Secure Semi-Supervised Vector Quantization for Dissimilarity Data

Xibin Zhu, Frank-Michael Schleif, and Barbara Hammer

CITEC - Centre of Excellence,
Bielefeld University, 33615 Bielefeld, Germany
{*xzhu, fschleif, bhammer*}*@techfak.uni-bielefeld.de*

Abstract. The amount and complexity of data increase rapidly, however, due to time and cost constrains, only few of them are fully labeled. In this context non-vectorial relational data given by pairwise (dis-)similarities without explicit vectorial representation, like score-values in sequences alignments, are particularly challenging. Existing semi-supervised learning (SSL) algorithms focus on vectorial data given in Euclidean space. In this paper we extend a prototype-based classifier for dissimilarity data to non i.i.d. semi-supervised tasks. Using conformal prediction the 'secure region' of unlabeled data can be used to improve the trained model based on labeled data while adapting the model complexity to cover the 'insecure region' of labeled data. The proposed method is evaluated on some benchmarks from the SSL domain.

Keywords: Semi-Supervised Learning, Proximity Data, Dissimilarity Data, Conformal Prediction, Learning Vector Quantization

1 Introduction

Big data are getting more and more challenging by means of storage and analysis requirements. Besides the amount of data, only few of these data are totally labeled, and labeling of all these data is indeed very costly and time consuming. Techniques of data mining, visualization, and machine learning are necessary to help people to analyse such data. Especially semi-supervised learning techniques, which integrate the structural and statistical knowledge of unlabeled data into the training, are widely used for this setting. A variety of SSL methods has been published [1]. They all focus on vectorial data given in Euclidean space or representations by means of positive semi-definite (psd) kernel matrices.

Many real world data are non-vectorial, often non-euclidean and given in the form of pairwise proximities between objects. Such data are also referred to as *proximity* or *relational data*, which are based on pairwise comparisons of objects providing some score-value of the (dis-)similarity of the objects. For such data, a vector space is not necessarily available and there is no guarantee of metric conditions. Examples of such proximity or (dis-)similarity measures are edit distance based measures for strings or images [5] or popular similarity

measures in bioinformatics such as scores obtained by the Smith-Waterman, FASTA, or blast algorithm [4]. Such partially labeled relational data are not widely addressed in the literature of SSL, yet. Only few methods consider SSL for classification of proximity data without an explicit underlying vector space and without requesting a metric space [9, 13], this is the topic of this paper.

In this paper we extend a prototype-based classifier proposed in [3] for semi-supervised tasks of non i.i.d. data employing conformal prediction [14] technique. For SSL tasks, conformal prediction is used to determine the *secure region* of unlabeled data, which can potentially enhance the performance of the training, and at the same time estimates a so-called *insecure region* of labeled data which helps to adapt the model complexity. The proposed method can directly deal with non-psd proximity multi-class data.

First we will review relational supervised prototype-based learning as recently introduced by the authors in a specific model, employing conformal prediction concepts as discussed in [11]. Thereafter we introduce an extension to semi-supervised learning. We show the effectiveness of our technique on simulated data, well-known vectorial data sets and biomedical dissimilarity data which are not psd. Finally we summarize our results and discuss potential extensions.

2 Semi-supervised prototype-based relational learning

Prototype-based relational learning for unsupervised and supervised cases has been investigated by [3]. For semi-supervised problems, first we will briefly review the idea of prototype-based learning for relational data, then we will give a short introduction about conformal prediction for prototype-based learning and finally show how to extend it for semi-supervised problems.

2.1 Prototype-based relational learning

As mentioned before, in the relational setting, data is not given as vectors, but as pairwise relation(s) between data points, e.g. distances between two points or some scores that describe some relations between the data. Let $\mathbf{v}_j \in \mathbb{V}$ be a set of objects defined in some data space, with $|\mathbb{V}| = N$. 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 $(\mathbf{v}_i, \mathbf{v}_j) \in \mathbb{V} \times \mathbb{V}$. 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 $\{i, j\}$.

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. We use a recently published prototype classifier for dissimilarity data [3] as basic method in the following. As detailed in [3], these data can always be embedded in pseudo-euclidean space in such a way that $d(\mathbf{v}_i, \mathbf{v}_j)$ is induced by a synthetic (but possibly not psd) bilinear form.

Classification takes place by means of k prototypes $\mathbf{w}_j \in W$ 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 $\mathbf{w}_j = \sum_i \alpha_{ji}\mathbf{v}_i$ with $\sum_i \alpha_{ji} = 1$. Then dissimilarities between data points and prototypes can be computed implicitly by means of

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

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ refers to the vector of coefficients describing the prototype $\mathbf{w}_j$, as shown in [3].

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 [10]. The corresponding cost function of the *relational prototype-based 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 prototypes are referred to, $\mathbf{w}^+$ and $\mathbf{w}^-$, respectively, corresponding to the coefficients α^+ and α^-, 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: component k of these vectors is adapted as

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

with

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

The partial derivative yields

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

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 pseudo-Euclidean space.

The prototypes are initialized as random vectors corresponding to random values α_{ij} which sum to one. It is possible to take class information into account by setting all α_{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 [3]: 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 $d(\mathbf{v}, \mathbf{w}_j) = D(\mathbf{v})^t \cdot \alpha_j - \frac{1}{2} \cdot \alpha_j^t D \alpha_j$.

2.2 Conformal Prediction for RPC

RPC can be effectively transferred to a conformal predictor which will be useful to extend it in a non-trivial way to semi-supervised learning. Conformal predictor introduced in [14] aims at the determination of confidence and credibility of classifier decisions. Thereby, the technique can be accompanied by a formal stability analysis. In the context of vectorial data, sparse conformal predictors have been recently discussed in [6], which we review now briefly.

Conformal prediction 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 $\mathbf{l}_{N+1}$, i.e. $\mathbf{z}_{N+1} := (\mathbf{v}_{N+1}, \mathbf{l}_{N+1})$. 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, the *conformal prediction* computes 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*[1] then

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

holds asymptotically for $N \to \infty$ for each distribution of $\mathbb{Z}$ [14].

To compute the conformal prediction region Γ^ϵ, a *non-conformity measure* is fixed $A(\mathcal{D}, \mathbf{z})$. It is used to calculate a non-conformity value μ that estimates how an observation $\mathbf{z}$ fits to given representative data $\mathcal{D}=\{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$. The conformal algorithm for classification is as follows: given a non-conformity measure A, significance level ϵ, examples $\mathbf{z}_1, \ldots, \mathbf{z}_N$, 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.

For given $\mathbf{z} = (\mathbf{x}, \mathbf{l})$ and a trained relational prototype-based model, we choose as non-conformity measure

$$\mu := \frac{d^+(\mathbf{x})}{d^-(\mathbf{x})} \tag{2}$$

[1] *exchangeability* is a weaker condition than data being i.i.d. which is readily applicable to the online setting as well, for example [14]

Algorithm 1 Conformal Prediction (CP)

```
1: function CP(𝒟, v_{N+1}, ϵ)
2:     for all l ∈ 𝕃 do
3:         z_{N+1} := (v_{N+1}, l)
4:         for i = 1, ..., N + 1 do
5:             𝒟_i := {z_1, ..., z_{N+1}}\{z_i}
6:             μ_i := A(𝒟_i, z_i)                          ▷ eq. 2
7:         end for
8:         r_l := |{i=1,...,N+1 | μ_i ≥ μ_{N+1}}| / (N+1)
9:     end for
10:    return Γ^ϵ := {l : r_l > ϵ}
11: end function
```

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. (1).

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 $\mathbb{L}$. 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 $\mathbf{v}_i$ two error rates are of particular interest, ϵ_1^i being the smallest ϵ and ϵ_2^i being the greatest ϵ so that $|\Gamma^\epsilon(\mathcal{D}, \mathbf{v}_i)| = 1$. 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

$$\text{confidence} : cf_i := 1 - \epsilon_1^i = 1 - r_{\mathbf{l}_{\text{2nd}}} \tag{3}$$

$$\text{credibility} : cr_i := \epsilon_2^i = r_{\mathbf{l}_{\text{1st}}} \tag{4}$$

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 (un)typical and not an outlier.

2.3 Semi-supervised Conformal RPC

In semi-supervised learning unlabeled data are used to enhance the learned model based on only labeled data (denoted as T_1). A very naive approach is

so-called *self-training*, which takes iteratively a part of the unlabeled data (denoted as T_2) as new training data into the retraining process until all labeled data are considered [15]. The problem of self-training is how to determine the labels of the unlabeled data which will be taken into the retraining, a simple idea is using *k-NN*, i.e. label the k nearest unlabeled data by the trained model and the predicted labels serve as 'true' labels of the unlabeled data in the retraining. For safety normally small k is used to avoid the degeneration of the learning performance, which can also cause very high computational effort for large data.

In order to get over this problem we combine the self-training approach with conformal prediction. First of all, to identify the unlabeled data with high confidence and credibility values defined by cc_i. For a given data $\mathbf{v}_i \in T_2$,

$$cc_i := cf_i \times cr_i \tag{5}$$

High *cc*-values of unlabeled data indicate that with high probability their predicted labels are the true underlying labels. That means only the unlabeled data with predicted labels of high probability will be taken into the next retraining. The region which consists of these unlabeled data with high cc_i is referred as '*secure region*' (denoted as $\mathcal{SR}$). Therefrom to identify $\mathcal{SR}$ we take a fraction (prc) of the top *cc*-values of the unlabeled data[2].

On the other hand in the retraining the '*insecure region*' ($\mathcal{ISR}$) of the training data can be found by

$$\mathcal{ISR} := \left\{ \mathbf{v}_i \in T_1 : cf_i \leq \left(1 - \frac{1}{L}\right) \vee cr_i \leq \frac{1}{L} \right\}. \tag{6}$$

and represented by a new prototype as the median of $\mathcal{ISR}$.This step automatically adapts the complexity of the model, i.e. the number of prototypes. For the next retraining this new prototype will be also trained with the new training data. The proposed method is referred to as *secure semi-supervised conformal relational prototype-based classifier* (SSC-RPC). See algorithm 2.

During the self-training process the training set T_1 is expanded by adding the secure region $\mathcal{SR}$ of unlabeled data to itself while the unlabeled data T_2 is shrunk by discarding its secure region $\mathcal{SR}$. The performance of the retaining is evaluated based on only labeled data. The method terminates if the improvement of the performance is not significant (less than 1%) after a given number of iterations ($win_{\text{max\_itr}}$) or the maximal iterations are reached (max_{itr}) or the insecure region ($\mathcal{ISR}$) is too small or the unlabeled set T_2 is empty. Since the size of $\mathcal{ISR}$ controls the complexity of the model, we found by some independent experiments, that $|\mathcal{ISR}| \leq 5$ is a good compromise between too dense or too sparse models.

[2] prc is customizable and in our experiments we set $prc = 5\%$ which is a good compromise between learning performance and efficiency.

Algorithm 2 secure semi-supervised conformal RPC

init: $W := \emptyset$, $W_{\text{new}} := \emptyset$, $W_{\text{best}} := \emptyset$, $\mathcal{ISR} := \emptyset$; $\mathcal{SR} := \emptyset$
$T_1 :=$ labeled data; $T_2 :=$ unlabeled data
$improve = 1\%$ ▷ threshold of improvement: default 1%
$EvalSet = T_1$ ▷ Evaluation set, i.e. labeled data
$itr = 0$ ▷ iteration counter
$ctn_{\text{best}} = 0$ ▷ counter for best result
$max_{\text{itr}} = 100$ ▷ maximal total iterations
$win_{\text{max\_itr}} = 10$ ▷ maximal iterations for a result as winner
$acc_{\text{best}} = 0$
repeat ▷ self-training process
  $W := W \bigcup W_{\text{new}}$
  $T_1 := T_1 \cup \mathcal{SR}$, $T_2 := T_2 \backslash \mathcal{SR}$
  $W :=$ train T_1 by RPC given W ▷ training with given prototypes
  $acc :=$ evaluation of W on $EvalSet$;
  if $acc - acc_{\text{best}} \geq improve$ **then**
    $W_{\text{best}} = W$, $acc_{\text{best}} = acc$, $ctn_{\text{best}} = 0$
  else
    $ctn_{\text{best}} = ctn_{\text{best}} + 1$
  end if
  $A_{T_1} := \{\mu_i, \forall i \in T1\}$ ▷ μ-values of T_1: eq. (2)
  $A_{T_2} := \{\mu_i, \forall i \in T2\}$
  $CF_{T_2} := \{cf_i, \forall i \in T_2\}$; $CR_{T_2} := \{cr_i, \forall i \in T_2\}$; ▷ eq. (3),(4)
  $CF_{T_1} := \{cf_i, \forall i \in T_1\}$; $CR_{T_1} := \{cr_i, \forall i \in T_1\}$;
  generate $\mathcal{ISR}$ of T_1 based on CF_{T_1} and CR_{T_1} ▷ eq. (6)
  generate $\mathcal{SR}$ of T_2 based on CF_{T_2} and CR_{T_2} ▷ eq. (5) and $prc = 5\%$
  generate W_{new} from $\mathcal{SR}$
  $itr = itr + 1$
until $|\mathcal{ISR}| \leq 5$ or $itr = max_{itr}$ or $ctn_{\text{best}} = win_{\text{max\_itr}}$ or $T_2 = \emptyset$
return W_{best};

3 Experiments

We compare SSC-RPC for SSL and RPC (trainded only on labeled data) on a large range of tasks including, five well-known UCI binary data sets[3], four SSL binary benchmark data sets[4], and two real life non-vectorial multi-class data sets from bioinformatic domain. Except for i.i.d. labeled data, we also demonstrate an artificial data set to show the ability of dealing with non i.i.d. labeled data of SSC-RPC. For vectorial data dissimilarity matrices D have been generated by using the squared-Euclidean distance. SSC-RPC has been initialized with one prototype per class, selected randomly from the labeled data set. In order to keep the comparisons fair we set the number of prototypes for each class for RPC to the number of prototypes for each class from SSC-RPC's final result.

Benchmarks and real life data sets

First we evaluate the methods on different UCI data sets, i.e. Diabetes(D1), German(D2), Haberman(D3), Voting(D4), WDBC(D5), and typical SSL benchmarks, i.e. Digit1(D6), USPS(D7), G241c(D8), COIL(D9) [1] [7]. For Digit1, USPS, G241c, COIL, the archive includes twelve data splits with 100 i.i.d. labeled data points. In oder to keep the same experimental setting, as for UCI

[3] http://archive.ics.uci.edu/ml/datasets.html
[4] http://www.kyb.tuebingen.mpg.de/ssl-book

Data	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
SSC-RPC	70.17	71.61	**73.30**	89.20	92.34	83.57	**79.47**	**73.64**	**59.24**	**81.06**	78.88
	(2.32)	(1.14)	(5.02)	(0.89)	(1.19)	(8.49)	(1.44)	(3.53)	(5.50)	(5.53)	(3.28)
RPC	70.00	71.44	70.27	89.20	92.29	83.55	78.25	72.31	57.00	79.37	78.78
	(2.20)	(1.30)	(7.29)	(0.90)	(1.64)	(8.62)	(2.43)	(5.13)	(2.89)	(4.78)	(3.70)

Table 1: Classification results for different vectorial and non-vectorial data.

data sets (as well as for the real life data sets later on), we randomly select 100 examples of the data to be used as labeled examples, and use the remaining data as unlabeled data. The experiments are repeated for 12 times and the average test-set accuracy (on the unlabeled data) and standard deviation are reported.

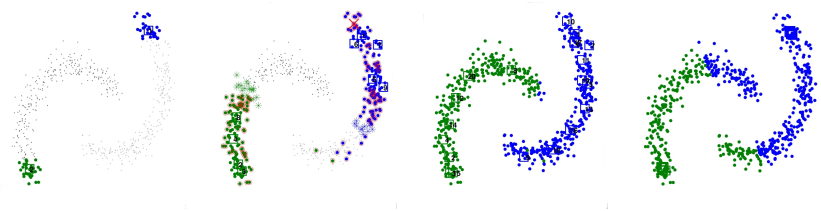
Further we evaluate the methods on two real life relational data sets, where no direct vector embedding exists and the data are given as (dis-)similarities. The *SwissProt* data set (D10) consists of $5,791$ samples of protein sequences in 10 classes taken as a subset from the popular SwissProt database of protein sequences [2] (release 37). The 10 most common classes such as Globin, Cytochrome b, etc. provided by the Prosite labeling. These sequences are compared using Smith-Waterman[4]. The *Copenhagen Chromosomes* data (D11) constitute a benchmark from cytogenetics [8]. 4,200 human chromosomes from 21 classes are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings can directly be compared using the edit distance based on the differences of the numbers and insertion/deletion costs 4.5 [8]. The classification problem is to label the data according to the chromosome type. The results are shown in Table 1. In half of all cases, semi-supervised learning improves the result, and in the remaining cases it never degenerates the learning performance, which is also an very important issue in SSL [12, 15].

Artificial data set: two banana-shaped data clouds

This data set contains two banana-shaped data clouds indicating two classes. Each banana consists of 300 2-D data points, Fig. 1(a). We select randomly non i.i.d. a small fraction (ca. 5%) of each banana as labeled data. RPC is trained only on labeled data with the same number of prototype for each class which SSC-RPC finally outcomes and can not learn the whole data space very well (see e.g. 1(d)). However, by means of $\mathcal{SR}$ of SSC-RPC the unlabeled data are considered iteratively by the self-training procedure. Figure 1(b), 1(c) shows some intermediate results up to convergence. The average accuracy (on unlabeled data) of 10 times randomly non i.i.d. selected labeled data is reported: SSC-RPC: **94.55%**(8.38), RPC: 77.29%(13.13).

4 Conclusions

We proposed an extension of conformal RPC for SSL by means of 'secure region' of unlabeled data to improve the classifier and 'insecure region' of labeled data to



(a) two bananas (b) 10. iteration (c) final SSC-RPC (d) final RPC

Fig. 1: (a) The data consist of green/blue labeled data and gray unlabeled data. Two prototypes are trained by only labeled data and marked with squares. (b) The secure region $\mathcal{SR}$ consists of the unlabeled data marked by stars and the insecure region $\mathcal{ISR}$ contains labeled data rounded by red circles. The new prototype taken from $\mathcal{ISR}$ is marked with a big red cross. During the self-training process additional prototypes are created. (c) the final result of SSC-RPC (d) the final result of RPC based only on labeled data

adapt the model complexity. It is a natural multi-class semi-supervised learner for vectorial and non-vectorial data sets. As a wrapper method it can also be integrated with other prototype-based methods. Our experiments show that the approach demonstrates in general superior results compared to standard RPC based on the labeled data alone, especially for non i.i.d. labeled data. Due to the lack of classical SSL benchmarks for non i.i.d. data, we will provide more detailed experiments for these relevant data in later work. Also additional parameter studies for SSC-RPC focusing on the *prc* parameter and sparsity aspects to address large scale problem will be addressed in the future.

Acknowledgments: Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded by the German Excellence Initiative is gratefully acknowledged. F.-M. Schleif was supported by the "German Sc. Found. (DFG)" (HA-2719/4-1).

References

1. O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
2. B. Boeckmann et. al. The swiss-prot protein knowledgebase and its supplement trembl in 2003,. *Nucleic Acids Research*, 31:365–370.
3. A. Gisbrecht, B. Mokbel, F.-M. Schleif, X. Zhu, and B. Hammer. Linear time relational prototype based learning. *J. of Neural Sys.*, 22(5):72–84, 2012.
4. Dan Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
5. B. Haasdonk and C. Bahlmann. Learning with distance substitution kernels. *Pattern Recognition - Proc. of the 26th DAGM Symposium*, 2004.
6. Mohamed Hebiri. Sparse conformal predictors. *Statistics and Computing*, 20(2):253–266, 2010.

7. Yu-Feng Li and Zhi-Hua Zhou. Towards making unlabeled data never hurt. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 1081–1088. Omnipress, 2011.
8. M. Neuhaus and H. Bunke. Edit distance based kernel functions for structural pattern classification. *Pattern Recognition*, 39(10):1852–1863, 2006.
9. O. Rajadell, P. Garcia-Sevilla, V.C. Dinh, and R.P.W. Duin. Semi-supervised hyperspectral pixel classification using interactive labeling. In *WHISPERS, 2011 3rd Workshop on*, pages 1–4, june 2011.
10. Atsushi Sato and Keiji Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *NIPS*, pages 423–429. MIT Press, 1995.
11. F.-M. Schleif, X. Zhu, and B. Hammer. A conformal classifier for dissimilarity data. In *Proceedings of AIAI 2012*, pages 234–243, 2012.
12. Aarti Singh, Robert D. Nowak, and Xiaojin Zhu. Unlabeled data: Now it helps, now it doesn't. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*, pages 1513–1520. Curran Associates, Inc., 2008.
13. Michael W. Trosset, Carey E. Priebe, Youngser Park, and Michael I. Miller. Semisupervised learning from dissimilarity data. *Computational Statistics and Data Analysis*, 52(10):4643 – 4657, 2008.
14. V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005.
15. Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. *Synthesis Lectures on Artif. Intell. and Machine Learning*, 3(1):1–130, 2009.