Self-training using a k-Nearest Neighbor as a base classifier reinforced by Support Vector Machines

M'bark Iggane
IRF – SIC
Faculty of Sciences, Ibn Zohr
University Agadir, MOROCCO

Abdelatif Ennaji
LITIS EA 4108
University of Rouen, France

Driss Mammass
IRF – SIC
Faculty of Sciences, Ibn Zohr
University Agadir, MOROCCO

Mostafa El Yassa
IRF – SIC
Faculty of Sciences, Ibn Zohr
University Agadir, MOROCCO

ABSTRACT
In supervised learning, algorithms infer a general prediction model based on previously labeled data. However, in many real-world machine learning problems, the number of labeled data is small, while the unlabeled data are abundant. Obviously, the reliability of the learned model depends essentially on the size of the training set (labeled data). Indeed, if the amount of labeled data is not high enough, the generalization errors of learned model may be important. In such situation, semi supervised learning algorithm may improve the generalization performance of this model by integrating unlabeled data in the learning process. One of the most classical methods of the semi-supervised learning is the self-training. An advantage of this method is that several traditional supervised learning algorithms are used to build the model in the self-training process.
In this paper, the *k-Nearest Neighbors* (k-NN) classifier was chosen in making decision during the self-training process. We also propose to reinforce self-training strategy by using a *Support vector machines* (SVM) classifier that can help the k-NN to label the unlabeled data.
Experimental results showed that Self-training based on k-NN and SVM can outperform the results with the Self-training based on k-NN classifier only. .

Keywords
Semi-supervised Learning, Self-training, k-NN, SVM

1. INTRODUCTION
In many real-world machine learning problems, the number of labeled data is small because the process of labeling data becomes very expensive or very difficult to achieve (eg labeling images or web pages), when the number of available data increase[7]. In such situation, we can label just a small part of the available data, and then try to exploit a huge amount of unlabeled data in the learning process. Therefore, the learning algorithm itself should find a way to take advantage of the unlabelled data.
Generally, the supervised learning algorithms have satisfactory performances when the number of labeled data is large enough. Otherwise the generalization performances of those algorithms decrease. In order to improve the learning algorithm's performance despite the lack of labeled data available, a new research theme has emerged in the recent years: *semi-supervised learning*.
The main idea of this approach is to take into account the information provided by the unlabeled data in addition to that provided by the labeled ones in order to build a reliable prediction model.
Among the possible approaches to this issue, one of the most natural is to adapt algorithms for supervised classification by allowing them to use information provided by the unlabelled examples. Include, for example, the use of the EM algorithm [6] using transductive inference algorithms [10], the use of two views of the examples [1; 6] and self-training [12].
In this paper, we propose to use the k-Nearest Neighbor classifier in making decision during the self-labeling process. We also propose to reinforce self-training strategy by using a Support vector machines (SVM) classifier that can help the k-NN to label the unlabeled data.
The remainder of this paper is organized as follows: In section 2, we describe a Self-training method, k-NN and SVM algorithm. We provide also the detail of our proposed procedure. The experimental resulted from our proposed method is presented, and compared with a single k-NN and Self-training on 5 datasets in section 3. Finally we conclude the paper in section 4.

2. SELF-TRAINING BASED ON K-NN AND SVM

2.1 Self-training algorithm
Self-Training is a classical technique used for semi-supervised learning [4]. The principle of this method is as follows:
In each iteration, the labeled data generate a classifier that assigns a class to each unlabelled data. The data that have the highest probabilities of classification are added to the labeled data with their predicted labels for the next iteration, and the process is then repeated until convergence.
The self-training procedure is described in Fig 1. It has been applied to several natural language processing tasks: for word sense disambiguation [4], for identifying subjective nouns [5], for classifying dialogues as 'emotional' or 'non-emotional'

[2], for object detection systems from images [3], for parsing and machine translation, etc [12].

Generally, the self-training technique gives good results in diverse applications. However, it is important to note that a classification mistake can reinforce itself, i.e. if we add a misclassified test example in a specific iteration, the subsequent iterations will also suffer and the classification accuracy of the learner goes down. In order to overcome this problem, several heuristics based essentially on the confidence level of the classifier used to make decision in self-training process have been proposed in the literature. For example, some algorithms try to avoid this by 'unlearn' unlabeled examples if the predictions confidence drops below a threshold [12].

Self-training has the advantage of working as a wrapper; in fact, it may employ any supervised classification algorithm in its core. Thus we are particularly focused on the strategy of using a second classifier with a high level of performance, in addition to the main classifier in the process of self-training. In our case we used Support Vector Machines (SVM). Thus, the final decision at each iteration of the main classifier will be taken with the help of the outputs of SVM classifier. To evaluate the effectiveness of this strategy, we have selected the k-NN classifier as the main classifier.

2.2 k-NN algorithm

The k-Nearest Neighbor (k-NN) is a simple but effective machine learning algorithm used for classification problems. It is a supervised learning algorithm where the result of new instance query is classified based on a majority of k-nearest neighbor category. The purpose of this algorithm is to classify a new object based on attributes and training samples. The classifiers do not use any model to fit and only based on memory. Given a query point, we find k number of objects or (training points) closest to the query point. The classification is using a majority vote among the classification of the k objects.

Here is a step by step description of the k-nearest neighbor k-NN algorithm:

1. Determine the parameter *k* = number of nearest neighbors.
2. Compute the distance between the query-instance and all the training samples.
3. Sort the distance and determine nearest neighbors based on the ***k***-th minimum distance.
4. Gather the label Y of the nearest neighbors.
5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance.

In the self-training process, the unlabelled data that are most confidently classified are added to the training data with, but the question how this confidence is measured, remains. The objective of the k-NN measures is to assign higher confidence to those data that are 'close' (i.e. with high similarity) to data of its predicted class, and are 'far' (i.e. low similarity) from data of a different class [13]. In this paper, the confidence measure for a k-NN classified data is calculated as follows:

Confidence(c)= k_i / k (c: class, k: the number of nearest neighbor and k_i : the number of data that are assigned to a class c within the k-nearset neighbors to the test data)

2.3 Support vector machines algorithm

The Support Vector Machine (SVM) is a classification technique based on statistical learning theory [9, 10, 11] that was applied with great success in many challenging non-linear classification problems processing large data sets.

The purpose of SVM during its training phase is to find the hyperplane that optimally separates the training set, so that new and unknown data points lie on the correct side of the hyperplane as much as possible.

A standard SVM classifier for two-class problem is defined as following:

Let consider a binary classification problem and a dataset $\{(x_1,y_1),(x_2,y_2),.....,(x_l,y_l)\}$ with $x_i \in R^d$ and $y_i \in \{-1,1\}$. In the feature space, the decision function given by an SVM is:

$$f(x) = sign\left[\omega'\phi(x)+b\right]$$

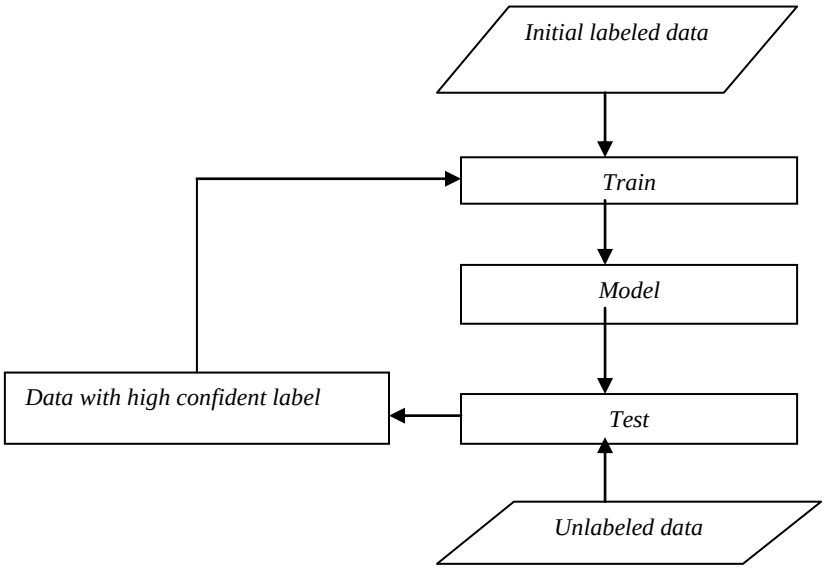

Fig 1: The procedure of self-training

Where ω is the weight vector, orthogonal to the hyperplane, "b" is a scalar that represents the margin of the hyperplane, "x" is the current tested sample, $\phi(x)$ is a function that transforms the input data into a higher dimensional feature space. Sign is the sign function.

ω and b are found by resolving the following optimization problem which expresses the maximization of the margin $1/\|\omega\|$ and the minimization of the training error :

$$\min_{\omega,b,\xi} \frac{1}{2}\omega'\omega + C\sum_{i=1}^{l}\xi_i$$

$$s.t: y_i\left[w'\phi(x)+b\right] \geq 1-\xi_i \quad \forall i = 1,.....,l$$

$$\xi_i \geq 0 \quad \forall i = 1,.....,l$$

2.4 The k-NN as a base classifier:

To describe the self-training based on k-NN, Let consider L and U denote the labeled and unlabeled data set. In standard self-training process, a learner keeps on choosing to label a small set of its most confident examples, from U and retraining itself on $L \bigcup U'$.

Given:

L: labeled training set

U: unlabeled set

While $(U \neq \varnothing)$

Allow ***k-NN*** to determine labels of ***U***

Determine ***U'***, a subset of ***U,*** whose elements are most confidently labeled

U = U \ U'

L = L + U'

End while

In detail, the self-training process starts by learning a hypothesis from the labeled data. In each self-training iteration, the main classifier selects unlabeled data on which it makes most confident prediction. Then it adds those examples and their predicted labels to the training set. It is important to note that we keep the same class distribution in the labeled training set L during the self-training process.
For instance, if there are 4 positive and 16 negative examples in L , then we will choose 1 positive and 4 negative examples to add to *L*.

2.5 The k-NN and SVM as base classifiers

Given:

L: labeled training set

U: unlabeled set

While $(U \neq \varnothing)$

Train ***SVM*** using ***L***

Allow ***k-NN*** to determine labels of ***U***

Allow ***SVM*** to determine labels of ***U***

Determine ***U'***, a subset of ***U,*** whose elements have the same class using *k-NN* and *SVM*

Determine ***V'***, a subset of ***U',*** whose elements are most confidently labeled by *SVM*

U = U \ V'

L = L + V'

End while

3. NUMERICAL RESULTS

In [8], a collection of state of the art algorithms for SSL are discussed by their authors and applied to a set of benchmark datasets. These datasets, including a detailed description of how they were obtained, are available at http://www.kyb.tuebingen.mpg.de/ssl-book. In order to test the performance of k-NN, Self-training based on k-NN and Self-training based on k-NN and SVM, five data sets are used. An overview of the five data sets can be found in Table 1.

Table 1. Datasets

Data Set	Points	Dimensions	Classes
g241c	1500	241	2
g241d	1500	241	2
Digit1	1500	241	2
USPS	1500	241	2
COIL$_2$	1500	241	2

The dataset g241c was generated by some processing of samples from a mixture of Gaussians. Digit1 contains images of '1' and the label is set according to the tilt of the digit in the image. USPS contains 150 images of each of the ten digits from the famous USPS set, with classes '2' U '5' vs. the rest. COIL is a processed version of the Columbia object image library (COIL-100), with 2 labels. All these data sets contain 1500 data points, each having 241 dimensions.
For each data set, 400 examples are kept aside to evaluate the performance of learned hypothesis, while the remaining 1100 examples are partitioned into labeled set (400 examples) and unlabeled set (700 examples).

We compare the classification accuracy of Self-training based on k-NN only and Self-training based on k-NN and SVM, with the classification accuracy of the k-NN algorithm (Table2).

Table 2: Performance of k-NN, Self-training based on k-NN and Self-training based on k-NN and SVM (averaged over 50 random realizations)

Data set	*k-NN*	Self-training based on *k-NN*	Self-training based on *k-NN* and *SVM*
g241c	65.36%	65.98%	68.20%
g241d	69.32%	69.12%	71.90%
Digit1	97.22%	97.51%	98.39%
USPS	94.45%	94.78%	96.35%
$COIL_2$	93.68%	96.09%	97.13%

The experimental results on 5 data sets show that Self-training based on k-NN benefit from the information provided by unlabeled examples, and perform better than the supervised k-NN algorithm.

However, due to the small size of the labeled data set (L), the generalization ability of the initial hypothesis may be poor. Consequently, the most confident examples set (L') may contain noise because of the fact that the learner may incorrectly assign labels to some unlabeled examples, and the generalization ability of the final hypothesis will be affected by the accumulation of such noise in each iteration of the training process (g241d). In this case a bad accuracy will be obtained at the end of the process because some data will be classified in the wrong class with high confident score. The introduction of SVM into the Self-training process improve the classification accuracy, so we remark that Self-training based on k-NN and SVM gave good result in comparison with Self-training based on k-NN only.

4. CONCLUSION

The Self-Training method proposed in this paper uses the k-NN classifier to make decisions during a self-training process. Experimental results on five datasets showed that Self-training based on k-NN can outperform the supervised k-NN. It presents significant gains on four datasets. There was one dataset in which statistically inferior results were obtained by this method: g241d.

The Self-training based on k-NN cannot guarantee that automatically generated labels are free of errors. Instead of using just k-NN into Self-training process, we propose to use SVM to help the main classifier (k-NN) in making decision.

As future work, we plan to use other methods to help the main classifier in making decision. We will also investigate the possibility of using more than two classifiers to make decision.

5. ACKNOWLEDGMENTS

This work is supported by a grant of Volubilis ‘Action intégrée Maroco-Française’ MA/2010/233

6. REFERENCES

[1] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual Conference on Computational Learning Theory, pages 92–100, Madison, WI, 1998. ACM Press

[2] B.Maeireizo, Litman, D., & Hwa, R. Co-training for predicting emotions with spoken dialogue data. The Companion Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), 2004.

[3] C. Rosenberg, et al., "Semi-supervised self-training of object detection models," Seventh IEEE Workshop on Applications of Computer Vision, 2005.

[4] D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In Meeting of the Association for Computational Linguistics, pages 189–196, 1995.

[5] E.Riloff, Wiebe, J., &Wilson, T. Learning subjective nouns using extraction pattern bootstrapping. Proceedings of the Seventh Conference on Natural Language Learning (CoNLL-2003).

[6] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2/3) :103–134, 2000.

[7] Ming Li, Zhi-hua Zhou - In: Proc. of the Advances in Knowledge Discovery and Data Mining (PAKDD 2005). LNAI 3518 , 2005

[8] O. Chapelle, B. Scholkopf, and A. Zien, editors. Semi-Supervised Learning. MIT Press, 2010.

[9] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the European Conference on Machine Learning. Springer, 1998.

[10] T. Joachims. Transductive inference for text classification using support vector machines. In Proceedings of ICML-99,16th International Conference on Machine Learning, pp. 200–209, Bled, SL. Morgan Kaufmann Publishers, San Francisco, US, 1999.

[11] Scholkopf Bernhard, Smola Alexander. Learning with Kernels, Support Vector Machine, MIT Press, London, 2002

[12] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2007.

[13] Wei Liu, Sanjay Chawla Class confidence weighted kNN algorithms for imbalanced data sets. Proceeding PAKDD'11 Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part II, pp. 345-356. 2011.