A Conformal Classifier for Dissimilarity Data

Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer

CITEC centre of excellence, Bielefeld University, 33615 Bielefeld, Germany
{fschleif,xzhu,bhammer}@techfak.uni-bielefeld.de

Abstract. Current classification algorithms focus on vectorial data, given in euclidean or kernel spaces. Many real world data, like biological sequences are not vectorial and often non-euclidean, given by (dis-)similarities only, requesting for efficient and interpretable models. Current classifiers for such data require complex transformations and provide only crisp classification without any measure of confidence, which is a standard requirement in the life sciences. In this paper we propose a prototype-based conformal classifier for dissimilarity data. It effectively deals with dissimilarity data. The model complexity is automatically adjusted and confidence measures are provided. 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

Learning for similarity and dissimilarity data is an active research field [2], since many data sets are naturally dealt with in terms of domain dependent measures. Examples include edit distance based measures for strings, images or popular similarity measures in bioinformatics (e.g. fasta, smith-waterman or blast [6]). Classifiers based on dissimilarity data assign a class label to a given example based on 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. If S is obtained from a valid inner-product, S can be considered as a kernel matrix. This can be processed by kernel-classifiers like Support Vector Machine (SVM) [16]. If S does not constitute a valid kernel, additional transformations are necessary to guarantee semi positive definiteness [2].

Some dedicated classification methods for dissimilarity data have been proposed in the last years, motivated by the work reported in [10]. In [4] a feature based dissimilarity space classification is proposed and combined with different classifiers. It was found that the dissimilarity representation is on average more effective than traditional feature representations. For new test data however all dissimilarities with respect to the training points have to be calculated, which can be prohibitive for large data sets. In [10] 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

L. Iliadis et al. (Eds.): AIAI 2012 Workshops, IFIP AICT 382, pp. 234–243, 2012.

methods are discussed in [11] but the approaches are not in closed form or are applicable for two class problems, only; additionally, results are quite limited. In [9] different techniques are compared, focusing on the determination/reduction of prototypes for dissimilarity learning, and they constitute a motivation for the approach proposed in the following, but with our strategy, reference vectors are obtained in a natural way.

The conformal prediction method which will be proposed in the following is based on a recent dissimilarity classifier introduced in [8]. Working directly on the dissimilarity matrix it arrives at a prototype-based classifier representing the data by a fixed number of prototypes. It replaces an explicit distance measure by an implicit form depending on the given dissimilarity matrix only. While very effective, the model is limited to crisp classification decisions, without additional information about confidence and the model complexity has to be pre-specified by a meta-parameter.

In the following we extend this relational prototype learner [8] by conformal prediction concepts, referred to as Conformal Relational Prototype Classifier (CRPC). CRPC directly deals with dissimilarity data, providing compact interpretable models, supporting each classification by a measure of confidence. In addition, the confidence is used for a dynamic adaptation of the model complexity during learning, increasing the model complexity as required by the resulting conformal regions. Now we first summarize some known facts about (dis-)similarity data. We shortly revisit the basic relational prototype based classifier, and then introduce the concept of conformal prediction in this context. The suitability of the technique is demonstrated using data from bioinformatics.

2 Preliminaries about Dissimilarity Data

Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects defined in some data space, with $|\mathbb{V}| = N$. We expect 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 dissimilarity matrix D, an associated similarity matrix S is induced by a process referred to as double centering: $S = -JDJ/2$ where $J = (I - \mathbf{1}\mathbf{1}^t/N)$ with identity matrix I and vector of ones $\mathbf{1}$. The costs of this operation are $\mathcal{O}(N^3)$. D is Euclidean if and only if S is positive semidefinite (psd). In case of psd (similarity) kernel matrices, standard kernel methods are available [14]. To guarantee psd, some preprocessings, outlined in [2] can be applied (e.g. clipping, flipping, shift, vector-representation). The idea is to change the eigenvalue decomposition of the similarity matrix S such that negative eigenvalues are avoided. Details about the transformations are discussed in [2], subsequently we focus on clipping and flipping, known to be effective in most cases. These operations are *only* applied for kernel methods but not for the used relational methods which do not rely on psd matrices.

Some further alternative transformations were proposed but in general severely affect the performance of optimization algorithms as discussed in [7,2]. The analysis in [10] indicates that for non-Euclidean dissimilarities corrections like above should be avoided.

Alternatively, techniques have been introduced which directly deal with possibly non-psd dissimilarities, embedding the data in a pseudo-Euclidean vector space (see e.g.

[7]). Then, vector operations can be directly transferred to this 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 vector operations only. One problem of this explicit transfer are the embedding costs of $\mathcal{O}(N^3)$, and, further, the fact that out-of-sample extensions to new data points are not immediate. Because of this fact, we are interested in efficient techniques using this embedding only implicitly.

3 Relational Prototype Based Learning

We assume a training set is given whose data points $\mathbf{v}_j$ are 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 $W \subseteq \mathbb{V}$. We build our new model based on a recently published prototype classifier for dissimilarity data [8].

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, using the bilinear form in this 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

$$\mathbf{w}_j = \sum_i \alpha_{ji} \mathbf{v}_i \text{ with } \sum_i \alpha_{ji} = 1 \,. \tag{1}$$

Then dissimilarities can be computed implicitly by means of

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

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\mathbf{w}_j$ implicitly, as shown in [7].

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 [13]. The corresponding cost function of the relational prototype classifier (RPC) becomes:

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

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

$$\Delta\alpha_k^+ \sim \; -\Phi'(\mu(\mathbf{v}_i)) \cdot \mu^+(\mathbf{v}_i) \cdot \frac{\partial \left([D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ \right)}{\partial \alpha_k^+}$$

$$\Delta\alpha_k^- \sim \; \Phi'(\mu(\mathbf{v}_i)) \cdot \mu^-(\mathbf{v}_i) \cdot \frac{\partial \left([D\alpha^-]_i - \frac{1}{2} \cdot (\alpha^-)^t D \alpha^- \right)}{\partial \alpha_k^-}$$

with

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

The partial derivative yields

$$\frac{\partial \left([D\alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j \right)}{\partial \alpha_{jk}} = d_{ik} - \sum_l d_{lk} \alpha_{jl}$$

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

The prototypes are initialized as random vectors corresponding to random values α_{ij} which sum to one.

Out-of-sample extension of the classification to new data is possible based on the following observation [7]: 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 α_j is

$$d(\mathbf{v}, \mathbf{w}_j) = D(\mathbf{v})^t \cdot \alpha_j - \frac{1}{2} \cdot \alpha_j^t D \alpha_j$$

4 Conformal Prediction

RPC is very effective as shown in [8] 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 sciences 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. A second drawback is that the complexity of the model in terms of the number of prototypes needs to be specified a priory.

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 [12], with a recent tutorial given in [15]. Conformal prediction aims at the determination of confidence and credibility of classifier decisions.

Conformal Prediction for RPC. We follow the general approach of conformal prediction as reviewed in [15]. 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 training 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* $\Gamma^\epsilon(\mathbf{z}_1,\ldots,\mathbf{z}_l,\mathbf{v}_{N+1}) \subseteq \mathbb{L}$ consisting of a number of possible label assignments. The applied method ensures that if the data $\mathbf{z}_i$ are exchangeable then

$$P(\mathbf{l}_{N+1} \notin \Gamma^\epsilon(\mathbf{z}_1,\ldots,\mathbf{z}_l,\mathbf{v}_{N+1})) \leq \epsilon \tag{3}$$

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 n times we will find with respect to statistical fluctuations that in less than $\epsilon \times n$ cases the real label $\mathbf{l}_{N+1}$ is not under the predicted labels of Γ^ϵ.

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 conformal algorithm for classification is as follows: given a nonconformity measure A, significance level ϵ, examples $\mathbf{z}_1$, ..., $\mathbf{z}_N$, object $\mathbf{v}_{N+1}$ and label $\mathbf{l}$, it is decided whether $\mathbf{l}$ is contained in $\Gamma^\epsilon(\mathbf{z}_1,\ldots,\mathbf{z}_N,\mathbf{v}_{N+1})$:

$$\begin{aligned}
&\text{set } \mathbf{z}_{N+1} := (\mathbf{x}_{N-1},\mathbf{l})\\
&\text{for } i = 1,\ldots,N+1 \text{ set } \alpha_i := A(\{\mathbf{z}_1,\ldots,\mathbf{z}_{N+1}\}\backslash\{\mathbf{z}_i\},\mathbf{z}_i)\\
&\text{set } r_l := \frac{|\{i=1,\ldots,N+1 \mid \alpha_i \geq \alpha_{N+1}\}|}{N+1}\\
&\text{include } \mathbf{l} \text{ if } r_l > \epsilon
\end{aligned}$$

Non Conformity Measure. As explained above, the non conformity measure $A(D,\mathbf{z})$ should evaluate whether a test example $\mathbf{z}$ fits to the representative 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 [1] but maybe with negative impact on the prediction efficiency. An obvious solution is to learn a prototype classifier with each individual D and match $\mathbf{z}$ against it. However, this method would entail high computational costs, because this procedure has to be done for all leave-one-out multi-sets for each of the L test objects $(\mathbf{v}_{N+1},\mathbf{l})$ in the conformal prediction algorithm. Our solution lies in the arbitrariness of A. We can ignore matching exactly against the data set but instead use the whole training data without $\mathbf{z}_{N+1}$, therefore learning must be performed only once. The information loss will be small if the number of training data is high, so that adding $\mathbf{z}_i$ but leaving out $\mathbf{z}_{N+1}$ will not significantly affect the learning results.

Thus, we assume that conformal prediction is used in the context of prototype based classifiers for which a sufficient number of training data is available and all information in the data D is implicitly represented by a trained prototype based classifier. Given $\mathbf{z} = (\mathbf{x},\mathbf{l})$, we choose

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

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

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. (2)

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 (3). 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 r-value is less or equal 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 r-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 largest ϵ so that $|\Gamma^\epsilon(D, \mathbf{v}_i)| = 1$. ϵ_2^i is the r-value of the best and ϵ_1^i is the r-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

$$\text{confidence} : 1 - \epsilon_1^i = 1 - r_{y_{\text{2nd}}} \tag{4}$$

$$\text{credibility} : \epsilon_2^i = r_{y_{\text{1st}}} \tag{5}$$

Confidence says something about being sure that the second best label and all worse ones are wrong. Credibility says something about being sure that the best label is right respectively that the data point is (un)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 $\epsilon_2^i - \epsilon_1^i$ being large for typical data $\mathbf{v}_i$. That means, that a good measure can give useful information already for an ensured (3) 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.

Complexity Adaptation in a Conformal Relational Prototype Classifier. We use the additional information provided by a conformal relational prototype classifier to automatically adapt the complexity of the model, i.e. the number of prototypes. We use 80% of the training set, denoted as T1 to train the model and 20%, denoted T2 to estimate the suitability of the current model by means of conformal prediction. For this subset, we compute α-values according to section 4. 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 $(1 - \epsilon_1^i) \leq (1 - \frac{1}{L})$ and a low credibility is observed for $\epsilon_2^i \leq \frac{1}{L}$. Hence we define

$$\mathcal{B} = \left\{ \mathbf{v}_i : \left(1 - \epsilon_1^i\right) \leq \left(1 - \frac{1}{L}\right) \vee \epsilon_2^i \leq \frac{1}{L} \right\} \tag{6}$$

If $|\mathcal{B}|$ is large, in our case we take the boundary ≥ 5, the complexity of the classifier is not yet sufficient. Hence, this parameter controls the sparsity of the model. We found by some independent experiments on simulated data, that $|\mathcal{B}| = 5$ is a good compromise between too dense $|\mathcal{B}| =\leq 5$ or to sparse models $|\mathcal{B}| \gg 5$. A new prototype is created and set to a representative data point in $\mathcal{B}$. The pseudo-code of the C-RPC algorithm is shown in Alg.:1.

The RPC algorithm represents prototypes indirectly by means of coefficient vectors which are not directly interpretable since they correspond to typical positions in a pseudo-Euclidean space. However, because of their representative character, we can approximate these positions in the 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 small values K in $\{1, 3\}$, 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.

The K-approximation is also extremely helpful in the test case because (dis-)similarities of the test point need only to be calculated for very few training samples.

Pseudocode of the C-RPC Method

5 Experiments

We evaluate our approach for a set of biomedical data:

- The *ProDom* dataset consists of 2604 protein sequences with 53 labels. It contains a comprehensive set of protein families compared by a pairwise structural alignment. Each sequence belongs to a group labeled by experts, here we use the data as provided in [3].
- The *Protein* data set 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 [2].
- The *SwissProt* data set (SWISS) consists of 5,791 samples of protein sequences in 10 classes taken as a subset from the popular SwissProt database of protein sequences [1]. The considered subset of the SwissProt database refers to the release 37. The 10 most common classes such as Globin, Cytochrome b, Protein kinase st, etc. provided by the Prosite labeling [5] 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, computing a local alignment of sequences [6].

Algorithm 1. Pseudocode of the CRPC algorithm

init: $prc := 20\%$; $credi\_threshod := \frac{1}{L}$, $confi\_threshod := 1 - \frac{1}{L}$; $W := \emptyset$;
$\mathcal{B} := \emptyset$;
$T1 :=$ randomly selected $1 - prc$ of training data;
$T2 :=$ the remaining training data
$improve := 1\%$; ▷ threshold of improvement: default 1%
$itr := 0$ ▷ iteration counter
$ctn\_best := 0$ ▷ counter for best result
$max\_itr := 100$ ▷ maximal total iterations
$max\_ctn\_best := 10$ ▷ maximal iterations for a result as winner
$W :=$ train $T1$ by RPC ; $W\_Best = W$;
$acc :=$ evaluation of W; ▷ accuracy w.r.t. $T1$
$A\_T1 := \{\alpha_i, \forall i \in T1\}$ ▷ α-values of $T1$: eq. (4)
$A\_T2 := \{\alpha_i, \forall i \in T2\}$ ▷ α-values of $T2$
$Confi := \{1 - \epsilon_1^i, \forall i \in T2\}$, $Credi := \{\epsilon_2^i, \forall i \in T2\}$ ▷ confidence/credibility of $T2$ by means of A_T1 and A_T2: eq. (4),(5),
generate $\mathcal{B}$ ▷ eq. (6)
while $|\mathcal{B}| \geq 5$ & $itr < max\_itr$ & $ctn\_best \leq max\_ctn\_best$ **do**
    $W := W \bigcup \{$new prototype(s) from $\mathcal{B}\}$
    $W :=$ train $T1$ by RPC given W; ▷ training with given prototypes
    $acc\_new :=$ evaluation of W; ▷ new accuracy
    $A\_T1 := \{\alpha_i, \forall i \in T1\}$; $A\_T2 := \{\alpha_i, \forall i \in T2\}$
    $Confi := \{1 - \epsilon_1^i, \forall i \in T2\}$; $Credi := \{\epsilon_2^i, \forall i \in T2\}$;
    generate $\mathcal{B}$,
    if $acc\_new - acc \geq improve$ **then**
        $W\_Best = W$; $acc = acc\_new$; $ctn\_best = 0$;
    else
        $ctn\_best = ctn\_best + 1$;
    end if
end while
return W_Best;

We compare our results with the reference method for dissimilarity learning, the kNN-Dissimilarity classifier (kNN Diss) [11] and a support vector machine (SVM) implementation [16]. 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 [3]. 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 [2]. The k in kNN-Diss was auto-optimized by the distools-Toolbox, typically resulting in $k = 5$. The initial prototypes for RPC and CRPC where 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.

[2] 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. Standard deviations are given in parenthesis, with the (mean) number of distinctive sample points or support vectors (rounded), used in the models. Full - indicates that roughly all training points belong to the model.

	ProDom (2604)	SWISS (5791)	PROTEIN (213)
RPC	95.00 (1.44—Full)	93.33 (0.96—Full)	97.91 (2.83—Full)
RPC_1	67.24 (4.73—53)	94.37 (0.83—10)	88.73 (3.22—4)
RPC_3	77.00 (2.12—159)	57.47 (2.54—30)	89.58 (7.52—12)
CRPC	96.09 (0.05—Full)	93.59 (1.18—Full)	98.15 (0.06—Full)
$CRPC_1$	92.16 (0.07—96)	94.12 (1.05—21)	94.11 (0.10—4)
$CRPC_3$	96.89 (0.08—253)	84.11 (0.17—54)	**99.08** (2.40—12)
kNN-Diss	**99.44** (0.00—Full)	98.08 (0.10—Full)	79.48 (0.45—Full)
SVM-flip	97.73 (1.02—782)	99.43 (0.36—712)	98.10 (3.33—140)
SVM-clip	98.00 (1.05—779)	**99.52** (0.25—699)	94.78 (5.70—165)

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. 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 vector in the full-class model (see Table 1).

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 1% 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 3% better compared with CRPC. Using K-approximation the results remain typically quite good[3]. Considering only $K = 1$ we obtain for CRPC 92.16% (ProDom), 94.12% (SWISS) and 94.11% (PROTEIN) which is $5 - 7\%$ worse compared to 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 0.4% of the points used by CRPC 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.

Interestingly the CRPC can compensate performance degradation caused by the K-approximation by additional prototypes, leading to only slightly more complex models, compared to RPC but with significantly improved prediction accuracy as compared to a direct K-approximation. The number of sample points in the model is often very relevant for dissimilarity data. As mentioned before the calculation of the scores, e.g. 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 a compact model is very desirable.

[3] With the exception of SWISS, where the α-matrix is already sparse and a rescaling of the remaining α values after $K-$approximation degraded the model.

6 Conclusions

We have defined the 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 provides 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 more compact models.

Acknowledgment. This work has been supported by the German Research Foundation (DFG) under grant number HA2719/4-1 and HA2719/6-1. Funding in the frame of the centre of excellence 'Cognitive Interaction Technologies' (CITEC) is gratefully acknowledged.

References

[1] Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: The swiss-prot protein knowledgebase and its supplement trembl in 2003. Nucl. Ac. Res. 31, 365–370

[2] Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: Concepts and algorithms. J. of Mach. Learn. Res. 10, 747–776 (2009)

[3] Duin, R.P.W.: PRTools (March 2012), `http://www.prtools.org`

[4] Duin, R.P.W., Loog, M., Pękalska, E.z., Tax, D.M.J.: Feature-Based Dissimilarity Space Classification. In: Ünay, D., Çataltepe, Z., Aksoy, S. (eds.) ICPR 2010. LNCS, vol. 6388, pp. 46–55. Springer, Heidelberg (2010)

[5] Gasteiger, E.: Expasy: the proteomics server for in-depth protein knowledge and analysis. Nuc. Ac. Res. 31(3784-3788) (2003)

[6] Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press (1997)

[7] Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation 22(9), 2229–2284 (2010)

[8] Hammer, B., Schleif, F.-M., Zhu, X.: Relational Extensions of Learning Vector Quantization. In: Lu, B.-L., Zhang, L., Kwok, J. (eds.) ICONIP 2011, Part II. LNCS, vol. 7063, pp. 481–489. Springer, Heidelberg (2011)

[9] Lozano, M., 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)

[10] Pekalska, E., Duin, R.P.W.: The dissimilarity representation for pattern recognition. World Scientific (2005)

[11] Pekalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39(2), 189–208 (2006)

[12] Proedrou, K., Nouretdinov, I., Vovk, V., Gammerman, A.: Transductive Confidence Machines for Pattern Recognition. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 381–390. Springer, Heidelberg (2002)

[13] Sato, A., Yamada, K.: Generalized learning vector quantization. In: NIPS, pp. 423–429 (1995)

[14] Schleif, F.-M., Villmann, T., Hammer, B., Schneider, P.: Efficient kernelized prototype based classification. Int. J. Neural Syst. 21(6), 443–457 (2011)

[15] Shafer, G., Vovk, V.: A tutorial on conformal prediction. JMLR 9, 371–421 (2008)

[16] Vapnik, V.: The nature of statistical learning theory. Stat. f. Eng. & Inf. Sc. Springer (2000)