Noname manuscript No.
(will be inserted by the editor)

Sparse conformal prediction for dissimilarity data

Frank-Michael Schleif · Xibin Zhu · Barbara Hammer

the date of receipt and acceptance should be inserted later

Abstract Existing classification algorithms focus on vectorial data given in Euclidean space or representations by means of positive semi-definite kernel matrices. Many real world data, like biological sequences are not vectorial, often non-euclidean and given only in the form of (dis-)similarities between examples, requesting for efficient and interpretable models. Vectorial embeddings or transformations to get a valid kernel are limited and current dissimilarity classifiers often lead to dense complex models which are hard to interpret by domain experts. They also fail to provide additional information about the confidence of the classification. In this paper we propose a prototype-based conformal classifier for dissimilarity data. It is based on a prototype dissimilarity learner and extended by the conformal prediction methodology. It (i) can deal with dissimilarity data characterized by an arbitrary symmetric dissimilarity matrix, (ii) offers intuitive classification in terms of sparse prototypical class representatives, (iii) leads to state-of-the-art classification results supported by a confidence measure and (iv) the model complexity is automatically adjusted. In experiments on dissimilarity data we investigate the effectiveness with respect to accuracy and model complexity in comparison to different state of the art classifiers.

1 Introduction

Similarity and dissimilarity based learning, or simply learning from proximities, constitutes a field of active research [6], since more and more data sets are naturally dealt with in terms of domain dependent similarities or dissimilarities. Examples include edit distance based measures for strings or images [14] or popular similarity measures in bioinformatics such as scores obtained by the Smith-Waterman, FASTA[1], or blast algorithm [13].

School of Computer Science, University of Birmingham, Birmingham B15 2TT UK,
CITEC centre of excellence, Bielefeld University, 33615 Bielefeld, Germany
E-mail: schleify@cs.bham.ac.uk,{xzhu|bhammer}@techfak.uni-bielefeld.de

[1] FASTA is an abbreviation of *fast all*. The predecessor of FASTA was FASTP which was only applicable to protein sequences

Classifiers based on dissimilarity data assign a class label to a given example based on the pairwise dissimilarities only without the need to consider an explicit vectorial embedding of data. Formally, data are characterized by a dissimilarity matrix D obtained from a set of objects where $d(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{R}$ constitutes a non-negative measure of the dissimilarity between the two objects.

A popular way to analyze dissimilarity data is to consider the related similarity matrix S which can be derived from D as a matrix of inner-products in some Hilbert space. This process is known as double centering and explained in more detail in Section 3. If S is obtained from a valid inner-product, S is a positive semi-definite matrix (psd) and can be considered as a kernel matrix. This can be processed by kernel-classifiers like the Support Vector Machine (SVM) [47]. If S is psd, it may be considered to be generated by a positive semi-definite inner product function $\kappa(\mathbf{v}_i, \mathbf{v}_j)$, fulfilling the Mercer conditions which can be expanded by means of its eigenvalues and eigenfunctions:

$$\kappa(\mathbf{v}_i, \mathbf{v}_j) = \sum_{i}^{\infty} \lambda_i \phi_i(\mathbf{v}_i)\phi_i(\mathbf{v}_j) = \langle \phi(\mathbf{v}_i), \phi(\mathbf{v}_j) \rangle \tag{1}$$

The values $\kappa(\mathbf{v}_i, \mathbf{v}_j)$ define the kernel matrix K $\forall \mathbf{v}_i, \mathbf{v}_j$. If S does not constitute a valid kernel, additional transformations are necessary to guarantee semi positive definiteness [6]. For a more detailed discussion about kernels and kernel classifiers see [43].

As detailed in the following the majority of the available methods to analyze proximity data has a lot of limitations. The available approaches are often quite complex with cubic complexity and are in general black box concepts with limited or no additional information about the model decisions. Prototype based methods offer interesting alternatives to obtain interpretable models [16] and have been recently extended to proximity learning [15,39]. Prototypes are representative points summarizing larger parts of the data sets. This can be done unsupervised, where the prototypes are often cluster centers or supervised, where the prototypes cover classes or sub-classes and may model larger parts of the data across individual clusters. Prototypes are interesting to get compact representations of a large set of points. Most often the prototypes are the main parameters of a prototype based model. For classification problems the decision functions, obtained by prototypes, are often comparably simple and can be communicated easily with domain experts. The analysis problem gets more accessible and domain knowledge can be integrated more easily [16]. The challenge is to identify the prototypes and to define models with good generalization capabilities using this rather sparse representation. If the prototypes are points of the original data, in contrast to for example linear combinations thereof, they are also called exemplars. Here we focus on classification tasks to solve two open problems common in the context of proximity learning. The first one is the model complexity and the second one is the reliability of the classification decision. Using concepts from inductive conformal prediction [49,48] we will address both issues for a family of prototype based learning algorithms.

The proposed method is based on a recent prototype based classifier for dissimilarity data [17]. This method extends popular prototype classifier techniques to gen-

eral relational data. It works directly on the dissimilarity matrix and it arrives at a prototype-based classifier which represents data by a fixed number of prototypes. Thereby, it replaces an explicit distance measure by an implicit form which depends on the given dissimilarity matrix only. While a very effective model with good classification accuracy and small model complexity arises, the sparseness of the resulting model, i.e. the number of prototypes, is treated as a meta-parameter and has to be chosen using cross-validation. Further the classification is done in a nearest prototype approach which does not directly provide additional information about the reliability of the classification decision.

In this contribution a relational prototype learner is proposed extended by conformal prediction concepts, referred to as Conformal Relational Prototype Classifier (CRPC). CRPC can be *directly* applied to dissimilarity data, providing sparse interpretable models where each classification is supported by a measure of confidence. In addition, the confidence is used for a dynamic adaptation of the model complexity during learning, growing the model complexity as required by the resulting conformal regions.

First we will give some background and related work around proximity learning and conformal predictors. Here we will also highlight alternative strategies to process the considered data by available methods and the current limitations. Subsequently we shortly revisit the basic relational prototype based classifier, and introduce the concept of conformal prediction in this context, afterwards. In the main part we derive a specific formulation of a relational prototype based learning algorithm coupled with concepts from conformal prediction addressing the two mentioned challenges. The suitability of the technique to arrive at sparse prototype-based models for dissimilarity data with an automatic adaptation of the model complexity is demonstrated using benchmark data from bioinformatics, afterwards.

2 Related work

Inspired by the work in [29], some dedicated classification methods which can directly deal with dissimilarity data have been proposed. In [9] a feature based dissimilarity space classification is proposed which makes use of the dissimilarity space strategy combined with different classifiers. It was found that the dissimilarity representation is in average more effective than traditional feature representations [29]. For new test data however all dissimilarities to the training points have to be calculated which can be prohibitive for large data sets. In [30] a density-based classifier is proposed which, again, is based on a dissimilarity space approach and requires the determination of a prototype set. Various prototype selection methods are discussed in [33] but the approaches are not in closed form or applicable for two class problems, only; additionally, results are quite limited. A good (sparse) representation of dissimilarity data is still an issue [31]. Another dissimilarity-classifier, employing Monte-Carlo simulation techniques, was proposed in [25]. This approach is, again, quite complex for the multi-class case. In [23] different techniques are compared, focusing on the determination/reduction of prototypes for dissimilarity learning. The strategies discussed in [23] are in parts heuristic focusing on a direct optimization

of the classification error in the training set and are not based on a cost function in a closed form. Especially they are not formulated as margin optimizers but in parts in an unsupervised manner optimizing a cluster model, widely unrelated to a supervised classification problem. The best results, obtained in [23] are competitive to the classical LVQ which is known to be sub-optimal compared to the generalized LVQ model (GLVQ). Our approach is based on [38] which also provides strong generalization bounds [3,18]. In the approach presented here the model complexity is adapted based on the conformal prediction approach.

Unlike [23] we will provide a strategy to obtain reference vectors of the dissimilarity matrix in a natural way, employing a cost function based margin optimizer and conformal prediction. It is a common requirement for classifiers to provide not only good generalization of the prediction on unseen data, but also to define a measure about the safety of this classification. In the field of proximity learning only very few methods provide such measures for example by a probabilistic classification [40,34]. For kernel classifiers the Probabilistic Classification Vector Machine [5] or the relevance vector machine [45], provide probabilistic outputs but are not directly applicable for dissimilarity data and also do not scale for large classification problems. These methods make some assumptions in the modeling step, for example regarding underlying data distributions which may lead to potentially biased probability estimates. Conformal prediction see [42] is an alternative to obtain measures of confidence and credibility regarding a model prediction and provides calibrated p-values. Conformal prediction is a very effective theoretical framework used for different problems in classification and regression [2,28] but also more recently in a wider context like feature selection [51] and kernel learning [1]. We will employ and detail this approach in the following to overcome some problems with classical prototype based relational learning. While the original conformal prediction is very costly for larger data sets, a more recent alternative called inductive conformal prediction [27,26,48] is also applicable for larger data sets which we will focus on.

3 Preliminaries about dissimilarity data

Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects defined in some data space, with $|\mathbb{V}| = N$ and $\mathbf{v}_j^T$ the transposed vector. We assume, there exists a dissimilarity measure such that $D \in \mathbb{R}^{N\times N}$ is a dissimilarity matrix measuring the pairwise dissimilarities $D_{ij} = d(\mathbf{v}_i, \mathbf{v}_j)$ between all pairs $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{V}$. Any reasonable (possibly non-metric) distance measure is sufficient. We assume zero diagonal $d(\mathbf{v}_i, \mathbf{v}_i) = 0$ for all i and symmetry $d(\mathbf{v}_i, \mathbf{v}_j) = d(\mathbf{v}_j, \mathbf{v}_i)$ for all i, j.

For every symmetric dissimilarity matrix D, an associated similarity matrix S is induced by a process referred to as double centering with cost $\mathcal{O}(N^2)$: $S = -JDJ/2$ where $J = (I - \mathbf{1}\mathbf{1}^T/N)$ with identity matrix I and vector of ones $\mathbf{1}$. D is Euclidean if and only if S is positive semi-definite (psd). Many classification techniques have been proposed to deal with such psd kernel matrices implicitly such as the support vector machine. In this case, preprocessing is required to guarantee psd. In [6] different strategies (such as clipping, flipping, shift, vector-representation) are discussed to obtain valid kernels from (dis-)similarity matrices. The idea is to change the eigen-

value decomposition of the similarity matrix S such that negative eigenvalues are avoided. More details can be found in the following.

Assuming we have a symmetric similarity matrix S, it has an eigenvalue decomposition $S = U^T \Lambda U$, with orthogonal matrix U and diagonal matrix Λ collecting the eigenvalues. In general, p eigenvectors of S have positive eigenvalues and q have negative eigenvalues; $(p, q, N-p-q)$ is referred to as the signature. If there are negative eigenvalues, i.e. $q \neq 0$, the vector space is called pseudo-Euclidean space [21].

Definition 1 (Pseudo-Euclidean space ([21])) A pseudo-Euclidean space $\xi = \mathbb{R}^{(q,p)}$ is a real vector space equipped with a non-degenerate, indefinite inner product $\langle ., . \rangle_\xi$. ξ admits a direct orthogonal decomposition $\xi = \xi_+ \oplus \xi_-$ where $\xi_+ = \mathbb{R}^p$ and $\xi_- = \mathbb{R}^q$ and the inner product is positive definite on ξ_+ and negative definite on ξ_-. The space ξ is therefore characterized by the signature (p, q).

The *clip*-operation sets all negative eigenvalues to zero, the *flip*-operation takes the absolute values, the *shift*-operation increases all eigenvalues by the absolute value of the minimal eigenvalue. The corrected matrix S^* is obtained as $S^* = U^T \Lambda^* U$, with Λ^* as the modified eigenvalue matrix using one of the above operations. The obtained matrix S^* can now be considered as a valid kernel matrix K. The cost of such transformation is $\mathcal{O}(N^3)$. As an alternative, data points can be treated as vectors where coefficients or variables are given by the pairwise similarities. These vectors can be processed using standard kernels. In [6] an extensive comparison of these preprocessing methods in connection to SVM is performed for a variety of benchmarks. Interestingly, some operations such as shift do not affect the location of global optima of important cost functions such as the quantization error [22], albeit the transformation can severely affect other performance measures of different optimization algorithms [15].

A further alternative is to embed the (dis-)similarity matrix into the so called dissimilarity space by taking all dissimilarities of a point $\mathbf{v}_i$ to all other points or a subset only, resulting in an at most N-dimensional vector representation of $\mathbf{v}_i$ [9]. However, this view is changing the original data representation and leads to a finite data space, limited by the number of samples. The analysis in [32] indicates that for non-Euclidean dissimilarities corrections like above should be avoided. A schematic view of the relations between S and D and its transformations is shown in Figure 1.

Alternatively, techniques have been introduced which directly deal with possibly non-psd similarity matrix S. Given a symmetric dissimilarity matrix D with zero diagonal, an embedding of D in a pseudo-Euclidean vector space determined by the eigenvector decomposition of S is always possible (see [29]). A symmetric bilinear form in this space is given by $\langle \mathbf{x}, \mathbf{y} \rangle_{p,q} = \mathbf{x}^T I_{p,q} \mathbf{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. Taking the eigenvectors of S together with the square root of the absolute value of the eigenvalues, we obtain vectors $\mathbf{v}_i$ in pseudo-Euclidean space such that $d_{ij}(\mathbf{v}_i, \mathbf{v}_j) = \langle \mathbf{v}_i - \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j \rangle_{p,q}$ holds for every pair of data points. If the number of data is not limited, a generalization of this concept to Krein spaces with according decomposition is possible [29].

Vector operations can be directly transferred to pseudo-Euclidean space, i.e. we can deal with prototypes as linear combinations of data in this space. Hence we can use prototype-based learning explicitly in pseudo-Euclidean space since it relies on

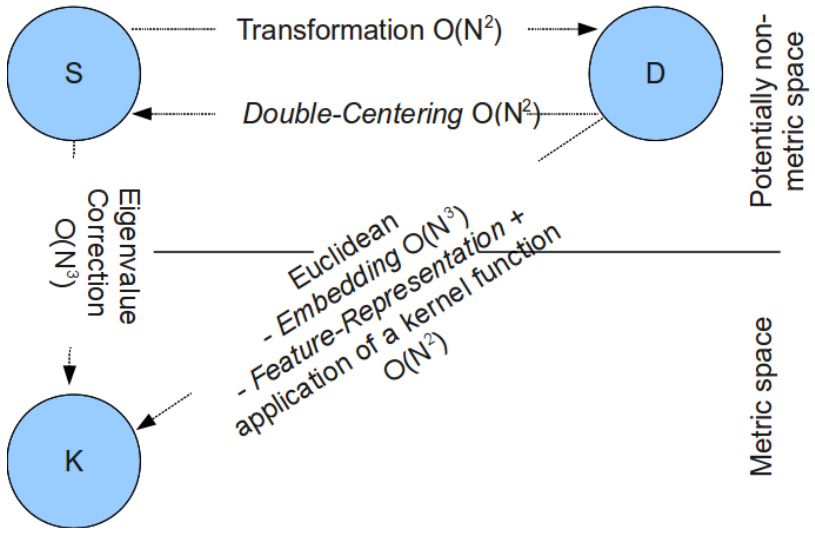


Fig. 1: Schema to illustrate the relation between similarities and dissimilarities.

vector operations only. One problem of this explicit transfer is given by the computational complexity of the embedding which is $\mathcal{O}(N^3)$, and further, by the fact that out-of-sample extensions to new data points characterized by pairwise dissimilarities are not immediate. Because of this fact, we are interested in efficient techniques which implicitly refer to this embedding only. As a side product, such algorithms are invariant to coordinate transforms in pseudo-Euclidean space.

4 Relational prototype based learning

We assume a training set is given where data point $\mathbf{v}_j$ is labeled $\mathbf{l}_j \in \mathbb{L}, |\mathbb{L}| = L$. The objective is to learn a classifier f such that $f(\mathbf{v}_k) = \mathbf{l}_k$ for any given data point. Thereby, $\mathbf{v}_k$ is represented implicitly by a vector of known dissimilarities with respect to $W \subseteq \mathbb{V}$. Subsequently, we review a recently published prototype classifier for dissimilarity data [17] which we use as the basic method in the following.

Classification takes place by means of k prototypes $\mathbf{w}_j$ in the pseudo-Euclidean space, which are priorly labeled. Typically, a winner takes all rule is assumed, i.e. a data point is mapped to the label assigned to the prototype which is closest to the data in pseudo-Euclidean space, taking the bilinear form in pseudo-Euclidean space to compute the distance. For relational data classification, the key assumption is to restrict prototype positions to linear combinations of data points of the form

$$\mathbf{w}_j = \sum_i \gamma_{ji}\mathbf{v}_i \text{ with } \sum_i \gamma_{ji} = 1 \quad \mathbf{w}_j \in W. \tag{2}$$

Then dissimilarities can be computed implicitly by means of

$$d(\mathbf{v}_i, \mathbf{w}_j) = [D \cdot \gamma_j]_i - \frac{1}{2} \cdot \gamma_j^T D \gamma_j \tag{3}$$

where $\gamma_j = (\gamma_{j1}, \ldots, \gamma_{jn})$ refers to the vector of coefficients describing the prototype $\mathbf{w}_j$ implicitly and can be randomly initialized, as shown in [15].

Using this observation, prototype classifier schemes which are based on cost functions can be transferred to the relational setting. We use the cost function defined in [38]. Alternatively a labeled Gaussian mixture model [41] could be used as shown in [17], which, however is very sensitive to hyper-parameters. The corresponding cost function of the relational prototype classifier (RPC) becomes:

$$E_{\mathrm{RPC}} = \sum_i \Phi \left(\frac{[D\gamma^+]_i - \frac{1}{2} \cdot (\gamma^+)^T D \gamma^+ - [D\gamma^-]_i + \frac{1}{2} \cdot (\gamma^-)^T D \gamma^-}{[D\gamma^+]_i - \frac{1}{2} \cdot (\gamma^+)^T D \gamma^+ + [D\gamma^-]_i - \frac{1}{2} \cdot (\gamma^-)^T D \gamma^-} \right), \tag{4}$$

where the closest correct and wrong prototypes are referred to, $\mathbf{w}^+$ and $\mathbf{w}^-$, respectively, corresponding to the coefficients γ^+ and $\gamma-$, respectively and $\Phi(x) = (1 + \exp(-x))^{-1}$. A simple stochastic gradient descent leads to adaptation rules for the coefficients γ^+ and γ^- in RPC: component k of these vectors is adapted as

$$\Delta\gamma_k^+ \sim -\Phi'(\mu(\mathbf{v}_i)) \cdot \mu^+(\mathbf{v}_i) \cdot \frac{\partial \left([D\gamma^+]_i - \frac{1}{2} \cdot (\gamma^+)^T D \gamma^+ \right)}{\partial \gamma_k^+} \tag{5}$$

$$\Delta\gamma_k^- \sim \Phi'(\mu(\mathbf{v}_i)) \cdot \mu^-(\mathbf{v}_i) \cdot \frac{\partial \left([D\gamma^-]_i - \frac{1}{2} \cdot (\gamma^-)^T D \gamma^- \right)}{\partial \gamma_k^-} \tag{6}$$

with

$$\mu(\mathbf{v}_i) = \frac{d(\mathbf{v}_i, \mathbf{w}^+) - d(\mathbf{v}_i, \mathbf{w}^-)}{d(\mathbf{v}_i, \mathbf{w}^+) + d(\mathbf{v}_i, \mathbf{w}^-)} \tag{7}$$

$$\mu^+(\mathbf{v}_i) = \frac{2 \cdot d(\mathbf{v}_i, \mathbf{w}^-)}{(d(\mathbf{v}_i, \mathbf{w}^+) + d(\mathbf{v}_i, \mathbf{w}^-))^2} \tag{8}$$

$$\mu^-(\mathbf{v}_i) = \frac{2 \cdot d(\mathbf{v}_i, \mathbf{w}^+)}{(d(\mathbf{v}_i, \mathbf{w}^+) + d(\mathbf{v}_i, \mathbf{w}^-))^2} \tag{9}$$

The partial derivative yields

$$\frac{\partial \left([D\gamma_j]_i - \frac{1}{2} \cdot \gamma_j^T D \gamma_j \right)}{\partial \gamma_{jk}} = d_{ik} - \sum_l d_{lk} \gamma_{jl} \tag{10}$$

Naturally, alternative gradient techniques can be used. After every adaptation step, normalization takes place to guarantee $\sum_i \gamma_{ji} = 1$. This way, a learning algorithm which adapts prototypes in a supervised manner is given for general dissimilarity data, whereby prototypes are implicitly embedded in pseudo-Euclidean space.

Initially the prototypes are indirectly modeled as random vectors corresponding to random values γ_{ij} which sum to one. It is possible to take class information into account by setting all γ_{ij} to zero which do not correspond to the class of the prototype.

Out-of-sample extension of the classification to new data is possible based on the following observation [15]: for a novel data point $\mathbf{v}$ characterized by its pairwise

dissimilarities $D(\mathbf{v})$ to the data used for training, the dissimilarity of $\mathbf{v}$ to a prototype modeled by γ_j can be calculated by

$$d(\mathbf{v}, \mathbf{w}_j) = D(\mathbf{v})^T \cdot \gamma_j - \frac{1}{2} \cdot \gamma_j^T D \gamma_j \tag{11}$$

Then by finding the closest/most similar prototype based on the distances/dissimilarities to all prototypes calculated by (11) the new data point will be classified by the label of the closest prototype.

5 Conformal prediction

RPC is very effective as shown in [17] but has two major limitations. RPC is a crisp classifier, where the classification function f predicts only the class label but no additional information about the confidence of the prediction is given. Especially in the life science some kind of reliability measure, similar to p- or q-values from statistics would be beneficial. Only few attempts exist to give reliability estimates for these methods (see [7,44]). A second drawback is that the complexity of the model in terms of the number of prototypes needs to be specified a priori.

In this approach, we propose to use conformal prediction to enhance classification results with a level of confidence, and to automatically grow a model which has suitable model complexity. Reliability, sometimes also referred as confidence, has been the subject of a theory called conformal prediction as introduced in [36,49]. See [42] for a recent tutorial on the topic. Conformal prediction aims at the determination of confidence and credibility of classifier decisions. Thereby, the technique can be accompanied by a formal stability analysis as provided in [49]. In the context of vectorial data, sparse conformal predictors have been recently discussed in [19].

5.1 Conformal prediction for RPC

We follow the general approach of conformal prediction as reviewed in [49,42]. Denote the labeled training data $\mathbf{z}_i = (\mathbf{v}_i, \mathbf{l}_i) \in \mathbb{Z} = \mathbb{V} \times \mathbb{L}$. Furthermore let $\mathbf{v}_{N+1}$ be a new data point with unknown label . The *conformal prediction* computes for given data $(\mathbf{z}_i)_{i=1,\ldots,N}$, an observed data point $\mathbf{v}_{N+1}$, and a chosen error rate ϵ an $(1-\epsilon)$-*prediction region* $\Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{v}_{N+1}) \subseteq \mathbb{L}$ consisting of a number of possible label assignments. The method ensures that if the data $\mathbf{z}_i$ are exchangeable [2] then

$$P(\mathbf{l}_{N+1} \notin \Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{v}_{N+1})) \leq \epsilon \tag{12}$$

holds asymptotically for $N \to \infty$ for each distribution of $\mathbb{Z}$. One says that the predictor is *asymptotically valid*. It is important to mention, that the probability is unconditional, such that if we repeat the process of drawing samples $\mathbf{v}_{N+1}$ and generating Γ^ϵ a number of k times we will find with respect to statistical fluctuations that in less than $\epsilon \times k$ cases the real label $\mathbf{l}_{N+1}$ is not under the predicted labels of Γ^ϵ.

[2] *Exchangeability* is a weaker condition than data being i.i.d. [49] which is also interesting for the online setting.

Algorithm 1 Conformal Prediction (CP)

function CP($\mathcal{D}$, $\mathbf{v}_{N+1}$, ϵ)
  for all $l \in \mathbb{L}$ **do**
    $\mathbf{z}_{N+1} := (\mathbf{v}_{N+1}, l)$
    for $i = 1, \ldots, N+1$ **do**
      $\mathcal{D}_i := \{\mathbf{z}_1, \ldots, \mathbf{z}_{N+1}\} \backslash \{\mathbf{z}_i\}$
      $\alpha_i^{\mathbf{l}} := A(\mathcal{D}_i, \mathbf{z}_i)$ ▷ non conformity of $\mathbf{z}_i$ against $\mathcal{D}_i$, using eq. 13
    end for
    $p_{N+1}^{\mathbf{l}} := \frac{|\{i=1,\ldots,N+1 \mid \alpha_i^{\mathbf{l}} \geq \alpha_{N+1}^{\mathbf{l}}\}|}{N+1}$
  end for
  return $\Gamma^\epsilon := \{\mathbf{l} : p_{N+1}^{\mathbf{l}} > \epsilon\}$
end function

Algorithm 2 Inductive Conformal Prediction (ICP)

function ICP($\mathcal{D}$, $\mathbf{v}_{N+1}$, ϵ)
  $\mathcal{D}_{tr} \cup \mathcal{D}_{cal} := \mathcal{D}$ ▷ split $\mathcal{D}$ into proper training set $\mathcal{D}_{tr}$ and calibration set $\mathcal{D}_{cal}$
  $W :=$ the model trained using $\mathcal{D}_{tr}$ ▷ train the model using $\mathcal{D}_{tr}$
  for all $\mathbf{z}_i \in \mathcal{D}_{cal}, i = 1, \ldots, |\mathcal{D}_{cal}|$ **do**
    $\alpha_i := A(W, \mathbf{z}_i)$ ▷ non conformity of the calibration set : using W eq. 13
  end for
  for all $l \in \mathbb{L}$ **do**
    $\mathbf{z}_{N+1} := (\mathbf{v}_{N+1}, l)$
    $\alpha_{N+1}^l := A(W, \mathbf{z}_i)$ ▷ non conformity of $\mathbf{z}_{N+1}$: using W eq. 13
    $p_{N+1}^l := \frac{|\{i=1,\ldots,n \mid \alpha_i \geq \alpha_{N+1}^l\}|}{N+1}$ ▷ p-value w.r.t label l using the non conformity of $\mathcal{D}_{cal}$
  end for
  return $\Gamma^\epsilon := \{\mathbf{l} : p_{N+1}^l > \epsilon\}$
end function

5.1.1 Computation of the prediction region

To compute the conformal prediction region, a non conformity measure is fixed $A(D, \mathbf{z})$. It is used to calculate a non conformity value α that estimates how an observation $\mathbf{z}$ fits to given representative data D=$\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. The original conformal algorithm for classification is as follows: given a nonconformity measure A, significance level ϵ, examples $\mathbf{z}_1, \ldots, \mathbf{z}_N$, for an new object $\mathbf{v}_{N+1}$ and a possible label $\mathbf{l}$, it is decided whether $\mathbf{l}$ is contained in $\Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z}_N, \mathbf{v}_{N+1})$: see algorithm 1.

However, the original method could yield high computational cost, duo to the necessity of considering all leave-one-out multi-sets for each new object and all possible labels, especially for large data sets. In order to overcome this problem some extensions have been investigated by [26]. One of these extensions is inductive conformal prediction (ICP), which split data into two subsets, i.e. *proper training set* and *calibration set*. The proper training set is used to train the model, and based on the trained model and a given non-conformity measure the non-conformity of the calibration set is calculated and used for the classification of new objects. See algorithm 2.

In this work we generally follow the concept of inductive conformal prediction, but with a small modification for adaptation of the model complexity. We split the

data set into multiple subsets, where we assume that each of them should be reasonably large to cover the data statistic. For the later crossvalidation scheme the data are split into three sets for training and one test set. The training sets are $T1 := \mathcal{D}_{tr}$ $(\{\mathbf{z}_1, \ldots, \mathbf{z}_n\})$ which is used to learn the classifier in a standard manner also known as the *proper training set* in inductive conformal prediction, $T2$ $(\{\mathbf{z}_{n+1}, \ldots, \mathbf{z}_m\})$ called *complexity set*, which is used to adapt the complexity of the model and $T3 := \mathcal{D}_{cal}$ $(\{\mathbf{z}_{m+1}, \ldots, \mathbf{z}_{N=m+q}\})$ the *calibration set* used during the prediction to calibrate the p-values. We will refer to the test set as $T4$ $(\{\mathbf{z}_{N+1}, \ldots, \mathbf{z}_{N+r}\})$ and assume that the labels of $T4$ are unknown and have to be predicted by the classifier.

In classical inductive conformal prediction the model is generated only *once* based on $T1$, providing a general classification rule, and the data of $T3$ are used to calculate the so called p-values which are taken to calculate the confidence and credibility measures for unknown data $T4$. In the original scheme $T2$ does not exist and is subsumed by $T1$.

Given a model trained using $T1$, for each entry of the calibration set $T3$ a non-conformity value is calculated (line 4-6 in algo. 2). Based on these non-conformity values a p-value is estimated for each possible label and test point from $T4$ (line 7-10). For classification using the conformal classifier, the label of a test item will be finally predicted as the label with the largest p-value. This refers to the label set provided by the conformal predictor which contains only one label. More complex schemes, by analyzing for example label sets with more than one label would be possible as well, but are not further considered here. The confidence value (cf) is given as one minus the second largest p-value (14) and the credibility (cr) is the largest p-value of this item (15) (for more details see section 5.1.3).

5.1.2 Non Conformity Measure

As explained above, the non conformity measure $A(D, \mathbf{z})$ should evaluate whether a test example $\mathbf{z}$ fits to data D. It is this part of the method that can incorporate detailed knowledge about the data distribution. Nevertheless one can use any real valued function [3] but maybe with negative impact on the prediction efficiency.

Thus, we assume that conformal prediction is used in the context of prototype based classifiers with a "sufficient number" of training data and where all information in the data D is implicitly represented by a trained prototype based classifier. Sufficient number should be understood in a statistical sense, as a number sufficient to describe the underlying data distribution, permitting any statistical conclusions from the data, this assumption is considered to be fulfilled in this work.

Given $\mathbf{z} = (\mathbf{x}, \mathbf{l})$, we choose

$$\alpha_i := \frac{d^+(\mathbf{x})}{d^-(\mathbf{x})} \tag{13}$$

with $d^+(\mathbf{x})$ being the distance between $\mathbf{x}$ and the closest prototype labeled $\mathbf{l}$, and $d^-(\mathbf{x})$ being the distance between $\mathbf{x}$ and the closest prototype labeled differently than $\mathbf{l}$ where distances are computed according to Eq. (3)

[3] *Any* measurable function on $\mathbb{Z}^{(*)} \times \mathbb{Z}$ taking values in the extended real line is a non conformity measure

5.1.3 Confidence and credibility

The prediction region $\Gamma^\epsilon(\mathbf{z}_1, \ldots, \mathbf{z_N}, \mathbf{v}_{N+1})$ stands in the center of conformal prediction. For a given error rate ϵ it contains the possible labels of $\mathbb{L}$ that ensure (12). But how can we use it for prediction?

Suppose we use a meaningful non conformity measure A. If the value ϵ is approaching 0, a conformal prediction with almost no errors is required, which can only be satisfied if the prediction region contains all possible labels. If we raise ϵ we allow errors to occur and as a benefit the conformal prediction algorithm excludes unlikely labels from our prediction region, increasing its information content. In detail those $\mathbf{l}$ are discarded for which the p-value is less than or equal to ϵ. Hence only a few $\mathbf{z}_i$ are as non conformal as $\mathbf{z}_{N+1} = (\mathbf{v}_{N+1}, \mathbf{l})$. This is a strong indicator that $\mathbf{z}_{N+1}$ does not belong to the distribution $\mathbb{Z}$ and so $\mathbf{l}$ seems not to be the right label. If one further raises ϵ only those $\mathbf{l}$ remain in the conformal region that can produce a high p-value meaning that the corresponding $\mathbf{z}_{N+1}$ is rated as very typical by A.

So one can trade error rate against information content. The most useful prediction is those containing exactly one label. Therefore, given an input $\mathbf{l}_i$ two error rates are of particular interest, ϵ_1^i being the smallest ϵ and ϵ_2^i being the greatest ϵ so that $|\Gamma^\epsilon(D, \mathbf{v}_i)| = 1$. ϵ_2^i is the p-value of the best and ϵ_1^i is the p-value of the second best label. Thus, typically, a conformal predictor outputs the label $\mathbf{l}$ which describes the prediction region for such choices ϵ, i.e. $\Gamma^\epsilon = \{\mathbf{l}\}$, and the classification is accompanied by the two measures

$$\text{confidence} : cf_i = 1 - \epsilon_1^i = 1 - p_{y_{\text{2nd}}} \tag{14}$$

$$\text{credibility} \ : cr_i = \epsilon_2^i = p_{y_{\text{1st}}} \tag{15}$$

Confidence says something about being sure that the second best label and all worse ones are wrong. Credibility says something about to be sure that the best label is right respectively that the data point is (a)typical and not an outlier.

The non conformity measure has a direct impact on the efficiency of the prediction region. A good, informative measure will exclude wrong labels for small error rates and will reject typical data only for great error rates, meaning that $\epsilon_2^i - \epsilon_1^i$ is large for typical data $\mathbf{v}_i$. That means, that a good measure can give useful information already for an ensured (12) small error rate ϵ_1^i and on the other hand one would have to face up a high average error rate ϵ_2^i to exclude the right label from the prediction region.

We would like to point out that the concept of conformal prediction permits pointwise measures of confidence which change for adapted training data, also in case of unchanged decision boundaries. This means, that similar as in classical statistics, more densely populated training regions permit more reliable confidence estimates during a decision. Some authors tried to obtain a kind of a probability from standard classifiers like SVM, by rescaling the distance to the decision boundary. However, with new training samples, far away from the decision boundary, a SVM model is *not* changed and hence the distance to the boundary is the same. An appropriate confidence measure of the classification can not be justified by such basic approaches, motivating again our strategy.

5.2 Complexity adaptation in a Conformal Relational Prototype Classifier

By traditional prototype methods the model complexity, i.e. number of prototypes, has to be defined beforehand, and it highly depends on the data distribution. It is difficult to find appropriate number of prototypes for different data sets. There are some extensions investigated to automatically adjust the number of prototypes by adding new prototypes or deleting redundant ones (see [12,20]), but most of them are restricted to vector space and based on heuristics, but not in a statistical sense. Especially, they can not be directly transferred to dissimilarity data.

Therefore, in this work we use the additional information provided by conformal prediction to automatically adapt the structural complexity of the model. Depending on the size of available data we can either use a full inductive conformal setting in the complexity adaptation and model prediction phase or limit inductive conformal prediction only to the model prediction. The last one means that the relational prototype classifier is not using inductive conformal prediction during the complexity adaptation but only in the prediction of the items from test set $T4$ using the calibration set $T3$. Accordingly, for model complexity adaptation based on the complexity set $T2$ we would not use a calibration set.

Alternatively we can use inductive conformal prediction also during the complexity adaptation of the relational prototype classifier. This however requires an additional calibration set in Algorithm 3, line 25. In the following we discuss the simplified case where the model complexity adaptation is based on $T1$ and $T2$ only.

As discussed before the available data are divided into multiple subsets, for training (T1), complexity (T2), calibration (T3) and test (T4). We use 80% of the training data as (T1) to train the model and 20%, denoted T2 to estimate the suitability of the current model, or the model complexity, by means of conformal prediction. Note that in this case we use a simplified version of conformal prediction in which we ignore all leave-one-out multisets and train the model on the whole $T1$. That means training only has to be performed once. The reasons thereof are: First, the locations of the prototypes depend on the whole data distribution, and will not be widely affected by a single data point. Secondly, the information loss will be minimal if the size of training data is sufficiently large, in this case adding a data point but leaving out another data point will not really affect the learning results. The calibration set $T3$ and the test set $T4$ are left out and used only in the prediction phase of the final trained model and not during the model complexity adaptation.

For T1 and T2, we compute α-values according to section 5.1.2. These values are used to calculate p-values for $T2$ (here an alternative calibration set can be used to get unbiased p-value estimates for $T2$ given a large data set). This provides point estimates for confidence and credibility of the classifier. We collect the set of points $\mathcal{B}$ with low credibility and/or confidence.

A low confidence is given if $(1 - \epsilon_1^i) \leq \zeta_1$, where ζ_1 is a user defined threshold, for example above the upper quartile of confidence values for the second best label. A low credibility is observed for $\epsilon_2^i \leq \zeta_2$, where ζ_2 is another threshold, e.g below the first quartile of confidence values for the best label.

Hence we define the so-called low confidence/credibility region $\mathcal{B}$

$$\mathcal{B} = \{\mathbf{v}_i : cf_i \leq \zeta_1 \vee cr_i \leq \zeta_2\} \tag{16}$$

If $|\mathcal{B}|$ is large (in our case we take a threshold of $\geq 1\% \cdot |T2|$), the complexity of the classifier is not yet sufficient. Hence, this parameter and ζ_1, ζ_2, control the sparsity of the model. For the data considered in the experimental section the threshold of $|\mathcal{B}|$ is in the range of $[4 - 10]$. A new prototype is created and set to the representative data point (median) in $\mathcal{B}$.

Pseudocode of the C-RPC method

Algorithm 3 Conformal relational prototype classifier.

1: **init:** $prc := 20\%$; $W :=$ randomly initialized (see Sec. 4)
2: Define Trainingset $T1 \bigcup T2$, a calibration set $T3$ and a testset $T4$
3: $\mathcal{B} := \{\emptyset\}$;
4: $T1 :=$ randomly selected $1 - prc$ of training data (proper set)
5: $T2 :=$ the remaining training data (complexity set)
6: $improve := 1\%$; ▷ threshold of improvement: default 1%
7: $itr := 0$ ▷ iteration counter
8: $ctn\_best := 0$ ▷ counter for best result
9: $max\_itr := 100$ ▷ maximal total iterations
10: $max\_ctn\_best := 10$ ▷ maximal iterations for a result as winner
11: $acc := 0$
12: **repeat**
13: $W := W \bigcup \{$new prototype representation(s) from $\mathcal{B}\}$ ▷ See description around Eq (16)
14: $W :=$ retrain W using RPC on $T1$;
15: $acc\_new :=$ evaluation of W; ▷ accuracy w.r.t. $T1$
16: **if** $acc\_new - acc \geq improve$ **then**
17: $W\_Best = W$; $acc = acc\_new$; $ctn\_best = 0$;
18: **else**
19: $ctn\_best = ctn\_best + 1$;
20: **end if**
21: ▷ adaptation of the model complexity step: see section 5.2
22: $\mathcal{A}_{T1} := \{\alpha_i, \forall i \in T1\}$, ▷ α-values of $T1$ w.r.t. W : eq. (13)
23: $\mathcal{A}^{\mathbb{L}}_{T2} := \{\alpha^l_i, \forall i \in T2, \forall l \in \mathbb{L}\}$ ▷ α-values of $T2$ for all possible labels w.r.t. W : eq. (13)
24: $P_{T2} := \{p^l_i, \forall i \in T2, \forall l \in \mathbb{L}\}$ ▷ p-values of T2 for all possible labels based on $\mathcal{A}_{T1}$ and $\mathcal{A}^{\mathbb{L}}_{T2}$
25: $CF := \{cf_i, \forall i \in T2\}, CR := \{cr_i, \forall i \in T2\}$
26: ▷ confidence/credibility of $T2$ by means of P_{T2}: eq. (14) (15)
27: generate $\mathcal{B}$ ▷ eq. (16)
28: **until** $|\mathcal{B}| < 1\% \cdot |T2|$ or $itr = max\_itr$ or $ctn\_best = max\_ctn\_best$
29: ▷ inductive conformal prediction process
30: $\mathcal{A}_{T3} := \{\alpha_i, \forall i \in T3\}$,
31: $\mathcal{A}^{\mathbb{L}}_{T4} := \{\alpha_i, \forall i \in T4, \forall l \in \mathbb{L}\}$, ▷ α-values of $T3$ and $T4$ w.r.t. W_Best : eq. (13)
32: $P_{T4} := \{p^l_i, \forall i \in T4, \forall l \in \mathbb{L}\}$ ▷ p-values of T4 for all possible labels based on $\mathcal{A}_{T3}$ and $\mathcal{A}^{\mathbb{L}}_{T4}$
return labels with largest p^l_i for each $i \in T4$

Algorithm 3 consists of three steps. Step one, covering lines 1-11, is the initialization phase where the data are divided into four datasets as described before and

some basic variables are initialized. Step two (lines 12-28) cover the training of RPC, which is repeated each time the model complexity is adapted. First a RPC model is learned, based on the current prototype representations given in W and using the training data $T1$. The optimized prototype representation is kept given there is a substantial improvement in the prediction accuracy on the training data $T1$. Further we test for data regions not well covered by the model using $T1$ and $T2$, see lines 21-27. This triggers a model complexity adaptation as described in more detail before. The algorithm iterates until the stopping criteria are met: line 28. The representation of the prototypes summarized in W is the matrix of the γ coefficients used in Eq. (3) and is based on $T1$ and $T2$. The size of this matrix (number of columns) is adapted in each complexity modeling step. Eventually, in step three, the obtained optimized prototype representation W_Best is used to predict the label and confidence or credibility values of the points from set $T4$ using $T3$ in accordance to the schema in algorithm 2 and by using Eq. (11) as the non-conformity measure.

5.3 Sparse approximation of prototypes

The RPC algorithm represents prototypes indirectly by means of coefficient vectors which are not directly interpretable since they correspond to typical positions in the pseudo-Euclidean space. However, because of their representative character, we can approximate these positions in pseudo-Euclidean space by its closest exemplars, i.e. data points originally contained in the training set. Unlike prototypes, these exemplars can be directly inspected in the same way as data. We refer to such an approximation as K-approximation if a prototype is substituted by its K closest exemplars, the latter being directly accessible to humans. We will see in experiments that the resulting classification accuracy is still quite good for the approximated models with $K = 1$ and we present an example showing the interpretability of the result. We refer to results obtained by a K-approximation by the subscript RPC_K or CRPC_K for the conformal classifier, respectively.

RPC (just as SVM) depends on the full proximity matrix and thus displays quadratic time and space complexity. Depending on the chosen dissimilarity, the main computational bottleneck is given by the computation of the dissimilarity matrix itself. Alignment of biological sequences, for example, is quadratic in the sequence length (linear, if approximations such as FASTA are used), such that a computation of the full dissimilarities for some thousand points as in the subsequent examples, would already lead to a computation time of more than some days (Intel Xeon QuadCore 2.5 GHz, alignment done by Smith-Waterman or FASTA) and a storage requirement of some 100 Megabyte. Efficient approximation strategies based on the Nyström approximation as introduced in [50] can be used. Here, we use the $K-$ approximation to obtain interpretable models and consider full similarity and dissimilarity matrices during training. The K-approximation is also extremely helpful in the test case because (dis-)similarities of the test point need only be calculated to very few training samples.

5.4 Theoretical complexity analysis

With respect to the runtime complexity, kernel methods, for example SVM needs $\mathcal{O}(N^2) - \mathcal{O}(N^3)$ operations to transfer the (dis-)similarity matrix into a valid kernel, as discussed before, and SVM training scales with $\approx \mathcal{O}(N^2)$ (using Sequential Minimal Optimization (SMO) [35]). Taking both operations together we still have a runtime complexity of $\mathcal{O}(N^2)$, for non-psd matrices.

The size of the model, given by the number of support-vector depends on the data sets. Often support vector models are large and may cover the whole training set. The relational prototype method on the other hand is trained on non-psd matrices directly and scales quadratic with the number of examples for the training [15] and the size of prototype representations is linear with respect to the number of examples.

For CRPC, due to the model adaptation CRPC has to retrain the model several times (denoted as k), normally $k \ll N$, so the retraining process of CRPC remains $\mathcal{O}(N^2)$. Additionally, the complexity of conformal prediction can be considered as linear $\mathcal{O}(N)$, since after each retraining the α-values with respect to all possible labels have to be calculated, i.e. $\mathcal{O}(k{\cdot}N{\cdot}|L|)$, and usually $|L| \ll N$, so the complexity of conformal prediction step is $\mathcal{O}(N)$.

The training time of kNN-Diss is $\mathcal{O}(N^2)$ with maximum model complexity. Again we would like to point out that the transformation to a valid psd matrix is not only costly, but also can degenerate the results as pointed out in [32]. The complexity of all methods is at least $\mathcal{O}(N^2)$, either due to the psd-correction or the training procedure. Our objective is not to obtain a faster training time, nor to achieve higher prediction accuracy. Instead we focus on *sparse, interpretable* models which can be trained in reasonable time and keep good generalization and query time for the test set, permitting pointwise measures of confidence.

6 Experiments

Initial experiments were done for the simulated checkerboard data, with known vector representation. It consists of two classes with 1250 points, in two dimensions and 5×5 clusters (see Figure 2 (left)). The dissimilarity matrix D was obtained using the Euclidean distance. RPC can learn this data only if the prototypes are initialized near the centers of the multi-modal distributions, provided a sufficient number of prototypes. The CRPC on the other hand automatically adapts its model complexity according to the introduced schema, leading to an effective model with a minimum initialization of 1 prototype per class only. We observe that the number of prototypes is slightly above the true number of 25 clusters, but the clusters are slightly overlapping and a number of 34 prototypes is considered a good result. The runtime behavior of the confidence and credibility measure during learning is shown in Figure 3.

We observe that at the initial point of learning, with only two prototypes, the number of points with a low confidence is very high but the credibility is in average quite good. This is an indicator that a large number of points is wrongly assigned, since the second label maybe similar likely. Due to the small number of prototypes a reasonable number of assignments are however considered to be correct, or the cred-

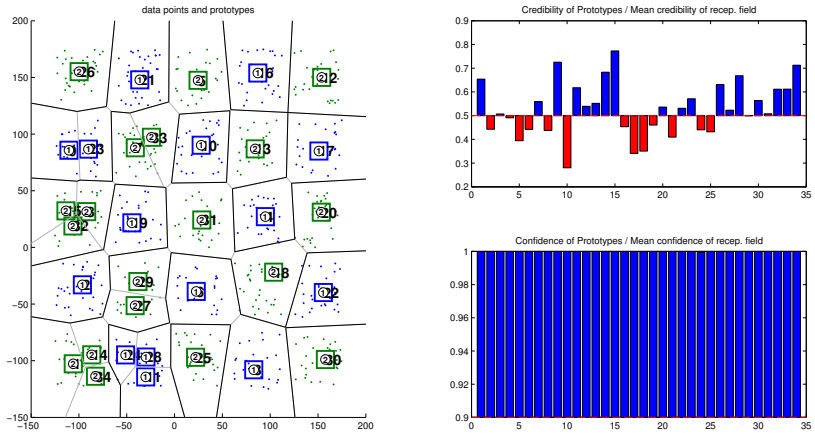


Fig. 2: Typical result of the CRPC for the two class checkerboard data, with 25 clusters. The initial model contained only 1 prototype per class, using the described conformal prediction schema the number of prototypes are auto-adjusted to 34 with almost perfect $(96.48\%(3.56))$ separation of the true clusters evaluated on the test. A standard RPC model with the same number of prototypes, as finally obtained by CRPC, was not able to learn the data and we got $50.72\%(2.59)$ accuracy on the test data. The class labels are given in the circled numbers. Right: statistic of the credibility and confidence for the different prototypes.

ibility is rather high, which is a natural consequence, because by chance $\approx 50\%$ got the correct label. To modify the model complexity, CRPC continuously analyzes the behavior of confidence and credibility. If one or both measures drop below the threshold an adaptation of the model complexity takes place. In the experiment above, the number of prototypes increased step wise, leading to a higher confidence on average. The credibility on the other hand suffers, because there are more similar prototypes (actually, those with the same label), which are alternative clusters for the considered point. Figure 3 also shows that the approach shows a convergent behavior which is also caused by limiting the minimal cluster size (if $|\mathcal{B}|<1\%\cdot|T2|$ terminates the program (Algorithm 2 line 27)). As the final prediction accuracy for CRPC_1 we get $96.48\% \pm 3.56$ which is a very good result.

We compare with SVM where, since data are Euclidean, a valid kernel results automatically[4]. The resulting prediction accuracy within a $10-$ fold cross validation is between $99-100\%$. The number of support vectors is $285-387$ which accounts to $23\%-31\%$ of the full data set. Also the kNN-Diss (see [29]) classifier performs very well with an accuracy of 100% on the test data but with all data in the model. Thus, the complexity of the alternative models is at least one order of magnitude larger as compared to CRPC_1 with automatic adjustment of the model complexity.

For further comparison, we test the algorithm on four biomedical data sets:

[4] We optimized the parameter C by a standard grid search.

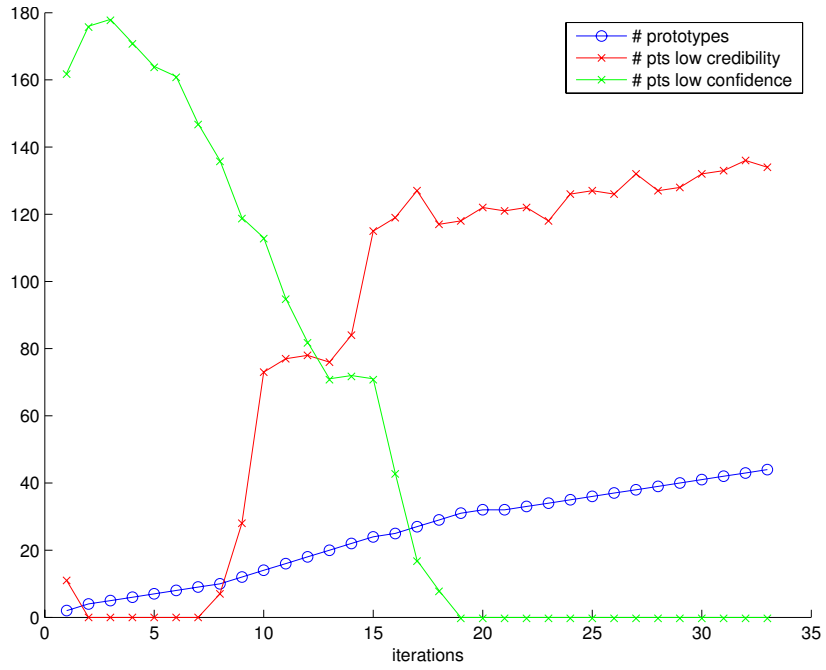


Fig. 3: Change of the number of low confidence/credibility data points during learning as well as the number of prototypes

– The *ProDom* dataset with signature $(1502, 682, 420)$ consists of 2604 protein sequences with 53 labels. It contains a comprehensive set of protein families and appeared first in the work of [37]. The pairwise structural alignments are computed by [37]. Each sequence belongs to a group labeled by experts, here we use the data as provided in [8].
– The *Protein* data set with signature $(209, 0, 4)$ consists of 213 data from 4 classes, representing globin proteins (heterogeneous globin, hemoglobin-A, hemoglobin-B, myoglobin) compared by an evolutionary measure, used already in [6].
– The *SwissProt* data set (SWISS), with a signature $(4578, 1212, 1)$, consists of 5,791 samples of protein sequences in 10 classes taken as a subset from the popular SwissProt database of protein sequences [4]. The considered subset of the SwissProt database refers to the release 37. A typical protein sequence consists of a string of amino acids, and the length of the full sequences varies between 30 to more than 1000 amino acids depending on the sequence. The 10 most common classes such as Globin, Cytochrome b, Protein kinase st, etc. provided by the Prosite labeling [11] where taken leading to 5,791 sequences. Due to this choice, an associated classification problem maps the sequences to their corresponding Prosite labels. These sequences are compared using Smith-Waterman which computes a local alignment of sequences [13]. This database is the standard source for

identifying and analyzing protein sequences such that an automated classification and processing technique would be very desirable.
- The *Bacteria* data set (Bacteria), with a signature $(2007, 0, 0)$, consists of 2007 samples of bacteria mass spec fingerprints in 30 classes taken as a subset from a commercial database provided by [24][5]. The selected bacteria classes are the most prominent ones, consisting of 22 up-to 203 entries. The underlying similarity measure and data generation are discussed in [24]. Basically, the similarities are measures of the alignment of two different spectra and the spectra encode a peptide snapshot of the considered bacterium.

We compare our results with the reference method for dissimilarity learning, the kNN-Dissimilarity classifier (kNN Diss) [33] and a support vector machine (SVM) implementation [47]. For SVM results for different preprocessing of the similarity-matrix are reported, as detailed before. The crossvalidation scheme, the kNN-Diss algorithm and the SVM have been implemented using prtools and distools [8]. The parameter C for the SVM was estimated in an internal cross validation on the training data, with a grid search $C \in [0.25, 2.5]$ with a step size of 0.25 using a linear kernel [6]. The k in kNN-Diss was auto-optimized by the distools-Toolbox, typically resulting in $k = 5$. The initial prototypes for RPC and CRPC were initialized within the class centers using random samples from the classes and optimized in the pre-described training procedure with up to 10 cycles (full training data sweeps). The initial number of prototypes is chosen according to the priorly known number of classes. We used 10 for SWISS and 21 for CHROMO and 4 for PROTEIN.

Experiments are done within a $10-$fold cross validation and with 10 repeats. We report the mean and standard deviation of the error on the test sets. For CRPC label prediction is based on the label with the highest p-value. Further we provide values for the model complexity, by means of the number of points used to represent the prototypes or, in case of SVM, the number of support vectors in the full-class model (see Table 1). For SVM we provide results where the proximity matrices have been processed as mentioned before to obtain metric similarities using clipping or flipping. This procedure has a complexity of $O(N^3)$ but is necessary for kernel machines. For comparison we also tried to obtain models without a costly eigenvalue correction (indicated by **no**) but failed for SVM due to convergence issues. Instead we provide some obtained results using a core vector machine (CVM) [46]. Theoretically CVM also can be used only for psd matrices but is less sensitive with respect to non psd matrices as long as the negative eigenvalues are small or not so relevant. For the ProDom data the negative eigenvalues are a substantial part of the data space, with a similar scale as the positive eigenvalues and it was not possible to run a kernel machine on the unprocessed proximity data.

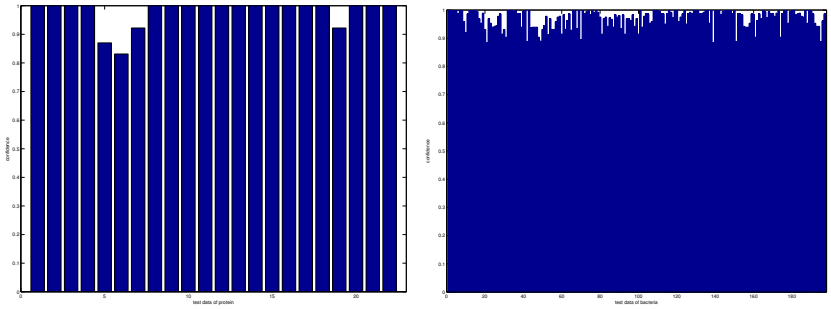
In Figure 4, as two examples, we show the important advantage of CRPC by providing the confidence measure. After the model adaptation process one can analyze the test data with respect to their high / low confidence and credibility values. Instead

[5] The database is not public but part of the sold product the article references to, here we use the version with 3034 bacteria groups. Details can be obtained by contacting the authors at Bruker.

[6] For the considered data we did not observe relevant improvements using an RBF kernel or similar, in particular since, in most cases, the Gram matrix is dealt with directly.

Table 1: Mean test set accuracies for different dissimilarity data using the knn-Dissimilarity classifier, SVM with clipping or flipping and (conformal) RPC. The standard deviations are given in parenthesis, together with the (mean) number of distinctive sample points or support vectors, rounded to whole numbers, used in the models. Full - indicates that roughly all training points belong to the model.

	ProDom (2604)	SWISS (5791)	PROTEIN (213)	Bacteria (2007)
RPC	95.00 (1.44—Full)	93.33 (0.96—Full)	97.91 (2.83—Full)	**91.96**(0.25-Full)
RPC_1	67.24 (4.73—53)	94.37 (0.83—10)	88.73 (3.22—4)	42.43(1.05-30)
CRPC	86.65 (1.84—Full)	93.74 (0.98—Full)	**98.18 (0.41—Full)**	88.71(0.38-Full)
$CRPC_1$	85.83 (2.31—88)	94.59 (1.12—12)	88.77 (1.14—4)	59.72(1.79-50)
kNN-Diss	**99.44** (0.00—Full)	98.08 (0.10—Full)	79.48 (0.45—Full)	91.85 (0.19 — Full)
CVM-no	-	97.27 (0.74—33)	76.73 (8.94—18.8)	72.54 (2.24 — 67)
SVM-flip	97.73 (1.02—782)	99.43 (0.36—712)	98.10 (3.33—140)	90.48 (2.24 — 1807)
SVM-clip	98.00 (1.05—779)	**99.52** (0.25—699)	94.78 (5.70—165)	90.38 (2.53 — 1807)



(a) confidence of protein (b) confidence of bacteria

Fig. 4: Confidence values after training for two exemplary data sets: (a) the confidence values of the test data of protein (b) the confidence values of the test data of bacteria. Point wise confidence and credibility values can be used to identify items which are not well classified, although the proposed label is correct.

of providing only a predicted label the pointwise measures for confidence and credibility also permit to identify the safety of this prediction and the consideration of alternative class label prediction (for example if a larger predicted label set, not only containing a single label, is similar likely). For the bacteria data set it is common see [24] to support the identification by a so called score measure. While this score is based on a simple non-metric measure of the similarity between the test item and a reference sample a conformal prediction is based on sound mathematical foundations. It would for example possible to identify regions of weak support or strong overlap in such databases.

Considering the different experiments we could not identify one single best method, with respect to the prediction accuracy. For PROTEIN, CRPC performed best with 20% better prediction compared to kNN-Diss and slightly better compared to SVM. For the SWISS data the best prediction result was obtained by SVM with 99.5% compared to 94.37% using RPC and 98.08% with kNN-Diss. The ProDom data have been

best predicted by kNN-Diss with 99.44% which is 1.5% better than with SVM and 4% better compared with RPC. As expected the Bacteria data are effectively modeled by all methods. Using K-approximation the results remain often quite good. Considering only $K = 1$ we obtain for $CRPC_1$ 86.65% (ProDom), 94.59% (SWISS) and 88.77% (PROTEIN) which is not as good as the best reported results, but with a significantly less number of sample points in the model. For ProDom only 3% of the points build the model, compared to $\approx 30\%$ using SVM. This effect is even more pronounced for SWISS with 0.2% of the points used by $CRPC_1$ and 12% by SVM and similar for PROTEIN $\approx 2\%$ with CRPC and 65% using SVM. The kNN-Diss classifier keeps roughly all points in the training data.

The reason why k-approximation for bacteria data set was found to be less effective is mainly due to the intrinsic dimensionality of bacteria data, which is very high. The intrinsic dimensionality can be estimated by looking at the ratio of the number of absolute eigenvalues of corresponding similarity matrix above a predefined threshold to the size of the matrix. In this case we transformed the dissimilarity to similarity matrix by using double centering and took 10% of the maximal eigenvalue as the threshold. For ProDom we obtained an intrinsic dimensionality 14.59%, for SwissProt 4.35%, for Protein 7.47%, and for Bacteria 90.43%. For high intrinsic dimensionality the prototypes can not be approximated well by using small k, they may depend on all data points. But still by means of conformal prediction we got some improvement compared to the standard approach.

The number of sample points in the model is often very relevant for dissimilarity data. As mentioned before the calculation of the scores, for example by the Smith-Waterman algorithm, is very costly. To map a new training point into the models, the (dis-)similarities to all points in the training data have to be calculated, hence a small number of sample points or sparse model is very desirable.

6.1 Interpretation of CRPC models

Considering the SWISS data set and (C-)RPC and a K-approximation of $K = 1$ we obtain a prediction accuracy of ≈ 94. This provides direct access to a very small number of associated data points, for which meta-information can be inspected.

Selecting the point associated with the $K = 1$ approximation of the *Zinc-finger* class we can track back the original swiss/uniprot reference number. Here, we get the ID 'O13124' as most representative for the group *Zinc-finger*. This leads directly to all associated meta information in the swiss-prot database. The expert can now consider the items represented by this prototype as very similar to the 'O13124' entry, revealing potential similar chemical properties within the group *Zinc-finger* modeled by (C)RPC.

For some dissimilarity data like mass-spectrometry scores in the context of bacteria identification [24] the K-approximated prototypes can be directly linked to median representations of the underlying data, here spectra. These databases are quite new and rapidly growing, requesting for inspection tools and interpretable classifiers to ensure validity of the stored results and data.

In contrast, the kNN-Diss classifier model is quite complex and an inspection is ineffective. In case of SVM the model parameters are the support vectors, which are close to the decision boundary, and hence, in general, atypical – limiting their usefulness for an interpretation.

For new points the model now also provides pointwise estimates of the confidence and credibility of the classification according Eq. (13). The classification is therefore accompanied by two values indicating the safety of the classification. Points which are probably assigned to the wrong class can be identified by, most often low confidence and low credibility values. But also cases where points are equally similar to two classes, for example, can be detected and appropriate analysis of the meta-data, more specific sub-models or reject operations can be applied.

7 Conclusions

We have defined the sparse conformal relational prototype classifier, an efficient classifier for dissimilarity data based on the relational prototype classifier and the conformal prediction concept. In addition to the good prediction accuracy, CRPC automatically adapts the model complexity and outputs measures of its accuracy by means of point wise confidence and credibility values, with a clear probabilistic interpretation. The experimental results show good performance compared to standard techniques but with easier access to much sparser models. In future work we will in more detail address the interpretability of the obtained models and how this can be linked to the supervised modeling of sequence databases and other application fields.

Acknowledgment

We would like to thank the anonymous reviewers for the helpful suggestions to improve this paper. This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication. Funding in the frame of the centre of excellence 'Cognitive Interaction Technologies' (CITEC) is gratefully acknowledged. The first author was kindly supported by a Marie Curie Intra-European Fellowship (IEF) FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS).

References

1. Balasubramanian, V., Chakraborty, S., Panchanathan, S., Ye, J.: Kernel learning for efficiency maximization in the conformal predictions framework. pp. 235–242 (2010)
2. Bhattacharyya, S.: Confidence in predictions from random tree ensembles. Knowledge and Information Systems **35**(2), 391–410 (2013)
3. Biehl, M., Ghosh, A., Hammer, B.: Dynamics and generalization ability of lvq algorithms. Journal of Machine Learning Research **8**, 323–360 (2007)
4. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: The swiss-prot protein knowledgebase and its supplement trembl in 2003,. Nucleic Acids Research **31**, 365–370

5. Chen, H., Tino, P., Yao, X.: Probabilistic classification vector machines. IEEE Transactions on Neural Networks **20**(6), 901–914 (2009)
6. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. Journal of Machine Learning Research **10**, 747–776 (2009)
7. Cordella, L.P., Foggia, P., Sansone, C., Tortorella, F., Vento, M.: Reliability parameters to improve combination strategies in multi-expert systems. Pattern Analysis and Applications **2**(3), 205–214 (1999)
8. Duin, R.P.: PRTools (2012). URL http://www.prtools.org
9. Duin, R.P.W., Loog, M., Pekalska, E., Tax, D.M.J.: Feature-based dissimilarity space classification. In: D. Ünay, Z. Çataltepe, S. Aksoy (eds.) ICPR Contests, *Lecture Notes in Computer Science*, vol. 6388, pp. 46–55. Springer (2010)
10. Elomaa, T., Mannila, H., Toivonen, H. (eds.): Machine Learning: ECML 2002, 13th European Conference on Machine Learning, Helsinki, Finland, August 19-23, 2002, Proceedings, *Lecture Notes in Computer Science*, vol. 2430. Springer (2002)
11. Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R., Bairoch, A.: Expasy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res **31**(3784-3788) (2003)
12. Grbovic, M., Vucetic, S.: Learning vector quantization with adaptive prototype addition and removal. In: Neural Networks, 2009. IJCNN 2009. International Joint Conference on, pp. 994–1001 (2009). DOI 10.1109/IJCNN.2009.5178710
13. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press (1997)
14. Haasdonk, B., Bahlmann, C.: Learning with distance substitution kernels. Pattern Recognition - Proc. of the 26th DAGM Symposium (2004)
15. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation **22**(9), 2229–2284 (2010)
16. Hammer, B., Mokbel, B., Schleif, F.M., Zhu, X.: Prototype-based classification of dissimilarity data. In: J. Gama, E. Bradley, J. Hollmén (eds.) IDA, *Lecture Notes in Computer Science*, vol. 7014, pp. 185–197. Springer (2011)
17. Hammer, B., Schleif, F.M., Zhu, X.: Relational extensions of learning vector quantization. In: B.L. Lu, L. Zhang, J.T. Kwok (eds.) ICONIP (2), *Lecture Notes in Computer Science*, vol. 7063, pp. 481–489. Springer (2011)
18. Hammer, B., Strickert, M., Villmann, T.: On the generalization ability of grlvq networks. Neural Processing Letters **21**(2), 109–120 (2005)
19. Hebiri, M.: Sparse conformal predictors. Statistics and Computing **20**(2), 253–266 (2010)
20. Kohonen, T., Kangas, J., Laaksonen, J., Torkkola, K.: Lvq pak: A program package for the correct application of learning vector quantization algorithms. pp. 725–730. IEEE (1992)
21. L., G.: A unified approach to pattern recognition. Pattern Recognition **17**(5), 575–582 (1984)
22. Laub, J., Roth, V., Buhmann, J.M., Müller, K.R.: On the information and representation of non-euclidean pairwise data. Pattern Recognition **39**(10), 1815–1826 (2006)
23. Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pekalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. Pattern Recognition **39**(10), 1827–1838 (2006)
24. Maier, T., Klebel, S., Renner, U., Kostrzewa, M.: Fast and reliable maldi-tof ms–based microorganism identification. Nature Methods (3) (2006)
25. Manolova, A., Guérin-Dugué, A.: Classification of dissimilarity data with a new flexible mahalanobis-like metric. Pattern Anal. Appl. **11**(3-4), 337–351 (2008)
26. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. In: Tools in Artificial Intelligence, Chap. 18, pp. 315–330. I-Tech (2008)
27. Papadopoulos, H., Proedrou, K., Vovk, V., Gammerman, A.: Inductive confidence machines for regression. In: Elomaa et al. [10], pp. 345–356
28. Papadopoulos, H., Vovk, V., Gammerman, A.: Regression conformal prediction with nearest neighbours. Journal of Artificial Intelligence Research **40**, 815–840 (2011)
29. Pekalska, E., Duin, R.: The dissimilarity representation for pattern recognition. World Scientific (2005)
30. Pekalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recognition Letters **23**(8), 943–956 (2002)
31. Pekalska, E., Duin, R.P.W.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Transactions on Systems, Man, and Cybernetics, Part C **38**(6), 729–744 (2008)

32. Pekalska, E., Duin, R.P.W., Günter, S., Bunke, H.: On not making dissimilarities euclidean. In: A.L.N. Fred, T. Caelli, R.P.W. Duin, A.C. Campilho, D. de Ridder (eds.) SSPR/SPR, *Lecture Notes in Computer Science*, vol. 3138, pp. 1145–1154. Springer (2004)
33. Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition **39**(2), 189–208 (2006)
34. Pekalska, E., Haasdonk, B.: Kernel discriminant analysis for positive definite and indefinite kernels. IEEE Trans. Pattern Anal. Mach. Intell. **31**(6), 1017–1032 (2009)
35. Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. pp. 185–208. MIT Press, Cambridge, MA, USA (1999)
36. Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive confidence machines for pattern recognition. In: Elomaa et al. [10], pp. 381–390
37. Roth, V., Laub, J., Buhmann, J.M., Müller, K.R.: Going metric: Denoising pairwise data. In: S. Becker, S. Thrun, K. Obermayer (eds.) NIPS, pp. 817–824. MIT Press (2002)
38. Sato, A., Yamada, K.: Generalized learning vector quantization. In: D.S. Touretzky, M. Mozer, M.E. Hasselmo (eds.) NIPS, pp. 423–429. MIT Press (1995)
39. Schleif, F.M., Villmann, T., Hammer, B., Schneider, P.: Efficient kernelized prototype based classification. Int. J. Neural Syst. **21**(6), 443–457 (2011)
40. Schneider, P., Geweniger, T., Schleif, F.M., Biehl, M., Villmann, T.: Multivariate class labeling in robust soft lvq. In: Proceedings of ESANN 2011, pp. 17–22 (2011)
41. Seo, S., Obermayer, K.: Soft learning vector quantization. Neural Computation **15**(7), 1589–1604 (2003)
42. Shafer, G., Vovk, V.: A tutorial on conformal prediction. Journal of Machine Learning Research **9**, 371–421 (2008)
43. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis and Discovery. Cambridge University Press (2004)
44. de Stefano, C., Sansone, C., Vento, M.: To reject or not to reject: that is the question: an answer in case of neural classifiers. IEEE Transactions on Systems, Man and Cybernetics Part C **30**(1), 84–93 (2000)
45. Tipping, M.E.: The relevance vector machine. In: S.A. Solla, T.K. Leen, K.R. Müller (eds.) NIPS, pp. 652–658. The MIT Press (1999)
46. Tsang, I.W., Kocsor, A., Kwok, J.T.: Simpler core vector machines with enclosing balls. In: Z. Ghahramani (ed.) ICML, *ACM International Conference Proceeding Series*, vol. 227, pp. 911–918. ACM (2007)
47. Vapnik, V.: The nature of statistical learning theory. Statistics for engineering and information science. Springer (2000)
48. Vovk, V.: Conditional validity of inductive conformal predictors. Journal of Machine Learning Research - Proceedings Track **25**, 475–490 (2012)
49. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic Learning in a Random World. Springer, New York (2005)
50. Williams, C., Seeger, M.: Using the nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems 13, pp. 682–688. MIT Press (2001)
51. Yang, M., Nouretdinov, I., Luo, Z., Gammerman, A.: Feature selection by conformal predictor. IFIP Advances in Information and Communication Technology **364 AICT**(PART 2), 439–448 (2011)