Conformal Feature-Selection Wrappers and ensembles for negative-transfer avoidance

Neurocomputing 397 (2020) 309–319

Contents lists available at ScienceDirect

Neurocomputing

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

Conformal Feature-Selection Wrappers and ensembles for negative-transfer avoidance

Shuang Zhou[a,*], Evgueni Smirnov[b], Gijs Schoenmakers[b], Ralf Peeters[b], Xi Wu[c]

[a] Department of PD & IGT, Philips Research China, China
[b] Department of Data Science and Knowledge Engineering, Maastricht University, the Netherlands
[c] School of Computing, Chengdu University of Information Technology, China

ARTICLE INFO

Article history:
Received 28 February 2019
Revised 29 June 2019
Accepted 9 September 2019
Available online 28 November 2019

Keywords:
Instance transfer
Conformal prediction
Feature Selection
Wrappers
Ensembles

ABSTRACT

In this paper we propose two methods for instance transfer based on conformal prediction. As a distinctive character, both of the methods are model independent and combine feature selection and source-instance selection to avoid negative transfer. The methods have been tested experimentally for different types of classification model on several benchmark data sets. The experimental results demonstrate that the new methods are capable of outperforming significantly standard instance transfer methods.

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 [1,2]. 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 [3], especially for small target data [4].

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 [4–7]. 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 distribution by estimating how probable is that the target data and source data are generated from the same distribution [8].

Following the results of both approaches, if we find that the source distribution is close to the target distribution, we transfer 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 do not transfer the source data. The reason is that in this case the transfer can be negative; i.e. it can cause a significant drop in the generalization performance of the target classification model. To avoid the negative transfer, 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 non-empty, we add it to the target data and then train the target classification model.
- *feature selection:* we select a subset of (input) features for which the source distribution is estimated to be close to the target distribution. If the subset is nonempty, the target and source

* Corresponding author.
E-mail addresses: shuang.zhou@philips.com (S. Zhou), smirnov@maastrichtuniversity.nl (E. Smirnov), gm.schoenmakers@maastrichtuniversity.nl (G. Schoenmakers), ralf.peeters@maastrichtuniversity.nl (R. Peeters), xi.wu@cuit.edu.cn (X. Wu).

[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.

https://doi.org/10.1016/j.neucom.2019.09.105
0925-2312/© 2019 Elsevier B.V. All rights reserved.

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. [9] proposed the only method available for mutually dependent feature and source-instance selection. The method realizes this property using decision trees [10] in an univariate manner. 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 two *model-independent* methods for the task of combining feature selection and source-instance selection: Conformal Feature-Selection Wrappers for Instance Transfer (CFSWIT) and Conformal Ensembles for Instance Transfer (CEIT). The methods are shown to be capable of avoiding negative transfer. Below we briefly summarize their similarities and differences.

The CFSWIT method is a wrapper method for feature selection [11]. Given a classification model that needs instance transfer, CFSWIT 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, the method first selects the *largest* relevant set of source instances using a conformal source-subset selection procedure proposed by Zhou et al. [12]. Then, it estimates the generalization performance of the classification model on the target data and selected source instances. Once the method has visited all the feature subsets according the chosen search strategy, 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.

The CFSWIT method starts the process of examining the space of feature subsets from the full set of features. This results in relatively *large* final subsets of features. Thus, the CFSWIT method outputs *large* subsets of features and the *largest* relevant subsets of source data that can be generated by the target distribution w.r.t. the selected features.

The CEIT method is an ensemble method. Given a classification model that needs instance transfer, CEIT (like CFSWIT) examines the space of feature subsets according to a chosen search strategy. However, when it evaluates a set of features, it considers only the target data represented by these features. If the generalization performance of the classification model in this case is acceptable on the target data, the method selects the *largest* relevant set of source instances using the conformal source-subset selection procedure. Then, it trains a classification model on the target data and largest relevant source subset, and adds that model to an ensemble. Once CEIT has visited all the feature subsets according the chosen search strategy, it outputs the ensemble. The ensemble consists of all the classification models generated while searching in the space of possible feature subsets. The models can be very different (i.e. diverse) due to the feature variety and instance transfer. The models' diversity can result in accurate ensemble rules from a repertoire of rules [13] from majority vote, score averaging etc.

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 a conformal test and its corresponding source-subset selection procedure. The wrapper method is given in Section 5. Sections 6 and 7 introduce the CFSWIT method and CEIT method, respectively. The experiments are provided in Section 8. Section 9 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 [12]. 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 [5] and Dynamic-TrAdaBoost [14]. These methods are similar to the AdaBoost algorithm [15] 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 [5]. 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 [16] 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 [17] 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 [18,19]. 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 [18]. 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 [20]. Then, MMDE applies Principal Component Analysis (PCA) [21] 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 [19]. 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 [22]. 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 [9]. They represent an instance-transfer method that combines feature selection and source-instance selection. The method employs the standard decision-tree algorithm [10] 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 relevant 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 [8]. 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 multi-variate 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 model independent methods.

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 $(h : X \to \mathbb{R}^{|Y|})$. We note that for the target classification task $h(x)$ is based on T. For 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 [8]. The test is used to decide 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 relevant source subset based on the CT.

4.1. Conformal test

The CT is proposed under the exchangeablity assumption of data generation [23].[2] It works with data sequences. Given a target data sequence T and a source data sequence S, it decides the

[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.

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 [24,25]. 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.

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

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 [8]. 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) [8]. 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 $m_S = |S|$.

We note that the p-value function defined in this way exhibits an analogy to the notion of Wilcoxon rank-sum test (see [26], Chapter 1). Hence, for big data sets, in which enumerating all combinations in $\mathcal{C}(T \cup S, m_S)$ is intractable, we propose to approximate the set p-value through Wilcoxon rank-sum test. More specifically, we assign ranks from 1 to $m_T + m_S$ to all instances in $T \cup S$ according to their nonconformity scores in ascending order. In this setting, the nonconformity score α_U of any subset U of $T \cup S$ with size m_S is replaced by the rank sum W that equals to $\sum_{(x_m,y_m)\in U} R_{(x_m,y_m)}$ where $R_{(x_m,y_m)}$ is the rank of nonconformity score $\alpha_{(x_m,y_m)}$ of instance $(x_m, y_m) \in U$. 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 $t(T, S)$; i.e., the p-value function can be implemented using the rank-sum test.

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 the latter definition of function t, if $S = \{(x_s, y_s)\}$ then $m_S = 1$ and $|\mathcal{C}(T \cup S, m_S)| = m_T + 1$ which implies that the number of possible individual p-values 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 relevant 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, it can be transferred. Interesting enough the expected p-value of this subset is equal to $\frac{1}{2}$ and, thus, it is known as relevant source subset $S^{\frac{1}{2}}$ (see [12]). 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 [12]. 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 [12], 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 relevant source subset S_n with the average individual p-value greater than or equal to $\frac{1}{2}$.

5. Feature Selection Wrappers

The wrapper method is a standard method for feature selection proposed in [11]. 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 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 $L \le R$ 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 $\bar{p}_n \ge \frac{1}{2}$ 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 $S_n$.
```

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 the classifier $h$
       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 $R = \emptyset$
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 introduce Conformal Feature-Selection Wrapper for Instance transfer (CFSWIT). Given the target data, the source data, and a classification model, CFSWIT 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 $\frac{1}{2}$. We note that this subset can be viewed as the one generated by the target distribution and thus it can be added to the target data [12].

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

```
Input:  Classifier $h$, $K$ input features $X^k$, Index set $\mathcal{K} \subseteq \{1, 2, \dots, K\}$, 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 classifier $h$ on $T \cup \hat{S}^{\frac{1}{2}}$ using a repeated validation;
4: Output estimate of the generalization performance of $h$.
```

Selecting the largest subset $\hat{S}^{\frac{1}{2}}$ is realized by the PASS algorithm (see Sub-Section 4.3). The PASS algorithm uses the general non-conformity function based on the classifier h [24]. Formally, given the target training data T and an 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 ith class in the class set Y produced by h trained on target training data T.

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 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

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 $\{1, 2, \dots, K\}$ (*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. Therefore, our CFSWIT method is initialized in the backward elimination mode with the initial index set $\mathcal{I}$ equal to the set $\{1, 2, \dots, K\}$. 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 time complexity of the CFSWIT method equals $O(FPW)$, where F is the number of feature-subsets visited according to the search algorithm SA, P is the time complexity of the PASS algorithm, and W is the time complexity of the validation process based on the target data T and transferred source data $S^{\frac{1}{2}}$ (step 3 of the Algorithm 3). We note that F is bounded by 2^K. Thus, computationally efficient search algorithms have to be used in combination with the CFSWIT method.

7. Conformal ensembles for instance transfer

Conformal Ensembles for Instance Transfer (CEIT) is an ensemble method which diversity is based on feature variety and instance transfer. The CEIT method searches in the space of possible combinations of the input features (similar to the CFSWIT method). Only if the generalization performance of the current feature subset $\{X^k\}_{k\in\mathcal{K}}$ is acceptable on the target data, CEIT determines the largest source subset $\hat{S}^{\frac{1}{2}}$ for that feature subset (in contrast to the CFSWIT method). Since $\hat{S}^{\frac{1}{2}}$ can be viewed as generated by the target distribution, the method trains a classifier on the target data and source subset $\hat{S}^{\frac{1}{2}}$, and adds that classifier to the final ensemble. Thus, the classifiers' diversity within the ensembles is realized due to different feature subsets selected and different source data transferred.

The pseudo-code for the CEIT method is given in Algorithm 4. Given a classifier h, all the input features X^k, target data T, source data S, a search algorithm SA and a performance threshold λ, the method operates as follows. It first initializes the visited index set V of features X^k equal to the set $\{1, 2, \dots, K\}$, and the final ensemble classifier set h_E equal to the empty set. Then, the CEIT method follows the same strategy as the CFSWIT method to expand the space of feature subsets and to examine each candidate feature set $\{X^k\}_{k\in\mathcal{K}}$ (Step 4 to 5). If the generalization performance (e.g., AUC) of a feature set $\{X^k\}_{k\in\mathcal{K}}$ is estimated to be higher or equal to the performance threshold λ (Step 8), the largest subset $\hat{S}^{\frac{1}{2}}$ of source data corresponding to $\{X^k\}_{k\in\mathcal{K}}$ is selected (Step 9 and 10). After that, a candidate classifier h is built on the target data and $\hat{S}^{\frac{1}{2}}$, and h is added to the final ensemble h_E (Step 11 and 12). The method repeats Steps 3 to 17 until there is no feature sets $\{X^k\}_{k\in\mathcal{K}}$ that can be visited using the search algorithm SA. When this happens the method outputs an ensemble h_E.

The ensemble h_E outputted by the CEIT method is a set of classifiers h. Thus, any ensemble classification rule is applicable (e.g., majority vote). In our experiments we applied the rule of averaging class probabilities [27].

The time complexity of the CEIT method equals $O(FPV)$, where F is the number of feature-subsets visited according to the search algorithm SA, P is the time complexity of the PASS algorithm, and V is the time complexity of the validation process based on the target data T (step 7 of the Algorithm 4). We note that we provide a worst-case time complexity indication since steps 9–12 are executed depending on the generalization performance of the current model h. If we compare the time complexities of the CFSWIT and CEIT methods, we observe that the CEIT method is more

Algorithm 4 CEIT: Conformal Ensembles for Instance Transfer.

```
Input: $K$ input features $X^k$, Target data $T$, Source data $S$
Classifier $h$, Search algorithm $SA$, Performance threshold $\lambda$,
Initial index set $\mathcal{I} \subseteq \{1, 2, \ldots, K\}$.
Output: Ensemble classifier $h_E$.
Set the set $V$ of the visited index sets equal to $\{\mathcal{I}\}$;
Set the ensemble classifier $h_E$ equal to $\{\}$;
repeat
    Determine the set $C$ of the candidate index sets from the members of $V$ according to the search algorithm $SA$;
    Determine the set $R$ of the index sets that are directly reachable from the index sets in $C$ according to the search algorithm $SA$;
    for any index set $\mathcal{K}$ in $R$ do
        Evaluate the generalization performance $P$ of $h$ on the feature subset $\{X^k\}_{k\in\mathcal{K}}$ and the target data $T$;
        if $P \geq \lambda$ then
            Represent the target data $T$ and the source data $S$ with the features $X^k$ for $k \in \mathcal{K}$;
            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 $h$);
            Train a candidate classifier $h_k$ on $T \cup \hat{S}^{\frac{1}{2}}$;
            Set $h_E$ equal to $h_E \cup h_k$;
        end if
    end for
    Retain in $R$ those index sets that result in a better generalization performance of $h$ compared with that for any index set in $C$;
    Set $V$ equal to $V \cup R$;
until $R = \emptyset$
if $h_E = \emptyset$ then
    Train a classifier $h$ on the target data $T$;
    Set $h_E$ equal to $h_E \cup h$;
end if
Output Ensemble classifier $h_E$.
```

computationally efficient. Even in the worst case the validation process is based on the target data only.

8. Experiments and results

This section presents our experimental set-up, results, and analysis. The instance-transfer tasks under study are described in Section 8.1. The experimental set-up is provided in Section 8.2. In Section 8.3, the generalization performance of the CFSWIT method and CEIT method as well as the generalization performance of other standard instance-transfer methods are evaluated and compared. Subsection 8.4 discusses the influence of performance-threshold parameter λ on the CEIT ensembles.

8.1. 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 specified in Table 1. The instance-transfer tasks are briefly described below.

- The first instance-transfer classification task is the landmine detection task [28]. 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 “Mine1” to “Mine15” correspond to regions that are relatively foliated while the data sets “Mine16” to “Mine29” correspond to regions that have bare earth. We used the data set “Mine29” as the target data, and use the data set “Mine1” 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.

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

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

- The second instance-transfer classification task is the wine quality task [29]. 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) [30,31]. 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 centres after the first follow-up period. We used the data from Center 14 as the target data set and data from the other four centres 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 [32]. 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.

8.2. Experimental set-up

The CFSWIT method was initialized as follows. The search method for the feature-subset space 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

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

Tasks	Baseline	CFSWIT	CEIT	CDTIT	MMDE	f-MMD	TrAda-Boost	Dynamic TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	0.55	0.58^{*}	**0.59^{*}**	**0.59^{*}**	0.56	0.52^{-}	0.57	0.56	0.56	0.57
Wine uality	0.60	0.64^{*}	**0.67^{*}**	0.66^{*}	0.58	0.59	0.62	0.63	0.64^{*}	0.66^{*}
TIME-CHF	0.58	0.64^{*}	**0.70^{*}**	0.66^{*}	0.55^{-}	0.61^{*}	0.60	0.60	0.64^{*}	0.64^{*}
Student 1	0.71	0.74^{*}	**0.81^{*}**	0.77^{*}	0.67^{-}	0.68^{-}	0.65^{-}	0.74	0.61^{-}	0.68^{-}
Student 2	0.71	0.74^{*}	0.75^{*}	**0.78^{*}**	0.70	0.71	0.71	0.74^{*}	0.75^{*}	0.71

algorithm employed the general nonconformity function based on the classifier used. The generalized performance of the feature subsets was evaluated using the Area Under the ROC Curve (AUC) [33]. The internal procedure for classifier evaluation in the CFSWIT method was 5-times repeated 5-fold cross validation.

The CEIT method was initialized analogously to the CDSWIT method. The additional parameter λ (performance threshold) was set to a value in the range of 0.1 $\pm$ *AUC* of the base classifier for which the generalization performance of that classifier is maximized.

The CFSWIT method and CEIT method were 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 proposed methods, 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) [10], support vector machines (SVM) [34] with linear kernel, and Naive Bayes classifiers [35]. When the base classifiers were C4.5 decision tree, all the methods were compared with conformal decision trees for instance transfer (CDTIT) (given in Section 2.3), 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 [35]. All the instance transfer methods were implemented in Java and embedded in the WEKA framework [27]. All the base classifiers were directly called from WEKA with the default parameters employed. One-against-all multi-class SVM were adopted for the Wine Quality task.

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 (linear kernel) 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.

8.3. 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 CEIT method achieves the best generalization performance for most of the instance-transfer classification tasks (4 out of 5). It achieves the maximal gain of 0.12 over the AUC of the C4.5 trees (baseline) for the TIME-CHF task. The CDTIT method achieves the second best generalization performance (2 out of 5 wins). The CFSWIT method has the third best generalization performance. It achieves significant better results than the baseline classifier, the methods based on feature selection, and most of the methods based on source-instance selection.

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 CEIT method with 3 wins out of 5 for SVMs and 1 wins out of 5 for Naive Bayes. Moreover, CFSWIT and CEIT never result in negative transfer while any other instance transfer method has at least one experiment with negative transfer.

If we analyse the results presented in Tables 2–4 we may conclude that the superior generalization performance of the CFSWIT method, the CEIT method, and the CDTIT method is due to the fact that these methods implement both feature selection and source-instance selection in contrast to other approaches to instance transfer. The three methods managed to find in all the experiments sufficiently large subset of features and the largest subset of source data that can be generated by the target distribution w.r.t. the selected features.

If we compare the CFSWIT method and the CDTIT method for the case of decision trees, we observe that CDTIT has a better generalization performance. This is mainly because CDTIT performs a multivariate instance transfer as a series of univariate instance transfers while CFSWIT performs just one non-decomposable multivariate instance transfer. This means that CDTIT is capable of extracting more relevant source information than CFSWIT.

If we compare the CEIT method and the CDTIT method for the case of decision trees, we observe that CEIT has a better generalization performance. This is mainly because CDTIT performs a multivariate instance transfer as a series of univariate instance transfers while CEIT performs a series of non-decomposable multivariate instance transfers. This means that CEIT is capable of extracting more diverse source information than CDTIT.

If we compare the CFSWIT method and the CEIT method, we may conclude that the CEIT method has more potential. This is due to three reasons. First, as mentioned above CEIT performs a multivariate instance transfer as a series of non-decomposable multivariate instance transfers while CFSWIT performs just one non-decomposable multivariate instance transfer. Second, the CEIT method is an ensemble method and thus it is capable of reducing the variance component of the error of the classifier.[3] Third, the CEIT method is more computationally efficient: in contrast to CFSWIT it transfers only for those feature sets which generalization performance is acceptable on the target data only.

To study the impact of instance transfer on the CFEWIT method and the CEIT method we performed an additional set of series of experiments. In the first series we compared the generalization performance of CFEWIT with that of standard Feature-Selection Wrappers (FSW) [11]. The results are provided in Table 5 for three types of base classifiers: C4.5, SVM and Naive Bayes. As is shown

[3] This explains that CEIT outperforms CFSWIT for high-variance classifiers such as decision trees.

Table 3
AUCs of CFSWIT, CEIT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing SVM as the base classifier. Best results are highlighted in bold. $*(-)$ denotes significantly better (worse) results w.r.t the baseline classifier.

Tasks	Baseline	CFSWIT	CEIT	MMDE	f-MMD	TrAda-Boost	Dynamic TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	0.59	0.62*	0.62*	0.62*	0.58	0.55	0.56	**0.64***	0.59
Wine Quality	0.72	**0.74**	0.73	0.67$^-$	0.72	0.67$^-$	0.66$^-$	0.70	**0.74**
TIME-CHF	0.68	0.70*	**0.72***	0.62$^-$	0.70*	0.64$^-$	0.64$^-$	0.67	0.69
Student 1	0.63	0.70*	**0.71***	0.64	0.65	0.63	0.65	0.67	**0.71***
Student 2	0.63	**0.80***	0.78*	0.72*	0.74*	0.63	0.64	0.78*	0.72*

Table 4
AUCs of CFSWIT, CEIT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing NaiveBayes as the base classifier. Best results are highlighted in bold. $*(-)$ denotes significantly better (worse) results w.r.t the baseline classifier.

Tasks	Baseline	CFSWIT	CEIT	MMDE	f-MMD	TrAda-Boost	Dynamic TrAda-Boost	TraBagg	Double-Bootstrap
Landmine	0.56	0.58*	0.57	**0.63***	0.59*	0.47$^-$	0.47$^-$	0.56	0.56
Wine Quality	0.72	**0.75**	0.73	0.66$^-$	0.73	0.69$^-$	0.69$^-$	0.74	**0.75**
TIME-CHF	0.71	0.74*	**0.76***	0.59$^-$	0.74*	**0.76***	**0.76***	0.72	0.74*
Student 1	0.68	**0.79***	0.74*	0.69	0.70	0.63	0.61$^-$	0.73*	0.71
Student 2	0.68	**0.77***	0.75*	0.66	0.71*	0.62	0.62	0.75*	0.73*

Table 5
AUCs of FSW and CFSWIT employing C4.5, SVM and NaiveBayes as base classifiers, respectively. Best results in every row are highlighted in bold. $*$ denotes significantly better results w.r.t the baseline classifier.

Task	C4.5			SVM			Naive Bayes		
	Baseline	FSW	CFSWIT	Baseline	FSW	CFSWIT	Baseline	FSW	CFSWIT
Landmine	0.55	0.55	0.58*	0.59	0.58	**0.62***	0.56	0.58*	0.58*
Wine Quality	0.60	0.64*	0.64*	0.72	0.73	0.74	0.72	0.71	**0.75**
TIME- CHF	0.58	0.58	0.64*	0.68	0.70*	0.70*	0.71	0.73*	**0.74***
Student 1	0.71	0.69	0.74*	0.63	0.72*	0.70*	0.68	0.67	**0.79***
Student 2	0.71	0.69	0.74*	0.63	0.72*	**0.80***	0.68	0.67	0.77*

Table 6
AUCs of FSE and CEIT employing C4.5, SVM and NaiveBayes as base classifiers, respectively. Best results in every row are highlighted in bold. $*$ denotes significantly better results w.r.t the baseline classifier.

Task	C4.5			SVM			Naive Bayes		
	Baseline	FSE	CEIT	Baseline	FSE	CEIT	Baseline	FSE	CEIT
Landmine	0.55	0.58*	0.59*	0.59	0.58	**0.62***	0.56	0.55	0.57
Wine Quality	0.60	0.64*	0.67*	0.72	0.70	**0.73**	0.72	0.71	**0.73**
TIME- CHF	0.58	0.62*	0.70*	0.68	0.72*	0.72*	0.71	0.75*	**0.76***
Student 1	0.71	0.73*	**0.81***	0.63	0.67*	0.71*	0.68	0.66	0.74*
Student 2	0.71	0.73*	0.75*	0.63	0.67*	**0.78***	0.68	0.66	0.75*

in the table, FSW outperforms the base classifiers in 7 out of 15 cases which is less than half of the experiments. However, CFEWIT outperforms the base classifiers in all the 15 experiments. This implies that instance transfer indeed helps wrapper feature selection.

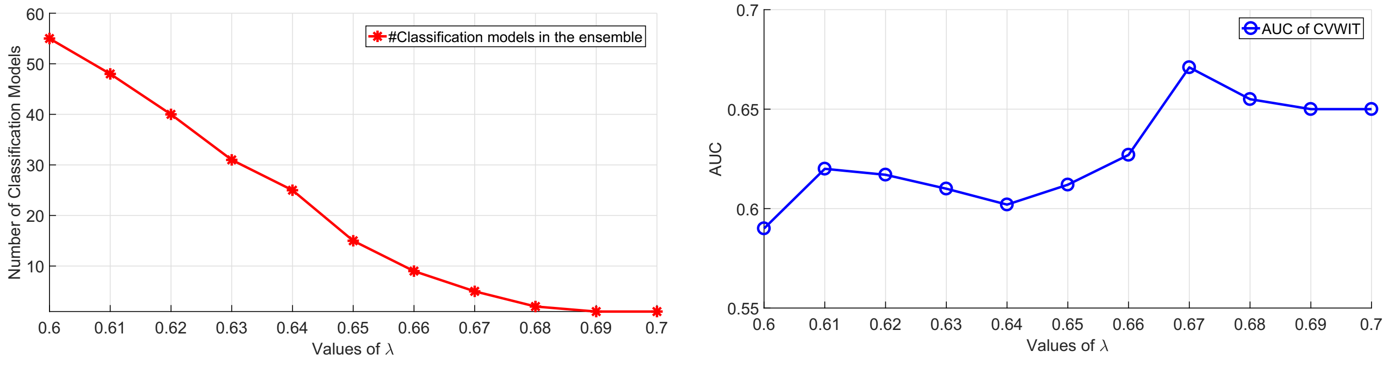
In the second additional series of experiments we compared the generalization performance of CEIT with that of Feature-Selection Ensembles (FSE). The latter are essentially CEIT ensembles that do not employ any instance transfer. The results are provided in Table 6 for three types of base classifiers: C4.5, SVM and Naive Bayes. As is shown in the table, FSE outperforms the baseline classifiers in most of the cases, especially for high-variance classifiers (due to reducing the variance component of the error). Comparing the performance of CEIT and FSE, CEIT achieves better results in all of the 15 cases, which demonstrates the benefit brought by instance transfer.

8.4. Study on the size of the CEIT ensembles

The size of the CEIT ensembles and thus their generalization performance are controlled by the performance-threshold parameter λ. Fig. 1(a) and (b) shows the number of classification models and the generalization performance (AUC) of a CEIT ensemble in the range of λ from 0.6 to 0.7 for the wine quality task. The plots show that the number of classification models decreases as the value of λ increases. The CEIT generalization performance demonstrates an upward trend when λ raises from 0.60 to 0.67. The reason is that the ensembles in this range contain classification models built on feature subsets with increasing discriminating power on the target data. For λ from 0.67 to 0.7 the CEIT generalization performance decreases, which is mainly due to the small number of classification models contained in the ensemble.

Fig. 2(a) and (b) shows the number of classification models and the generalization performance (AUC) of a CEIT ensemble for the TIME-CHF task. The plots show similar patterns and can be explained analogously as for the wine quality task.

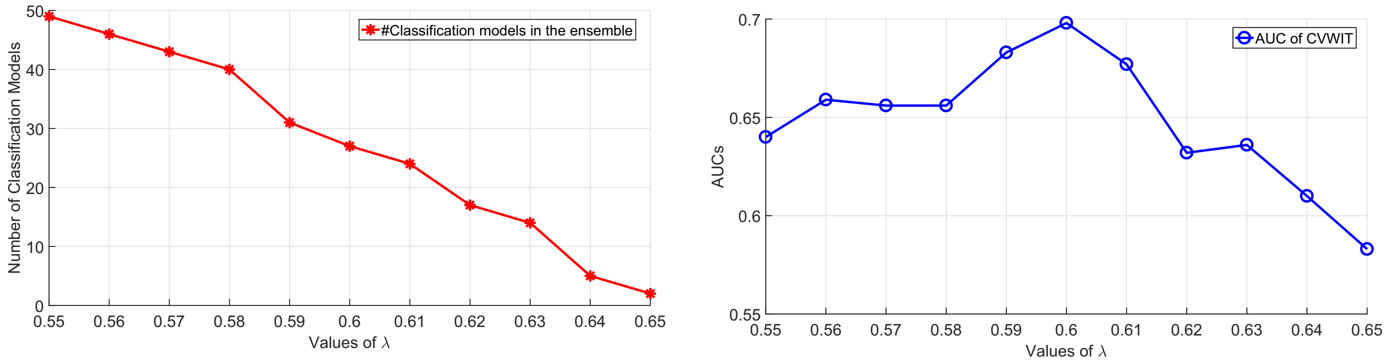
When comparing the plots for the wine quality task and the TIME-CHF task, we find the maximal generalization performance of the CEIT ensembles is achieved with 5 classification models for the wine quality task and with 27 classification models for the TIME-CHF task. The reason for this big difference in the number of classification models is the different relevance of the source data w.r.t. the target data (computed by the p-value function t). For the TIME-CHF task the relevance is higher, and thus diversity that instance transfer brings to the classification models is lower. Thus, more classification models are needed. For the wine quality task the situation is opposite: the relevance of the source data is lower,



(a) Number of Classification Models

(b) AUCs

Fig. 1. Number of classification models and AUCs of with different λ for the wine quality task.



(a) Number of Classification Models

(b) AUCs

Fig. 2. Number of classification models and AUCs of with different λ for the TIME-CHF task.

and thus diversity that instance transfer brings to the classification models is higher. Thus, less classification models are needed.

9. Conclusion

In this paper we propose two methods for instance transfer: Conformal Feature-Selection Wrappers for Instance Transfer (CFSWIT) and Conformal Ensembles for Instance Transfer (CEIT). The methods share many similarities like search in the space of possible feature subsets. However, they are different in several aspects:

(1) final results: CFSWIT outputs a feature set and a source subset while CEIT outputs an ensemble.
(2) instance-transfer usage: CFSWIT uses the transferred source data (and target data) for estimating the generalization performance of the candidate feature sets. CEIT uses the transferred source data for (additionally) diversifying the classification models in the final ensemble.
(3) computational efficiency: CEIT transfers source data only for those feature subsets whose generalization performance is estimated to be acceptable on the target data; i.e., the number of source-subset selection procedure performed is significantly less.

The experiments showed that CFSWIT and CEIT are capable of outperforming several instance-transfer methods. To the best of our knowledge these methods are the only known *model-independent* methods that avoid negative transfer by *combining feature selection and source-instance selection*.

Future research will focus on new model-independent methods for negative-transfer avoidance. In the context of feature selection in addition to wrappers (e.g., the CFSWIT method) we need to develop univariate filters and multivariate filters that employ source-instance selection. The filters are not necessary to be better in terms of the final results, however, there should be definitely a gain in computational efficiency. In the context of ensemble learning in addition to the majority-voting ensembles (e.g., the CEIT method) it is worth exploring other types of ensembles such as boosting, stacking, error-correcting output codes,.etc. Im this way, we expect faster methods with better capabilities of avoiding negative transfer.

References

[1] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[2] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, J. Big Data 3 (1) (2016) 9.
[3] L. Torrey, J. Shavlik, Transfer learning, Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, andTechniques., IGI Global, 2009, p. 242.
[4] W. Dai, Q. Yang, G.-R. xue, Y. Yu, Boosting for transfer learning, in: Proceedings of the Twenty-forth International Conference on Machine Learning, ACM, 2007, pp. 193–200.
[5] W. Dai, G.-R. Xue, Q. Yang, Y. Yu, Transferring Naive Bayes classifiers for text classification, in: Proceedings of the National Conference on Artificiall Intellegence, 22, 2007, p. 540.
[6] S. Zhou, E. Smirnov, R. Peeters, Conformal region classification with instance–transfer boosting, Int. J. Artif. Intell. Tools 24 (6) (2015) 1560002.
[7] B. Tan, Y. Song, E. Zhong, Q. Yang, Transitive transfer learning, in: Proceedings of the Twenty-firth SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1155–1164.

[8] S. Zhou, E. Smirnov, G. Schoenmakers, K. Driessens, R. Peeters, Testing exchangeability for transfer decision, Pattern Recogn. Lett. 88 (2017) 64–71.
[9] S. Zhou, E. Smirnov, G. Schoenmakers, R. Peeters, Conformal decision-tree approach to instance transfer, Ann. Math. Artif. Intell. 81 (1–2) (2017) 85–104.
[10] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., 1993.
[11] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.
[12] S. Zhou, E. Smirnov, G. Schoenmakers, R. Peeters, Conformity-based source subset selection for instance transfer, Neurocomputing 258 (2017) 41–51.
[13] O. Sagi, L. Rokach, Ensemble learning: a survey, Wiley Interdiscipl. Rev. Data Mining Knowl. Discovery 8 (4) (2018).
[14] S. Al-Stouhi, C.K. Reddy, Adaptive boosting for transfer learning using dynamic updates, in: Proceedings of the Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 60–75.
[15] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann, 1996, pp. 148–156.
[16] D. Lin, X. An, J. Zhang, Double-bootstrapping source data selection for instance-based transfer learning, Pattern Recogn. Lett. 34 (11) (2013) 1279–1285.
[17] T. Kamishima, M. Hamasaki, S. Akaho, Trbagg: A simple transfer learning method and its application to personalization in collaborative tagging, in: Proceedings of the Ninth IEEE International Conference on Data Mining, IEEE, 2009, pp. 219–228.
[18] S.J. Pan, J.T. Kwok, Q. Yang, Transfer learning via dimensionality reduction., in: Proceedings of the AAAI, 8, 2008, pp. 677–682.
[19] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210.
[20] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Schölkopf, A.J. Smola, Integrating structured biological data by kernel maximum mean discrepancy, Bioinformatics 22 (14) (2006) e49–e57.
[21] I.T. Jolliffe, Principal component analysis, in: Proceedings of the International Encyclopedia of Statistical Science, 2011, pp. 1094–1096.
[22] S. Uguroglu, J. Carbonell, Feature selection for transfer learning, in: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 430–442.
[23] D. Aldous, Exchangeability and Related Topics, Springer, 1985.
[24] G. Shafer, V. Vovk, A tutorial on conformal prediction, J. Mach. Learn. Res. 9 (2008) 371–421.
[25] V. Vovk, The basic conformal prediction framework, in: Conformal Prediction for Reliable Machine Learning Theory, Adaptations and Applications, Elsevier, 2014, pp. 1–20.
[26] E.L. Lehmann, H.J. D'Abrera, Nonparametrics: Statistical Methods based on Ranks, Springer New York, 2006.
[27] P. Christopher, H. Mark, F. Eibe, W. Ian, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016.
[28] Y. Xue, X. Liao, L. Carin, B. Krishnapuram, Multi-task learning for classification with dirichlet process priors, J. Mach. Learn. Res. 8 (Jan) (2007) 35–63.
[29] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, Decision Supp. Syst. 47 (4) (2009) 547–553.
[30] H.P. Brunner-La Rocca, P.T. Buser, R. Schindler, A. Bernheim, P. Rickenbacher, M. Pfisterer, TIME-CHF-Investigators, et al., Management of elderly patients with congestive heart failuredesign of the trial of intensified versus standard medical therapy in elderly patients with congestive heart failure (time-chf), Am. Heart J. 151 (5) (2006) 949–955.
[31] M. Pfisterer, P. Buser, H. Rickli, M. Gutmann, P. Erne, P. Rickenbacher, A. Vuillomenet, U. Jeker, P. Dubach, H. 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) (2009) 383–392.
[32] P. Cortez, A. Silva, Using data mining to predict secondary school student performance, in: Proceedings of the EUROSIS, 2008.
[33] A.P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recogn. 30 (7) (1997) 1145–1159.
[34] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM, 1992, pp. 144–152.
[35] T.M. Mitchell, Machine Learning, McGraw-Hill, 1997.

Shuang Zhou received a bachelors degree in software engineering from University of Electronic Science and Technology of China, UESTC, in 2009. She received a masters degree in artificial intelligence and Doctor Degree from Department of Data Science and Knowledge Engineering (DKE), Maastricht University, the Netherlands, in 2012 and 2017, respectively. Currently, she works as a data scientist in Philips Research China. Her research interests include transfer learning, conformal prediction, and their applications.

Evgueni Smirnov is an assistant professor of Artificial Intelligence at the Department of Knowledge Engineering, Maastricht University. He graduated in Computer Science from the Technical University of Sofia in 1988, and he earned his Ph.D. degree in Artificial Intelligence at Maastricht University in 2001. Research interests include: Data mining (reliable prediction, feature selection); Machine learning (transfer learning, ensemble learning, kernel methods, version spaces); Applications of data mining to medicine, transportation, machinery-automation systems, and education. Evgueni Smirnov has co-edited several books on reliable data mining (Springer, IEEE). He supervised/executed six commercial data-mining projects. His team of Ph.D. students does research on transfer learning, ensemble learning, and medical data mining.

Gijs Schoenmakers is an Assistant Professor at the Department of Knowledge Engineering (DKE) at Maastricht University, The Netherlands. My main research field is Game Theory. Within this field my research focuses primarily on equilibria and equilibrium refinements in repeated and stochastic games. The most important result he achieved in this field is establishing the existence of subgame perfect equilibria in recursive perfect information games (joint work with Jeroen Kuipers, Janos Flesch and Koos Vrieze). Last year he was asked to join DKEs research on Machine Learning. His task here primarily consists of uncovering mathematical structures within the concepts that are being researched.

Ralf Peeters (1964) is a full professor in Applied Mathematics at Maastricht University. He graduated at Delft University of Technology (1988) and received his Ph.D. degree from the Free University, Amsterdam (1994). He currently is vice-chair and Research Director of the Department of Data Science and Knowledge Engineering. His research interests include: system identification and machine learning, signal processing, data science, optimization, and applications of knowledge engineering to medicine and the life sciences.

Dr. Xi Wu is the Professor and dean of Department of Computer Science, Chengdu University of Information Technology, and he is also the deputy director of Collaborative Innovation Center for Image and Geospatial Information of Sichuan Province, P.R. China. His main research area is the development of novel methods for analysis of imaging data. He has been also involved in cognitive studies cooperated with Computational intelligence since 2008 when he joined the Sichuan University and Vanderbilt University Institute of Imaging Science, Vanderbilt University for Ph.D. study. In 2012, He was with Oxford Centre for Functional MRI of the Brain, University of Oxford as a research intern.