A Simple Hybrid Method for Semi-Supervised Learning

Hernán C. Ahumada[1,2,⋆] and Pablo M. Granitto[1]

[1] CIFASIS, French Argentine International Center for Information and Systems Sciences, UPCAM, France / UNR-CONICET, Argentina
Bv. 27 de Febrero 210 Bis, 2000, Rosario, Argentina
[2] Facultad de Tecnología y Ciencias Aplicadas - Universidad Nacional de Catamarca
Maximio Victoria 55, 4700, Catamarca, Argentina
{ahumada,granitto}@cifasis-conicet.gov.ar

Abstract. We introduce and describe the Hybrid Semi-Supervised Method (HSSM) for learning. This is the first hybrid method aimed to solve problems with both labeled and unlabeled data. The new method uses an unsupervised stage in order to decompose the full problem into a set of simpler subproblems. HSSM applies simple stopping criteria during the unsupervised stage, which allows the method to concentrate on the difficult portions of the original problem. The new algorithm also makes use of a simple strategy to select at each subproblem a small subset of unlabeled samples that are relevant to modify the decision surface. To this end, HSSM trains a linear SVM on the available labeled samples, and selects the unlabeled samples that lie within the margin of the trained SVM. We evaluated the new method using a previously introduced setup, which includes datasets with very different properties. Overall, the error levels produced by the new HSSM are similar to other SSL methods, but HSSM is shown to be more efficient than all previous methods, using only a small fraction of the available unlabeled data.

Keywords: Semi-supervised learning, Hybrid methods, Classification.

1 Introduction

Semi-supervised learning (SSL) is a learning paradigm that recently has gained interest by researchers [3]. The main feature of SSL is its ability to use a few labeled examples together with many unlabeled examples. SSL has a high practical value in many real world applications where giving a label to an example is an expensive and consuming time task [14].

The goal of SSL methods is to improve the performance with respect to supervised methods when labeled data is scarce or expensive [14]. However, SSL methods usually need a large number of unlabeled examples to obtain similar or better results than supervised methods. Therefore, SSL methods are normally

⋆ Author to whom all correspondence should be addressed. Authors acknowledge grant support from ANPCyT PICT 237.

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 138–145, 2012.

used on large datasets, which implies a high computational cost. Another problem, remarked by Singh et al. [12], is the fact that in many cases the addition of unlabeled examples is counterproductive to the learning process.

In this work we introduce a hybrid strategy for semi-supervised binary classification problems, which combines unsupervised, supervised and semi-supervised learning stages. This new strategy aims at dividing the dataset in many subproblems and then, in a smart way, choosing those unlabeled examples that are more relevant to the generation of the hypothesis. We evaluate the effectiveness and efficiency of the new method over some datasets proposed by Chapelle et al. [3] as a benchmark set for semi-supervised methods.

The rest of the paper is organized as follows. In the next section we shortly review previous works on semi-supervised problems. In Section 3 we introduce the HSSM method, which we evaluate in Section 4. Finally, in Section 5 we discuss the results and future lines of research.

2 Related Works

Semi-supervised methods can be divided in generative, graph based models and low density separation [14]. Generative methods try to model directly the probability density function that generated the data. They use the unlabeled examples to extract information that can help to find the best parameters for this task. Graph based methods build a graph whose nodes are the examples (with and without label) and whose arcs have weights proportional to the similarity among them. The idea of such methods is to spread labels from labeled nodes (examples) to nearby unknown label nodes [2]. Low density separation methods use unlabeled examples to find regions of low density, in which they place the decision surface [10,8]. Joachims [9] introduced one of the best known methods in this area, the semi-supervised support vector machine (S3VM), which seeks to maximize the margin of the solution considering both labeled and unlabeled examples. To this end, the method evaluates different assignments of labels to the unlabeled examples. Published methods differ in their strategy to assign the labels and to find the minimum of the cost function, including SVM-light [9], Branch and Bound [4] or the Low Density Separation (LDS) method [5].

Even if some semi-supervised methods can cope with big datasets, there are good reasons to select a reduced set of relevant unlabeled samples to use in SSL: accuracy and efficiency [11]. For example, Delalleau et al. [7] propose to start only with the supervised set, and to add unlabeled samples that are far away from each other, in order to cover the manifold in a uniform way. Then, in a second step, they propose to discard samples that are far away from the decision surface and to replace them with samples near the border. Li and Zhou [11] discuss the problem of the decrease in accuracy from a supervised method to a SSL method that can be observed in same datasets. Rather than selecting the unlabeled data used for training, their method selects the samples that will be predicted with the SSL classifier, switching to a simpler supervised method when appropriate.

HSSM

Input:
D^l : The set of labeled samples.
D^u : The set of unlabeled samples.
$Cl()$: A clustering algorithm.
$DF_{sup}()$: A classifier.
$DF_{ssl}()$: A SSL method.
$SC()$: A stopping criteria.
$IS()$: An unlabeled samples selection function.

Function HSSM($D^l, D^u, Cl, DF_{sup}, DF_{ssl}, SC, IS$):
1. Apply $Cl()$ to $D^l \cup D^u$ to create $(D^l \cup D^u)_1$ and $(D^l \cup D^u)_2$
2. For $i = 1$ to 2:
IF SC(D_i^l, D_i^u) THEN
Train a classifier $DF_{sup}(D_i^l)$
Apply $IS(D_i^u, DF_{sup})$ to produce S_i^u
Train a SSL $DF_{ssl}(D_i^l, S_i^u)$
ELSE
Call $HSSM(D_i^l, D_i^u, Cl, DF_{sup}, DF_{ssl}, SC, IS)$:

Fig. 1. The pseudocode of HSSM

Ahumada et al. [1] proposed the use of a hybrid method to solve multiclass problems. The method has two steps, first it uses a clustering algorithm to construct a hierarchy of easier subproblems, and then it trains several SVMs to solve each individual subproblem. The authors claim that the clustering stage produce several easy-to-solve problems, including a high number of clusters containing only one class. To the best of our knowledge there are no applications of this kind of hybrid methods to SSL.

3 The Hybrid Semi-Supervised Method

The new Hybrid Semi-Supervised Method (HSSM) is a hybrid combination of unsupervised, supervised and semi-supervised methods. As with other hybrid methods [1], the objective is to decompose the original problem into a set of smaller and simpler subproblems, that can be solved more efficiently than the original problem. Also, when faced with a semi-supervised subproblem, the method uses a new simple strategy to select a subset of all the available unlabeled data.

Figure 1 shows the pseudocode of HSSM. The method is a recursion that builds an unsupervised decision tree. It starts by applying a given clustering method to all the input data (labeled and unlabeled samples together) in order to find two clusters. Then the method checks each cluster against the Stopping criteria. If the criteria is met, then the node is considered a leaf and the method fits a classifier to that cluster. In the opposite case the method is called in a recursive way on the corresponding cluster.

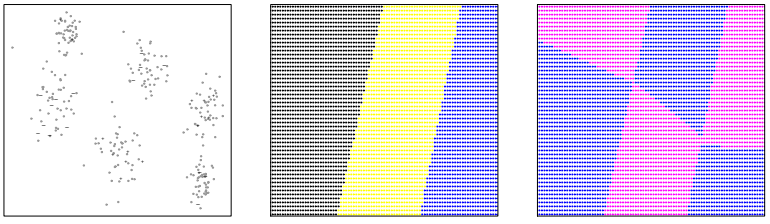

Fig. 2. HSSM on an artificial problem. Left panel shows the partially labeled training set, center panel shows the clustering solution produced by the method and the right panel shows the classification output.

Any clustering algorithm can be used to create the hierarchy. For example, a partitioning method can be called at each recursion step to produce two clusters, or a hierarchical method can be used to build the complete hierarchy with a single call. In this work we use two well-known methods [13]: Average Linkage (AL) and Single Linkage (SL). In both cases the clustering method produces the complete hierarchy in an agglomerative way, thus in practice the HSSM method descends the tree checking where to prune according to the stopping criteria, and then replacing nodes with leaves (classifiers).

We used three simple conditions as stopping criteria. First, the division process is stopped if the "father node" has labeled samples from both classes but both "children nodes" have labeled samples from only one class. In this case it is highly probable that the decision surface goes through the "father node", so the last splitting is canceled and an SSL method is used on the "father node", as there is a potential advantage in adding unlabeled data to this learning problem. Second, when one (and only one) of the clusters has all its labeled samples from the same class, it is considered a leaf and there is no further splitting. In this case it is highly probable that the cluster is far away from the decision border, and all the samples in the cluster can be safely assigned to the present label. Third, if a cluster contains only unlabeled data, it is deleted from the tree structure, as it cannot help in determining the border and the most probable situation is that it contains a group of outliers. More elaborated stopping criteria can be used if needed, but our simple criteria gives a good balance between accuracy and efficiency, as we show in the next section.

Once a node has been selected as a leaf, we adjust a classifier to the corresponding cluster of labeled plus unlabeled data. If the node comes from the second stopping criteria, we use a trivial classifier that assigns all points to the only class present in the cluster. On the other hand, when the node comes from criteria i) we train a full SSL method on the data, but selecting only a subset of unlabeled data that can be helpful to the learning process. Again, any combination of SSL method plus selection method for the subset of unlabeled data can be used. In this work we use SVM-light [9] as the SSL method. Due to this choice, we use a linear SVM [6] to select the subset of unlabeled data. We fit the SVM to all the labeled data in the node, and then select all the unlabeled data that lie within the margin of the classifier to be used for the SSL process.

Table 1. Details of the six datasets used in this work. Row "n" shows the number of samples, "p" the number of variables

Dataset	g241c	g241d	digit1	USPS	BCI	text
n	1500	1500	1500	1500	400	1500
p	241	241	241	241	117	11960

In order to predict a new example, it is driven down the tree until it reaches a leaf. At each node, the example is assigned to a branch using the same criteria as the clustering method (for example, with single linkage the example is assigned to the clustering corresponding to its nearest neighbor) [1]. Once in a leaf, the corresponding classifier assigns a class to the example.

In Figure 2 we show a working example of HSSM. We produced an artificial binary dataset, where each class is formed by three normal distributions with the same variance. We sampled a total of 15 labeled samples for each class and 270 unlabeled samples. As can be seen in the figure, the clustering stage decompose the non-linear problem into 3 subproblem that can be easily solved by a linear SSL method.

4 Experimental Evaluation

To evaluate the new HSSM method we followed as close as possible the setup proposed in the book by Chapelle et al. [3], chapter 21, for SSL methods. We selected the six binary datasets described in Table 1. The evaluation setup includes two experiments for each dataset, one using 10 labeled points in total and the other using 100 labeled points. All the remaining samples are used as unlabeled data. For each experiment, the setup includes 12 replications with fixed labeled subsets [3]. The test set is always the full set of unlabeled samples.

Table 2. Classification errors for diverse methods on the six datasets considered in this work, with two different setups: 10 or 100 labeled points

	Dataset	SVM-L	Light-L	Light-NL	LDS	HSSM-AL	HSSM-SL
10	g241c	46.91	20.99	21.40	24.71	44.63	45.22
	g241d	45.56	46.48	46.87	50.08	44.52	45.86
	digit1	43.39	20.59	20.54	17.77	21.14	21.54
	USPS	19.98	30.70	30.59	25.20	20.64	18.75
	BCI	49.00	50.02	49.76	49.15	50.09	49.70
	text	49.41	28.60	28.80	31.21	32.81	41.25
100	g241c	23.67	18.18	18.93	18.46	41.09	44.36
	g241d	25.76	23.76	30.70	22.42	33.83	43.67
	digit1	49.19	18.05	16.95	6.15	8.20	7.30
	USPS	20.01	21.12	13.56	9.77	8.61	6.89
	BCI	38.33	42.67	42.72	33.25	45.94	47.69
	text	28.35	22.31	22.30	24.52	29.51	27.52

Table 3. Details on the tree structure created by HSSM using two different clustering methods (AL and SL) and two different number of labeled samples (10 and 100). For each possible type of leaf (One or two class leaves), the columns show the average number of leaves created by the method and, in brackets, the average percentage of samples used by those classifiers. The "Discarded" column shows the average percentage of samples that were discarded during the training process.

	Dataset	One class leaves	Two class leaves	Discarded
10 - AL	g241c	7.1 (4.9%)	1.0 (2.3%)	92.7%
	g241d	5.1 (22.6%)	1.6 (2.2%)	75.2%
	digit1	2.1 (37.6%)	1.5 (36.0%)	26.4%
	USPS	2.8 (32.0%)	1.2 (15.8%)	52.3%
	BCI	4.3 (50.6%)	2.0 (12.0%)	37.4%
	text	3.6 (13.5%)	1.5 (11.0%)	75.5%
10 - SL	g241c	6.8 (0.6%)	1.0 (0.5%)	98.9%
	g241d	7.1 (0.6%)	1.0 (0.3%)	99.1%
	digit1	5.8 (15.9%)	1.0 (16.7%)	67.5%
	USPS	2.3 (0.4%)	1.0 (1.0%)	98.5%
	BCI	6.1 (14.6%)	1.4 (4.1%)	81.3%
	text	6.2 (2.1%)	1.0 (1.3%)	96.6%
100 - AL	g241c	59.8 (39.9%)	11.8 (7.3%)	52.8%
	g241d	68.0 (42.3%)	8.5 (5.1%)	52.6%
	digit1	14.3 (87.0%)	3.9 (6.0%)	7.0%
	USPS	14.1 (72.1%)	3.5 (7.1%)	20.8%
	BCI	31.4 (45.7%)	21.0 (22.7%)	31.7%
	text	36.0 (63.4%)	11.7 (10.2%)	26.4%
100 - SL	g241c	96.1 (8.6%)	1.2 (0.8%)	90.7%
	g241d	93.4 (8.7%)	2.3 (0.8%)	90.5%
	digit1	49.7 (57.5%)	2.2 (6.4%)	36.1%
	USPS	39.4 (27.9%)	1.2 (2.9%)	69.2%
	BCI	60.5 (40.1%)	13.3 (11.7%)	48.2%
	text	76.0 (25.2%)	3.0 (1.3%)	73.5%

We evaluated two version of HSSM, one using Average Linkage as clustering method (HSSM-AL from here on) and the other using Single Linkage (HSSM-SL). In both cases we used a linear SVM as supervised method, and Joachims' SVM-light as SSL method. For the SVM, the C constant was fixed to 100, as it is difficult to do a model selection with 10 labeled points. For SVM-light, all constants where taken as suggested by Joachims [9].

As a comparison with the new method, we also included classification results using other methods, always evaluated with the same experimental setup. First, we included the results obtained with a linear SVM, without taking into account the unlabeled data. Then, we considered two version of Joachims' SVM-light, one with a linear kernel (Light-L in the tables) and another one with a Gaussian kernel (Light-NL). We also included the results obtained with Chapelle & Zien's Low Density Separation (LDS) method [5], another non-linear SSL method.

Table 2 show the corresponding results for the two different setups. As in Chapelle et al. [3], the results are shown as the average percentage of classification

error over the 12 runs of each method. The six datasets are very different in structure and origin, as explained in [3]. The first two datasets, g241c and g241d, are artificial and were designed to fool SSL methods in some way. HSSM results are not very good on these problems. Only in g241d with 10 labeled samples the new method shows a good performance. Nevertheless, the artificial digit1 dataset has an structure that favors SSL methods. In this case the HSSM shows good results, equal or better than other SSL methods. The last 3 datasets are real world problems with different properties. The USPS dataset is highly imbalanced. In this problem again both HSSM versions show the best results. In the other two real world datasets the results are mixed. The BCI problem is highly difficult, and all methods show near random results (except for the LDS method with 100 points). The text dataset is sparse and high dimensional. HSSM is clearly better than the base supervised method but never better than the other SSL methods.

On Table 3 we show some details about the tree structures created by HSSM. Overall, it is remarkable the low number of SSL classifiers needed by the HSSM, and the small proportion of samples used by them. Comparing SL and AL structures, it is interesting to note that SL clustering always discard a higher proportion of samples, and always uses a lower number of SSL classifiers (two class leaves), which combined produce more efficient HSSM classifiers. This is a consequence of SL tendency to form small clusters, which are discarded or assigned a single class by HSSM most of the time. Another interesting finding is that the HSSM method always uses more unlabeled data on the digit1 than on other datasets, which is the artificial problem aimed to favor SSL methods, where it also shows the best results among all methods. As a last remark, the best result on the unbalanced USPS dataset is produced by HSSM-SL in both setups, using in most cases only one SSL classifier and less than 3% of the data, on average, to train it.

5 Conclusions

In this work we presented the HSSM for semi-supervised learning. This is the first hybrid method aimed to solve problems with labeled plus unlabeled data. The new method follows a typical strategy in hybrid methods, using an unsupervised stage in order to decompose the full problem into a set of simpler subproblems. Using simple stopping criteria during the unsupervised stage, we allow the method to concentrate on the difficult portions of the original problem.

We also introduced a simple method to select at each subproblem a small subset of unlabeled samples that are relevant to modify the decision surface. To this end we trained a linear SVM on the available labeled samples, and selected the unlabeled samples that lie within the margin of the trained SVM.

We evaluated the new method using a setup introduced by Chapelle et al. [3], which includes dataset with very different properties. Overall, the error levels produced by the new HSSM are similar to other SSL methods. The new method seems to be more aggressive than previous SSL methods, as it shows better results in problems appropriate for SSL, but also worst results in problems aimed at fooling SSL methods. We showed that HSSM is more efficient than other SSL

methods, using in all cases only a small fraction of the available unlabeled data to produce equivalent results than other methods. As a last analysis, the use of the SL clustering methods always produced simpler and more efficient classifiers than the use of AL clustering, with a similar performance in classification accuracy.

Further work is needed in order to find better ways to regularize the method and improve its performance on problems less favorable to SSL methods.

References

1. Ahumada, H.C., Grinblat, G.L., Granitto, P.M.: Unsupervized Data-Driven Partitioning of Multiclass Problems. In: Honkela, T. (ed.) ICANN 2011, Part I. LNCS, vol. 6791, pp. 117–125. Springer, Heidelberg (2011)
2. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: ICML 18, pp. 19–26. Morgan Kaufmann, San Francisco (2001)
3. Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge (2006)
4. Chapelle, O., Sindhwani, V., Keerthi, S.: Branch and bound for semi-supervised support vector machines. In: NIPS 19. MIT Press, Cambridge (2007)
5. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: AISTATS 2005, pp. 57–64 (2005)
6. Cristianini, N., Shawe–Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
7. Delalleau, O., Bengio, Y., Le Roux, N.: Large-scale algorithms. In: Chapelle, O., Schölkopf, B., Zien, A. (eds.) Semi-Supervised Learning, pp. 333–341. MIT Press, Cambridge (2006)
8. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: Actes de CAP 2005, pp. 281–296 (2005)
9. Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML 16, pp. 200–209. Morgan Kaufmann Publishers, San Francisco (1999)
10. Lawrence, N.D., Jordan, M.I.: Semi-supervised learning via gaussian processes. In: NIPS 17, pp. 753–760. MIT Press, Cambridge (2004)
11. Li, Y.-F., Zhou, Z.-H.: Improving semi-supervised support vector machines through unlabeled instances selection. In: Burgard, W., Roth, D. (eds.) AAAI. AAAI Press (2011)
12. Singh, A., Nowak, R.D., Zhu, X.: Unlabeled data: Now it helps, now it doesn't. In: NIPS 21, pp. 1513–1520 (2008)
13. Sneath, P.H.A., Sokal, R.R.: Numerical Taxonomy. W.H. Freeman and Company, San Francisco (1973)
14. Zhu, X., Goldberg, A.B.: Introduction to Semi-Supervised Learning. Morgan & Claypool Publishers, California (2009)