RBF Networks for Learning from Partially Labeled Data

Abdelhamid Bouchachia HAMID@ISYS.UNI-KLU.AC.AT
Department of Informatics, University of Klagenfurt, Austria

Abstract

This paper investigates the problem of learning from labeled and unlabeled data. A neural approach that relies on radial basis function neural networks (RBFNs) is proposed. This latter is based on a supervised learning rule and to adapt it for learning unlabeled data, first, labeled data is applied to generate some labeled clusters by means of a fully supervised clustering algorithm. Then, these clusters are used to (partly) assign labels to the unlabeled data. For this purpose, three methods are explored. The first applies Fuzzy C-means (FCM) to estimate the class of the unlabeled data, the second uses a general distance measure, and the third is a combination of the two methods. The prototypes resulting from the supervised and eventually refined by FCM are used as centers of the radial basis functions of the network. The training data consists of known labeled data and the unlabeled data whose labels have been estimated. The numerical evaluation, conducted on two data sets, has shown how unlabeled data can help enhancing the accuracy of the neural classifier and that this latter outperforms other semi-supervised classifiers.

1. Introduction

The combination of labeled and unlabeled data to train a classifier has recently gained much attention from the research community. The review of the literature shows that there are several methods to approach this problem. Some of them are:

- Seeding and constrained K-Means (Basu et al., 2002), (Bensaid & Bezdek, 1996)
- Pre-labeling (Amini & Gallinari, 2003), (Nigam et al., 2000)
- Co-training and Ensemble learning (Blum & Mitchell, 1998), (Ghani, 2002), (Nigam & Ghani, 2000)
- Active learning (Abe & Mamitsuka, 1998), (Klein et al., 2002), (Warmuth et al., 2003),
- Objective function optimization (Bouchachia & Pedrycz, 2003),, (Demiriz et al., 2002), (Pedrycz & Waletzky, 1997)

Of course some hybrid methods fall in more than one class. This set of methods involves various computational models but it seems that a large body of the work in this domain of learning with partial supervision relies on probabilistic model, especially the expectation-maximization technique and its variants, compared with other machine learning paradigms like genetic algorithms, support vector machines, and neural networks.

In this work, we investigate the application of a neural approach to deal with situation where data is only partially labeled. Basically, we will use radial basis function neural networks. This type of neural networks is fully supervised and therefore, the unlabeled data is not directly used. We follow the second class of methods which is pre-labeling. However, we do not exclusively rely on pre-labeling but we also use the seeding approach as will be explained in Section 3. From a general view, the scheme applied here is to use the labeled samples to guide the classification process and to boost its accuracy using the unlabeled data. We will show, via three pre-labeling methods based on *supervised clustering*, how using unlabeled data helps enhancing the accuracy of the classification of real world data using radial basis function networks. Furthermore, a comparative study is conducted using two other methods, namely the Seeding-based (Basu et al., 2002) and Expectation-maximization (Nigam et al., 2000) methods.

The rest of this paper is organized as follows. Section 2 explains briefly radial basis functions networks. Section 3 discusses the way RBF networks are fitted to the problem of learning with partial supervision. In particular, this section introduces the fully supervised algorithm used to generate the labeled clusters and the three methods used to estimate the labels of the unlabeled data. The evaluation of

Appearing in *Proc. of the 22^{st} ICML Workshop on Learning with Partially Classified Training Data*, Bonn, Germany, August 2005.

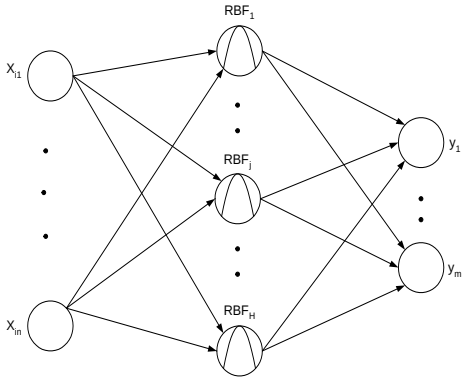

Figure 1. Architecture of an RBF neural network

the approach on two data sets is presented in section 4. A comparative study is developed in section. 4.1. Finally, section 5 concludes the paper.

2. Radial Basis Function Neural Networks

Inspired by research in regions of the cerebral cortex and the visual cortex, RBF networks have been proposed by Moody and Darken (1989) as supervised learning neural networks. A RBF network is a two-layer architecture where each unit in the hidden layer represents a radial basis function (see Fig. 1). These units measure the degree of overlap (or matching) between input vectors and a set of prototypes drawn from the training set.

A RBFN is a mapping $M : R^n \rightarrow R^m$ such that each input vector $x_i \in R^n$ is of dimension n and vectors $C_j \in R^n$ ($j = 1..H$) representing the prototypes of the input vectors. The output space of the mapping is of *m*-dimensions (i.e., size of the output vectors). The output of each RBF unit (called also receptive field) is given as:

$$\phi_j(x_i) = \phi_j(||x_i - C_j||) \tag{1}$$

where $||.||$ is the Euclidean norm on the input space to compute the distance between the n-dimensional input i and a hidden unit j. The function ϕ has various forms. Here, the Gaussian function is considered. Therefore, ϕ has the following form:

$$\phi_j(x_i) = exp(-\frac{||x_i - C_j||^2}{\sigma_j^2}) \tag{2}$$

where σ_j is the width of the jth RBF unit. Note that if $x_i = C_j$, $\phi_j(x_i) = 1$ yielding maximum overlap.

The kth output, $y_j(x_i)$, of a RBF network according to the weighted sum option is:

$$y_k(x_i) = \sum_{j=0}^{H} \phi_j(x_i) \cdot w(k, j) \tag{3}$$

where $\phi_0() = 1$, $w(k, j)$ is the weight of the jth receptive field to the kth output and $w(k, 0)$ is the bias of the kth output.

The key problem in RBF networks is the design of the parameters of the receptive fields: the prototypes C_j and the widths σ_j. Generally, prototypes representing the subregions (or classes) of the input space are found using clustering algorithms. It is important, however, to notice that these algorithms determine clusters independently of any semantical information about the real classes of the input. In this work, the prototypes are determined via a fully supervised clustering algorithm. These are directly used or refined using the FCM algorithm depending on the method used as will be explained in Section 3. Here, a class can be represented by more than one cluster. Using the centers found, the widths, which are the radii of the Gaussian basis functions, are searched. Radii should be set so that the Gaussian from one center overlaps with near centers to a certain extent to ensure smoothness across the input space. The other possibility is to apply isotropic Gaussian functions whose width is fixed according the spread of the centers. In this work, the width is computed by considering the maximal spread of data points within the desired region.

During the training stage, for each data point x_i, $y_k(x_i)$ is computed. This can be expressed in a matrix form as:

$$Y = \Phi W \tag{4}$$

The goal of the training stage is to find the weight W. This can be done in two ways; either through a repetitive adjustment of the weight using the delta learning rule or by computing W directly, i.e.

$$W = \Phi^{-1} Y \tag{5}$$

provided that Φ is nonsingular. To avoid the singularity problem, a small value λ is added to the diagonal terms, i.e., if we let $\varphi = \Phi + \lambda I$, then:

$$W = \varphi^{-1} Y \tag{6}$$

I is the identity matrix.

The direct computation is easier and provides instantaneous training of the network.

3. RBF for Learning with Partial Supervision

As mentioned earlier, the approach that we are interested in, is the pre-labeling approach based on seeding. In general, according to the pre-labeling approach, the set of labeled data points are used to design a first version of the classifier. This latter is then used to estimate the label of the

unlabeled data. The final classifier is then constructed using both data, the originally labeled and the newly labeled via the previous version of the classifier. This approach is applied in several research works (Amini & Gallinari, 2003), (Blum & Mitchell, 1998), (Nigam et al., 2000).

In this paper, a different method is applied. We still use the labeled data to estimate the classes of the unlabeled data, but we refrain from performing the estimation of labels using a classifier built via the given labeled data. We rather estimate the labels relying on clustering techniques before training the classifier which is in this case a RBF network using both sets of data points.

The investigated method, as graphically portrayed in Fig. 2, consists of three steps:

- Clustering with full supervision
- Label estimation for the unlabeled data
- Training and testing the neural network

To perform the first step, we designed a supervised clustering algorithm that performs a partitioning of labeled data. In the second step, the labels of the unlabeled data are estimated. To achieve this goal, we will apply three methods as will be explained later. The last step is concerned with training and testing the network. Here, the given labeled, pre-labeled data (whose labels are estimated), and the clusters' prototypes resulting either from the second step (or eventually from the first step) are used as input to the network. The first two types of input are used to train the net and the last is used as the center of the radial basis functions corresponding to the neurones of the hidden layer of the RBF network.

In the following, the details of the first two steps are given.

3.1. Clustering with full Supervision

Let $X = [x]_{kp}$ $k = 1..N,\ p = 1..n$ be a set of data points to be classified, where N is the size of X and n is the dimensionality of data. This data consists of two sets: X^l of size N^l and X^u of size N^u where the former designates the labeled data and the latter the unlabeled data such that $X = X^l \cup X^u$.

The labeled data X^l is used to get some initial data representatives which are actually the prototypes of some clusters. Naturally, a class can consist of many clusters. Therefore, the class distribution will be completely covered. If the data points of some class are separated by some data points of another class, clusters will be generated systematically without any topographic constraint. We aim, through this algorithm, at generating pure clusters independently

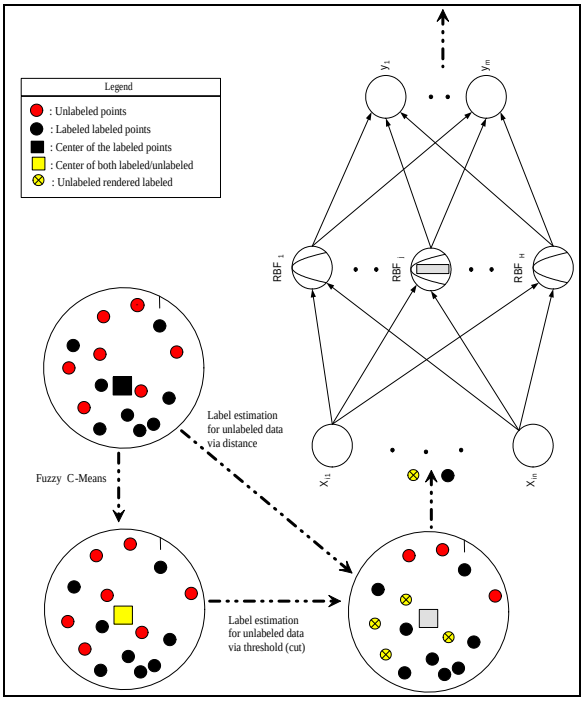

Figure 2. Process of training the RBF net

of the distribution of classes in the space. The algorithm, which is one-pass algorithm, is described as follows:

1. First data point x_1 is assigned to the first cluster whose prototype, M_1, is that data point.

2. Initialize a value σ that indicates the maximal distance between a data point x_i and the prototype M_j of its cluster C_j (it indicates the spread of the cluster). The larger σ, the larger will be the clusters. Therefore, the value of σ has to be reasonable so that the size and the number of clusters get also reasonable. Note that the number of clusters corresponding to the number of radial basis functions is usually chosen as 50%-60% of the size of the training data. Therefore, σ is chosen to get such a number of clusters.

3. For each next data point x_i do:

   (a) Compute the distance $d(x_i, M_j)$ between the data point at hand and each cluster of the same class, hence the supervision aspect of this algorithm.

   (b) Retain the computed distance d and the index q of the cluster that allows for the smallest distance to that data point.

   (c) If $d \leq \sigma$, x_i is assigned to cluster q

   (d) If $d > \sigma$, a new cluster is created.

(e) Recompute the prototype of the cluster to which the new point x_i is assigned as follows:

$$M_q = \frac{1}{N^q} \sum_{x_i \in C_q} x_i \tag{7}$$

where N^q is the current number of data points belonging to cluster q. (Note that M_q is an n-dimensional vector).

It is important to stress again that the number of centers is not equal to the number of classes and each cluster has a known label. For the neural representation of data, a class is represented by more than one radial basis function. The number of centers M_j resulting from this step will be used in the next step to (partly) label the unlabeled data.

3.2. Labeling the Unlabeled Data

To assign a label to the unlabeled data points, three methods are applied:

3.2.1. Prototypicality-based Method

According to this method, the centers M_j are used as seeds to cluster the whole data, labeled and unlabeled, using the Fuzzy C-Means algorithm (FCM). The idea here is that the prototypes to be generated are guided by the labeled data in their move towards an optimal position within the whole data. The FCM algorithm (Bezdek, 1981) is performed through the alternating calculation of the membership values of data points $\mu_{ji}, i = 1..N$ and the cluster centroids $v_j,\ i = 1..C$. The cluster centroids are calculated by

$$v_j = \frac{\sum_{i=1}^{N} \mu_{ji}^m x_i}{\sum_{i=1}^{N} \mu_{ji}^m} \tag{8}$$

while the partition matrix that represents the membership values is computed by:

$$\mu_{ji} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ji}}{d_{ki}}\right)^{2/(m-1)}} \tag{9}$$

where the superscript m indicates the fuzziness degree. The higher m, the fuzzier the clusters' borders. d_{ij} is the Euclidean distance between the data point x_i and the prototype of the cluster j.

After running FCM of the whole data, the initial centers M_j are refined and transformed into v_j. Now, clusters contain labeled and unlabeled data. Therefore, the labels of the most prototypical unlabeled data can be estimated. This is done by assigning the label of the cluster to which the data point strongly belongs. This is computed by determining the winning cluster:

$$L(x_i) = \arg\max_j(\mu_{ji}), \quad j = 1, ..., C \tag{10}$$

where x_i is a data point, L indicates the class label, and μ is the membership function that expresses the degree of belongingness of data points to clusters standing for a fuzzy set. Note that if a data point belongs with the same strength to two or more clusters having different labels, this data point will not be labeled. The resulting estimated labeled data is X^e.

Therefore, the data set used to train the RBF net will consist of the labeled data and the unlabeled data for which a label has been estimated, hence, $X^{training} = X^l \cup X^e$. The centers v_j computed by Eq. 8 will be used as prototypes for the hidden RBF units (see Eq. 2).

3.2.2. Distance based Method

According to this method, the labels of the unlabeled data are estimated directly by using a distance measure. In this paper, we use a more general distance investigated by Gustafson and Kessel (1979) and which is defined as follows:

$$d^2(x_i, v_j) = (x_i - v_j)^T \Sigma_j (x_i - v_j) \tag{11}$$

where Σ_j is the norm-inducing matrix of the cluster j. By setting Σ_j to the inverse of the covariance matrix associated with the cluster j, the distance becomes the Mahalanobis distance.

Once, the distance of the unlabeled data to the clusters generated by the supervised clustering algorithm is computed, those points for which the distance is lower than a given threshold are selected and labeled accordingly. The goal is to consider only the unlabeled data that falls in the nearest neighborhood of the known prototypes representing the labeled data. Here also, only a subset of the unlabeled data is assigned a label $X^e \subset X^u$. Therefore, the training data will consist only of the given and the pre-labeled X^e. The centers M_j computed by means of Eq. 7 are used as prototypes for the hidden RBF units.

3.2.3. Combined Method

This method is actually a combination of the distance-based method and the prototypicality-based method. Initially, the fully supervised clustering algorithm is performed producing some prototypes. These prototypes are used as seed to the FCM algorithm to cluster both labeled and unlabeled data. A preliminary label for the entire unlabeled data is assigned by computing the the maximum membership value of each data point to all clusters. Afterwards, the distance-based method is applied. Of course, some points which have been considered to be candidates for labeling will be discarded. The second step can be seen as a filter that allows to label a subset of the unlabeled data points. Once selected, these data points and the given labeled data points are used to train the neural network.

4. Numerical Evaluation

To evaluate the approach presented here, two data sets are used: the cancer and the wine data set (Hettich et al., 1998). The cancer data consists of 683 instances with 9 features, while the wine data set consists of 178 instances with 13 features. Both data sets are splitted into three parts: training set, testing set, and the unlabeled set used. First, an amount of data is randomly selected to be considered as the labeled set. In our experimental setup, this set varies from 2%, 4%, 6%, 8%, 10%, 25%, 40%, 55%, and 70%. Once the amount of labeled data is determined, the remaining part of the data is dynamically and randomly divided into two parts: used and temporarily not used. Because a 10-cross validation split is applied to test the classifier, the first part will be used for training and testing the classifier. This part is variable depending on the amount of labeled and unlabeled data that we are interested in. In fact, to examine the effect of increasing the unlabeled data, we will vary the amount of unlabeled (to be effectively used) from 0%, 10%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, to 100%.

We are primarily interested in evaluating four aspects:

- The effect of the unlabeled data on the accuracy of the classifier. This is done by varying the amount of unlabeled data.
- The effect of the labeled data on the classification accuracy by increasing the amount of labeled data.
- The effectiveness of each of the pre-labeling methods
- Comparing the proposed algorithm against other approaches (see Sec. 4.1).

It is worth mentioning that in our data split strategy, we preserve the uniformity, so that no class is omitted during the training phase. All classes are represented by some samples.

The results obtained on the wine data set are plotted in Figs. 3, 5, 6, 9, and 10. Those related to the cancer data set are plotted in Figs. 4, 7, 8, 11, 12, and 13. Each of the figures shows the percentage of labeled and unlabeled data used, and the evolution of the accuracy when increasing the amount of each of these. The dashed lines illustrate the performance ratio when the RBF network is trained on only the given number of labeled data. In addition, the effectiveness of the three proposed methods is illustrated.

Figure 3 and Fig. 4, related to the wine and the cancer data set respectively, show a typical evolution of the classification performance as the number of unlabeled data increases although the size of labeled data is very small lying in the range of [2%, 10%] of the whole data. The difference that

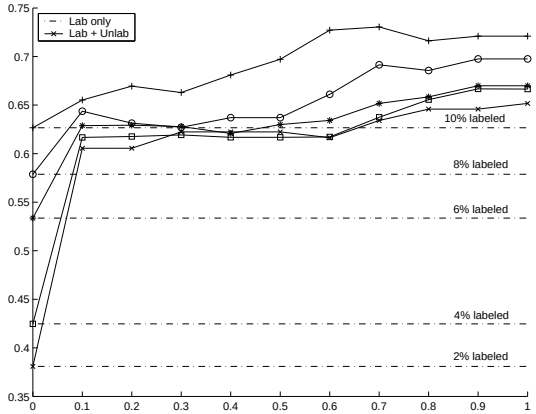

Figure 3. Prototypicality-based, wine data, labeled in [2%,10%], unlabeled in [0%,100%]

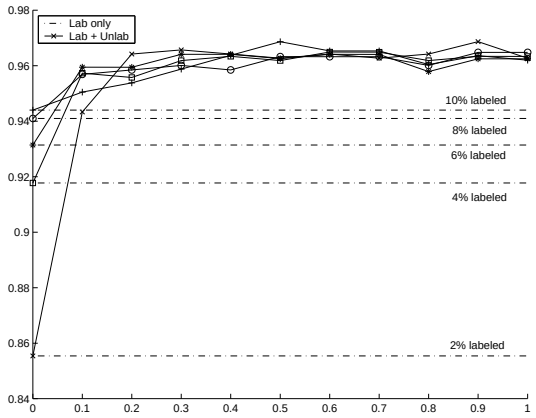

Figure 4. prototypicality-based method, cancer data, labeled in [2%,10%], unlabeled in [0%,100%]

can be noticed is that the level of accuracy is high when using the cancer data compared with the wine data. The highest performance value obtained on this latter is 73.04% obtained when the classifier is "boosted" with an amount of unlabeled data that is 70% of the available data which is equivalent to 102 data points and when 10% (equivalent to only 16 data points) of the labeled data is involved. The highest performance ratio for the cancer data set is 96.86% obtained when an amount of unlabeled data that is 50% of the available data which is equivalent to 277 data points and when 10% (equivalent to only 61 data points) of the labeled data is involved.

Furthermore, it is worth stressing that for the wine data set, the first pre-labeling method, that is the prototypicality-based method, has allowed to obtain the best results compared with the distance-based and the combined method when evaluated on the same range of labeled data (i.e., [2%, 10%]) as shown in Figs. 5, and 6. However, the combined method with a performance ratio of 69% outperforms

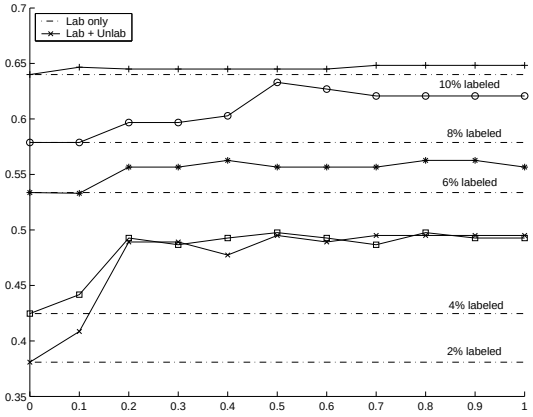

Figure 5. Distance-based method, wine data, labeled in [2%,10%], unlabeled in [0%,100%]

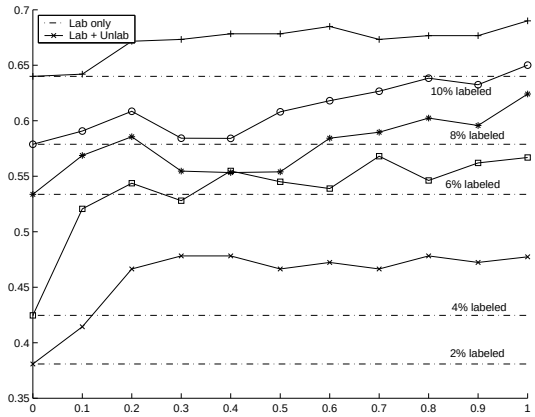

Figure 6. Combined method, wine data, labeled in [2%,10%], unlabeled in [0%,100%]

the distance-based method that achieves in the best case 65%. But, the successive application of the distance criterion after fulfilling the prototypicality condition has worsen the good results obtained using the prototypicality-based method only, given the difference between them in terms of accuracy level. More importantly, the highest improvement of accuracy is achieved when only 2% of the data is labeled. In fact, an improvement of 27% is obtained using the prototypicality-based method on only 2% of labeled data, while for the combined method, we obtained an improvement of 14% when using 4% of labeled data. The smallest improvement, 11%, has been achieved through the distance-based method on 2% of labeled data as shown in Fig. 5. These results show the over-performance of the prototypicality-based method.

Very similar results have been obtained on the cancer data set although the level of accuracy is higher approaching 97.31%. This value is achieved by the combined method. This latter has allowed to get the highest improvement of accuracy, namely 11.76%. However, close performance and performance improvement values are also obtained by the prototypicality-based and distance-based methods, 96.86%, 97.31% with a maximum improvement of 11.31% and 11.61% respectively (see Figs. 4, 7, 8). Again all these values result after using only 2% of the labeled data. In summary, the combined method, applied on the cancer data with an amount of labeled data lying in the range of [2%, 10%], produced the best results.

Now, let us examine the evolution of the classifier when the labeled data is in the interval [10%,70%] of the whole data. As illustrated in Figs. 9, 10, 11, 12, 13, the rhythm of the performance progress decreases slightly. For the wine data set, a ratio of 9.43% accuracy improvement is obtained when using the prototypicality-based method with 10% labeled data only. The highest classification performance,

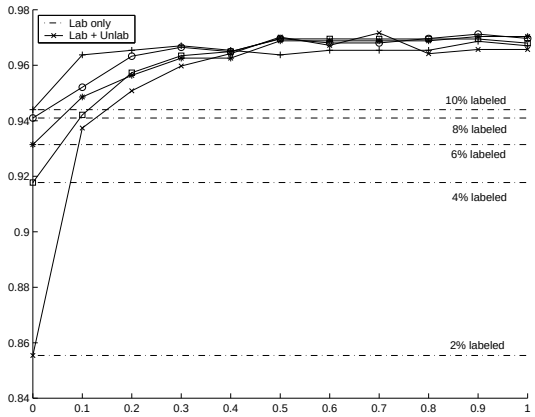

Figure 7. Distance-based method, cancer data, labeled in [2%,10%], unlabeled in [0%,100%]

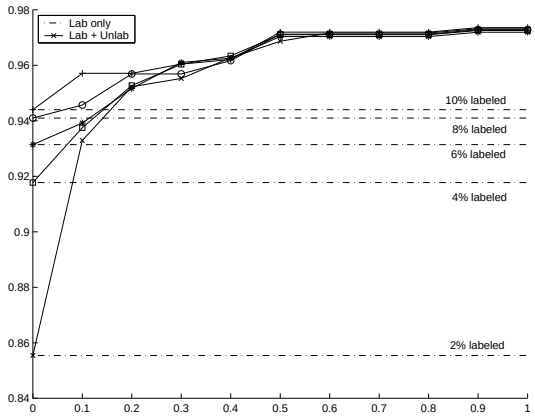

Figure 8. Combined method, cancer data, labeled in [2%,10%], unlabeled in [0%,100%]

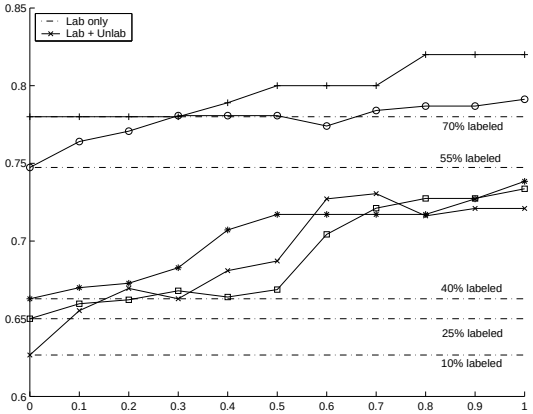

Figure 9. prototypicality-based method , wine data, labeled in [10%,70%], unlabeled in [0%,100%]

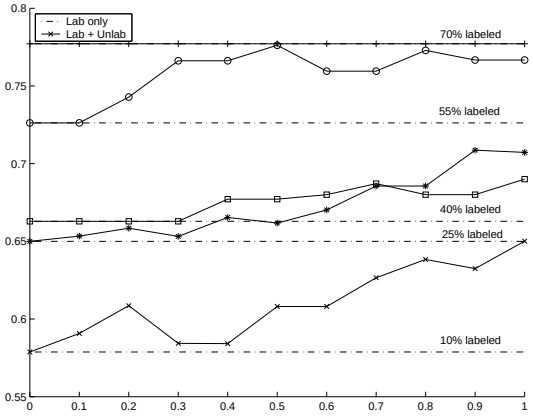

Figure 10. prototypicality-based method, wine data, labeled in [10%,70%], unlabeled in [0%,100%]

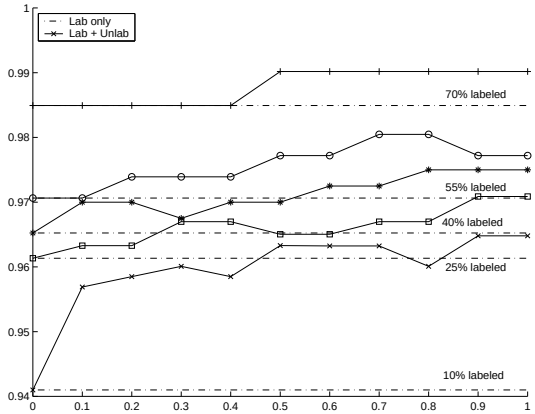

Figure 11. prototypicality-based method , cancer data, labeled in [10%,70%], unlabeled in [0%,100%]

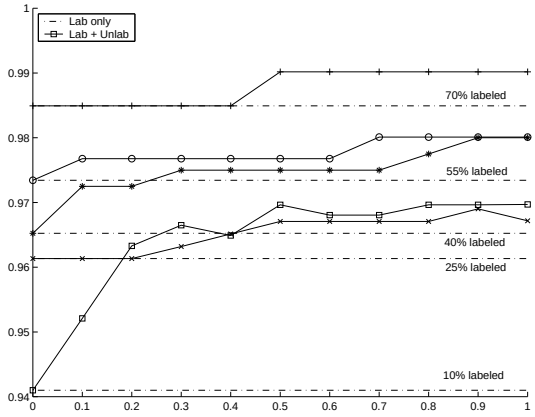

Figure 12. Distance-based, cancer data, labeled in [10%,70%], unlabeled in [0%,100%]

82%, is achieved when the amount of labeled data is set to 70% of the whole data using this method. The two remaining methods achieve 8.13% improvement with 10% labeled data and a maximum accuracy of 64.27% and 65% respectively. With the cancer data set, the maximum classification accuracy, which is 99.01%, is obtained by all methods when the amount of labeled data amounts to 70% of the whole data. The maximum accuracy improvement, which is 3.18%, is achieved by the combined method.

The set of experiments conducted have shown that the accuracy level of the RBF classifier can be boosted using unlabeled data. This is true until certain level. If the amount of labeled data that is available is big, then learning with unlabeled data does not contribute too much. The classifier, proposed in this paper, works quite well when only few labeled data is available. Furthermore, two of the prelabeling methods, prototypicality and combined methods, have shown better results compared with the distance-based method.

4.1. Comparative Study

In this section, we will compare the approach suggested here against two further methods: Seeded-Kmeans proposed in (Basu et al., 2002) and the basic expectation maximization method as proposed in (Nigam et al., 2000) using both data sets. The first algorithm uses the labeled data to generate some seed for each class. The unlabeled data is then assigned to clusters based on their similarity to the clusters' prototypes which are consequently updated. The second algorithm consists of two steps. First, an EM-classifier is built using only the labeled data. The basic assumption is that the data can be represented as a mixture of Gaussians. The characteristics - mean and variance- are then computed. During the second step of the algorithm, the unlabeled data is used to retrain the classifier as follows. In the E-step, the labels of the unlabeled data are estimated. Then, in the M-step, both the originally labeled data and the unlabeled data are used to re-adjust the characteristics of the Gaussians accordingly. This process is

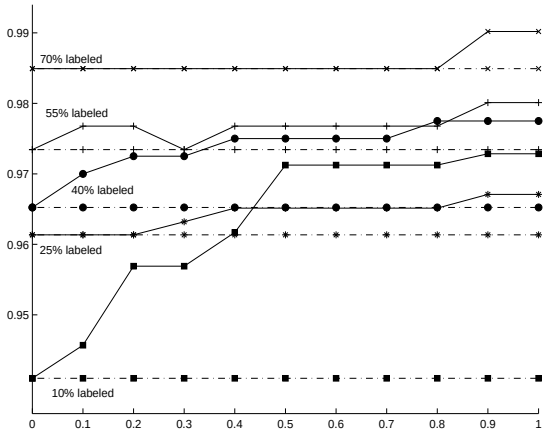

Figure 13. Combined method , cancer data, labeled in [10%,70%], unlabeled in [0%,100%]

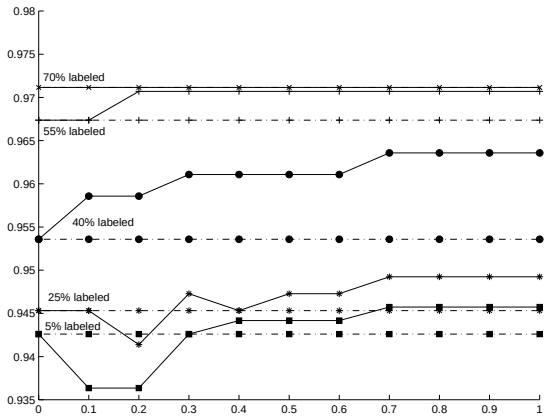

Figure 14. EM on cancer data

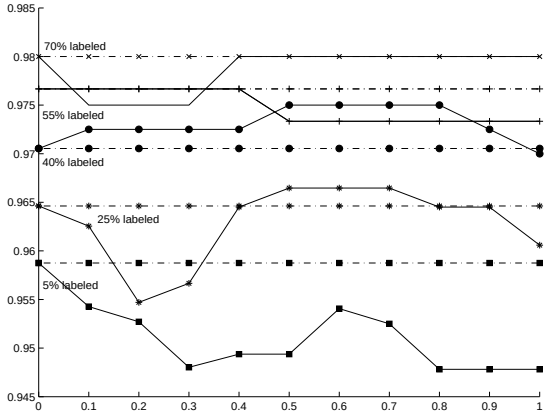

Figure 15. Seeded-Kmeans on cancer data

repeated until convergence.

Figures 13, 14, and 15 portray the results of the three algorithms, the neural-based algorithm of this paper, Seeded-Kmeans algorithm, and the EM algorithm. The plots show that the algorithm proposed in this paper has performed much better than the EM and Seeded-Kmeans algorithms. It is also noticeable that although the Seeded-Kmeans outperforms the EM in terms of general accuracy, this latter looks more stable. In fact, as the amount of unlabeled data increases, the performance of the EM algorithm improves. On the contrast, the Seeded-Kmeans algorithm provides higher accuracy as the amount of labeled data increases. For instance, for 5% labeled data, the accuracy is always higher than 95.03% while with the EM, the accuracy is always less than 94.52%. This observation can be generalized to other amounts of labeled data (see Fig. 14, and 15). Note that very similar results are obtained with the wine data set.

As to the proposed neural-based algorithm, an accuracy of 99.11% is achieved when 70% of the labeled data and 80% of the available unlabeled data are used. This accuracy has not been achieved by any of the other algorithms. In general, as labeled or unlabeled data increases, the accuracy level of the neural-based algorithm increases. In summary, the proposed method has shown to be better than the EM and seeding-based methods.

5. Conclusion

This paper is concerned with the problem of using both labeled and unlabeled data to train a radial basis function network. The usefulness and the contribution of unlabeled data is shown. Three methods are proposed and evaluated. Two of them have shown a better performance in the context of learning from hybrid data. Furthermore, the paper shows that neural networks, and in particular, radial basis function can work very well in this context. In fact, compared with seed-based and expectation-maximization methods, our approach has shown better performance.

As a future work, we are interested in comparing the approach discussed here with further methods like those based on support vector machines and genetic algorithms. So far, there is a large body of methods dealing with learning from labeled and unlabeled data but there is no comparative study. It is also interesting to check other types of neural networks in terms of adaptation and accuracy compared with the RBF networks.

References

Abe, N., & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. *Proc. of the* 15^{th} *International Conference on Machine Learning (ICML'98)* (p. 110).

Amini, M., & Gallinari, P. (2003). Semi-supervised learning with explicit misclassification modeling. *Proc. of the* 18^{th} *International Joint Conference on Artificial Intelligence (IJCAI'03)* (pp. 555–561).

Basu, S., Banerjee, A., & Mooney, R. (2002). Semi-supervised clustering by seeding. *Proc. of the* 18^{th} *International Conference on Machine Learning (ICML'02)* (pp. 19–26). Sydney, Australia.

Bensaid, A., & Bezdek, J. (1996). Partial supervision based on point-prototype clustering algorithms. *The* 4^{th} *European Congress on Intelligent Techniques and Soft Computing (EUFIT'96)* (pp. 1402–1406). Aachen, Germany.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabaled data with co-training. *Proc. of the* 11^{th} *Annual Conference on Computational Learning Theory)* (pp. 92–100).

Bouchachia, A., & Pedrycz, W. (2003). A semi-supervised clustering algorithm for data exploration. *Proc. of the International Fuzzy Systems Association World Congress (IFSA'03)* (pp. 328–337). Istanbul, Turkey.

Demiriz, A., Bennett, K., & Embrechts, M. (2002). A genetic algorithm approach for semi-supervised clustering. *Journal of Smart Engineering System Design, 4*, 35–44.

Ghani, R. (2002). Combining Labeled and Unlabeled Data for MultiClass Text Categorization. *Proceedings of the* 19^{th} *International Conference on Machine Learning*.

Gustafson, D., & Kessel, W. (1979). Fuzzy clustering with a fuzzy covariance matrix. *Proc. of the IEEE Conference on Decision and Control* (pp. 761–766).

Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Klein, S., Kamvar, S., & Manning, C. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. *Proc. of the* 18^{th} *International Conference on Machine Learning (ICML'02)*. Sydney, Australia.

Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation*, *1*, 284–294.

Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. *Proc. of the* 9^{th} *International Conference on Information and Knowledge Management* (pp. 86–93).

Nigam, K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using expectation-maximization. *Machine Learning, 39*, 103–134.

Pedrycz, W., & Waletzky, J. (1997). Fuzzy clustering with partial supervision. *IEEE Transaction on SMCB, 27*, 787–795.

Warmuth, M., Liao, J., Raetsch, G., Mathieson, M., Putta, S., & Lemmenk, C. (2003). Support vector machines for active learning in the drug discovery process. *Chemical Information Sciences, 42*, 667–673.