are provided in Table 5. We highlight the highest number in each column. As shown in the table, our p-value function outperforms the other measures in most of the experiments.

4.3.2. Text data

Fig. 2 presents the results of our p-value function for the 20-newsgroups instance-transfer classification tasks. It shows the correlation between the p-values of the source data and the AUC performance of the instance transfer classifiers. On the x-axis, the instance-transfer text classification tasks are again sorted descendingly according the p-values. The y-axis shows the gain in AUC of

[3] http://www.cse.ust.hk/TL/.
[4] https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

Table 2
Wine quality instance-transfer classification tasks.

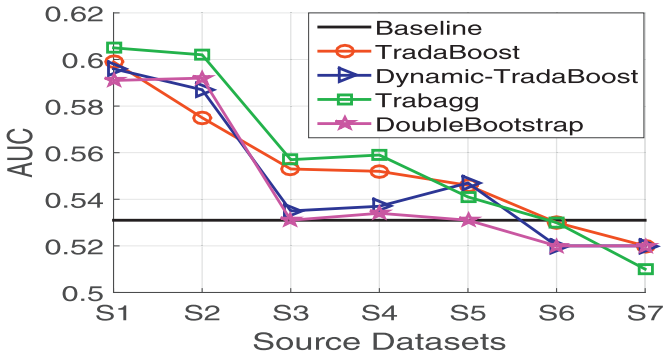
Datasets		Description	Size	*p*-value
Wine	T	Red wine	1599	
	S1	White wine that *alcohol* < 0.5	1540	0.327
	S2	White wine that *total sulphur dioxide* > 0.2	1499	0.286
	S3	White wine that *alcohol* < 0.5 and 0.16 < *acid* < 0.22	1499	0.241
	S4	Random sample of white wine	1469	0.208
	S5	White wine that *total sulphur dioxide* > 0.2 and 0.16 < *acid* < 0.22	1548	0.199
	S6	Source data S5 by shifting the mean of the feature *sulphates* to 0	1548	0.106
	S7	Source data S5 by shifting the mean of the feature *alcohol* to 0	1548	0.053

Table 3
20-Newsgroups instance-transfer classification tasks.

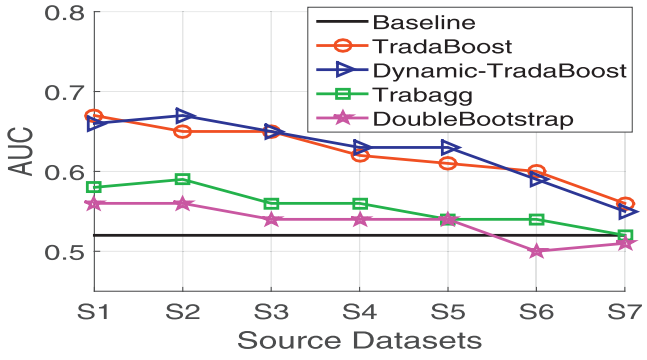
Datasets	Tasks	Sample Size		*p*-value
		$\|T\|$	$\|S\|$	
20-Newsgroups	comp vs talk	4482	3652	0.390
	rec vs sci	3961	3965	0.343
	ci vs talk	3374	3828	0.340
	rec vs talk	3669	3561	0.320
	comp vs sci	3930	4900	0.303

Table 4
Reuters-21578 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



(a) Landmine



(b) Wine

Fig. 1. AUCs of TrAdaBoost, Dynamic-TrAdaBoost, Trabagg, and DoubleBootstrap on non-text data.

the instance-transfer classifiers w.r.t. non-transfer SVM classifier[5]. Fig. 2(a)–(d) show that the AUC gain over SVM decreases with the *p*-value of the source data.

Fig. 3 presents the results for the Reuters-21578 instance-transfer classification tasks. Analogously, it shows the correlation between the *p*-values and the gain in AUC of the instance-transfer classifiers. As shown in those sub-figures, all those instance-transfer classifiers result in negative transfer for the "people vs places" task which associated with a very lower *p*-value (0.146). This result shows that our *p*-value provides a good prediction for negative transfer. The improvement for the "orgs vs people" task is not significant, although it corresponds to a high *p*-value. That is because the baseline classifier has already a good AUC which limits the benefit of instance transfer.

We again compared the *p*-value function with methods discussed in Section 2.2. The Mahalanobis distance was excluded, since it produces inaccurate results for data having bigger number of features than that of instances. The results are provided in Table 6 and they show that on most of these tasks our *p*-value function *t* outperforms the previous state of the art.

[5] The use of AUC gain is due the fact that the instance-transfer text classification tasks differ in the target data and the source data making AUCs on different tasks incomparable.

Table 5
Capacity to predict the success of instance-transfer classifiers for the Landmine and wine tasks.

Measures	Landmine				Wine			
	TrAdaBoost	Dynamic TrAdaBoost	TraBagg	Double-bootstrap	TrAdaBoost	Dynamic TrAdaBoost	TraBagg	Double-bootstrap
Conformity test (*p*-value)	**0.933**	**0.925**	**0.991**	**0.981**	**0.936**	**0.882**	0.811	**0.920**
KL-divergence	0.804	0.820	0.940	0.920	0.522	0.864	**0.858**	0.801
Mahalanobis distance	0.856	0.792	0.944	0.882	0.572	0.856	0.789	0.684
$\mathcal{A}$-distance	0.920	0.784	0.882	0.810	0.618	0.682	0.851	0.790
Discrepancy distance	0.782	0.886	0.835	0.882	0.920	0.722	0.740	0.712
TrCV	0.895	0.821	0.960	0.889	0.872	0.764	0.680	0.782

Table 6
Capacity to predict the success of instance-transfer classifiers for the 20-Newsgroups and Reuters-21578 tasks.

Measures	20-Newsgroups				Reuters-21578			
	TrAda-Boost	Dynamic TrAdaBoost	TraBagg	Double-bootstrap	TrAda-Boost	Dynamic TrAdaBoost	TraBagg	Double-bootstrap
Conformity test (p-value)	**0.884**	**0.889**	**0.855**	0.793	**0.914**	**0.972**	**0.999**	0.965
KL-divergence	0.271	0.261	0.164	0.197	0.334	0.157	0.441	0.336
$\mathcal{A}$-distance	0.569	0.566	0.714	0.472	0.664	0.789	0.894	0.986
Discrepancy distance	0.864	0.875	0.670	**0.971**	0.774	0.876	0.954	**1**
TrCV	0.716	0.714	0.761	0.620	0.341	0.506	0.667	0.857

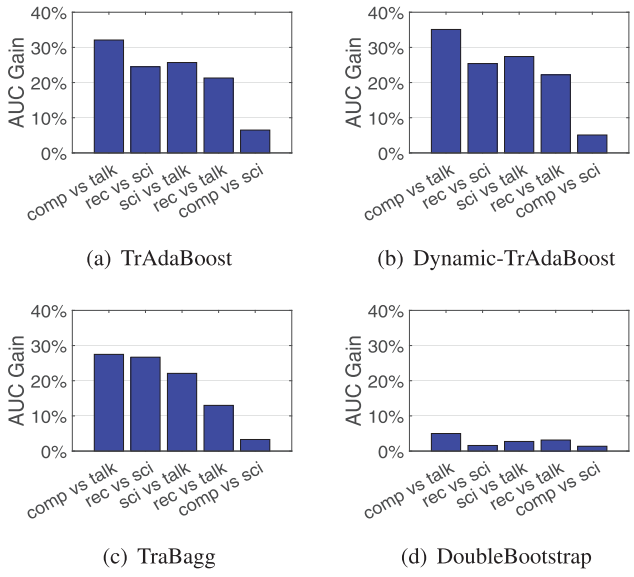


Fig. 2. AUC gain on the 20-Newsgroups instance-transfer classification tasks.

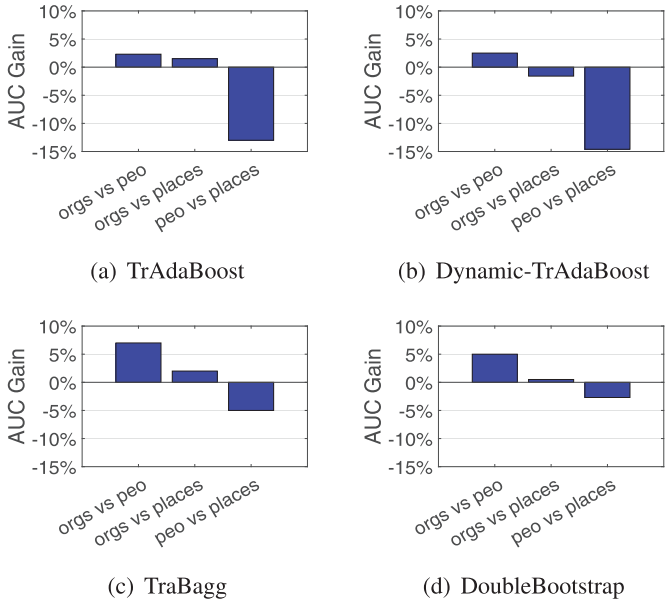


Fig. 3. AUC gain on the Reuters-21578 instance-transfer classification tasks.

5. Conclusion

We proposed a non-parametric test to decide on instance transfer. The test is based on the conformal prediction framework [20] and tests whether the target data and the source data have been generated from the target distribution under the exchangeability assumption [2].

Our test overcomes several drawbacks of the existing methods for transfer decisions. More precisely, it does not employ probability-distribution estimation and it provides decisions in statistical sense. On the top of that, we emphasize two practical advantages:

- Our test is applicable under the exchangeability assumption that is known to be a weaker assumption than the randomness assumption. It broadens the scope of applicability.
- Our test can be either dependent or independent on/of the predictive models used. The model dependency of the test is important for instance transfer. For example, we can do the test using a classifier that latter will be used for instance transfer; i.e., the decision for instance transfer depends is tailored to the classifier used.

References

[1] S. Al-Stouhi, C.K. Reddy, Adaptive boosting for transfer learning using dynamic updates, in: Machine learning and knowledge discovery in databases, Springer, 2011, pp. 60–75.
[2] D.J. Aldous, Exchangeability and related topics, Springer, 1985.
[3] V. Balasubramanian, S.-S. Ho, V. Vovk, Conformal prediction for reliable machine learning: Theory, adaptations and applications, Elsevier, 2014.
[4] L. Breiman, Random forests, Mach Learn 45 (1) (2001) 5–32.
[5] A. Church, On the concept of a random sequence, Bull Am Math Soc 46 (2) (1940) 130–135.
[6] W. Dai, G.-R. Xue, Q. Yang, Y. Yu, Transferring naive bayes classifiers for text classification, in: Proceedings of the national conference on artificiall intelligence, 22, 2007, p. 540.
[7] W. Dai, Q. Yang, G.-R. xue, Y. Yu, Boosting for transfer learning, in: Proceedings of the 24th international conference on machine learning, ACM, 2007, pp. 193–200.
[8] R. De Maesschalck, D. Jouan-Rimbaud, D.L. Massart, The mahalanobis distance, Chemometr Intell Lab Syst 50 (1) (2000) 1–18.
[9] T. Fawcett, An introduction to ROC analysis, Pattern Recognit Lett 27 (8) (2006) 861–874.
[10] V. Fedorova, A.J. Gammerman, I. Nouretdinov, V. Vovk, Plug-in martingales for testing exchangeability on-line, in: Proceedings of the 29th international conference on machine learning, 2012.
[11] T. Kamishima, M. Hamasaki, S. Akaho, Trbagg: a simple transfer learning method and its application to personalization in collaborative tagging, in: Proceedings of the 9th IEEE international conference on data mining, IEEE, 2009, pp. 219–228.
[12] D. Kifer, S. Ben-David, J. Gehrke, Detecting change in data streams, in: Proceedings of the 13th international conference on very large databases, vol. 30, VLDB Endowment, 2004, pp. 180–191.
[13] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif Intell 97 (1–2) (1997) 273–324.
[14] S. Kullback, R.A. Leible, On information and sufficiency, Ann Math Stat 22 (1) (1951) 79–86.
[15] E.L. Lehmann, H.J. D'Abrera, Nonparametrics: statistical methods based on ranks, Springer New York, 2006.
[16] D. Lin, X. An, J. Zhang, Double-bootstrapping source data selection for instance-based transfer learning, Pattern Recognit Lett 34 (11) (2013) 1279–1285.
[17] Y. Mansour, M. Mohri, A. Rostamizadeh, Domain adaptation: learning bounds and algorithms, in: Proceedings of the 22nd conference on learning theory, 2009.
[18] P. Martin-Löf, The definition of random sequences, Inf Control 9 (6) (1966) 602–619.
[19] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans Knowl Data Eng 22 (10) (2010) 1345–1359.
[20] G. Shafer, V. Vovk, A tutorial on conformal prediction, J Mach Learn Res 9 (2008) 371–421.
[21] B. Tan, Y. Song, E. Zhong, Q. Yang, Transitive transfer learning, in: Proceedings of the 21th SIGKDD international conference on knowledge discovery and data mining, ACM, 2015, pp. 1155–1164.

[22] L. Torrey, J. Shavlik, Transfer learning, in: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques, 1, 2009, p. 242.
[23] Vladimir Vovk, Transductive conformal prediction, Int J Artif Intell Tools 24 (06) (2015) 1560001. World Scientific
[24] V. Vovk, A. Gammerman, C. Saunders, Machine-learning applications of algorithmic randomness, in: Proceedings of the 16th international conference on machine learning, 1999, pp. 444–453.
[25] V. Vovk, A. Gammerman, G. Shafer, Algorithmic learning in a random world, Springer, 2005.
[26] E. Zhong, W. Fan, Q. Yang, O. Verscheure, J. Ren, Cross validation framework to choose amongst models and datasets for transfer learning, in: Machine learning and knowledge discovery in databases, Springer, 2010, pp. 547–562.
[27] S. Zhou, E. Smirnov, R. Peeters, Conformal region classification with instance–transfer boosting, Int J Artif Intell Tools 24 (6) (2015) 1560002.