Proceedings of Machine Learning Research 91:1–18, 2018 Conformal and Probabilistic Prediction and Applications

Conformal Feature-Selection Wrappers for Instance Transfer

Shuang Zhou S.ZHOU@CUIT.EDU.CN
School of Controlling Engineering, Chengdu University of Information Technology
Evgueni Smirnov SMIRNOV@MAASTRICHTUNIVERSITY.NL
Gijs Schoenmakers GM.SCHOENMAKERS@MAASTRICHTUNIVERSITY.NL
Ralf Peeters RALF.PEETERS@MAASTRICHTUNIVERSITY.NL
Department of Knowledge Engineering,
Maastricht University,
P.O. Box 616, 6200 MD, Maastricht, The Netherlands
Tao Jiang JIANG@CUIT.EDU.CN
School of Controlling Engineering, Chengdu University of Information Technology

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Ralf Peeters

Abstract

In this paper we propose a new method of conformal feature-selection wrappers for instance transfer (CFSWIT). Given target and source data, the method optimally selects features and source data that are relevant for a classification model. The CFSWIT method is model-independent. It was tested experimentally for several types of classifiers. The experiments show that the CFSWIT method is capable of outperforming standard instance transfer methods.

Keywords: Instance Transfer, Conformal Prediction, Feature Selection, Wrappers

1. Introduction

Instance transfer was proposed to improve classification models for a *target* domain of interest by making use of the data borrowed from an auxiliary *source* domain (Pan and Yang, 2010; Weiss et al., 2016). The target and source domains share the same input feature space and the same class-label set but differ in the underlying probability distributions. If the source domain is relevant to the target domain; i.e., the source distribution is close to the target distribution, instance transfer can significantly improve the classification models for the target domain (Torrey and Shavlik, 2009), especially for small target data (Dai et al., 2007b).

Estimating the closeness of the source distribution to the target distribution is a difficult problem. This is due to the fact that the target and source probability distributions are usually unknown. There exist two main approaches to this problem that are both data-driven. The first approach measures the closeness of the source distribution to the target distribution by first estimating the parameters of the distribution functions from the target and source data (Dai et al., 2007a,b; Zhou et al., 2015; Tan et al., 2015). Then, it computes the distances between estimated distribution functions to approximate the distribution closeness. The second approach measures the closeness of the source distribution to the target distri-

bution by estimating how probable is that the target data and source data are generated from the same distribution (Zhou et al., 2017a).

Following the results of both approaches, if we find that the source distribution is close to the target distribution, we add the source data to the target data and then train the target classification model. However, if we find that the source distribution is not close to the target distribution, we can follow one of the three scenarios given below:

- **no instance transfer:** we cancel the instance transfer and train the target classification model on the target data only.
- **source-instance selection:** we select a subset of the source instances that corresponds to a component of the source distribution estimated to be close to the target distribution [1]. If the subset is nonempty, we add it to the target data and then train the target classification model.
- **feature selection:** we select a subset of features for which the source distribution is estimated to be close to the target distribution. If the subset is nonempty, the target and source data are represented by the selected features only. The source data is added the target data, and, then, the target classification model is trained.

When the last two scenarios fail, we can follow a forth scenario of combining feature selection and source-instance selection. In this scenario we select a subset of features and a subset of source data that corresponds to a component of the source distribution estimated to be close the target distribution on the selected features. This task assumes that selecting features and selecting source data are mutually dependent, and thus cannot be realized by a mechanical combination of the instance-transfer methods based on feature selection and instance-transfer methods based on source-instance selection. So far, Zhou et al. (2017b) proposed the only method available for mutually dependent feature and source-instance selection. The method realizes this property using decision trees (Quinlan, 1993) in an univariate manner by imposing an additional restriction that the final features have a good predictive power. The experiments showed that this method outperforms the existing instance transfer methods based on either source-instance selection or feature selection. However, this method is tailored to decision trees; i.e., it is *model-dependent*.

In this paper we propose a *model-independent* method for the task of combining feature selection and source-instance selection. The method is essentially a wrapper method for feature selection (Kohavi and John, 1997). Given a classification model that needs feature selection, our method examines the space of feature subsets according to a chosen search strategy. When it evaluates a set of features, it considers both target and source data represented by these features only. Under this constraint, our method first finds the *largest* relevant set of source instances that can be selected using a conformal source-subset selection procedure proposed by (Zhou et al., 2017c). Then, it estimates the generalization performance of the classification model on the target data and selected source instances. Once our

1. The source-instance selection implicitly assumes that the source distribution is a mixture distribution. The selected instances are expected to be those that are generated by a component of the source distribution that is close to the target distribution.

method has visited and evaluated all the feature subsets according the search strategy chosen, it determines a subset of features with the maximal generalization performance. This subset is outputted together with the corresponding largest relevant set of source instances.

We note that our method assumes that the process of examining the space of feature subsets starts from the set of all the features. That is why, the final subset of features is relatively *large*. Thus, our method outputs a *large* subset of features and the *largest* subset of source data that corresponds to a component of the source distribution estimated to be close the target distribution on the selected features.

The remainder of this article is structured as follows. Section 2 provides an overview of the related work. The classification task in context of instance transfer is formulated in Section 3. Section 4 explains the conformal test and its corresponding source-subset selection procedure. The wrapper method is given in Section 5. Section 6 introduces the main contribution of this article, namely the conformal feature-selection wrappers for instance transfer. The experiments are provided in Section 7. Section 8 concludes the article.

2. Related Works

As it was stated in the previous section there exist two types of methods for instance transfer when the relevance of the source domain is not sufficient for the target domain: methods based on source-instance selection and methods based on feature selection. In this section we provide an overview of these two types of methods as well as the only combined method.

2.1. Methods based on Source-Instance Selection

Methods based on source-instance selection transfer relevant source instances to improve classification models for the target domain (Zhou et al., 2017c). Source-instance selection can be done in two ways: soft selection and hard selection. The soft selection picks the source instances implicitly. It assigns weights to source instances proportionally to their relevance to the target data. In this way the influence of the less relevant source instances is restricted compared with that of most relevant ones when the final classification model is being trained. The hard selection picks the source instances explicitly. It directly selects source instances depending on their relevance to the target data. In this way only the most relevant source instances influence training of the final classification model.

The soft selection was implemented in several boosting-based methods, e.g., TrAdaBoost (Dai et al., 2007a) and Dynamic-TrAdaBoost (Al-Stouhi and Reddy, 2011). These methods are similar to the AdaBoost algorithm (Freund and Schapire, 1996) but employ two opposite weight-update schemes depending on the type of the instances: (1) the weights of misclassified target instances are increased, and (2) the weights of misclassified source instances are decreased. In theory the average weighted training loss of boosting-based algorithms on the source data is guaranteed to converge to 0 as the number of iterations approaches infinity (Dai et al., 2007a). This implies that in this case the relevant source instances will be classified correctly and the irrelevant source instances will receive a weight of 0; i.e., there will be a perfect selection of the source instances. However, in practice when most of the source instances are irrelevant, these algorithms are likely to stop at very first iterations because the training error on target data exceeds 0.5 in early iterations. In this case, the irrelevant

source instances are not filtered out and cause a negative effect on the final classification model.

The hard selection is implemented in several bagging-based methods. There are two types of implementations: direct and indirect. Double-Bootstrap (Lin et al., 2013) is an example of direct implementation. It first constructs an ensemble of classification models trained on bootstrap samples from the target data. Then the ensemble classifies the source instances and those of them that are correctly classified are selected. Thus, when most of the source instances are irrelevant, this method tends not to select source instances; i.e., the instance transfer process stops.

TrBagg (Kamishima et al., 2009) is an example of an indirect implementation of the hard instance selection. It first randomly generates a set of bootstrap samples from the combined target and source data, and then trains several base classification models on those samples. Finally, a subset of the base classification models are selected by minimizing the empirical error on the target data. The latter means that source subsets that are contained in the bootstrap samples are indirectly selected through selecting the base models. Although TraBagg is simple, it has similar problem as the boosting methods when the source data is rather irrelevant. In this case TrBagg requires a large number of bootstrap iterations to filter out irrelevant source instances which makes it computationally inefficient.

2.2. Methods based on Feature Selection

Methods based on feature selection aim at finding relevant features for which the source distribution becomes closer to the target distribution. Historically, in instance transfer these methods were preceded by feature transformation methods (Pan et al., 2008, 2011). That is why, for the sake of completeness of the presentation we first consider feature transformation methods and then feature selection methods.

The feature transformation methods operate as follows. First they search for a low-dimensional feature space where the target data and source data are relevant. Then, they train classification models on the target data and source data in that space. The Maximum Mean Discrepancy Embedding (MMDE) is of one of the first representative of the feature transformation methods (Pan et al., 2008). It first learns a kernel matrix corresponding to a nonlinear transformation that projects the target data and source data to a latent space in which the distance between the two data sets is minimized. The distance between the data sets is measured by Maximum Mean Discrepancy (MMD) score (Borgwardt et al., 2006). Then, MMDE applies Principal Component Analysis (PCA) (Jolliffe, 2011) on the learned kernel matrix to obtain a low-dimensional feature space for the target data and source data. The new space allows any classification algorithm to be trained on the target and source data. Recently the computational inefficiency of MMDE was addressed in (Pan et al., 2011). As a result a new feature transformation method was proposed, namely Transfer Component Analysis (TCA). TCA has proven itself as effective as MMDE but much more computationally efficient.

Maximum Mean Discrepancy (f-MMD) is a feature selection methods that was proposed in (Uguroglu, 2011). It is based on the MMD score as well. However, instead of finding a low-dimensional representation for the target data and source data jointly, f-MMD identifies a subset of features (called variant features) which contribute the most to the MMD score

and excludes them. The problem of finding variant features is formulated as a convex optimization problem. More precisely, a weight matrix, the diagonal of which corresponds to the weights of all the features, is incorporated in the MMD calculation. The variant features are expected to receive higher weights after optimization, since they minimize the negative MMD score in the objective function. That is to say the variant features are defined as those that contribute most to maximizing the MMD between data sets.

Analyzing the methods considered in this subsection we note mainly two drawbacks. First, these methods may impair geometric or statistical properties of the original target and source data due to the dimensionality reduction. Second, these methods learn the low-dimensional space in an unsupervised manner and dismiss the relevance of the input features for the class labels. Some of the removed features may have a strong class relevance and influence the performance of resulting classification models.

2.3. Conformal Decision Trees for Instance Transfer

Conformal decision trees for instance transfer (CDTIT) were proposed in (Zhou et al., 2017b). They represent an instance-transfer method that combines feature selection and source-instance selection. The method employs the standard decision-tree algorithm (Quinlan, 1993) to construct trees. Univariate instance transfer is performed on the level of feature selection for test nodes of decision trees. More precisely, at each test node the method first selects for every feature the largest source subset which is relevant to the target data when only considering this feature. The relevance of source instances is decided by a statistical test, namely conformal test (Zhou et al., 2017a). Then, the method estimates the predictive power of this feature on the target data and the selected source subset using some measures. Once the predictive power of all features were estimated, the method selects the feature with the highest predictive power for this test node (i.e. the best feature is determined based on the target data and most relevant source instances and its predictive power). We note that constructing a decision tree consists of a series of such steps of univariate instance transfer and feature selection. Thus, the conformal decision trees are essentially an embedded multivariate feature selection method for instance transfer based on univariate source instance selection and feature selection.

The conformal decision trees demonstrated the power of combining feature selection and source-instance selection for instance transfer. However, the results are restricted to decision trees only. In this paper we address this issue by developing a model independent method.

3. Classification Tasks and Solutions

Let X be a instance space defined by K input features $X^k, k \in \{1, 2, \ldots, K\}$ and Y be a finite class set. A domain is defined as a tuple consisting of a labeled space $(X \times Y)$ and a probability distribution P over $(X \times Y)$. We consider first a domain $\langle (X \times Y), P_T \rangle$ that we call a target domain (domain of interest). The target data set T is a multi set of m_T instances $(x_t, y_t) \in X \times Y$ drawn from the target distribution P_T under the i.i.d assumption. Given a test instance $x_{m_T+1} \in X$, *the target classification task* is to find an estimate $\hat{y} \in Y$ for the true class of x_{m_T+1} according to P_T.

Let us consider a second domain $\langle (X \times Y), P_S \rangle$ that we call a source domain. The source data S is a multi set of m_S instances $(x_s, y_s) \in X \times Y$ drawn from the source distribution

P_S under the i.i.d assumption. Assuming that the source domain is relevant to the target domain (i.e. P_S is close to P_T), *the instance-transfer classification task* is to find an estimate $\hat{y} \in Y$ for the true class of x_{m_T+1} according to P_T using source data S as an auxiliary training data.

To solve the classification tasks defined above we train a classifier $h(x)$ in a hypothesis space H of classifiers h ($h : X \to 2^{\mathbb{Y}}$). We note that for the target classification task $h(x)$ is based on T. For the the instance-transfer classification task the classifier $h(x)$ is based on T and selected source instances from S. Once the classifier is available, it outputs for any test instance x_{m_T+1} a posterior distribution of scores $\{s_y\}_{y \in Y}$. The class y with the highest posterior score s_y is the estimated class $\hat{y}$ for the instance x.

4. Conformal test for source relevance

The conformal feature-selection wrappers for instance transfer that we propose in this paper are based on the conformal test (CT) introduced in (Zhou et al., 2017a). The test is used to estimate the relevance of the source data to the target data. In the following, we first describe the CT and the p-value function it employs. Then, we explain and compare two different ways to use the CT for source relevance estimation. Finally, we introduce the algorithm we used for selecting the largest source subset based on the CT.

4.1. Conformal Test

The CT is proposed under the exchangeablity assumption of data generation (Aldous, 1985) [2]. It works with data sequences. Given a target data sequence T and a source data sequence S , it decides the relevance of S to T by testing the null hypothesis that the concatenated data sequence TS was generated by the target distribution P_T under the exchangeability assumption.

To test the null hypothesis, CT makes use of the conformal prediction framework that was introduced in (Shafer and Vovk, 2008; Vovk, 2014). The test employs the nonconformity scores of subsequences of TS as statistics for the null hypothesis. The nonconformity score of a subsequence can be computed based on the nonconformity scores of the instances contained in the subsequence. Given the concatenated sequence TS, the nonconformity score α of an instance $(x, y) \in TS$ is a positive real number that indicates how strange the instance (x, y) is for the sequence T. To compute the instance nonconformity scores we need an *instance* nonconformity function A. If $(X \times Y)^{(*)}$ represent the set of all sequences defined over $(X \times Y)$, the instance nonconformity function A is a mapping from $(X \times Y)^{(*)} \times (X \times Y)$ to $\mathbb{R}^+ \cup \{+\infty\}$ that measures the degree of strangeness of an instance in relation to a sequence.

To compute the sequence nonconformity scores we need a *sequence* nonconformity function. Given the concatenated sequence TS and a subsequence U of some elements of $T \cup S$, the sum sequence nonconformity function returns a score α_U indicating how strange the subsequence U is with respect to all subsequences with size $|U|$ of the data sequence TS.

2. The exchangeability assumption is a weaker assumption than the randomness assumption. It holds for a sequence of random variables if and only if the joint probability distributions of any two permutations of those variables coincide.

Definition 1 (Sum Sequence Nonconformity Function) *Given an instance nonconformity function A, data sequences T and S, and a subsequence 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*

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

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 CT employs sequence nonconformity scores as test statistics. The p-value function of the CT is defined as follows.

Definition 2 (p-Value Function) *The p-value function is a function $t : (X \times Y)^{(*)} \times \mathbb{N} \to [0, 1]$ defined as:*

$$t(T, S) = \frac{|\{U \in \mathcal{P}(TS, m_S) : \alpha_U \geq \alpha_S\}|}{|\mathcal{P}(TS, m_S)|},$$

where $\mathcal{P}(TS, m_S)$ is the set of all subsequences of TS with length $|S| = m_S$, α_U and α_S are sequence nonconformity scores returned by $A^*(T, U)$ and $A^*(T, s)$, respectively.

The validity of the p-value function t was proven in (Zhou et al., 2017a). The p-value returned by the function t indicates the likelihood that the sequence TS was generated by the target distribution P_T under the exchangeability assumption. The higher the p-value is, the more relevant the source sequence is to the target sequence. Therefore, this p-value can be viewed as a non-symmetrical measure of relevance of the source data to the target data.

The CT employs the p-value function t for testing the exchangeability of the concatenated data sequence TS. The source data sequence is relevant to the target data sequence at the significance level $\epsilon_t \in [0, 1]$ if and only if the returned p-value is greater than or equal to ϵ_t.

The CT was extended for data sets (since the sum sequence nonconformity function $A^*(T, U)$ is independent of the ordering of the sequence U) (Zhou et al., 2017a). The p-value function t is redefined as follows:

$$t(T, S) = \frac{|\{U \in \mathcal{C}(T \cup S, m_S) : \alpha_U \geq \alpha_S\}|}{|\mathcal{C}(T \cup S, m_S)|},$$

where T and S are the target and source data sets, respectively, and $\mathcal{C}(T \cup S, m_S)$ is the set of all subsets of $T \cup S$ with size $m_S = |S|$.

4.2. Measure individual relevance and set relevance by the p-value function

As it was mentioned in the previous subsection, the p-value returned by the function t can be viewed as a non-symmetrical measure of relevance of the source data to the target data. Since the p-value function t can be applied to source data with arbitrary size, it allows for measuring the relevance of source data in two different ways. When the size of the source data S equals 1 ($m_S = 1$), function t estimates the individual relevance of a source instance (x_s, y_s) with value $t(T, \{(x_s, y_s)\})$. When the size of the source data is greater than 1 ($m_S > 1$), function t estimates the relevance of the source set as a whole with value $t(T, S)$.

Comparing to individual relevance, set relevance is more precise in terms of source relevance estimation. According to function t, if $S = \{(x_s, y_s)\}$ then $m_S = 1$, and $|\mathcal{C}(T \cup S, m_S)| = m_T + 1$, whereas the numerator is a positive integer. Consequently, the number of possible individual p-value is bounded by $m_T + 1$. If $m_S > 1$, the number of possible set p-value is bounded by $|\mathcal{C}(T \cup S, m_S)|$, which quickly grows much larger than $m_T + 1$. Therefore, the set p-value can better distinguish sets with different nonconformity scores.

Source-subset selection based on individual relevance is computationally more efficient than that based on set relevance. Assume that all instances in the source data S are sorted in increasing order of nonconformity scores. According to Definition 2, we have that the individual relevance of the source instance with index $s(s > 1)$ is always less than or equal to that of the source instance with index $s - 1$, i.e., $t(T, \{(x_s, y_s)\}) \leq t(T, \{(x_{s-1}, y_{s-1})\})$. That is to say the individual relevance is a decreasing function of the index s, and through the index s, it is also a decreasing function of the nonconformity score. When individual relevance is employed to select the largest subset of source instances that passes the CT at a significance level ϵ_t, we can simply apply binary search on the sorted source set to quickly find the last instance that has p-value no less than ϵ_t. The largest source subset is then formed by adding all the instances before this instance and the instance itself.

The set relevance in general is not a monotonic function of the index s, and is not a monotonic function of the nonconformity scores as well. Let S_s be a subset consisting of first $s(s > 1)$ instances of the sorted data S. For each s we may have either $t(T, S_s) \leq t(T, S_{s-1})$ or $t(T, S_s) \geq t(T, S_{s-1})$. To better illustrate this claim, we provide the following example. Assume that TS consists of target instance t_1, t_2, t_3 associated with nonconformity scores 1,4,5, and source instances s_1, s_2, s_3 associated with nonconformity scores 2,3,6 (note that the source instances are sorted by increasing order of the nonconformity scores). In this case, we have $t(T, S_1) = 0.75$, $t(T, S_2) = 0.8$ and $t(T, S_3) = 0.5$. Due to the non-monotonicity, source-subset selection based on set relevance is computationally inefficient.

4.3. Pre-training Approximate Selection for the Relevant Source Subset

If a source subset is generated by the target distribution, the expected p-value of this subset is equal to $\frac{1}{2}$. We call such a subset as relevant source subset $S^{\frac{1}{2}}$. Due to the non-monotonicity of the source relevance finding the largest relevant source subset $S^{\frac{1}{2}}$ may involve repeated application of the function t. To reduce the computational overhead, a pre-training approximate selection algorithm for the relevant source subset (denoted as PASS) was proposed in (Zhou et al., 2017c). The algorithm finds a close approximation $\hat{S}^{\frac{1}{2}}$ of the largest relevant subset $S^{\frac{1}{2}}$ at a small computational cost.

To illustrate the key idea behind the PASS algorithm assume that the source data S is sorted in increasing order of the nonconformity scores $\alpha_{(x_s, y_s)}$ and S_n is a subset consisting of the first n instances of the ordered source data S. By Theorem 3 from (Zhou et al., 2017c), if the average of individual p-values of all instances in the source subset S_n equals $\frac{1}{2} + \frac{1}{2(m_T+1)}$, then the set p-value of S_n is approximately equal to $\frac{1}{2}$. For large target data the term $\frac{1}{2(m_T+1)}$ can be ignored. Therefore, the PASS algorithm finds the largest subset S_n

with the average individual p-value equals $\frac{1}{2}$, which in this case is the approximate subset $\hat{S}^{\frac{1}{2}}$.

The PASS algorithm is presented in Algorithm 1. Given a target data set T, a source data set S, and an instance nonconformity function A, it first computes the nonconformity scores $\alpha_{(x_s,y_s)}$ for the source instances $(x_s, y_s) \in S$ using the instance nonconformity function A. Then, the source data set S is sorted in increasing order of the nonconformity scores $\alpha_{(x_s,y_s)}$; i.e. it becomes sorted in decreasing order of the individual p-values. This implies that the average $\bar{p}_n$ of individual p-values of the instances in S_n is decreasing with the index n. Therefore, the PASS algorithm employs the binary-search method on the sorted source data S to generate the largest source subset S_n with the average individual p-value greater than or equal to $\frac{1}{2}$.

Algorithm 1 PASS: Pre-training selection algorithm based on individual relevance

Input: Target data T, Source data S, Instance nonconformity function A.
Output: Largest source subset S_n with the mean individual p-value $\bar{p}_n$ equal to $\frac{1}{2}$.

for each source instance $(x_s, y_s) \in S$ **do**
  Set the nonconformity score $\alpha_{(x_s,y_s)}$ equal to $A(T, (x_s, y_s))$;
end for
Sort the source data S in increasing order of the nonconformity scores $\alpha_{(x_s,y_s)}$;
Set the left counter L equal to 1 and the right counter R equal to $m_S - 1$;
while $L \leq R$ **do**
  Set the middle index n equal to $\lfloor \frac{L+R}{2} \rfloor$;
  Set $\bar{p}_n$ as the mean of the individual p-values of the instances in S_n;
  Set $\bar{p}_{n+1}$ as the mean of the individual p-values of the instances in S_{n+1};
  if $\bar{p}_n \geq \frac{1}{2}$ and $\bar{p}_{n+1} < \frac{1}{2}$ **then**
    break;
  else if $\bar{p}_n > \epsilon$ **then**
    Set L equal to $n + 1$;
  else
    Set R equal to $n - 1$;
  end if
end while
output S_n.

5. Feature Selection Wrappers

The wrapper method is a standard method for feature selection proposed in (Kohavi and John, 1997). The method examines the space of all possible combinations of the input features X^k according to a chosen search algorithm. The goal is to find that feature combination for which generalization performance of a given classifier is maximized.

To formally introduce the wrapper method we observe that any possible combination of the input features X^k is given by a index set $\mathcal{K} \subseteq \{1, 2, \ldots, K\}$, where K is the number of the features. Hence, the space of all possible combinations of the input features X^k can be uniquely represented by the power set $\mathcal{P}(\{1, 2, \ldots, K\})$. We note that the power set

$\mathcal{P}(\{1,2,\ldots,K\})$ is a partially-ordered set and, thus, it can be systematically examined using any search algorithm. When the search algorithm visits any index set $\mathcal{K} \in \mathcal{P}(\{1,2,\ldots,K\})$, the wrapper method estimates the generalization power of a classifier on the input features X^k for $k \in \mathcal{K}$. Once the search algorithm stops, the wrapper method outputs that index set $\mathcal{K}$ that specifies a set $\{X^k\}_{k\in\mathcal{K}}$ of features for which the generalization power of the classifier is maximized (see Algorithm 2).

Algorithm 2 FSW: Feature Selection Wrapper

Input: K input features X^k, Target data T, Classifier h, Search algorithm SA, Initial index set $\mathcal{I} \subseteq \{1,2,\ldots,K\}$.

Output: index set $\mathcal{K} \subseteq \{1,2,\ldots,K\}$ so that the generalization performance of h is maximized for $\{X^k\}_{k\in\mathcal{K}}$.

1: Set the set V of the visited index sets equal to $\{\mathcal{I}\}$;
2: **repeat**
3: Determine the set C of the candidate index sets from the members of V according to the search algorithm SA;
4: Determine the set R of the index sets that are directly reachable from the index sets in C according to the search algorithm SA;
5: Evaluate the generalization performance of h on the feature subset $\{X^k\}_{k\in\mathcal{K}}$ defined by any index set $\mathcal{K}$ in R;
6: Retain in R those index sets that result in a better generalization performance of h compared with that for any index set in C;
7: Set V equal to $V \cup R$;
8: **until** $R \neq \emptyset$
9: **Output** index set $\mathcal{K}$ in V that results in a maximal generalization performance of h.

6. Conformal Feature-Selection Wrappers for Instance Transfer

In this section we describe the proposed method, namely conformal feature-selection wrappers for instance transfer (CFSWIT). Given the target data, the source data, and a classification model, the method selects a large subset of features and the largest subset of source data that corresponds to the selected features. The distinctive characteristic of the CFSWIT method is that the selection of features and source instances is realized with respect to the classification model. Thus, the CFSWIT method is indeed a wrapper and its pseudo-code is similar to that given in Algorithm 2. The main difference is the way how the generalization performance of the classifier h is estimated for a set features $\{X^k\}_{k\in\mathcal{K}}$ with $\mathcal{K} \subseteq \{1,2,\ldots,K\}$ (see line 5 in Algorithm 2).

The pseudocode for conformal instance-transfer estimation of the classifier's generalization performance (CITCGP) for a set features is given in Algorithm 3. Given a classifier h, all the input features X^k, a particular index set $\mathcal{K}$, target data T and source data S, the algorithm estimates the generalization performance of h for the input features $\{X^k\}_{k\in\mathcal{K}}$ as follows. First, it represents the target data T and the source data S with the features X^k for $k \in \mathcal{K}$ only. Then, the algorithm selects the largest subset $\hat{S}^{\frac{1}{2}}$ of the source data S with set p-value close to $\frac{1}{2}$. For that purpose it employs the PASS algorithm. The PASS algorithm

uses the general non-conformity function based on the classifier h (Shafer and Vovk, 2008). Formally, given the target training data T and a source instance (x, y), the function outputs a score α equal to:

$$\sum_{y_i \in Y, y_i \neq y} s_{y_i},$$

where s_{y_i} is the score of the *i-th* class in the class set Y produced by h trained on target training data T.

Algorithm 3 CITCGP: Conformal Instance-Transfer Estimation of Classifier Generalization Performance

Input: Classifier h, K input features X^k, Index set $\mathcal{K} \subseteq \{1, 2, \ldots, K\}$, Target data T, Source data S.

Output: an estimate of the generalization performance of h for $\{X^k\}_{k \in \mathcal{K}}$.

1: Represent the target data T and the source data S with the features X^k for $k \in \mathcal{K}$;
2: Select the largest subset $\hat{S}^{\frac{1}{2}}$ of the source data S with set p-value close to $\frac{1}{2}$ (using the PASS algorithm with the general non-conformity function based on h);
3: Estimate the generalization performance of the classifier h on $T \cup \hat{S}^{\frac{1}{2}}$ using a repeated cross validation;
4: **Output** estimate of the generalization performance of h.

The general non-conformity function based on the classifier h is used in order to tailor selecting relevant source instances ($\hat{S}^{\frac{1}{2}}$) to the specific set of features $\{X^k\}_{k \in \mathcal{K}}$ through the classifier h. Once the the largest source subset $\hat{S}^{\frac{1}{2}}$ was selected, the generalization performance of the classifier h is estimated on the union of T and $\hat{S}^{\frac{1}{2}}$. The estimation process is implemented using a repeated cross-validation procedure. When it stops, the estimated generalization performance of the classifier h is outputted for the features $\{X^k\}_{k \in \mathcal{K}}$.

Depending on the application (target) domain any useful evaluation criterion can be used to measure the generalization performance of the classifier h. In our experiments we employed area under ROC curve (AUC) (Bradley, 1997).

As it is suggested above our CFSWIT method is represented by Algorithm 2 where the evaluation of the classifier generalization performance is realized by Algorithm 3. The method outputs a feature subset and the largest relevant source-subset for the selected features. In this context, we note that the CFSWIT output is sensitive to the initialization procedure (see Algorithm 2, line 1). If we start with the initial index set $\mathcal{I}$ equal to the set $\{1, 2, \ldots, K\}$ (*backward elimination mode*), the wrappers usually produce relatively large index sets $\mathcal{K}$ (i.e feature sets $\{X^k\}_{k \in \mathcal{K}}$). If we start with the initial index set $\mathcal{I}$ equal to the empty set $\emptyset$ (*forward selection mode*), the wrappers usually result in relatively small feature sets $\{X^k\}_{k \in \mathcal{K}}$. In instance transfer is advisable to be more conservative, i.e. to have larger feature sets $\{X^k\}_{k \in \mathcal{K}}$ to represent the data. In this way we preserve more information from the target data and use hopefully more relevant information from the source data. That is why, our CFSWIT method is initialized in the backward elimination mode with the initial index set $\mathcal{I}$ equal to the set $\{1, 2, \ldots, K\}$. This means that the method aims at finding the largest feature sets $\{X^k\}_{k \in \mathcal{K}}$ which however depends on the search algorithm used.

From the above we may conclude that the CFSWIT method aims at selecting a large subset of features and the largest relevant subset of source data that corresponds to a component of the source distribution estimated to be close the target distribution on the selected features. The distinctive characteristic of the method is that all the estimation procedures are implemented using a given classification model. This means that the final results are tailored to the classification model and aim at boosting its generalization performance. The CFSWIT method can be applied for any type of classification models. Thus, as a wrapper it is a model-independent method.

7. Experiments and Results

This section presents our experimental set-up, results, and analysis. The instance-transfer tasks under study are described in Subsection 7.1. The experimental set-up is provided in Subsection 7.2. In Subsection 7.3, the generalization performance of the CFSWIT method and other standard instance-transfer methods is evaluated and compared.

7.1. Instance-transfer Classification Tasks

In the experiments, we considered five instance-transfer classification tasks defined on real-world data sets that are commonly used in transfer learning research. Each task is given with a target data set and a source data set, described in Table 1. The instance-transfer tasks are defined below.

- The first instance-transfer classification task is the landmine detection task (Xue et al., 2007). The landmine detection data is a collection of data sets related to detecting landmine 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, the data sets “Mine 1" to “Mine 15" correspond to regions that are relatively foliated while the data sets “Mine 16" to “Mine 29" correspond to regions that have bare earth. We used the data set “Mine 29" as the target data, and use the data set “Mine 1" as the source data. To guarantee that the target data and the source data are distributed differently for some features, we manipulated the marginal distribution of the feature with the highest information-gain ratio for the source data by adding random noise generated from the standard uniform distribution.

- The second instance-transfer classification task is the wine quality task (Cortez et al., 2009). The wine quality data consists of 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. We used a random sample from the red wine data as the target data and used a random sample of the white wine data as the source data. To guarantee that the target data and the source data are distributed differently for some features, random noise generated from the standard uniform distribution was added to two features with the highest information-gain ratios for the source data.

- The third instance-transfer classification task is the survival prediction task from the Trial of Intensified versus Standard Medical Therapy in Elderly Patients With Congestive Heart Failure (TIME-CHF)(Brunner-La Rocca et al., 2006; Pfisterer et al., 2009).

Each patient instance is described by 18 bio-markers, and a class label indicating the survival or death of a patient within 5.5 years follow-up. The patient bio-markers and class labels are collected from five different medical centers after the first follow-up period. We used the data from Center 14 as the target data set and data from the other four centers were combined together in a source data set.

- The fourth and fifth instance-transfer classification tasks are defined on the exam records of students from two Portuguese schools: Gabriel Pereira and Mousinho da Silveira (Cortez and Silva, 2008). Each exam record is considered as an instance that is represented by a series of demographic, social, and school related features and a binary grade (pass or no pass). In the experiments, we defined a binary classification task on the grades. The two instance-transfer tasks are defined as follows: the fourth task (referred to as Student 1) use the students' Mathematics exam records of school Mousinho da Silveira as the target data, and use the Portuguese exam records of the same group of students as the source data; the fifth task (referred to as Student 2) employ the same target data as the first task, but use the students' Mathematics exam records of school Gabriel Pereira as the source data.

Table 1: Descriptions of the data sets for instance-transfer classification tasks

Task	Number of Classes	Data set size	
		$\|T\|$	$\|S\|$
Landmine	2	449	690
Wine Quality	3	159	1499
TIME-CHF	2	81	453
Student 1	2	46	46
Student 2	2	46	349

7.2. Experimental Set-up

The CFSWIT method was initialized as follows. The search method employed was the best-first search method. The algorithm for selecting the largest relevant source subset was the algorithm PASS (described in Section 4.3). The algorithm employed the general nonconformity function based on the classifier used. The predictive power of the feature subsets was evaluated using the Area Under the ROC Curve (AUC) (Bradley, 1997). The internal procedure for classifier evaluation in the CFSWIT method was 5-times repeated 5-fold cross validation.

The CFSWIT method was compared with the seven instance-transfer methods presented in Section 2. The methods based on feature selection were represented by the MMDE method and the f-MMD method. The methods were initialized as follows: (1) the dimension size of the reduced feature space for the MMDE method was set equal to 10; (2) the features for the f-MMD method with weights higher than 0.1 were excluded. The methods based on source-instance selection were represented by the TrAdaBoost method, the Dynamic-

TrAdaBoost method, the TraBagg method, and the DoubleBootStrap method. The methods were initialized for iteration number equal to 100.

The CFSWIT method, the methods based on feature selection , and the methods based on source-instance selection were applied for three types of base classifiers: C4.5 decision trees (DT) (Quinlan, 1993), support vector machines (SVM) (Boser et al., 1992) with RBF kernel, and Naive Bayes classifiers (Mitchell, 1997). When the base classifiers were C4.5 decision tree, all the methods were compared with conformal decision trees for instance transfer (CDTIT), since this a method that combines both feature selection and source-subset selection. The implementation of CDTIT was that based on the C4.5 decision trees.

The external procedure of evaluation for all the methods was 10-times repeated 10-fold cross validation on the target data; i.e., the source data was used as auxiliary training data only. The generalization performance of all the methods was evaluated using AUC. The performance of C4.5, SVM and NaiveBayes for the case of no instance transfer was used as baseline. A paired t-test is performed with significance level 0.05 to find significantly better (or worse) results with respect to the corresponding baseline classifier.

7.3. Results

The results when the C4.5 trees were used as baseline classifiers are presented in Table 2. From the table we see that the CDTIT method achieves the best generalization performance for all the instance-transfer classification tasks. It gains a margin of 0.08 over the AUC of the baseline classifier in the best case (the TIME-CHF task). The proposed CFSWIT method has the second best generalization performance (achieves 3 wins out of 5). It achieves significant better results than the baseline classifier over all the tasks, but a bit worse than CDTIT. This is due to the fact that CDTIT is more flexible than CFSWIT: CDTIT performs a multivariate instance transfer as a series of several univariate instance transfers while CFSWIT performs a non-decomposable multivariate instance transfer.

Tasks	Baseline	CFSWIT	CDTIT	MMDE	f-MMD	TrAda-Boost	Dynamic TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	0.55	0.58*	0.59*	0.56	0.52$^-$	0.57	0.56	0.56	0.57
Wine Quality	0.60	0.64*	0.66*	0.58	0.59	0.62	0.63	0.64*	0.66*
TIME-CHF	0.58	0.64*	0.66*	0.55$^-$	0.61*	0.60	0.60	0.64*	0.64*
Student 1	0.71	0.74*	0.77*	0.67$^-$	0.68$^-$	0.65$^-$	0.74	0.61$^-$	0.68$^-$
Student 2	0.71	0.74*	0.78*	0.70	0.71	0.71	0.74*	0.85*	0.71

Table 2: AUCs of CFSWIT, CDTIT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing C 4.5 as the base classifier. $*(-)$ denotes significantly better (worse) results w.r.t the baseline classifier.

The results when SVMs and Naive Bayes were used as baseline classifiers are presented in Tables 3 and 4, respectively. From the tables we see that the CFSWIT method has the best generalization performance compared with the other instance transfer methods: it achieves 3 wins out of 5 for both SVMs and Naive Bayes. The second best is the Double-BootStrap method with 2 wins out of 5 for SVMs only. The CFSWIT method does not result in

negative transfer while any other instance transfer method has at least one experiment with negative transfer.

If we analyze Tables 3 and 4 we observe that the CFSWIT method does not fail because the component of the source distribution on the final subset of features that corresponds to the selected source instances is indeed close to the target distribution for all the experiments. This does not always happen for other methods because they either select features or source instances; i.e. they are much less flexible.

Tasks	Baseline	CFSWIT	MMDE	f-MMD	TrAda-Boost	Dynamic TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	0.59	0.62^{*}	0.62^{*}	0.58	0.55	0.56	0.64^{*}	0.59
Wine Quality	0.72	0.74	0.67^{-}	0.72	0.67^{-}	0.66^{-}	0.70	0.74
TIME-CHF	0.68	0.70^{*}	0.62^{-}	0.70^{*}	0.64^{-}	0.64^{-}	0.67	0.69
Student 1	0.63	0.70^{*}	0.64	0.65	0.63	0.65	0.67	0.71^{*}
Student 2	0.63	0.80^{*}	0.72^{*}	0.74^{*}	0.63	0.64	0.78^{*}	0.72^{*}

Table 3: AUCs of CFSWIT, EmbedSR-DT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing SVM as the base classifier. $*(-)$ denotes significantly better (worse) results w.r.t the baseline classifier.

Tasks	Baseline	CFSWIT	MMDE	f-MMD	TrAda-Boost	Dynamic TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	0.56	0.58^{*}	0.63^{*}	0.59^{*}	0.47^{-}	0.47^{-}	0.56	0.56
Wine Quality	0.72	0.75	0.66^{-}	0.73	0.69^{-}	0.69^{-}	0.74	0.75
TIME-CHF	0.71	0.74^{*}	0.59^{-}	0.74^{*}	0.76^{*}	0.76^{*}	0.72	0.74^{*}
Student 1	0.68	0.79^{*}	0.69	0.70	0.63	0.61^{-}	0.73^{*}	0.71
Student 2	0.68	0.77^{*}	0.66	0.71^{*}	0.62	0.62	0.75^{*}	0.73^{*}

Table 4: AUCs of CFSWIT, EmbedSR-DT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing NaiveBayes as the base classifier. $*(-)$ denotes significantly better (worse) results w.r.t the baseline classifier.

If we compare the results for the methods in Tables 2-4, we observe that the generalization performance is lowest when the base classifiers are the C4.5 decision trees and highest when base classifiers are the Naive Bayes classifiers. This shows the importance of the option of choosing the base classifiers. This option is provided by the model-independent methods such as the CFSWIT method.

8. Conclusion

In this paper we proposed the new method of conformal feature-selection wrappers for instance transfer (CFSWIT). Given a classification model in the presence of target data and source data, the method performs feature selection and source-subset selection for this model.

For that purpose the CFSWIT method examines the space of feature subsets according to a search algorithm chosen. When it evaluates a set of features, it considers both target and source data represented only by these features. Under this constraint, the method first finds the largest relevant set of source instances that can be selected using a conformal source-subset selection procedure from (Zhou et al., 2017c). Then, it estimates the generalization performance of the classification model on the target data and selected source instances. Once the search algorithm completes the examination of the space of feature subsets, the CFSWIT method outputs a large subset of features and the largest relevant subset of source data for these features for which the generalization performance of the classification model is maximized. In this respect the CFSWIT method is similar to the conformal decision trees for instance transfer (Zhou et al., 2017b) and is different from any mechanical combination of instance-transfer methods based on either feature selection or source-instance selection.

The experiments showed that the CFSWIT method is capable of outperforming several instance-transfer methods. This means that practically combining feature selection and source-instance selection is a powerful approach to instance transfer which the CFSWIT method exemplifies as a model-independent method.

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 Aldous. *Exchangeability and related topics*. Springer, 1985.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

Hans Peter Brunner-La Rocca, Peter Theo Buser, Ruth Schindler, Alain Bernheim, Peter Rickenbacher, Matthias Pfisterer, TIME-CHF-Investigators, et al. Management of elderly patients with congestive heart failure—design of the trial of intensified versus standard medical therapy in elderly patients with congestive heart failure (time-chf). *American heart journal*, 151(5):949–955, 2006.

Paulo Cortez and Alice Maria Gonçalves Silva. Using data mining to predict secondary school student performance. 2008.

Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Transferring naive bayes classifiers for text classification. In *Proceedings of the National Conference on Artificiall Intellegence*, volume 22, page 540, 2007a.

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, 2007b.

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

Ian T. Jolliffe. Principal component analysis. In *International Encyclopedia of Statistical Science*, pages 1094–1096. 2011.

Toshihiro Kamishima, Masahiro Hamasaki, and Shotaro 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*, pages 219–228. IEEE, 2009.

Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

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.

Tom M. Mitchell. *Machine learning*. McGraw-Hill, 1997.

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

Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

Matthias Pfisterer, Peter Buser, Hans Rickli, Marc Gutmann, Paul Erne, Peter Rickenbacher, André Vuillomenet, Urs Jeker, Paul Dubach, Hansjürg Beer, et al. Bnp-guided vs symptom-guided heart failure therapy: the trial of intensified vs standard medical therapy in elderly patients with congestive heart failure (time-chf) randomized trial. *Jama*, 301 (4):383–392, 2009.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

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

Ben Tan, Yangqiu Song, Erheng Zhong, and Qiang Yang. Transitive transfer learning. In *Proceedings of the 21th SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1155–1164. ACM, 2015.

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

Carbonell Jaime Uguroglu, Selen. Feature selection for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 430–442. Springer, 2011.

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

Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016.

Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(Jan): 35–63, 2007.

Shuang Zhou, Evgueni Smirnov, and Ralf Peeters. Conformal region classification with instance-transfer boosting. *International Journal on Artificial Intelligence Tools*, 24(6): 1560002, 2015.

Shuang Zhou, Evgueni Smirnov, Gijs Schoenmakers, Kurt Driessens, and Ralf Peeters. Testing exchangeability for transfer decision. *Pattern Recognition Letters*, 88:64–71, 2017a.

Shuang Zhou, Evgueni Smirnov, Gijs Schoenmakers, and Ralf Peeters. Conformal decision-tree approach to instance transfer. *Annals of Mathematics and Artificial Intelligence*, 81 (1-2):85–104, 2017b.

Shuang Zhou, Evgueni Smirnov, Gijs Schoenmakers, and Ralf Peeters. Conformity-based source subset selection for instance transfer. *Neurocomputing*, 258:41–51, 2017c.