Compatibility-based Similarity Learning for Case Base Prediction

Chunyang Fan[1,2][0009−0003−9921−6353], Fadi Badra[2][0000−0002−2437−8230], and Marie-Jeanne Lesot[1][0000−0002−3604−6647]

[1] Sorbonne UniversitÃĭ, CNRS, LIP6, F-75005 Paris, France
firstname.lastname@lip6.fr
[2] UniversitÃĭ Sorbonne Paris Nord, Sorbonne UniversitÃĭ, INSERM, Limics, 93000, Bobigny, France firstname.lastname@univ-paris13.fr

Abstract. The Analogical Transfer Principle, on which Case Base Prediction relies, states that similarity in certain components, typically data features, implies similarity in other components, typically data labels. It has been implemented as a compatibility constraint both for prediction tasks and, more recently, for similarity learning in the CCoSiL algorithm. This paper proposes an in-depth study of this approach, conducting a thorough experimental analysis to characterize its convergence and its algorithmic complexity. The paper then proposes a generalized version, named eXtended CCoSiL (XCoSiL), that disentangles two conceptual roles dually played by the data. XCoSiL allows to decrease significantly the computational cost, while maintaining or increasing the performance as compared to CCoSiL.

Keywords: Analogical Transfer · Similarity Measure Learning · Metric Learning · Case-Based Reasoning.

1 Introduction

Case Base Prediction (CBP, see e.g. [7, 2, 12] for surveys) solves machine learning tasks, e.g. classification or regression, leveraging the Analogical Transfer Principle (ATP), according to which similarity in certain components (typically, data features) implies similarity in others (typically, data labels). It is for instance applied by the k-NN algorithm. The choice of similarity measures is crucial for CBP, and it is thus connected the topic of Metric Learning (MeL) [4, 14]. Recent work [5] proposed to exploit ATP, implemented as a compatibility constraint [1], for Metric Learning, in the so-called Continuous Compatibility-based Similarity Learning method (CCoSiL): this joint framework of CBP and MeL achieves the objective of learning similarity measures tailored for specific CBP tasks.

This paper first conducts a thorough experimental analysis of CCoSiL properties, characterizing precisely its convergence, regarding the loss function, the accuracy when combined with CBP classifiers and the obtained similarity measure, its convergence speed and its algorithmic complexity. A precise interpretation of the dual role played by the training data/case base within CCoSiL then leads us

to the proposition of a generalized version, named eXtended Compatibility based Similarity Learning (XCoSiL), that disentangles two conceptual data natures. It allows to decrease significantly the computational cost while maintaining or even increasing the performance as compared to CCoSiL, as empirically shown in the conducted experiments.

Section 2 briefly reviews related works, Section 3 describes the experimental analysis of CCoSiL properties. Section 4 presents the proposed XCoSiL algorithm, experimentally studied in Section 5, before Section 6 concludes.

2 Context and Related Work

This section introduces the background of Metric Learning and reviews related work on Case Base Prediction, detailing the CCoSiL method upon which this paper contributions rely.

Metric Learning, MeL. MeL aims at learning appropriate metrics for machine learning tasks, e.g. classification, clustering or information retrieval. A large variety of techniques have been proposed [4, 14], among which one can mention e.g., linear, non-linear, local and histogram-based methods. Deep MeL employs neural networks to project data into embedding spaces, leading to approaches that can be categorized into pair-based methods using contrastive loss [8], triplet-based methods [9], which aim at bringing similar samples closer while pushing dissimilar samples apart, similarity cloud method [6] and clustering-based methods capturing global data structures [13]. Additional approaches e.g. integrate unsupervised hyperbolic methods for hierarchical data [16].

In the case-based reasoning domain, it has e.g. been proposed to learn weights to combine local measures in a data-driven approach [10]. The Large Margin Nearest Neighbor (LMNN) method [15] learns the positive semi-definite matrix that parametrizes the Mahalanobis distance to optimize k-NN performance.

Incompatibility-based CBP and Induced MeL. Among the large variety of Case Base Prediction methods (see e.g. [7, 2, 12] for surveys), this section focuses on incompatibility based ones, using the following notations: CB denotes a case base composed of pairs $c = (s, r) \in \mathcal{S} \times \mathcal{R}$, where $s \in \mathcal{S}$ is the situation (i.e., the input/features) and $r \in \mathcal{R}$ the corresponding outcome (i.e., the label/target). The effectiveness of the analogical transfer crucially depends on two similarity measures, $\sigma_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ and $\sigma_{\mathcal{R}} : \mathcal{R} \times \mathcal{R} \to \mathbb{R}^+$.

The Analogical Transfer Principle (ATP) basically states that if two cases have similar situations, according to $\sigma_{\mathcal{S}}$, they should have similar outcomes, according to $\sigma_{\mathcal{R}}$. The latter is often predefined, e.g. as a binary outcome equality check for classification tasks; in the following, we omit mentioning $\sigma_{\mathcal{R}}$ in the parameters. On the other hand, $\sigma_{\mathcal{S}}$ can be learned, in a metric learning framework.

The Compatibility-based Analogical Transfer algorithm (CoAT) [1] interprets ATP as an ordinal constraint: it considers that a case triplet (c_i, c_j, c_k) violates

ATP according to the considered $\sigma_{\mathcal{S}}$ and $\sigma_{\mathcal{R}}$ measures if situation s_i is more similar to s_k than to s_j but its outcome r_i is more similar to r_j than to r_k. This can be formalized as $\mathbb{1}\{\sigma_{\mathcal{S}}(s_i,s_k)-\sigma_{\mathcal{S}}(s_i,s_j))\mathbb{1}\{\sigma_{\mathcal{R}}(r_i,r_j)>\sigma_{\mathcal{R}}(r_i,r_k)\}$. For a new situation s, it predicts the outcome r that minimizes the discrete global *incompatibility function* that counts the number of such violating triplets in $\mathrm{CB}\cup\{(s,r)\}$. This paper studies the continuous extension of CoAT called CCoAT [5], that takes into account the violation extents, beyond their numbers:

$$\mathrm{Ext}(c_i,c_j,c_k;\sigma_{\mathcal{S}}) := \max(0,\sigma_{\mathcal{S}}(s_i,s_k)-\sigma_{\mathcal{S}}(s_i,s_j))\mathbb{1}\{\sigma_{\mathcal{R}}(r_i,r_j)>\sigma_{\mathcal{R}}(r_i,r_k)\} \quad (1)$$

For a given case triplet (c_i,c_j,c_k), Ext can be viewed as a local measure of the incompatibility between $\sigma_{\mathcal{S}}$ and $\sigma_{\mathcal{R}}$; reciprocally, for a given $\sigma_{\mathcal{S}}$, Ext can be viewed as the incompatibility between the three cases. Given a case (s,r) and a data set $\mathcal{D}$s, the average local incompatibility can be interpreted in the energy function framework [11], as shown in [3] and [5]:

$$E((s,r);\sigma_{\mathcal{S}},\mathcal{D}) := \frac{1}{|\mathcal{D}^{\otimes}(r)|}\sum_{(c_j,c_k)\in\mathcal{D}^{\otimes}(r)}\mathrm{Ext}((s,r),c_j,c_k;\sigma_{\mathcal{S}}) \quad (2)$$

where $\mathcal{D}^{\otimes}(r) := \{(c_j,c_k)\in\mathcal{D}^2 \mid r_j=r, r_k\neq r\}$ (taking into account all triplets using $\mathcal{D}\cup\{(s,r)\}$ has been shown to be much more expensive for a comparable performance [5]). $\mathcal{D}^{\otimes}(r)$ only contains so-called valid triplets (c_i,c_j,c_k) such that, denoting r the outcome associated to c_i, c_j also has outcome r whereas c_k has a different outcome [5]. E measures the violation extent averaged over such valid triplets formed by the case (s,r) and $\mathcal{D}$. As such, it can be viewed as a global incompatibility, measuring either the incompatibility of (s,r) with $\mathcal{D}$ for given $\sigma_{\mathcal{S}}$ and $\sigma_{\mathcal{R}}$ or the incompatibility of $\sigma_{\mathcal{S}}$ and $\sigma_{\mathcal{R}}$ at $\mathcal{D}$ level. Thus, it can be utilized for both outcome prediction, in the CBP framework, and for $\sigma_{\mathcal{S}}$ learning, in the MeL framework, resp. in the CCoAT and CCoSiL algorithms [5].

More precisely, in the CBP case, given $\mathcal{D}$ and $\sigma_{\mathcal{S}}$, for a new situation s, CCoAT predicts the outcome r that leads to the minimal energy E:

$$\hat{r} = \mathrm{CCoAT}(s;\sigma_{\mathcal{S}},\mathcal{D}) := \operatorname*{argmin}_{r\in\mathcal{R}} E((s,r);\sigma_{\mathcal{S}},\mathcal{D}) \quad (3)$$

In this CBP framework, $\mathcal{D}$ corresponds to the traditional case base, CB.

In the MeL case, for a given (s,r), the CCoSiL algorithm additionally considers the energy gap between the correct and the incorrect outcomes, in a traditional loss form [11]:

$$\ell\left((s,r);\sigma_{\mathcal{S}},\mathcal{D}\right) := \psi\left(E((s,r);\sigma_{\mathcal{S}},\mathcal{D}) - \min_{r'\in\mathcal{R}\setminus\{r\}} E((s,r');\sigma_{\mathcal{S}},\mathcal{D})\right) \quad (4)$$

where ψ is an increasing function, e.g. the Minimum Classification Error (MCE) loss $\psi_{\mathrm{M}}(x)=\frac{1}{1+e^{-x}}$ considered in the following. Then, given $\mathcal{D}$, CCoSiL learns $\sigma_{\mathcal{S}}$ by minimizing the total loss, averaged over all cases in $\mathcal{D}$:

$$L(\sigma_{\mathcal{S}};\mathcal{D}) := \frac{1}{|\mathcal{D}|}\sum_{(s_i,r_i)\in\mathcal{D}}\ell\left((s_i,r_i);\sigma_{\mathcal{S}},\mathcal{D}\right) \quad (5)$$

$$\hat{\sigma}_{\mathcal{S}} := \mathrm{CCoSiL}(\mathcal{D}) := \operatorname*{argmin}_{\sigma_{\mathcal{S}}} L(\sigma_{\mathcal{S}};\mathcal{D}) \quad (6)$$

Table 1. Dataset characteristics.

	Iris	User	Breast	PIMA	Wine	Seeds	Waveform (WF)
# Features	4	5	30	8	13	7	21
# Classes	3	4	2	2	3	3	3
# Points	150	403	569	768	178	210	5000

where $\sigma_{\mathcal{S}}$ is considered as parametrized by $\theta \in \Theta$ in practice, thus the minimum in Eq. (6) is searched over the parameter space Θ. In this framework, $\mathcal{D}$ corresponds to the training data of the metric learning task.

When CCoSiL is applied to optimize the performance of a CBP classifier, e.g. CCoAT, the latter can be written as $\text{CCoAT}(s; \text{CCoSiL}(\mathcal{D}), \mathcal{D})$, highlighting the fact that $\mathcal{D}$ simultaneously plays the role of case base for the CBP step and training set for the MeL step.

3 Experimental Analysis of CCoSiL

This section proposes an experimental analysis of CCoSiL properties: it shows that CCoSiL loss function converges as training progresses and is closely associated with improvement in classification performance. In addition, it provides evidence for the convergence speed and consistency of the learned similarity measure. Finally, the algorithmic complexity of CCoSiL is studied.

The experimental setup reproduces the one in [5], comparing CCoSiL with the reference LMNN [15]. The aim is to learn $\sigma_{\mathcal{S}}$, defined as the opposite of the Mahalanobis distance $\|L^\top(s_i - s_j)\|_2^2$, parameterized with $\theta = L$. Experiments are conducted using 10-fold cross-validation, i.e. setting $\text{CB} = \mathcal{D} = 90\%$ of the dataset, for the 7 UCI datasets whose characteristics are shown in Tab. 1.

3.1 Loss and Accuracy Convergence

Figure 1 shows the evolution of the loss value (blue curve), the accuracy on the training (green) and test data (red) along training epochs, with their standard deviations computed on the cross-validation folds. The first 7 graphs (in the upper part) show the results for CCoSiL, the lower part, commented in Sec. 5, shows them for the XCoSiL method proposed in Sec. 4, to ease the comparison.

As the blue curves in the upper part of Fig 1 show, the loss function of CCoSiL exhibits a clear downward trend across most datasets: it shows a continuous, smooth, and convex downward trend within 50 epochs, indicating that the convergence stabilizes. A detailed examination of the exceptions (Breast Cancer, PIMA and Waveform) reveals that the loss values decrease rapidly within 5 epochs, and present an oscillatory behavior later on.

The classification accuracy on the training set (green) exhibits a steady upward trend, and, for the test set (red), generally shows an upward trend, albeit with more volatility. On Breast Cancer, PIMA, Wine and Waveform, the accuracies on test sets show an even higher volatility and reach their peaks within

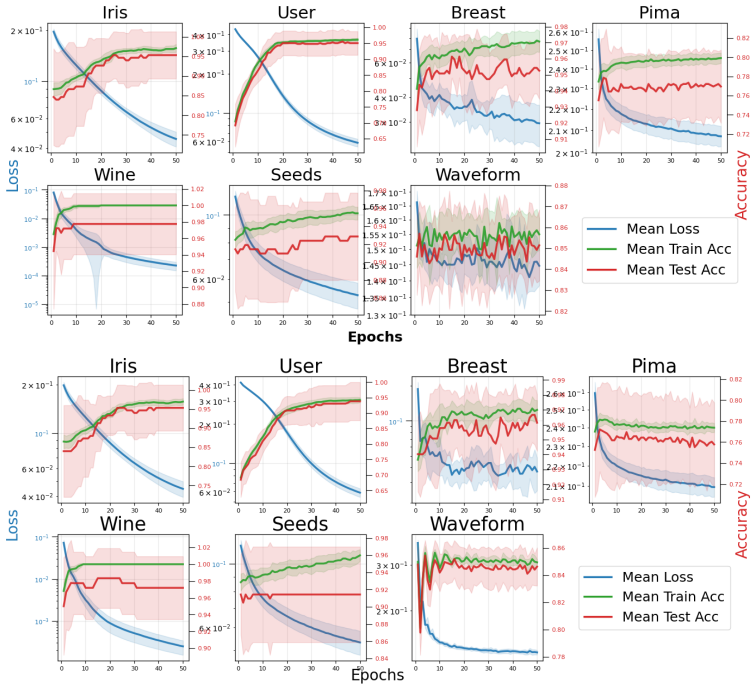


Fig. 1. Losses (blue) and accuracies on train (green) and test sets (red) along training epochs on UCI datasets using CCoSiL (upper) and XCoSiL (lower).

the first 5 epochs, which is consistent with the rapid decline of the loss values in the same period. In practice, one may use early stopping to terminate the training process at its peak. We leave this possibility as a direction for future work and, for the purpose of trend analysis, we keep training for 50 epochs.

3.2 Learned Metric Convergence

This section studies the learned similarity measure, examining its convergence and speed. Let $\hat{\theta}_{i,j}^{D,C}$ denote the matrix learned on dataset D using MeL configuration C at the j-th epoch of the i-th fold of the cross-validation.

Convergence Consistency. We first examine the consistency of the final matrices across different folds, testing whether $\hat{\theta}_{i,50}^{D,C} \approx \hat{\theta}_{i',50}^{D,C}$ holds for all cross validation folds i, i' between 1 and 10. The matrices $\hat{\theta}$ are vectorized and compared through their cosine similarity, averaged across all i, i':

$$\mathrm{Cos}\left(\hat{\theta}_{i,50}^{D,C}, \hat{\theta}_{i',50}^{D,C}\right) := \left\langle \mathrm{vec}\left(\hat{\theta}_{i,50}^{D,C}\right), \mathrm{vec}\left(\hat{\theta}_{i',50}^{D,C}\right) \right\rangle / \left(\left\| \mathrm{vec}\left(\hat{\theta}_{i,50}^{D,C}\right) \right\|_2 . \left\| \mathrm{vec}\left(\hat{\theta}_{i',50}^{D,C}\right) \right\|_2 \right)$$

This choice is motivated by its scale invariance and directional sensitivity. Indeed, the scale of the Mahalanobis distance does not influence the classification results

Table 2. Average cosine similarities between learned matrices over folds.

MeL	Iris	User	Breast	PIMA	Wine	Seeds	WF
LMNN	0.9818	0.9983	0.9316	0.9978	0.9310	0.9662	0.9994
CCoSiL	0.9889	0.9912	0.8978	0.9766	0.9859	0.9835	0.9280
XCoSiL	0.9882	0.9325	0.9227	0.9704	0.9856	0.9862	0.9939

(i.e., matrices θ and $c \cdot \theta$ for $c > 0$ yield identical results). Furthermore, being bounded, its numerical values are easy to interpret and compare.

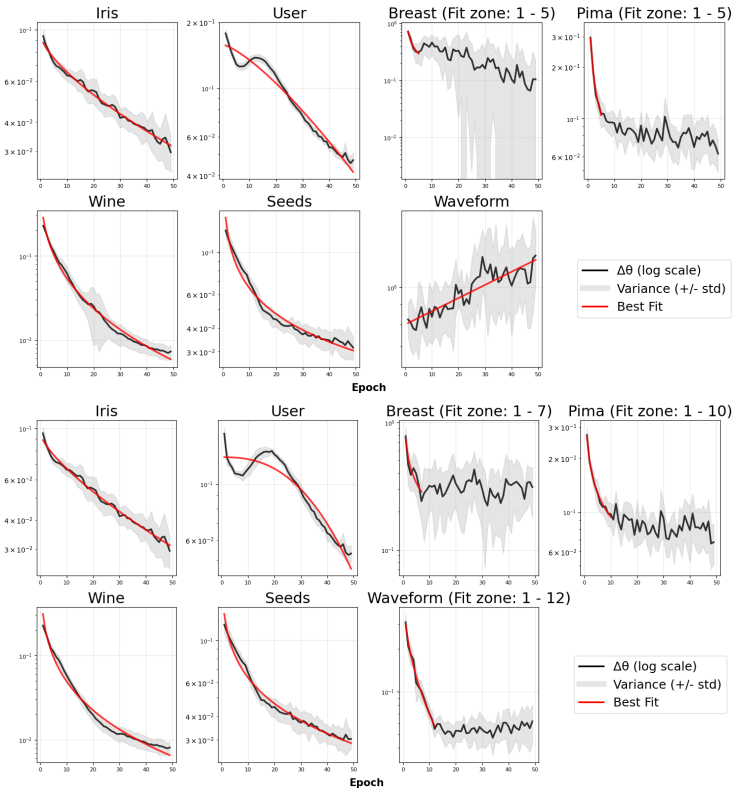
The results presented in Table 2 show that the learned parameters $\hat{\theta}$ by CCoSiL across different folds exhibit a high consistency on all datasets except for Breast Cancer: the average cosine similarity almost always exceeds 0.95 and many values approach 1, in addition they are always higher than the ones obtained by LMNN. Breast Cancer is the only dataset with value (slightly) lower than 0.9. This is most likely because CCoSiL's rapid convergence causes $\hat{\theta}$ to reach an optimal plateau within the first few epochs, as discussed above. Therefore, after more than 40 epochs of fluctuations, $\hat{\theta}$ varies across folds.

Convergence Speed To investigate the convergence speed of the MeL process in CCoSiL, we utilize the Frobenius norm $\|\cdot\|_F$ to calculate, in each fold, the variation of the θ matrix from one epoch to the following one. Specifically, for all epochs j in $\{2, \dots 50\}$, we calculate $\Delta\hat{\theta}^{D,C}_{i,j} := \left\|\hat{\theta}^{D,C}_{i,j} - \hat{\theta}^{D,C}_{i,j-1}\right\|_F$. Here, we use the Frobenius norm over cosine similarity because, for space $\mathbb{R}^{d\times d}$ where $\hat{\theta}$ resides, it forms a complete normed vector space: $\Delta\hat{\theta}^{D,C}_{i,j}$ converges to 0 if and only if the sequence $\hat{\theta}^{D,C}_{i,j}$ for all j converges. This implies that observing the evolution of $\Delta\hat{\theta}^{D,C}_{i,j}$ with respect to j allows to analyze the MeL process convergence.

The black curves in the upper left part of Fig. 2 show that for 6 out of 7 datasets, $\Delta\hat{\theta}^{D,C}_{i,j}$ decreases as j increases, which indicates that $\hat{\theta}$ tends to converge as training progresses. For Waveform, $\Delta\hat{\theta}^{D,C}_{i,j}$ augments with high volatility. Considering the fact that the accuracy on the corresponding test set reaches its peak after the second epoch, we presume that the MeL process has already reached a local optimal plateau at this point, while the Adam optimizer continues to update $\hat{\theta}$ with high volatility attempting to escape the plateau.

In order to characterize the convergence speed more precisely, we fit the curves using the Bayesian Information Criterion (BIC) to select the most appropriate function type among 5 possibilities: (i) Sub-polynomial: $f(j) = -\alpha \log(j) + \beta$; (ii) Polynomial: $f(j) = j^{-\alpha} + \beta$; (iii) Intermediate: $f(j) = -\alpha e^{-\gamma j} + \beta$, with $\gamma \in (0, 1)$; (iv) Exponential: $f(j) = -\alpha e^{j} + \beta$; (v) Super-exponential: $f(j) = -\alpha e^{\gamma j} + \beta$, with $\gamma > 1$.

The best fit is shown as red curve in Fig. 2 with its parameters and R^2 value in the right part of the figure. For all datasets except Breast Cancer and PIMA, the R^2 values are close to 1, indicating a good fit. For the two exceptions, it is lower than 0.9, which can be explained by the previous analysis according to



Data set	Type
Iris	Inter 0.9898
User	SupExp 0.9659
Breast	Poly 0.9677
PIMA	Poly 0.9922
Wine	SupExp 0.9910
Seeds	SupExp 0.9732
WF	↑ Exp 0.7849

Data set	Type
Iris	Inter 0.9912
User	SupExp 0.9158
Breast	Poly 0.8557
PIMA	Poly 0.9928
Wine	SupExp 0.9805
Seeds	Inter 0.9822
WF	SupExp 0.9880

Fig. 2. Similarity measure convergence speed for CCoSiL (7 upper graphs) and XCoSiL (7 lower graphs): (black) Frobenius distance between successive parameter values, (red) best fit whose parameters are in the table on the right.

which the effective learning period occurs within the first 5 epochs. The results thus report the best fit on the first 5 epochs. For Waveform, the previous analysis showed an even shorter effective learning period, limited to the first 2 epochs, the later ones bringing little loss decrease. They appear to increase the differences between the successive parameter values, leading to a divergent behavior in terms of similarity metric, advocating for the integration of an early stopping criterion, as already mentioned. The right part of Fig. 2 shows that, on most datasets, the learning process exhibits a rapid convergence rate, typically higher than polynomial convergence, or even super-exponential convergence. This suggests that CCoSiL possesses favorable convergence performance in MeL.

Combined with the analysis of the loss function and accuracies, this experimental analysis allows to conclude that the CCoSiL loss function effectively guides the MeL process to enhance classification performance. In some case, it can suffer from local optimal plateaus, but generally the parameter matrix converges to a stable state with a favorable convergence rate.

3.3 Algorithmic Complexity

Computational complexity. CCoSiL has a cubic computation complexity, as the computation amount of Ext per epoch is $O(n^3)$, denoting $n = |\mathcal{D}|$. Indeed, computing $L(\sigma_{\mathcal{S}}; \mathcal{D})$ (Eq. (5)) requires a double summation over triples (c_i, c_j, c_k): it iterates over $c_i \in \mathcal{D}$ and, for each of them, $\ell(c_i; \sigma_{\mathcal{S}}, \mathcal{D})$ iterates over $(c_j, c_k) \in \mathcal{D}^{\otimes}(r_i)$ due to E expression (Eq. (2)). For a classification task with K classes (i.e. $|\mathcal{R}| = K$) and $\overline{n} = n/K$ average number of cases per class, $|\mathcal{D}^{\otimes}(r_i)| = \overline{n} \times (K-1)\overline{n} = n^2(K-1)/K^2 = O(n^2)$. Therefore, the total number of computations of Ext is $O(n^3)$.

Spatial complexity. Due to the large volume of triple calculations, modern machine learning frameworks (e.g., pytorch) typically employ parallel computing to accelerate the training process. However, the parallel computing in CCoSiL also entails a high spatial complexity: calculating $E((s_i, r_i); \sigma_{\mathcal{S}}, \mathcal{D})$ requires storing the Ext values for all $(c_j, c_k) \in \mathcal{D}^{\otimes}(r_i)$, which is $O(n^2)$. Therefore, training simultaneously b data in a batch leads to a $O(b \cdot n^2)$ overall spatial complexity.

To avoid memory overflow caused by quadratic spatial complexity, the CCoSiL code adopts a compromise: the calculation of $E((s_i, r); \sigma_{\mathcal{S}}, \mathcal{D})$ is parallelized, but for each batch, the code forces the computation of the energy corresponding to only one (s_i, r) at a time. While this reduces the spatial complexity, it also sacrifices a portion of the overall computational efficiency.

4 Proposition of XCoSiL

This section proposes a generalized version of CCoSiL, named XCoSiL for eXtended Compatibility based Similarity Learning. It is based on a thorough interpretation of the roles played by the cases involved in Ext (Eq. 1). Distinguishing between 2 roles allows to separate the cases into 2 data sets and consequently to decrease the computational cost of the approach, as discussed below.

4.1 Proposed Loss Function

In the local incompatibility function Ext defined in Eq. (1), the first case, c_i, plays a different role from the two other ones, c_j and c_k, that have a symmetrical status (indeed, $\sigma_{\mathcal{S}}(s_i, s_j) - \sigma_{\mathcal{S}}(s_i, s_k) = -(\sigma_{\mathcal{S}}(s_i, s_k) - \sigma_{\mathcal{S}}(s_i, s_j))$, the same applies to $\sigma_{\mathcal{R}}$): c_i serves as a subject to be compared with c_j and c_k used as references to evaluate the extent of the ATP violation. This difference can also be observed in the definition of the energy function E (see Eq. 2) that highlights the fact that (s, r) may be taken from a set different from $\mathcal{D}$.

Therefore, we propose to modify the definition of the loss function defined in Eq. (5), to average on cases (s, r) from a set $\mathcal{D}_1$ that may differ from the set $\mathcal{D}_2$ used to compute the local incompatibilities in function Ext, used in the energy function E (Eq. (2)) on which ℓ (Eq. (4)) depends:

$$L(\sigma_{\mathcal{S}}; \mathcal{D}_1, \mathcal{D}_2) := \frac{1}{|\mathcal{D}_1|} \sum_{(s_i, r_i) \in \mathcal{D}_1} \ell\left((s_i, r_i); \sigma_{\mathcal{S}}, \mathcal{D}_2\right) \tag{7}$$

We keep the notation L, changing the number of parameters it takes. Minimizing $L(\sigma_{\mathcal{S}}; \mathcal{D}_1, \mathcal{D}_2)$ allows to find the similarity measure $\sigma_{\mathcal{S}}$ that maximizes the average energy gap between the correct and the incorrect outcomes for cases in $\mathcal{D}_1$, relative to the reference cases in $\mathcal{D}_2$.

The constraints regarding the choices of $\mathcal{D}_1$ and $\mathcal{D}_2$ and their required relationships raises interesting questions. Following the above discussion, $\mathcal{D}_2$, that corresponds to reference cases, needs to contain high quality data that may for instance be selected by an expert, allowing for a knowledge-driven component in the MeL task. Obviously, an inappropriate choice for $\mathcal{D}_2$, e.g. containing noisy data, would prevent from learning a meaningful similarity measure. On the other hand, $\mathcal{D}_1$ needs to be representative of the data that will be processed after the learning step has been completed, featuring a data-driven component. It may be the case that they satisfy an inclusion relation ($\mathcal{D}_2 \subset \mathcal{D}_1$) or that they are distinct ($\mathcal{D}_2 \cap \mathcal{D}_1 = \emptyset$). Still, they both need to be related to the real distribution of the unknown cases, but it is not required that $\mathcal{D}_2$ contains i.i.d. data.

This distinction aligns with the common understanding of training set in machine learning and case base in CBP: in machine learning, the training set ($\mathcal{D}_1$) typically requires i.i.d. condition to reflect the real distribution of the unknown cases; on the other hand, in CBP, the case base CB can be expert-selected as reference in prediction inference, which is a natural source of $\mathcal{D}_2$ when combining XCoSiL with CBP. Hence, the two datasets, respectively taking the roles of training set and reference set, can make XCoSiL a both data-driven and knowledge-driven learning framework, naturally adapted to optimize the performance of CBP classifiers.

4.2 Complexity Analysis

The most direct benefit of distinguishing between $\mathcal{D}_1$ and $\mathcal{D}_2$ is the enhanced flexibility in controlling the complexity. Indeed, denoting $n_1 = |\mathcal{D}_1|$ and $n_2 = |\mathcal{D}_2|$, the computational complexity of $L(\sigma_{\mathcal{S}}; \mathcal{D}_1, \mathcal{D}_2)$ is $O(n_1 \cdot n_2^2)$, corresponding to the number of computations of $\mathrm{Ext}(c_i, c_j, c_k)$ per epoch, as the reference cases (c_j, c_k) iterate over the $O(n_2^2)$-sized set $\mathcal{D}_2^{\otimes}(r)$. Now, in the expected case where $n_2 \ll n_1$, this quantity is much smaller than the $O(|\mathcal{D}|^3)$ complexity of CCoSiL.

Furthermore, for a single case c, the computational complexity of calculating $E(c; \sigma_{\mathcal{S}}, \mathcal{D}_2)$ is $O(n_2^2)$, i.e. it only depends on the size of the reference set. When $n_2 \ll n_1$, this low spatial complexity enables XCoSiL to further parallelize the computation of more energy functions, e.g. simultaneously calculating them for multiple cases within the same batch, thereby enhancing overall training efficiency.

5 Experimental Analysis of XCoSiL

We conduct experiments to study empirically the algorithmic complexity of XCoSiL as compared to CCoSiL, to compare the relevance of the similarity measures, in terms of the accuracy it leads to when combined with CBP classifiers and to examine its performance in terms of convergence and stability.

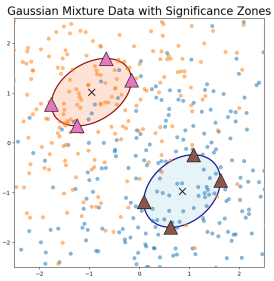


Fig. 3. Illustration of the D_{S} built for a two dimension data set, that contains the 8 points represented as pink and brown triangles.

5.1 Experimental Protocol

Given a training set denoted $\mathcal{T}$ and situations s taken from a test set, we compare the 6 following configurations: defining CB, $\mathcal{D}, \mathcal{D}_1$ and $\mathcal{D}_2$ from $\mathcal{T}$ as discussed and justified below, for a a prediction algorithm CBP $\in \{k\text{-NN}, \mathrm{CCoAT}\}$ and a metric learning algorithm MeL $\in \{\mathrm{LMNN}, \mathrm{CCoSiL}\}$, we consider

- CBP(s, MeL($\mathcal{D}$), CB), that leads to 4 combinations (see below)
- CBP(s, XCoSiL ($\mathcal{D}_1, \mathcal{D}_2$), CB)
- CBP(s, Id, CB), that corresponds to the reference configuration based on the Euclidean distance (the MeL step denoted Init in the result table Tab. 3).

The experiments are conducted on the UCI benchmark data used in Sec. 3, split into training and test sets in a classical 10-fold cross-validation setting.

We first discuss the definition of $\mathcal{D}_2$, interpreted as a high-quality reference set (see Sec. 4.1). In order to simulate expert knowledge, we propose to build a reference set D_{S} containing representative points as follows: we assume that $\mathcal{T}$ can be considered as drawn from a Gaussian mixture distribution, whose parameters can be estimated from $\mathcal{T}$. For a chosen significance level $\alpha \in (0, 1)$, D_{S} is defined as containing, for each class r, the boundary points of the projection of $100 \cdot \alpha\%$ central region of this distribution onto each dimension. They can be computed analytically (formula omitted for space constraints) and are illustrated, for a toy data set in Fig. 3. Denoting K and d the number of classes and of features resp., $|D_S| = 2 \cdot K \cdot d$. For the considered 7 data sets, see Tab. 1, it ranges from 24 to 126, whereas $|\mathcal{T}|$ ranges from 135 to 4500, for Iris and Waveform resp. We then set $\mathcal{D}_2 = D_S$ and $\mathcal{D}_1 = \mathcal{T}$ (note that, with probability 1, $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$).

As discussed in Sec. 4.1, based on the interpretation of their respective roles, CB is expected to be equal to $\mathcal{D}_2$, leading to set CB $= D_{\mathrm{S}}$. However, in order to test the generality of the proposed learning process, we additionally examine the results obtained when setting CB $= \mathcal{T}$.

Finally, regarding $\mathcal{D}$, used by the MeL competitors LMNN and CCoSiL, we examine 3 configurations: D_{S}, $\mathcal{T}$ and $D_{\mathrm{S}} \cup \mathcal{T}$. The latter ensures fairness, insofar as it provides the competitors with the exact same information as XCoSiL. We do not consider setting $\mathcal{D}$ to D_{S}, as the latter does not contain enough data.

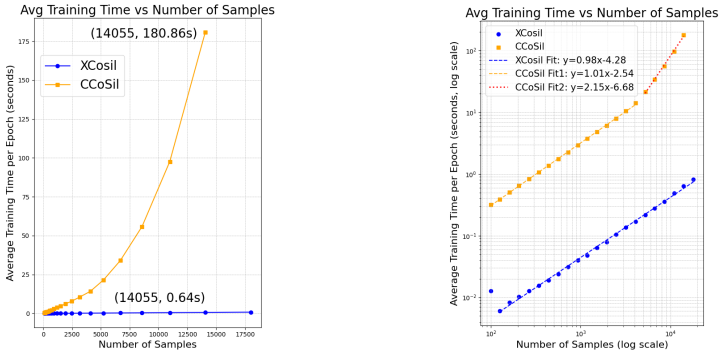


Fig. 4. Training time of CCoSiL and XCoSiL on normal scale (left) and log-log scale (right) for synthetic Gaussian datasets.

5.2 Empirical Complexity Analysis

In order to compare empirically the computational time of CCoSiL and XCoSiL, we consider synthetic data, generated as five classes with Gaussian distribution with 10 features, varying the number of points $\mathcal{T}$ contains, along a geometric sequence with 22 points from 100 to 18000. The results are shown in Fig. 4.

For XCoSiL, excluding the fluctuation at $n = 100$, the slope of fitting line is approximately 0.98. This observation is consistent with the previously analysis: since $n_2 = |D_S| = 2Kd$ is a constant independent of n, with $n_1 = n$, the overall computational complexity of XCoSiL $O(n_1 n_2^2) = O(n)$.

For CCoSiL, the largest configuration with 18000 points cannot be processed due to the high spatial complexity (see Sec. 3.3). The fitting lines is decomposed into two parts, with slopes approximately 1.01 (resp. 2.15) at smaller (resp. larger) scales and further increases, because the CCoSiL implementation leverages parallel computing for individual energy function evaluations to mitigate time cost. Once the data size exceeds a certain threshold, such that the single-pass capacity of the parallel processing units, the time complexity gradually converges toward the theoretical computational complexity of $O(n^3)$.

5.3 Accuracy Analysis

The experimental results of classification accuracy are summarized in Tab. 3. As detailed in the following, they show that XCoSiL effectively learns similarity metrics that are well-adapted to the characteristics of D_S and that its competitive accuracy with CCoAT(D_S) stems from its hybrid learning mode.

Considering the overall best accuracy achieved by each MeL method across all classifiers and datasets, we first observe that XCoSiL($\mathcal{T}, D_S$) with CCoAT(D_S) achieves a competitive average accuracy of 91.20%, which is only 0.15 ppt. lower than the best combination, CCoSiL($D_S \cup \mathcal{T}$) with CCoAT($\mathcal{T}$), with the advantage of a highly reduced computational time, both at the MeL and the CBP levels.

Table 3. Classification accuracy (%) of various combinations of CBP and MeL algorithms for various input data sets, under 10-fold cross-validation. The best results in each block (resp. each colum) are in **bold** (resp. **bold and underlined**).

Classifier	MeL	Iris	User	Breast	PIMA	Wine	Seeds	WF	Avg.
k-NN (D_S)	Init	**95.33**	72.67	74.86	72.79	96.05	88.57	67.22	81.07
	LMNN($\mathcal{T}$)	94.67	93.03	**78.20**	72.39	97.19	**93.33**	67.76	85.22
	LMNN($D_S \cup \mathcal{T}$)	94.67	93.03	**78.20**	72.39	97.19	**93.33**	67.90	**85.24**
	CCoSiL($\mathcal{T}$)	94.67	96.02	53.10	**75.26**	97.22	91.90	75.90	83.44
	CCoSiL($D_S \cup \mathcal{T}$)	94.67	**96.52**	55.03	**75.26**	97.22	92.86	76.54	84.01
	XCoSiL($\mathcal{T}, D_S$)	94.67	95.28	52.21	73.18	**97.78**	91.90	**76.78**	83.11
CCoAT (D_S)	Init	85.33	72.18	67.83	74.86	92.19	91.90	63.52	78.26
	LMNN($\mathcal{T}$)	94.67	91.27	74.33	74.48	**99.44**	92.38	63.54	84.30
	LMNN($D_S \cup \mathcal{T}$)	94.67	91.27	74.33	74.48	**99.44**	92.38	63.54	84.30
	CCoSiL($\mathcal{T}$)	**96.00**	**96.51**	92.10	75.78	97.19	**93.33**	74.48	89.34
	CCoSiL($D_S \cup \mathcal{T}$)	**96.00**	**96.51**	93.15	75.78	98.33	92.86	73.40	89.43
	XCoSiL($\mathcal{T}, D_S$)	**96.00**	96.02	**95.08**	**77.47**	96.63	91.90	**85.28**	**91.20**
k-NN ($\mathcal{T}$)	Init	94.00	83.12	**97.01**	73.83	95.56	91.90	79.82	87.89
	LMNN($\mathcal{T}$)	**95.33**	92.55	96.48	74.10	**99.44**	92.86	79.94	90.10
	LMNN($D_S \cup \mathcal{T}$)	**95.33**	93.04	96.31	**74.23**	98.33	**94.29**	79.96	90.21
	CCoSiL($\mathcal{T}$)	93.33	95.29	96.14	71.10	97.78	92.38	**83.74**	89.97
	CCoSiL($D_S \cup \mathcal{T}$)	94.00	**95.54**	95.61	72.28	98.33	92.86	**83.74**	**90.34**
	XCoSiL($\mathcal{T}, D_S$)	93.33	95.28	95.96	71.61	96.63	91.90	82.64	89.62
CCoAT ($\mathcal{T}$)	Init	84.67	62.24	89.80	75.13	90.52	92.38	75.26	81.43
	LMNN($\mathcal{T}$)	94.00	87.06	92.62	74.48	**99.44**	91.43	75.94	87.85
	LMNN($D_S \cup \mathcal{T}$)	94.00	87.07	92.79	74.21	97.22	92.38	75.90	87.65
	CCoSiL($\mathcal{T}$)	95.33	95.03	95.25	76.95	97.75	**92.86**	**85.14**	91.19
	CCoSiL($D_S \cup \mathcal{T}$)	**96.00**	**95.52**	95.26	**77.21**	98.33	92.38	84.70	**91.34**
	XCoSiL($\mathcal{T}, D_S$)	95.33	92.29	**95.60**	75.13	97.75	**92.86**	84.62	90.51

More precisely, the former outperforms the latter on 3 datasets, is outperformed on 3 datasets and tied on 1, showing they can be considered as equivalent. Both configurations outperform LMNN with either k-NN or CCoAT.

Impact of D_S We examine the impact of D_S, to study whether the relevance of XCoSiL is due to these reference data. At MeL level, it can be observed that using $D_S \cup \mathcal{T}$ generally has a very limited impact, for LMNN as for CCoSiL, as compared to $\mathcal{T}$, whatever CBP algorithm is used. This indicates that, as expected, the D_S addition does not as such provide significantly useful information for MeL. Indeed, in the considered experimental setting, D_S is derived from $\mathcal{T}$.

At classification level, taking D_S as CB generally leads to a lower accuracy for both k-NN and CCoAT across all tested MeL methods, except XCoSiL. This can be explained by the fact that the built D_S can be too extreme to be used as CB: it may not contain enough points to represent the data complexity and the basic Gaussian mixture assumption may not hold. This is especially noticeable for k-NN that appears more affected by the choice of CB than CCoAT.

Table 4. Comparison of the similarity metrics learned by XCoSiL and CCoSiL.

Cosine Similarity	Iris	User	Breast	PIMA	Wine	Seeds	WF
XCoSiL($\mathcal{T}, D_S$) vs. CCoSiL($\mathcal{T}$)	0.999	0.877	0.956	0.974	0.996	0.998	0.745
XCoSiL($D_S \cup \mathcal{T}, D_S$) vs. CCoSiL($D_S \cup \mathcal{T}$)	0.993	0.870	0.955	0.970	0.986	0.973	0.748

Impact of the classifier We study the impact of the CBP classifier used after XCoSiL, to examine whether XCoSiL is general or dedicated to a specific classifier, as compared to the other MeL algorithms. In the less favorable configuration where CB = D_S, k-NN with XCoSiL achieves an average accuracy lower than CCoSiL and much lower than LMNN. In contrast, CCoAT with XCoSiL achieves the highest average accuracy among all tested MeL methods. Furthermore, XCoSiL with CCoAT achieves a significant improvement compared to other best configurations: it outperforms CCoSiL($D_S \cup \mathcal{T}$) with CCoAT by 1.76 ppt. and LMNN($D_S \cup \mathcal{T}$) with k-NN by 5.95 ppt. This suggests that XCoSiL can effectively leverage the selected reference set D_S to learn metrics that are particularly well-suited for prediction using D_S.

In the favorable configuration where CB = $\mathcal{T}$, for k-NN, XCoSiL achieves a slightly lower average accuracy than CCoSiL and LMNN; for CCoAT, it achieves an average accuracy slightly lower than CCoSiL but higher than LMNN. Overall, with CCoAT it still achieves competitive accuracy compared to LMNN with k-NN and slightly lower than CCoSiL with CCoAT. This indicates that in addition to a significant reduction in algorithmic complexity, XCoSiL can learn an effective similarity metrics comparable to those learned by traditional methods.

5.4 Convergence Analysis

Tab. 4 shows the Cosine Similarity of the metrics learned by XCoSiL and CCoSiL: they are very close one to another, with values greater than 0.9, except for User and Waveform, showing that they globally agree on the learned metrics.

In addition, we conduct the same analysis of loss evolution and convergence described in Sec. 3 and shown on the 7 lower graphs of Fig. 1 and 2. XCoSiL shows a very similar pattern to that of CCoSiL, except for the Waveform dataset, where it exhibits a smoother loss decrease and accuracy increase. XCoSiL also demonstrates a convergence speed very similar to that of CCoSiL.

6 Conclusion

This paper studied compatibility-based metric learning, conducting an experimental analysis of the CCoSiL method and proposing a generalized approach XCoSiL: while CCoSiL exhibits desirable convergence properties, it overloads the datasets with different conceptual roles. The dual source principle of XCoSiL disentangles them, distinguishing between a training role, to reflect the data distribution, and a reference role, to provide representative cases. The conducted experiments demonstrate that, as compared to LMNN and CCoSiL, XCoSiL is

able to learn competitive similarity measures even with a small reference set, that allows to decrease significantly the computational cost: by combining the quantity advantage of a large training set and the quality advantage of a selected reference set, XCoSiL allows the integration of expert knowledge to enable efficient learning with lower computational cost.

This approach raises the crucial question of the reference set D_S definition: applying XCoSiL to real CBP tasks may allow for building D_S integrating expert knowledge. The proposition of strategies to take into account several expressions of expert knowledge, as well as the development of data driven strategies when no expert is available, constitute research directions to further improve compatibility based metric learning that leverage the Analogical Transfer Principle.

Acknowledgements Funded by the SMeLT project, ANR22-CE23-0032-03.

References

1. Badra, F.: A Dataset Complexity Measure for Analogical Transfer. In: IJCAI'20 (2020)
2. Badra, F., Lesot, M.J.: Case-based prediction âĂŞ A survey. IJAR (2023)
3. Badra, F., Lesot, M.J., Marquer, E., Couceiro, M.: Some Perspectives on Similarity Learning for Case-Based Reasoning and Analogical Transfer. In: IARML@IJCAI'23 (2023)
4. Bellet, A., Habrard, A., Sebban, M.: Metric Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, Springer (2015)
5. Fan, C., Badra, F., Lesot, M.J.: Case-Based Prediction Using a Continuous Compatibility Measure. In: ICAART'26 (2026)
6. Gabel, T., Godehardt, E.: Top-Down Induction of Similarity Measures Using Similarity Clouds. In: ICCBR'15 (2015)
7. Gilboa, I., Schmeidler, D.: Case based Predictions: Introduction. In: Maskin, E. (ed.) Introduction to Case-Based Prediction. World Scientific Publishers (2010)
8. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality Reduction by Learning an Invariant Mapping. In: CVPR'06 (2006)
9. Hoffer, E., Ailon, N.: Deep Metric Learning Using Triplet Network. In: Similarity-Based Pattern Recognition (2015)
10. Jaiswal, A., Bach, K.: A Data-Driven Approach for Determining Weights in Global Similarity Functions. In: ICCBR'19 (2019)
11. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.J.: A Tutorial on Energy-Based Learning. MIT press (2006)
12. Prade, H., Richard, G.: Analogical proportion-based induction: from classification to creativity. Journal of Applied Logics (2024)
13. Song, H.O., Xiang, Y., Jegelka, S., Savarese, S.: Deep Metric Learning via Lifted Structured Feature Embedding. CVPR'16 (2016)
14. Suárez, J.L., García, S., Herrera, F.: A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges. Neurocomputing (2021)
15. Weinberger, K.Q., Blitzer, J., Saul, L.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. In: NIPS'05 (2005)
16. Yan, J., Luo, L., Deng, C., Huang, H.: Unsupervised Hyperbolic Metric Learning. In: CVPR'21 (2021)