Self-Training with Combination of Three Different Support Vector Machines Classifiers

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—**Ensemble learning is the machine learning paradigm concerned with utilizing multiple base classifiers which trained and then combined to achieve a strong generalization. This technique can be beneficial to semi- supervised learning, which exploits unlabeled data in addition to the labeled data, to achieve the best possible classification performance. One of the most common methods of semi-supervised learning is Self-training in which one base classifier is used to make decisions during a learning process. Instead of using just one base classifier into self-training process, an ensemble made up of three Support Vector Machines (SVM) classifiers with different kernels, which is denoted by SELF3SVM, is used in this paper. The experimental results with real and artificial data demonstrate that combining three SVM classifiers into self-training process is often much more accurate than the standard Self-training based on just one base classifier.**

Keywords-Self-training; SVM; Ensemble learning; Semi-supervised learning

I. Introduction

The main goal of machine learning is to construct learners with a strong generalization ability. In the Ensemble Learning approach, this gaol can be achieved by training sets of learners whose decisions are combined.

Empirical studies **[**6, 19, 24**]** and theoretical explanations **[**9, 18, 7**]** have demonstrated that a set of classifiers very often attain higher performance than any of the classifiers in the set. Over the past two decades, many ensemble learning methods have been proposed, such as Boosting **[24]**, Bagging **[**12**]**, Stacking **[**4**]**, Random Forests **[13]** etc. Most of these methods are supervised ensembles which rely on the availability of large labeled data sets without considering unlabeled data [5, 15]. However, for many practical classification applications such as image analysis, and web pages classification, labeled data may be very scarce but unlabeled data is more readily available. The Semi-supervised learning tries to solve this problem by reducing the needed amount of labeled data and exploiting the unlabeled data. There are several different methods for Semi-Supervised learning based on different assumptions. Among these methods [16, 22] we can cite several categories such as generative methods [**11**], S3VMs (Semi-Supervised Support Vector Machines), graph-based methods, the Co-training [1] or the Self-training [23].

This paper focuses more specifically on the Self-training method in which one base classifier is used to make decisions during a learning process. The main idea of this method is first to train a base classifier on labeled data set. The classifier is then used to classify the unlabeled data set. After classification, the most confident examples are added to the labeled data set to form a new labeled data set for the next iteration. The process is then repeated for several times or until a stop condition is met. At each iteration, the most confidently labeled examples are supposed to be correct. However, the base classifier may erroneously label some unlabeled examples. Thus, it is possible that erroneous labels are introduced to labeled data set. These errors are often reinforced and will affect the base classifier on the subsequent iterations. Instead of using just one base classifier into self-training process, an ensemble made up of three *Support Vector Machines* (SVM) classifiers with different kernels, which is denoted by SELF3SVM, is proposed to overcome the problem encountered by the standard Self-training method.

This paper is organised as follows. Section 2 proposes an overview of Ensemble learning. Section 3 describes SELF3SVM procedure. Experimental results are presented and analyzed in Section 4. Finally, Section 5 concludes and suggests issues for future works.

II. ENSEMBLE LEARNING

A. *Why to combine multiple classifiers?*

An *ensemble of classifier* (***EoC***) is a set of classifiers whose individual decisions are combined to predict new examples. The main idea behind the ***EoC*** is to use several individual classifiers, and combine them in order to obtain a classifier that outperforms every one of them. In particular, it has been demonstrated through several research works that the ***EoC*** are an effective solution to many problems which confront the algorithms that induce single classifiers. Dietterich provides in [20] a categorization of these problems, in three types of limitation:

- Statistical: Learning algorithms may have a high variance when it can generate, in a given space H, several hypotheses that appear equally approximate the true classifier with respect to the available training data. However, they may have different generalization performance. By combining these hypotheses, it is possible to reduce the variance. Therefore, we might get a good approximation of the unknown true classifier.

- Representational: In many machine learning problems, the true classifier may not be correctly approximated in the space H of hypotheses. It is then possible to expand this space by adding a combination of hypotheses drawn from H. In this way, we can approximate the true classifier outside the space of hypotheses. Thus, it is possible that combining several hypotheses can approximate the true classifier more than any single hypothesis could do.

- Computational: Many learning algorithms apply local optimization techniques that may get stuck in local optima. In this situation, it is possible to reduce the risk of choosing the wrong classifier by combining multiple suboptimal classifiers - locally optimal - . So, a set of classifiers obtained by running several heuristic researches from different points in space H may provides a better approximation of the true classifier.

B. *Architecture of multiple classifier combination*

There are several different schemes to combine classifiers which have different interests. Generally, three approaches for combining classifiers can be considered: parallel approach, sequential approach and hybrid approach ([2, 20, 14]).

- Sequential approach: In the sequential combination, called also serial combination, two or more classifiers are arranged in a chain where the individual classifiers are evaluated in sequential order. Each classifier takes into account the prediction of the upstream classifier. Such an approach can be seen as progressive filtering of decisions. Generally, this will reduce the error rate overall the chain. Nevertheless, a combination of this type is particularly sensitive to the order in which the classifiers are placed. Indeed, even if they do not need to be the most efficient, the first classifiers in the chain must be robust.

- Parallel approach: In contrast to the sequential approach, the parallel organization of classifiers requires that each individual classifier produces an output simultaneously. All of these outputs are then fused using a combination operator, such as a simple majority vote, to produce a final decision. In this approach the order in which the classifiers are placed is not involved.

- Hybrid approach: The idea of the hybrid approach consists in combining the above two approaches in order to retain the advantages of both. Methods belonging to this category are generally designed for specific applications, as it is the case for example of the method proposed by Kim et al. in [8] for the recognition of cursive words extracted from bank checks.

Many studies show that the combination of classifiers (sequential, parallel or hybrid) improves significantly the performance of an EoC compared to each individual classifier. However, among these three approaches, the one which arouses the greatest interest of the scientific community is the parallel combination of classifiers.

C. *Combining rules*

Several combining rules can be applied to combine different classifiers [9, 10]. These rules are usually categorized into two classes, i.e., fixed and trained rules. In this paper, some well known simple fixed rules, such as the product rule, sum rule, min rule, max rule and vote rule will be summarized.

Thus, consider n individual classifiers $\left(C_j, j = 1, ..., n\right)$ that estimate a posteriori probabilities of m classes $\left(w_i, i = 1, ..., m\right)$. Each classifier C_j produces a real vector of de form

$$P_j = \left[p1j, p2j, ..., pmj \right] \qquad (1)$$

Where

$$p_{ij}(x) = p\left(w_i / x_j\right) \qquad (2)$$

(2) denotes the a posteriori probabilities that classifier C_j has that x belongs to class w_i . Generally, we can classify x by choosing the largest posterior probability:

$$\text{Assign } x \; to \; w_k \; if \; p\left(w_k / x_j\right) = \max_{i=1}^{m}\left(p\left(w_i / x_j\right)\right) \qquad (3)$$

The combined prediction from the different classifiers is done by:

- ***Maximum rule:***

$$\text{Assign } x \text{ to } w_j \text{ if } j = \arg\max_i \left(\max_{k=1}^{n} \left(p\left(w_i / x_k\right)\right)\right) \quad (4)$$

- ***Minimum rule:***

$$\text{Assign } x \text{ to } w_j \text{ if } j = \arg\max_i \left(\min_{k=1}^{n} \left(p\left(w_i / x_k\right)\right)\right) \quad (5)$$

- ***Product rule:***

$$\text{Assign } x \text{ to } w_j \text{ if } j = \arg\max_i \ p(w_i) \prod_{k=1}^{n} p\left(w_i / x_k\right) \quad (6)$$

- ***Median rule:***

$$\text{Assign } x \text{ to } w_j \text{ if } j = \arg\max_i \ \frac{1}{n} \sum_{k=1}^{n} p\left(w_i / x_k\right) \quad (7)$$

III. SELF3SVM METHOD

SELF3SVM is a procedure that uses an ensemble made up of three Support Vector Machines (SVM) classifiers with different kernels into Self-training process. Before presenting the description of SELF3SVM procedure, we first briefly present the Support Vector Machine (SVM) algorithm.

A. *Overview of Support Vector Machines*

The Support Vector Machine (SVM) is a method of supervised classification based on statistical learning theory [21]. The SVM aims to find the best hyperplane (ω, b) that optimally separates the data set into two classes. The classification of a new data point x is given by its position relative to the hyperplane, i.e the sign of:

$$\omega . x + b \quad (8)$$

This method was applied with great success in many non-linear classification problems. This is done by means of kernel functions. A standard SVM classifier for two-class problem is defined as following:

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

$$f(x) = sign\left[\omega' \phi(x) + b\right] \quad (9)$$

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, $\phi(.)$ defines the nature of nonlinear kernel that transforms the input data into a higher dimensional feature space. *Sign* is the sign function.
The optimization problem of SVM is formalized as follows:

$$\min_{\omega,b,\xi} \frac{1}{2}\omega'\omega + C\sum_{i=1}^{l}\xi_i$$
$$s.t : y_i\left[w'\phi(x)+b\right] \geq 1-\xi_i \quad \forall i = 1,.....,l \quad (10)$$
$$\xi_i \geq 0 \quad \forall i = 1,.....,l$$

Solving this problem, allows us to determine the value of ω and b .

For this experiment, we trained three SVM classifiers with three different kernels: Linear, polynomials and radial basis functions. Those kernels are mathematically defined as:

- Linear kernel:

$$K(x_i, x_j) = x_i^T . x_j \quad (11)$$

- RBF kernel:

$$K(x_i, x_j) = \exp(-\ g \left\| x_i - x_j \right\|^2) \quad (12)$$

- Polynomial kernel:

$$K(x_i, x_j) = [< x_i, x_j > + r]^d \quad (13)$$

The accuracy of an SVM model is largely dependent on the selection of the kernel parameters. Indeed, all these kernels share one common cost parameter C that controls the penalty degree for the classification error of training data. In addition to the parameter C, the RBF kernel has a second parameter g namely gamma.

B. *Description of SELF3SVM method*

In the learning process, we first train three SVM classifiers, using different kernels, on the same data learning. Then, they made separately predictions on the unlabeled data set. The outputs of those base classifiers are combined using five classifier combining rules: maximum, majority vote, median, mean and product fusing rule. The combination of the outputs from each of the classifiers produces a final result for each example. Then the examples that are classified with high confidence scores are added to the data learning set incrementally. Three models are retrained, and the whole process is iterated until convergence is achieved.

SELF3SVM Algorithm:

```
Given:
    Labeled training set: L,  Unlabeled set: U,  Test set: T
    Classifiers:  SVM1 (linear kernel),  SVM2 (Radial Basis kernel)
                  SVM3 (Polynomial kernel)
    Rules: R (Maximum, Product, Majority, Median, Mean)
While (U ≠ ∅)
    1. Train SVM1, SVM2 and SVM3 using L
    2. Allow SVM1 to determine labels of U (Outputs1)
    3. Allow SVM2 to determine labels of U (Outputs2)
    4. Allow SVM3 to determine labels of U (Outputs3)
    5. Produce F the set of final outputs (overall results):
    6.  F = Outputs1 R Outputs2 R Outputs3
    7. Determine F' a subset of F, whose elements are the most
confident
    8. L = L + F'
    9. U = U + F'
End while
```

IV. Experiments

Experiments are performed on 8 data sets listed in Table I. The first column lists the names of data sets, the second the size of data sets, the third the number of features and the last the number of classes in the problem.

for each data set, 25% data are randomly chosen to form the test set, while the remaining 75% data are partitioned into the labeled and unlabeled sets where 10%(of the 75%) are used as labeled examples while the remaining 90% (of the 75%) are used as unlabeled data.

TABLE I. Data set summary

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
Australian	690	15	2
Wdbc	569	31	2
Pima	768	9	2

For comparison, the standard Self-training, denoted by SELF, is run on the same labeled/unlabeled/test splits as those used for SELF3SVM procedure. In our experiments, we use the software LIBSVM to classify the data sets [3]. The specific cost parameter C and the other kernel parameters values are adjusted using grid search.

Table II, Table III, Table IV and Table V present the best accuracy of SELF using SVM(with different kernel: linear, RBF and Polynomial), and the accuracy of SELF3SVM applying maximum, mean, product, median and majority fusing rules on each data sets. For our experiments, we made ten different evaluation sets for each data set by random selection.

TABLE II. Classification accuracies on g241c and g241d

		Data sets	
		g241c	**g241d**
	Best SELF	87.00%(2.31)	84.36%(4.45)
SELF3SVM	Maximum	85.75%(4.62)	82.46%(1.26)
	Median	**88.25**%(1.43)	**84.99**%(3.95)
	Mean	**87.25**%(5.20)	**85.21%(**5.35**)**
	Majority	86.75%(0.71)	**85.10**%(4.37)
	Product	86.75%(3.32)	83.14%(5.52)

TABLE III. Classification accuracies on Dgit1 and Usps

		Data sets	
		Digit1	**USPS**
	Best SELF	96.49%(3.54)	92.25%(5.33)
SELF3SVM	Maximum	96.24%(0.54)	91.16%(4.23)
	Median	**97.12**%(2.75)	**93.12**%(1.36)
	Mean	**96.99**%(3.36)	**92.50**%(2.97)
	Majority	96.49%(6.05)	**92.50**%(3.76)
	Product	96.14%(4.43)	91.53%(2.81)

TABLE IV. Classification accuracies on Coil and Australian

		Data sets	
		COIL$_2$	**Australian**
	Best SELF	89.00%(3.67)	87.45%(0.34)
SELF3SVM	Maximum	87.75%(2.95)	**87.93**%(4.32)
	Median	89.00%(2.78)	**88.25**%(2.03)
	Mean	**89.25**%(1.45)	**88.50**%(3.55)
	Majority	**89.25**%(3.30)	86.12%(5.76)
	Product	88.00%(1.79)	87.45%(3.64)

TABLE V. Classification accuracies on Wdbc and Pima

		Data set	
		Wdbc	**Pima**
	Best SELF	92.80%(0.98)	75.66%(4.39)
SELF3SVM	Maximum	92.80%(3.28)	74.60%(3.26)
	Median	**93.52**%(3.05)	**77.24**%(2.20)
	Mean	**93.32**%(1.15)	**76.19**%(2.87)
	Majority	92.80%(1.00)	**75.75**%(1.56)
	Product	92.02%(1.98)	75.66%(2.25)

Comparing the experimental results in Table2 II, Table III , Table IV and Table V we can see that for most data sets, the SELF3SVM applying mean , majority or median fusing rule outperform the best SELF while product and maximum fusing rule perform less. Indeed, the best SELF3SVM applying mean or median fusing rule achieves improvement in the range of 0.25% to 1.58% over the best SELF. We also not that the product and maximum fusing rule never exceeds the best performance obtained by the best SELF. For instance, the Best SELF running on g241c data set achieves a classification accuracy of 87.00% and outperforms the SELF3SVM using product fusing rule by 0.25%.

In general, the maximum and product rules even perform worse than some single classifiers because of their sensitivity to noise as it is stated in [17].

V. Conclusion

In this paper, the SELF3SVM procedure which uses three SVM into the Self-training process is proposed. These three SVM are combined using five combining rules: maximum, mean, median, product and majority vote. Based on our experiments results, we show that SELF3SVM using median or mean fusing rule may outperform the performance of SELF based on just one SVM classifier. We also show that the maximum and product fusing rules never exceeds the best performance obtained by the best SELF.

As future work, we plan to use some trained combining rules, such as Dempster-Shafer method. We also intend to use more than three classifiers in the Self-training process to enhance the performance accuracy.

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.

[2] A. Rahman et M. Fairhurst. A Study of Some Multi-Expert Recognition Strategies for Industrial Applications : Issues of Processing Speed and Implementability. International Conference on Vision Interface, 1999.

[3] C.-C.Chang,C.-J.Lin,LIBSVM:a library for support vector machines. Available from World Wide Web: (http://www.csie.ntu.edu.tw/~cjlin/libsvm).

[4] D. H. Wolpert. Stacked generalization. Neural Networks 5(2):241–259, 1992.

[5] D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In M. Mozer, M. I. Jordan, and T. Petsche, editors, Advances in Neural Information Processing Systems 9, pages 571–577. MIT Press, Cambridge, MA, 1997.

[6] E. Bauer and R.. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning, 36(1/2):525–536, 1999.

[7] E.M. Kleinberg. A Mathematically Rigorous Foundation for Supervised Learning. In J. Kittler and F. Roli, editors, Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy, volume 1857 of Lecture Notes in Computer Science, pages 67–76. Springer-Verlag, 2000.

[8] J. Kim, K. Kim, C. Nadal et C. Suen. A Methodology of Combining HMM and MLP Classifiers for Cursive Word Recognition. International Conference Document Analysis and Recognition, vol. 2, pages 319–322, 2000

[9] J. Kittler, M. Hatef, R.P.W. Duin, and Matas J. On combining classifiers. IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(3):226–239, 1998.

[10] J. Kittler and F. M. Alkoot. Sum versus vote fusion in multiple classifier systems. IEEE Transaction on Pattern Analysis ans Machine Intelligence,110–115, 2003.

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

[12] L. Breiman. Bagging predictors. Machine Learning 24(2):123–140, 1996

[13] L. Breiman. Random forests. Machine Learning 45(1):5–32, 2001

[14] L.I. Kuncheva. Combining pattern recognition. methods and algorithms. John Wiley and Sons, 2004

[15] M. L. Zhang, Z. H. Zhou. Exploiting unlabeled data to enhance ensemble diversity. Proceedings of the 10th IEEE International Conference on Data Mining, Sydney, Australia, pp 619–628, 2010

[16] O. Chapelle, B. Schôlkopf, and A. Zien, editors. Semi-Supervised Learning (Adaptive Computation and Machine Learning). The MIT Press, September 2006.

[17] R. Duin et D. Tax. Experiments with Classifier Combining Rules. First Workshop on Multiple Classifier Systems, vol. 1857, pages 16–29, 2000.

[18] R.E. Schapire. A brief introduction to boosting. In Thomas Dean, editor, 16th International Joint Conference on Artificial Intelligence, pages 1401–1406. Morgan Kauffman, 1999.

[19] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision tress: Bagging, boosting and randomization. Machine Learning, 40(2):139–158, 2000.

[20] T.G. Dietterich. Ensemble Methods in Machine Learning. First Workshop on Multiple Classifier Systems, vol. 1857, pages 1–15, 2000

[21] Vapnik, V. N. The Nature of Statistical Learning Theory. Statistics for Engineering and Information Science. Springer.1995

[22] X. Zhu. Semi-supervised learning literature survey, 2006.

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

[24] Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In Proceedings of the 13th International Conference on Machine Learning, pages 148–156. Morgan Kauffman, 1996.