Theoretical and Experimental Study of a Complexity Measure for Analogical Transfer

Fadi Badra[1], Marie-Jeanne Lesot[2], Aman Barakat[2], and Christophe Marsala[2]

[1] Université Sorbonne Paris Nord, LIMICS, Sorbonne Université, INSERM, Bobigny, France, badra@univ-paris13.fr
[2] Sorbonne Université, CNRS, LIP6, Paris, France, firstname.name@lip6.fr

Abstract. Analogical transfer addresses classification and regression tasks, performing a plausible inference according to which similar instances are likely to be associated with similar labels. This paper proposes a detailed study of the ordinal implementation of this principle by the so-called `CoAT` algorithm, that is based on a data set complexity measure quantifying the number of inversions observed when ranking the data according to their instance or label similarities. At a theoretical level, it establishes an upper bound of the complexity measure, providing a reference value to which the observed one can be compared. At an algorithmic level, it proposes an optimization that allows decreasing the computational complexity by one order of magnitude. At an experimental level, it studies the correlation of the complexity measure with the accuracy of the conducted label inference, as well as with the classification task difficulty, as captured by the class overlapping degree.

Keywords: computational analogy, analogical transfer, similarity-induced ranking, data set complexity

1 Introduction

Analogical reasoning [7] has been recognized by psychologists to be at the core of human thoughts [6]; computational analogy proposes, among others, to transpose its principles to artificial intelligence and machine learning models [3, 15]. In particular, computational analogical transfer addresses prediction tasks, such as classification and regression, and implements a special type of plausible inference according to which similar instances are likely to be associated with similar labels. Since the pioneering logical formulation of analogical transfer [4], many formulations of this principle have been proposed: a first type of approach consists in looking for local alignment of the instance and the labels similarities [2, 8, 13, 14]. A second one expresses the similarity principle as a negative constraint that excludes labels that are not similar enough to the ones of the similar instances [10]. A third approach, related to the second one, consists in measuring the extent to which a potential label is supported as compared to the

other available data [5, 9]. Other works study how to take into account domain knowledge in the inference [11, 12].

This paper focuses on another approach, implemented in the Complexity-based Analogical Transfer algorithm `CoAT` [1], presented in more details in Section 2. `CoAT` relies on an ordinal formalization of the above mentioned principle according to which the ranking induced by the similarity measure applied to the data instances should be identical to the ranking induced by the similarity measure applied to their associated labels. As a consequence, the predicted label must be such that it does not entail ranking inversions. More precisely, the `CoAT` algorithm relies on the definition of a complexity measure that counts the number of so-called inversions, i.e. case triples violating the ordering requirement.

This paper proposes an extended study of the `CoAT` method, from three complementary points of view, namely theoretically, computationally and experimentally: it first establishes an upper bound of the complexity measure, providing a reference value to assess the relevance of the analogical assumption for the considered data set and similarity measures. It then addresses algorithmic concerns about the `CoAT` method and shows that it can be optimized to reduce its computational complexity to a tractable value. Finally, it presents an experimental study of the complexity measure, in particular regarding its correlation with the accuracy of the conducted label inference, its correlation with the prediction task difficulty for classification tasks (measured as the class overlap). Experimental results about its computational cost are also presented.

The paper is structured as follows: Section 2 recalls the principle of the `CoAT` algorithm, Section 3 presents the established upper bound on the complexity measure, successively for regression and classification tasks. Section 4 discusses the algorithmic optimization that allows reducing the computational cost of the `CoAT` approach for inference. Section 5 describes the conducted experimental study. Finally, Section 6 concludes the paper.

2 Reminder on Complexity-based Analogy

This section recaps the principles of the Complexity-based Analogical Transfer algorithm `CoAT` [1] studied in this paper. After introducing the notations used throughout the paper, it recalls the definition of the complexity measure on which `CoAT` relies and then describes the inference algorithm itself.

2.1 Notations

Throughout the paper, D denotes a case base containing n instances, or *cases*, c_i, $i = 1..n$, defined as ordered pairs (s_i, r_i): s_i denotes the *situation*, i.e. a feature-based description of the considered instance, or problem, and r_i its associated label, or *outcome* or solution, either a numerical value, in the case of regression tasks, or a categorical value, in the case of classification tasks. Throughout the paper, the situation and outcome that constitute a case are identified by their identical index.

In addition, σ_S denotes a similarity measure applied to the situations, for instance derived from the Euclidean distance in case all features are numerical. σ_R denotes a similarity measure applied to the outcomes. Formally, a similarity measure applied to elements from a domain $\mathcal{X}$ is defined as a function $\sigma : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ that satisfies a reflexivity constraint: there exists a value $M \in \mathbb{R}^+$ such that $\forall (x, y) \in \mathcal{X}$, $\sigma(x, y) \leq \sigma(x, x) = M$ and $\sigma(x, x) = \sigma(y, y)$, i.e. the similarity of any object to itself is maximal. In addition, a similarity measure is most of the time supposed to be symmetrical.

2.2 Ordinal Analogical Principle: Complexity Definition

As mentioned in the introduction, the `CoAT` algorithm [1] relies on an ordinal understanding of the basic analogical principle, according to which similar instances are to be associated with similar labels: the complexity measure at the core of the algorithm quantifies the extent to which the ordering induced by the situation similarity measure σ_S is similar to the ordering induced by the outcome similarity measure σ_R.

More formally, the following qualitative continuity constraint is tested on each triple of cases (c_0, c, c'), with $c_0 = (s_0, r_0)$, $c = (s, r)$, and $c' = (s', r')$:

$$\text{if } \sigma_S(s_0, s) \geq \sigma_S(s_0, s'), \text{ then } \sigma_R(r_0, r) \geq \sigma_R(r_0, r')$$

This constraint expresses that each time a situation s_0 is more similar to a situation s than to a situation s', this order is preserved on their associated outcomes. Any violation of the constraint is called an inversion of similarity, defined as a Boolean value:

$$inv(c_0, c, c') = (\sigma_S(s_0, s) \geq \sigma_S(s_0, s')) \wedge (\sigma_R(r_0, r) < \sigma_R(r_0, r'))$$

It leads to the following definition of *inversion set* for a given case c with respect to a case base D and two similarity measures σ_S and σ_R (omitted in the notation to ease it) using the same function name, with a different arity:

$$inv(c) = \{(c_i, c_j) \in D \times D \mid inv(c, c_i, c_j) \text{ holds } \} \tag{1}$$

Finally, the complexity measure for a case base D and two similarity measures σ_S and σ_R counts the number of such inversions observed in the case base:

$$\Gamma(D, \sigma_S, \sigma_R) = \sum_{c \in D} |inv(c)| \tag{2}$$

2.3 Ordinal Analogical Inference Algorithm

The complexity measure Γ defined in Eq. (2) can then be used either to select appropriate similarity measures, as the ordered pair (σ_S, σ_R) among a list of candidates that minimizes Γ, or to perform inference for a new situation [1]. In the latter case, for a new situation s, the transfer inference consists in predicting

Algorithm 1 Complexity-based Analogical Transfer `CoAT` [1]

inputs: s (new situation), D (case base), R (set of candidate outcomes), σ_S, σ_R (situation and outcome similarity measures)
output: predicted outcome r_s for the new situation s
$min\Gamma = \Gamma(D \cup \{(s, r_0)\}, \sigma_S, \sigma_R)$ for a specific arbitrary $r_0 \in R$
for $r \in R$ **do**
  $D' = D \cup \{(s, r)\}$
  $\Gamma' = \text{computeGamma}(D', \sigma_S, \sigma_R)$ using Alg. 2
  if $\Gamma' < min\Gamma$ **then**
    $min\Gamma = \Gamma'$
    $r_s = r$
  end if
end for
return r_s

Algorithm 2 Complexity computation : computeGamma [1]

inputs: D (case base), σ_S, σ_R (situation and outcome similarity measures)
output: value of $\Gamma(D, \sigma_S, \sigma_R)$
$\Gamma = 0$
for $c_0 \in D$ **do**
  for $c \in D$ **do**
    for $c' \in D$ **do**
      if $inv(c_0, c, c')$ **then**
        $\Gamma = \Gamma + 1$
      end if
    end for
  end for
end for
return Γ

the outcome r_s that minimizes Γ applied to the considered data set enriched with the additional case (s, r_s):

$$r_s = \arg\min_{r \in R} \Gamma(D \cup \{(s, r)\}, \sigma_S, \sigma_R)$$

where R denotes a set of candidate outcomes. R is e.g. defined as the set of possible classes in case of classification tasks or as an interval of possible numerical values for regression tasks.

The `CoAT` inference algorithm [1] reproduced in Algorithm 1 then consists in testing all possible values r for the outcome and outputting the one that leads to the minimal complexity when the considered data set is completed with the candidate case (s, r).

3 Theoretical Property of the Complexity Measure: Upper Bound

This section proposes a theoretical study of the complexity measure introduced in [1] and recalled in the previous section, establishing the value of its upper bound for a given case base and given similarity measures. Such a bound provides a reference value that defines a quality scale for the observed complexity value. Indeed, the `CoAT` algorithm relies on comparisons of candidate complexity values and selects the smallest one. The same principle is applied when the complexity measure is used to select appropriate similarity measures, as the ones that minimize Γ. However, such comparative approaches do not give indication about the quality of these selections, i.e. whether these best choices are actually relevant ones. Establishing a complexity upper bound provides a reference value, corresponding to the worst case, to define a quality scale.

This section establishes the values of this bound, first in the general case, then focusing on the case of binary classification. Indeed, for the latter, a tighter bound can be provided: the worst case established in general appears to be too pessimistic in the case of binary classification.

3.1 General Case

The following theorem states the value of the complexity upper bound and shows it can be considered as tight, stating the case in which the bound can be attained.

Theorem 1. *For a case base D and two similarity measures σ_S and σ_R, the maximum value that the complexity $\Gamma(D, \sigma_S, \sigma_R)$ can take is $\Gamma_{max} = \frac{n^2(n-1)}{2}$ where $n = |D|$.*

There exists a pair (σ_S, σ_R) such that this bound can be attained if all outcome values observed in D are distinct.

To demonstrate this theorem, we first show that this value is an upper bound, and exhibit the configuration in which it can be attained.

Value of the Upper Bound Establishing the value of the upper bound relies on the observation that, for any case c, among the two ordered pairs (c_i, c_j) and (c_j, c_i), at most one belongs to the set of inversions $inv(c)$. Indeed, assuming without loss of generality that $(c_i, c_j) \in inv(c)$, then by definition $\sigma_S(s, s_i) \geq \sigma_S(s, s_j)$ and $\sigma_R(r, r_i) < \sigma_R(r, r_j)$. Having $(c_j, c_i) \in inv(c)$ would require that $\sigma_R(r, r_j) < \sigma_R(r, r_i)$, which contradicts this hypothesis.

In addition, $c_i \neq c_j$ since an equality would lead to $\sigma_R(r, r_i) = \sigma_R(r, r_j)$, which is not compatible with the inversion definition

As a consequence, the cardinality of $inv(c)$ is bounded by the number of combinations of 2 elements of the case base that are distinct, i.e. $n(n-1)/2$. Summing over the n possible values for c leads to the expected inequality $\Gamma \leq n^2(n-1)/2$.

Tightness of the Upper Bound The bound can be shown to be tight by exhibiting similarity measures σ_S and σ_R for which it is attained: in the case where all outcome values observed in D are distinct, which can for instance occur in the case of regression tasks, consider

- $\sigma_S(s_i, s_j) = 1$ for all pairs of situations s_i and s_j
- $\sigma_R(r_i, r_j) = e^{-f(i,j)}$ where f is a pairing function, i.e., is such that the values it assigns for the pairs (i, j) are all distinct. An example of such function is the Cantor pairing function $f(i, j) = \frac{1}{2}(i + j)(i + j + 1) + j$.

Then for any pair of distinct cases (c_i, c_j), the fact that all values assigned by σ_R are distinct entails that exactly one of the two possible ordered pairs belongs to $inv(c)$ for any c, the one with the minimal index $min(i, j)$. Indeed, considering without loss of generality that $i < j$, then it holds that $\sigma_S(s, s_i) = \sigma_S(s, s_j) = 1$ and $\sigma_R(r, r_i) < \sigma_R(r, r_j)$ due to the above definitions of σ_S and σ_R. The total number of inversions for a given c thus equals $\frac{n(n-1)}{2}$ and the upper bound is attained.

It can be observed that this bound remains a pessimistic one, insofar as these similarity measures are of course not relevant ones (indeed, considering all situations are fully similar does not constitute a relevant choice). However, it allows to set a scale and a reference value to compare the complexity values of relevant similarity measures.

3.2 Binary Classification Case

In the case of binary classification, by definition, the outcomes r can only take two values. The bound established in Theorem 1 can thus not be attained (except in the extreme case where D contains 2 instances, one of each class), and is not tight enough. This section establishes a tighter bound:

Theorem 2. *For a case base D in which all outcomes values observed in D can only take two distinct values, for two similarity measures σ_S and σ_R, the maximum value that the complexity $\Gamma(D, \sigma_S, \sigma_R)$ can take is $\Gamma_{cls} = \frac{n^3}{4}$, where $n = |D|$.*

There exists a pair (σ_S, σ_R) such that this bound can be attained if the two classes have the same number of elements.

Value of the Upper Bound Let n_0 and n_1 denote the number of cases in the case base associated with each of the two outcomes respectively.

For any triple (c, c_i, c_j), the inequality $\sigma_R(r, r_i) < \sigma_R(r, r_j)$ can hold only if $r \neq r_i$ and $r = r_j$ because of the reflexivity property of a similarity measure, as recalled in Section 2.1. As a consequence, the maximal number of such triples is $n_0 n_1$, which in turn is bounded by $\frac{n^2}{2}$. Indeed, $n_0 n_1 = n_0(n - n_0)$ which is a quadratic function of n_0 whose maximum is reached for $n_0 = \frac{n}{2}$ and equals $\frac{n^2}{4}$.

Summing over all n situations c leads to the expected result.

Tightness of the Upper Bound In the case where σ_S is the same as in the previous section, i.e. $\sigma_S(s, s') = 1$ for all pairs of situations and where $\sigma_R(r, r') = 1$ if $r = r'$ and 0 otherwise, the bound is attained if the two classes have the same number of elements, $n_0 = n_1 = \frac{n}{2}$.

Indeed, for a given case c, an inversion is observed for any ordered pair (c_i, c_j) where c_j has the same class as c and c_i the opposite class: $\sigma_S(c, c_i) = \sigma_S(c, c_j) = 1$ and $\sigma_R(r, r_i) = 0 < \sigma_S(r, r_j) = 1$. There are $n_0 n_1 = \frac{n^2}{4}$ such triples for a given c and thus $\frac{n^3}{4}$ inversions altogether.

Note that the order of magnitude of this bound is the same as the one in the general case, cubic in the number of cases, but it is halved (also note they are equal in the extreme case where $n = 2$). As previously, it is a pessimistic bound as it relies on non-relevant similarity measures.

4 Algorithmic Optimisation

This section studies the `CoAT` algorithm from an algorithmic point of view and shows that its computational complexity can be decreased drastically. It first presents the justification, and then describes the proposed optimized version of the `CoAT` algorithm.

4.1 Principle

The computational complexity of the `CoAT` algorithm, as recalled from [1] in Algorithm 1, is high, as it computes the complexity measure of the candidate modified data set D' for each candidate outcome value: the complexity is $O(n^3 \times |R|)$, i.e. not tractable for real data sets. Indeed, computing the complexity of a data set is a cubic function of its number of cases as it considers all triples (see the triple loop in Algorithm 2).

However, one can decrease the complexity by one order of magnitude when observing that actually not all triples are needed to identify the optimal r value minimizing $\Gamma(D', \sigma_S, \sigma_R)$: indeed all triples that do not involve the new situation are not needed, as they are common to all candidate modified data sets D'. Formally,

$$\begin{aligned}\Gamma(D \cup \{(s,r)\}, \sigma_S, \sigma_R) &= \sum_{i=1}^{n} |inv(c_i)| + |inv((s,r))| \\ &= \Gamma(D, \sigma_S, \sigma_R) + \\ &\quad \sum_{i=1}^{n} |\{c_j \in D | \sigma_S(s_i, s) \geq \sigma_S(s_i, s_j) \wedge \sigma_R(s_i, r) < \sigma_R(s_i, r_j)\}| \\ &\quad + |inv((s,r))|\end{aligned}$$

As the term $\Gamma(D, \sigma_S, \sigma_R)$ does not depend on the candidate outcome r, its computation is not needed: denoting

$$\Delta\Gamma(s, r, D, \sigma_S, \sigma_R) = \Gamma(D \cup \{(s,r)\}, \sigma_S, \sigma_R) - \Gamma(D, \sigma_S, \sigma_R)$$

Algorithm 3 Optimized variant of the `CoAT` algorithm

inputs: s (new situation), D (case base), R (set of candidate outcomes), σ_S, σ_R (situation and outcome similarity measures)
output: predicted outcome r_s for the new situation s
$min\Delta\Gamma = n^2(n-1)/2$ with $n = |D|$
for $r \in R$ **do**
  $\Delta\Gamma' = \text{computeDeltaGamma}(D, s, r, \sigma_S, \sigma_R)$ using Alg. 4
  if $\Delta\Gamma' < min\Delta\Gamma$ **then**
    $min\Delta\Gamma = \Delta\Gamma'$
    $r_s = r$
  end if
end for
return r_s

it holds that

$$\arg\min_{r \in R} \Gamma(D \cup \{(s,r)\}, \sigma_S, \sigma_R) = \arg\min_{r \in R} \Delta\Gamma(s, r, D, \sigma_S, \sigma_R)$$

Now the computation of $\Delta\Gamma(s, r, D, \sigma_S, \sigma_R)$ is quadratic in the number of cases contained in the considered case base D, as both its terms are. Indeed, the first term of the sum defining $\Delta\Gamma(s, r, D, \sigma_S, \sigma_R)$ is a sum over all n cases contained in D, which altogether has a quadratic complexity in n. The second term needs to go through all pairs of cases in D to compute $|inv((s,r))|$, which also has a quadratic complexity.

4.2 Proposed Optimized Algorithm

Algorithm 3 shows how the principle discussed in the previous subsection can be implemented. First the minimal value of the candidate $min\Delta\Gamma$ can be initialized to the upper bound established in Section 3. The algorithm then goes through all possible candidate outcome values $r \in R$, but each value is used to compute $\Delta\Gamma$ instead of Γ.

This computation is performed in Algorithm 4, which relies on a double loop, where the computation of Γ performed in Algorithm 2 relies on a triple loop. Indeed, it only considers the triples that contain the considered candidate case $c_0 = (s, r)$, testing all three possibilities, depending on whether c_0 is in first, second or third position in the triple.

This procedure can be further optimized if the best observed value for $min\Delta\Gamma$ is given as additional argument to the function computeDeltaGamma described in Algorithm 4. Indeed, it is then possible to take advantage of an early discard principle, i.e. to exit the double loop on c and c' as soon as the current $\Delta\Gamma$ value becomes greater than the already observed best value stored in this additional parameter $min\Delta\Gamma$. This principle can be especially useful when `CoAT` is applied to regression tasks, i.e. when the number of candidate outcome values is high.

Algorithm 4 Complexity increase computation : computeDeltaGamma

inputs: D (case base), s additional situation, r associated candidate outcome, σ_S, σ_R (situation and outcome similarity measures)
output: value of $\Delta\Gamma(s, r, \sigma_S, \sigma_R)$ with respect to D
$c_0 = (s, r)$
$\Delta\Gamma = 0$
for $c \in D$ **do**
  for $c' \in D$ **do**
    if $inv(c_0, c, c')$ **then**
      $\Delta\Gamma = \Delta\Gamma + 1$
    end if
    if $inv(c, c_0, c')$ **then**
      $\Delta\Gamma = \Delta\Gamma + 1$
    end if
    if $inv(c, c', c_0)$ **then**
      $\Delta\Gamma = \Delta\Gamma + 1$
    end if
  end for
end for
return $\Delta\Gamma$

A further direction for computational optimization, left for future works, is to define a relevant order, possibly based on a heuristic criterion, for testing the candidate outcome values: if the optimal one is tested first, all other candidates can be discarded early without fully computing their associated $\Delta\Gamma$ values.

5 Experimental Study

The experimental study of the `CoAT` algorithm and its optimization described in this section rely on their Python implementation available on github[3].

The experimental study consists in three parts: the first one, described in Section 5.1, illustrates the decrease of computational complexity offered by the algorithmic optimization proposed in Section 4; the second one (Sec. 5.2) shows there is a correlation between the data set complexity and the accuracy of the inferred label; the third one shows there is a correlation between the data set complexity and the classification task difficulty, as captured by the class overlapping degrees (Sec. 5.3).

5.1 Computational Cost

This section describes the experimental study of the computational cost of the `CoAT` algorithm and its optimization proposed in Section 4. Note that the latter is correct in the sense that its output is always identical to that of `CoAT`.

[3] https://github.com/fadibadra/coat_iccbr

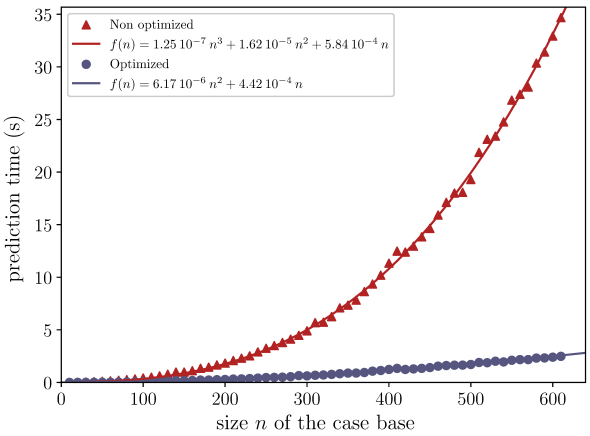

Fig. 1. `CoAT` prediction time for its non-optimized and optimized versions, according to the size of the case base.

Experimental Protocol The experiment is conducted on case bases of increasing sizes obtained as random subsamples of the UCI Balance Scale dataset[4] that defines a 3-class classification task. For each case base, the prediction time of both the non-optimized and the optimized versions of the `CoAT` algorithm is measured on 10 test instances. The similarity measure σ_S is defined as a decreasing function of the Euclidean distance, and the similarity measure σ_R is the class membership similarity: for two classes r_i and r_j, $\sigma_R(r_i, r_j) = 1$ if $r_i = r_j$, and 0 otherwise.

Results Figure 1 shows the average prediction time of the `CoAT` algorithm as a function of the size of the case base: the red triangles correspond to the non-optimized version of `CoAT`, the blue dots to the optimized version. The latter clearly show the decrease in the computational time.

A polynomial regression shows that the computing time of non-optimized version of `CoAT` on a case base of size n can be approximated by the function $f(n) = 1.25\,10^{-7}\,n^3 + 1.62\,10^{-5}\,n^2 + 5.84\,10^{-4}\,n$ (red line on the figure), with a root mean square error of 0.3; the computing time of optimized version of `CoAT` can be approximated by the function $f(n) = 6.17\,10^{-6}\,n^2 + 4.42\,10^{-4}\,n$ (blue line on the figure), with a root mean square error of 0.04. . These results are consistent with the theoretical analysis of their respective computational costs (see Section 4).

[4] https://archive.ics.uci.edu/ml/datasets/balance+scale

5.2 Correlation between Case Base Complexity and Performance

This experiment shows that there is a correlation between the value of the complexity measure and the performance of the `CoAT` prediction algorithm.

Experimental Protocol The experiment is conducted on 200 instances extracted from the Balance Scale data set. As the instances of these data sets are described only by d numeric features, each situation can be represented by a vector of $\mathbb{R}^d$. Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ be two such vectors. The performance of the `CoAT` algorithm is measured for each dataset by generating 100 different classification tasks $\{(D, \sigma_i, \sigma_R)\}_{1 \leq i \leq 100}$, each of which is obtained by keeping D and σ_R fixed, and choosing for σ_S a decreasing function of a randomly weighted Euclidean distance. More precisely, a set of random linear maps $\{L_i : \mathbb{R}^d \longrightarrow \mathbb{R}^d\}_{1 \leq i \leq 100}$ are generated, and for each map L_i, σ_i is defined as a decreasing function of the Euclidean distance computed in the L_i's embedding space:

$$\sigma_i(\mathbf{x}, \mathbf{y}) = e^{-d_i(\mathbf{x},\mathbf{y})} \text{ with } d_i(\mathbf{x}, \mathbf{y}) = \|L_i\mathbf{x} - L_i\mathbf{y}\|_2 = \sqrt{(\mathbf{x}-\mathbf{y})^T L_i^T L_i (\mathbf{x}-\mathbf{y})}$$

The performance is also measured on the task (D, σ_E, σ_R), in which $\sigma_E(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|_2}$ is a decreasing function of the Euclidean distance, which amounts to taking as linear map the identity matrix. The similarity measure σ_R is the class membership similarity. For each task, the performance is measured by the prediction accuracy, with 10-fold cross validation.

Results Fig. 2 shows for each classification task the average accuracy and standard deviation of the `CoAT` algorithm according to the dataset complexity. The blue points correspond to the randomly generated σ_i similarity measures. The red point gives the results for the σ_E similarity measure based on the standard Euclidean distance. The green line shows the result of a linear regression on the data. The Pearson's coefficient is -0.97. On these datasets, the results clearly show a correlation between the dataset complexity and the performance of the `CoAT` algorithm. The dataset complexity values range between 6.10% and 15.58% of the upper bound $\Gamma_{max} = 3,980,000$. The complexity upper bound thus provides a reference value, that can be used to define a quality scale for the similarity measure σ_S.

5.3 Correlation between Complexity and Task Difficulty

This experiment shows the correlation between the data set complexity and the classification task difficulty, as captured by the class overlapping degree.

Experimental Protocol The experiment is conducted on a set of $2D$ synthetic data sets, whose instances are equally split into two classes (blue and orange). A set of classification tasks $\{(D_i, \sigma_S, \sigma_R)\}_{1 \leq i \leq 500}$ is generated, in which both

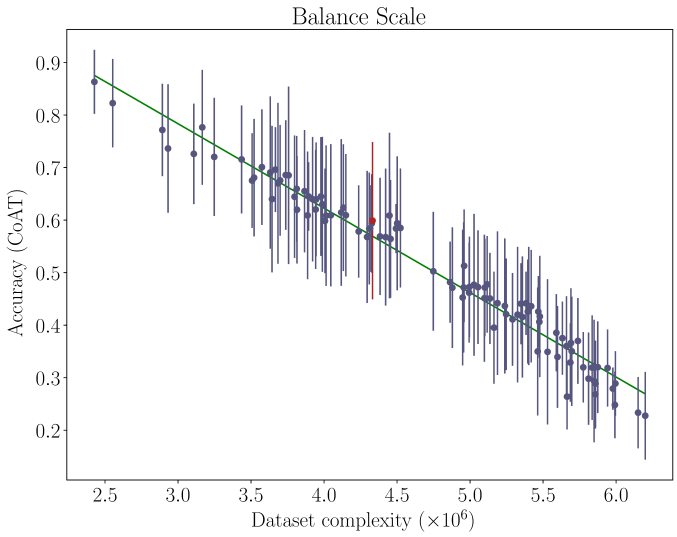

Fig. 2. Relation between `CoAT` performance (accuracy) and dataset complexity on the Balance Scale data set.

the size $|D_i| = 100$ of the case bases and the two similarity measures σ_S and σ_R are fixed, but the overlapping degree of the two classes vary. Random samples are drawn for each class from a multivariate normal distribution centered on the point $(-10, 0)$ for the blue class and $(10, 0)$ for the orange class, and the covariance matrix of each normal distribution for the data set D_i is the identity matrix multiplied by the scale factor $\texttt{scale} = i$. This parameter thus controls the overlapping degree of the two classes The similarity measure σ_S is a decreasing function of the Euclidean distance, and the similarity measure σ_R is the class membership similarity. The data set complexity $\Gamma(D_i, \sigma_S, \sigma_R)$ is computed for each classification task $(D_i, \sigma_S, \sigma_R)$.

Results Figure 3 shows the data distribution and ratios $\Gamma(D_i, \sigma_S, \sigma_R)/\Gamma_{max}$ obtained for different data sets D_i. When the value of the `scale` parameter is very small, the two classes are well separated, but as its value increases, the two classes start to overlap. When the two classes are well separated, no instance is more similar to an instance of a different class than it is to an instance of the same class, hence, $\Gamma = 0$. When the instances of the two classes get closer to each other, or overlap, some class similarities happen to be lower than some intra-class similarities, leading to a non-zero data set complexity Γ. Note however that the data set complexity Γ does not exactly measure the overlapping degree of the two classes, since it can be non-zero when the two class are close to one another, but non overlapping.

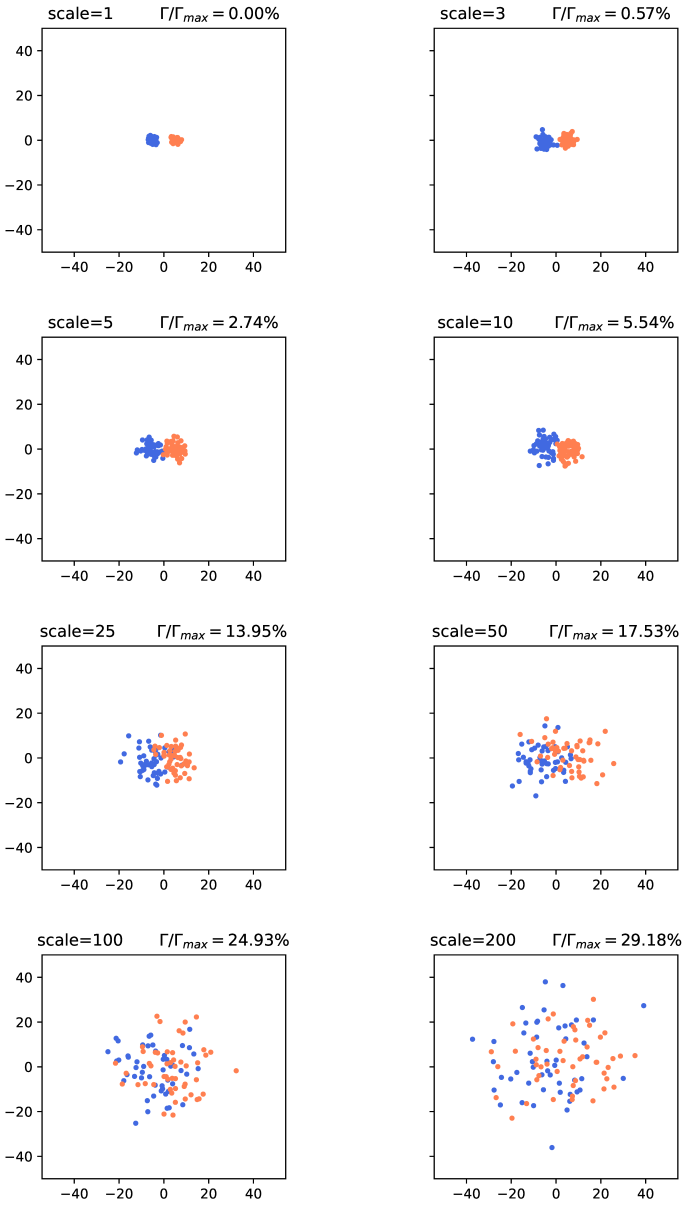

Fig. 3. Data distribution and ratio Γ/Γ_{max} for different synthetic data sets.

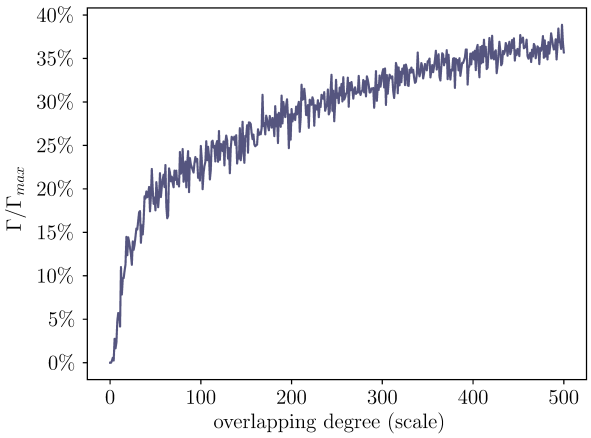

Fig. 4. Relation between data set complexity and classification task difficulty.

As shown on Figure 4, the more the two classes overlap, the higher the data set complexity. The ratio $\Gamma(D_i, \sigma_S, \sigma_R)/\Gamma_{max}$ thus provides a scale on which to estimate the difficulty of the task, as captured by the overlapping degree `scale` of the two classes on the data set D_i. This experiments also illustrates that Γ_{max} is not a tight bound: even for highly overlapping classes, the complexity value is still lower that 40% of Γ_{max}.

6 Conclusion and Future Works

This paper proposes an extended study of the `CoAT` algorithm that provides an ordinal implementation of the analogical transfer principle for classification and regression tasks. From a theoretical point of view, the paper establishes an upper bound of the complexity measure `CoAT` relies on. From an algorithmic point of view, an efficient optimization of the `CoAT` algorithm is proposed, that allows decreasing its computational cost by one order of magnitude: the optimized variant has a quadratic complexity, equivalent to that of any relational learning method. This property is illustrated experimentally. The paper also provides an experimental characterization of the complexity measure, regarding its relation to the inference performance and to the inference difficulty. When D and σ_R are fixed, the complexity measure is an intrinsic indicator of the quality of the similarity measure σ_S. When σ_S and σ_R are fixed, the complexity measure is an indicator of the inference difficulty, as captured by D's class overlapping degree.

The results obtained in this study provide additional arguments regarding the relevance of the ordinal approach to analogical transfer and motivations for further developing this principle of plausible inference. Future directions of research in particular include working on tighter bounds of the complexity measure, e.g.

focusing on more realistic (and thus less pessimistic) similarity measures and on the case of non binary classification. From an algorithmic point of view, future works will aim at proposing methods, possibly heuristics, to optimize the order in which the candidate outcomes are tested: as mentioned earlier, if the optimal value is considered first, other candidates can be discarded early, further decreasing the computational cost.

References

1. Badra, F.: A Dataset Complexity Measure for Analogical Transfer. In: IJCAI Int. Joint Conf. on Artificial Intelligence. pp. 1601–1607 (2020)
2. Badra, F., Sedki, K., Ugon, A.: On the Role of Similarity in Analogical Transfer. In: Cox, M.T., Funk, P., Begum, S. (eds.) Case-Based Reasoning Research and Development, vol. 11156, pp. 499–514. Springer International Publishing, Cham (2018)
3. David H Helman (ed.): Analogical Reasoning: Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy. Springer (2010)
4. Davies, T.R., Russell, S.J.: A logical approach to reasoning by analogy. In: IJCAI International Joint Conference on Artificial Intelligence (1987)
5. Dubois, D., Hüllermeier, E., Prade, H.: Flexible control of case-based prediction in the framework of possibility theory. In: EWCBR. pp. 61–73. Springer-Verlag (2000)
6. Hofstadter, D.R., Sander, E.: Surfaces and Essences: Analogy as the Fuel and Fire of Thinking. Basic Books, New York (2013)
7. Holyoak, K.J., Thagard, P.: The Analogical Mind. The American psychologist 52(1), 35–44 (1997)
8. Hug, N., Prade, H., Richard, G., Serrurier, M.: Analogical classifiers: A theoretical perspective. In: 22nd European Conference on Artificial Intelligence – ECAI. vol. 285, pp. 689–697 (2016)
9. Hüllermeier, E.: Possibilistic instance-based learning. Artificial Intelligence 148(1-2), 335–383 (2003)
10. Hüllermeier, E.: Credible case-based inference using similarity profiles. IEEE Transactions on Knowledge and Data Engineering 19(6), 847–858 (2007)
11. Lieber, J.: Application of the Revision Theory to Adaptation in Case-Based Reasoning: The Conservative Adaptation. In: Weber, R.O., Richter, M.M. (eds.) Case-Based Reasoning Research and Development, vol. 4626, pp. 239–253. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
12. Lieber, J., Nauer, E., Prade, H.: When Revision-Based Case Adaptation Meets Analogical Extrapolation. In: Sánchez-Ruiz, A.A., Floyd, M.W. (eds.) Case-Based Reasoning Research and Development. vol. 12877, pp. 156–170. Springer International Publishing, Cham (2021)
13. Miclet, L., Bayoudh, S., Delhay, A.: Analogical dissimilarity: Definition, algorithms and two experiments in machine learning. Journal of Artificial Intelligence Research 32, 793–824 (2008)
14. Ontañón, S., Plaza, E.: On knowledge transfer in case-based inference. In: Díaz-Agudo, B., Watson, I. (eds.) ICCBR Proceedings. pp. 312–326. Springer-Verlag Berlin Heidelberg (2012)
15. Prade, H.: Reasoning with Data - A New Challenge for AI? In: Schockaert, S., Senellart, P. (eds.) Scalable Uncertainty Management, vol. 9858, pp. 274–288. Springer International Publishing, Cham (2016)