Ensembles based on Conformal Instance Transfer

Shuang Zhou SHUANG.ZHOU@PHILIPS.COM
Department of Big Data and AI, Philips Research, Shanghai, China

Evgueni Smirnov SMIRNOV@MAASTRICHTUNIVERSITY.NL
Gijs Schoenmakers GM.SCHOENMAKERS@MAASTRICHTUNIVERSITY.NL
Department of Data Science and Knowledge Engineering, Maastricht University, The Netherlands

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

Abstract

In this paper we propose a new ensemble method based on conformal instance transfer. The method combines feature selection and source-instance selection to avoid negative transfer in a model-independent way. It was tested experimentally for different types of classifiers on several benchmark data sets. The experiment results demonstrate that the new method is capable of outperforming significantly standard instance transfer methods.

Keywords: Instance Transfer, Conformal Prediction, Ensembles

1. Introduction

Instance transfer received a significant attention in the last decade (Pan and Yang, 2010; Weiss et al., 2016). It aims at improving the generalization performance of classification models for a *target* domain of interest using the data from an auxiliary *source* domain (Pan and Yang, 2010; Weiss et al., 2016). In this paper we consider the case when the target and source domains share the same input feature space and the same class-label set but differ in the underlying probability distributions. In this context, if the source domain is found to be relevant to the target domain; i.e. the source distribution is close to the target distribution, the source data can be transferred to the target data and a classification model can be trained for the target domain. This can significantly improve the model's generalization performance (Torrey and Shavlik, 2009), especially for small target data (Dai et al., 2007b).

In many practical situations, however, the source distribution is not close enough to the target distribution. In this case we either do not use the source data or we do transfer the source data which causes usually a drop in the generalization performance of the target classification model, an indication of negative transfer. To avoid negative transfer, we can follow one of the three scenarios that we summarize below (Zhou et al., 2018):

- **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.

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.

- **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 data are represented by the selected features only. The source data is added to the target data, and, then, the target classification model is trained.
- **feature selection and source-instance selection:** 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 scenario 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 two methods were proposed that combine feature selection and source-instance selection in a mutual dependent way. The first method is *model dependent* (Zhou et al., 2017b). It builds decision trees (Quinlan, 1993) by selecting features on the target and selected source data. The second method is *model independent* (Zhou et al., 2018). It is essentially a wrapper method that selects subsets of features based on the target and selected source data. We note than both methods implement source-data selection using procedures from conformal instance transfer (Zhou, 2017). This is due to the fact that the conformal instance transfer provides statistical guarantees for transfer decisions.

In this paper we propose a new *model-independent* method for combining feature selection and source-instance selection: Ensembles based on Conformal Instance Transfer (ECIT). Given a classification model that needs instance transfer, ECIT examines the space of feature subsets according to a chosen search strategy. 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 on the target data is acceptable, the method selects the *largest* relevant set of source instances using the conformal source-subset selection procedure from (Zhou et al., 2017c) [2]. Then, it trains a classification model on the target data and largest relevant source subset, and adds that model to an ensemble. Once ECIT 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 (Sagi and Rokach, 2018) from majority vote, score averaging etc.

The remainder of this paper 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 provides basics of the conformal instance transfer: it introduces a conformal test for transfer decisions and its corresponding source-subset selection procedure employed by the ECIT method. The method itself is introduced in Sections 5. The experiments are provided in Section 6. Section 7 concludes the paper.

2. We note that source-subset selection can be implemented using other approaches for reliable prediction such as version spaces (Smirnov, 1992; Smirnov et al., 2004, 2006b) and ROC analysis (Vanderlooy et al., 2006).

2. Related Works

As it was stated in the previous section there exist three types of methods for instance transfer when the relevance of the source domain is not sufficient for the target domain. In this section we provide an overview of the methods within these three types.

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; Zhou et al., 2015) 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 and Carbonell, 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. Methods based on Feature Selection and Source-Instance Selection

As it is stated in the previous section there exist two methods that combine feature selection and source-instance selection. Below we summarize the two methods.

Decision Trees based on Conformal Instance Transfer (DTCIT) are a model dependent method. They employ the standard decision-tree algorithm (Quinlan, 1993). 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 (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.

Feature-Selection Wrappers based on Conformal Instance Transfer (FSWCIT) are a model independent method. Given a classification model that needs instance transfer, FSWCIT 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., 2017c). 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 FSWCIT 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 FSWCIT 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.

In this paper we propose a new method for combining feature selection and source-instance selection. The method is *model-independent* in contrast to the DTCIT method and it is *computationally efficient* in contrast to the FSWCIT method. Below we provide a necessary background information and description of the next 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 ($h : X \to \mathbb{R}^{|Y|}$). 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 Instance Transfer

As it is stated in Section 1 the ECIT method that we propose in this paper is based on the conformal instance transfer. In this section we first introduce the conformal test (CT) for transfer decision (Zhou et al., 2017a). 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 exchangeability assumption of data generation (Aldous, 1985) [3]. 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. There exist several nonconformity functions defined in a general way and in a model-specific way (Shafer and Vovk, 2008; Smirnov and Kaptein, 2006; Smirnov et al., 2006a, 2009).

3. 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.

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 $t : (X \times Y)^{(*)} \times (X \times Y)^{(*)} \to [0, 1]$ 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 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 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 close to $\frac{1}{2}$ and, thus, it is known as relevant source subset $S^{\frac{1}{2}}$ (see (Zhou et al., 2017c)). 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 relevant 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. Ensembles based on Conformal Instance Transfer

Ensembles based on conformal instance transfer (ECIT) form an ensemble method which diversity is based on feature variety and instance transfer. The ECIT method searches in the search space of possible combinations of the input features. If the generalization performance of the current feature subset is acceptable on the target data, ECIT determines the largest source subset $\hat{S}^{\frac{1}{2}}$ for that feature subset. 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 ECIT method is given in Algorithm 2. 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: (a) the set V of index sets of the visited feature subsets equal to an index set $\mathcal{I} \subseteq \{1, 2, \ldots, K\}$ [4], and (b) the final ensemble classifier set h_E equal to the empty set. Then, the ECIT method employs the search algorithm A to determine the index sets $\mathcal{K}$ of the feature subsets $\{X^k\}_{k\in\mathcal{K}}$ that will be visited next (Steps 4 to 5). If the generalization performance (e.g., AUC) of the classifier h on a feature set $\{X^k\}_{k\in\mathcal{K}}$ is estimated to be higher or equal to the performance threshold λ (Steps 7 and 8), the feature set is considered as useful. In this case the largest subset $\hat{S}^{\frac{1}{2}}$ of source data corresponding to $\{X^k\}_{k\in\mathcal{K}}$ is selected (Steps 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 ECIT method is a set of classifiers h. Thus, any ensemble classification rule is applicable (e.g., majority vote) (Sagi and Rokach, 2018). In our experiments we applied the rule of averaging class probabilities (Pal et al., 2016).

6. Experiments and Results

This section presents our experimental set-up, results, and analysis. The instance-transfer tasks under study are described in Subsection 6.1. The experimental set-up is provided in Subsection 6.2. In Subsection 6.3, the generalization performance of the FSWCIT method and ECIT method as well as the generalization performance of other standard instance-transfer methods are evaluated and compared. Subsection 6.4 discusses the influence of performance-threshold parameter λ on the ECIT ensembles.

6.1. 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.

4. We note that any feature subset is represented by index set $\mathcal{I} \subseteq \{1, 2, \ldots, K\}$ that contains the indices of the features in the subset.

Algorithm 2 ECIT: Ensembles based on Conformal Instance Transfer

Input: K input features X^k, Target data T, Source data S
Classifier h, Search algorithm SA, Performance threshold λ,
Initial index set $\mathcal{I} \subseteq \{1, 2, \ldots, K\}$.
Output: Ensemble classifier h_E.

Set the set V of the index sets of the visited feature 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 in the search space from the index sets in C according to the search algorithm SA;
  for all index sets $\mathcal{K}$ in R **do**
    Evaluate the generalization performance P of the classifier h trained 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.

- 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). 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 5 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.

6.2. Experimental Set-up

The ECIT method was initialized as follows. The search method for the feature-subset space was the best-first search method. The method employed the general nonconformity

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

function based on the classifier used. The generalized performance of the feature subsets was evaluated using the Area Under the ROC Curve (AUC) (Bradley, 1997). The internal procedure for classifier evaluation in the ECIT method was 5-times repeated 5-fold cross validation (see Step 7 in Algorithm 2). The parameter λ (performance threshold) was set to a value in the range of $AUCBC \pm 0.1$ for which the generalization performance of that classifier is maximized, where $AUCBC$ is the AUC of a base classifier.

The ECIT method was compared with the nine 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 methods based on feature selection and source-instance selection were represented by the FSWCIT method.

The methods from the experiments were applied for three types of base classifiers: C4.5 decision trees (DT) (Quinlan, 1993), support vector machines (SVM) (Boser et al., 1992) with linear kernel, and Naive Bayes classifiers (Mitchell, 1997). When the base classifiers were C4.5 decision tree, all the methods were compared with Decision trees based on conformal instance transfer (DTCIT) (given in Subsection 2.3), since this a method that combines both feature selection and source-subset selection. The implementation of DTCIT was that based on the C4.5 decision trees (Mitchell, 1997).

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 was performed with significance level 0.05 to find significantly better (or worse) results with respect to the corresponding base classifier.

6.3. Results

The results when the C4.5 trees were used as base classifiers are presented in Table 2. From the table we see that the ECIT 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 (base) for the TIME-CHF task. The DTCIT method achieves the second best generalization performance (2 out of 5 wins). The FSWCIT method has the third best generalization performance. It achieves significant better results than the base classifier, the methods based on feature selection, and most of the methods based on source-instance selection.

Tasks	Base	FSWCIT	ECIT	DTCIT	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 Quality	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*	0.85*	0.71

Table 2: AUCs of FSWCIT, ECIT , DTCIT, 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 base classifier.

The results when SVMs and Naive Bayes were used as base classifiers are presented in Tables 3 and 4, respectively. From the tables we see that the FSWCIT 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 ECIT method with 3 wins out of 5 for SVMs and 1 wins out of 5 for Naive Bayes. Moreover, FSWCIT and ECIT never result in negative transfer while any other instance transfer method has at least one experiment with negative transfer.

If we analyze the results presented in Tables 2, 3, and 4 we may conclude that the superior generalization performance of the ECIT method, the FSWCIT method, and the DTCIT 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 ECIT method and the DTCIT method (for the case of decision trees), we observe that ECIT has a better generalization performance. This is mainly because DTCIT performs a multivariate instance transfer as a series of univariate instance transfers while ECIT performs a series of non-decomposable multivariate instance transfers. This means that ECIT is capable of extracting more diverse source information than DTCIT.

If we compare the ECIT method and the FSWCIT method, we may conclude that the ECIT method has more potential. This is due to three reasons. First, as mentioned above ECIT performs a multivariate instance transfer as a series of non-decomposable multivariate instance transfers while FSWCIT performs just one non-decomposable multivariate instance transfer. Second, the ECIT method is an ensemble method and thus it is capable of reducing

the variance component of the error of the classifier [5]. Third, the ECIT method is more computationally efficient: in contrast to FSWCIT it transfers only for those feature sets which generalization performance is acceptable on the target data only.

Tasks	Base	FSWCIT	ECIT	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 3: AUCs of FSWCIT, ECIT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing SVM as the base classifier. $*(-)$ denotes significantly better (worse) results w.r.t the base classifier.

Tasks	Base	FSWCIT	ECIT	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 4: AUCs of FSWCIT, ECIT, MMDE, f-MMD, TrAdaBoost, Dynamic-TrAdaBoost, TraBagg and DoubleBootStrap employing NaiveBayes as the base classifier. $*(-)$ denotes significantly better (worse) results w.r.t the base classifier.

To understand the impact of ensemble method and instance transfer on the performance of ECIT, we compared the performance of ECIT with that of Feature-Selection Ensemble (FSE) which is essentially ECIT without instance transfer. The results are given for C4.5, SVM and Naive Bayes in Table 5. As we can see from the table, FSE outperforms the base classifiers in most of the cases, especially for high-variance classifiers. It confirms our conclusion that an ensemble method is capable of reducing the variance component of the classification error. Comparing the performance of ECIT and FSE, ECIT achieves better results in all of the cases, which demonstrates the benefit brought by instance transfer.

6.4. Study of the Size of the ECIT Ensembles

The size of the ECIT ensembles and thus their generalization performance are controlled by the performance-threshold parameter λ. Figures 1(a) and 1(b) show the number of classification models and the generalization performance (AUC) of a ECIT ensemble in the range of λ from 0.6 to 0.7 for the wine quality task.

5. This explains that ECIT outperforms FSWCIT for high-variance classifiers such as decision trees.

Task	C4.5			SVM			Naive Bayes		
	Base	FSE	ECIT	Base	FSE	ECIT	Baseline	FSE	ECIT
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*

Table 5: AUCs of FSE and ECIT employing C4.5, SVM and NaiveBayes as base classifiers, respectively. $*$ denotes significantly better results w.r.t the base classifier.

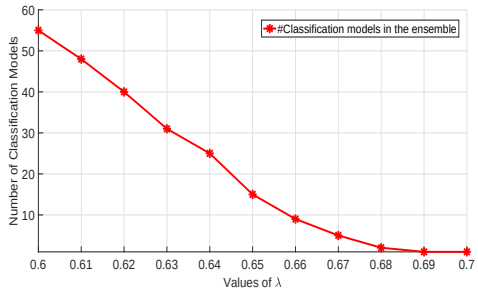
The plots show that the number of the classification models in the ECIT ensembles increases as the value of λ decreases. The ECIT generalization performance first grows with the number of the classification models (λ decreases from 0.7 to 0.67). The growth is due to the increasing number of the classification models which diversity is boosted by instance transfer. Then we observe a pick and a non-monotonic decrease of that performance (λ decreases from 0.67 to 0.6). The decrease can be explained by the fact that the classification models become too diverse; i.e. the number of very different selected source subsets based on different feature subsets becomes too big.

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

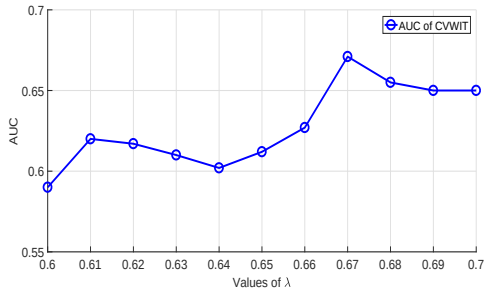
When comparing the plots for the wine quality task and the TIME-CHF task, we find the maximal generalization performance of the ECIT ensembles is achieved with 5 classification models for the wine quality task and with 27 classification models for the TIME-CHF task. The reason for this big difference 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, and thus diversity that instance transfer brings to the classification models is higher. Thus, less classification models are needed.

7. Conclusion

In this paper we propose a new method Ensembles based on Conformal Instance Transfer (ECIT). The distinctive feature of the method is that it employs instance transfer for ensemble diversification. The ECIT method belongs to the family of methods that combine feature selection and source-instance selection to avoid negative transfer. In contrast with the other members of that family the ECIT method is simultaneously model independent and computationally efficient.

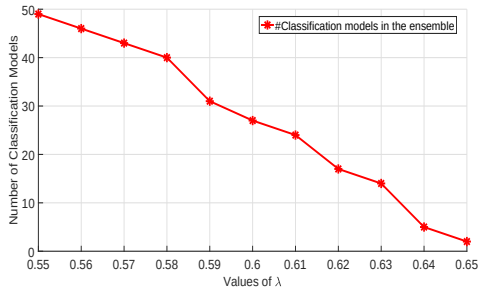


(a) Number of Classification Models

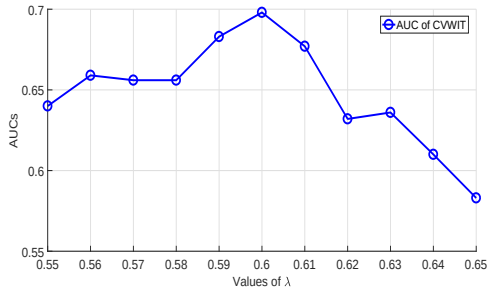


(b) AUCs

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



(a) Number of Classification Models



(b) AUCs

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

Future research will focus on speeding up the ECIT method, especially the conformal part of source selection. We plan to employ for this purpose cross conformal predictors (Vovk, 2015) and their faster version (Beganovic and Smirnov, 2018).

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.

Dorian Beganovic and Evgueni Smirnov. Ensemble cross-conformal prediction. In *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops, ICDM Workshops*, pages 870–877. IEEE Computer Society, 2018.

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.

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.

Christopher Pal, Mark Hall, Eibe Frank, and Ian Witten. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2016.

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.

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

Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(4), 2018.

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

Evgueni Smirnov. Space fragmenting - a method of disjunctive concept acquisition. In *Proceedings of the Fifth International Conference on Artificial Intelligence: Methodology, Systems, Applications, AIMSA 1992*, pages 97–104. Elsevier, 1992.

Evgueni Smirnov and Rianne Kaptein. Theoretical and experimental study of a meta-typicalness approach for reliable classification. In *Workshops Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 739–743. IEEE Computer Society, 2006.

Evgueni Smirnov, Ida Sprinkhuizen-Kuyper, and Georgi Nalbantov. Unanimous voting using support vector machines. In *Proceedings of the Sixteenth Belgium-Netherlands Conference on Artificial Intelligence*, pages 147–152, 2004.

Evgueni Smirnov, Ida Sprinkhuizen-Kuyper, Georgi Nalbantov, and Stijn Vanderlooy. Meta-typicalness approach to reliable classification. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI 2006)*, volume 141, pages 811–812. IOS Press, 2006a.

Evgueni N. Smirnov, Ida G. Sprinkhuizen-Kuyper, Georgi I. Nalbantov, and Stijn Vanderlooy. Version space support vector machines. In *Proceedings of the 17th European Conference on Artificial Intelligence, (ECAI 2006)*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 809–810. IOS Press, 2006b.

Evgueni N. Smirnov, Georgi I. Nalbantov, and A. M. Kaptein. Meta-conformity approach to reliable classification. *Intell. Data Anal.*, 13(6):901–915, 2009. doi: 10.3233/IDA-2009-0400. URL https://doi.org/10.3233/IDA-2009-0400.

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

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

Stijn Vanderlooy, Ida Sprinkhuizen-Kuyper, and Evgueni Smirnov. An analysis of reliable classifiers through ROC isometrics. In *Proceedings of the ICML 2006 Workshop on ROC Analysis (ROCML 2006)*, pages 55–62, 2006.

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

Vladimir Vovk. Cross-conformal predictors. *Ann. Math. Artif. Intell.*, 74(1-2):9–28, 2015.

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. *Bridging Conformal Prediction and Instance Transfer*. PhD thesis, Maastricht University, 2017.

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.

Shuang Zhou, Evgueni N. Smirnov, Gijs Schoenmakers, Ralf Peeters, and Tao Jiang. Conformal feature-selection wrappers for instance transfer. In *In Proceedings of the 7th Symposium on Conformal and Probabilistic Prediction and Applications, COPA 2018, 11-13 June 2018, Maastricht, The Netherlands.*, volume 91 of *Proceedings of Machine Learning Research*, pages 96–113. PMLR, 2018.