A Preliminary Study on Transductive Extreme Learning Machines

Simone Scardapane, Danilo Comminiello, Michele Scarpiniti, and Aurelio Uncini

Department of Information Engineering, Electronics and Telecommunications (DIET),
"Sapienza" University of Rome,
via Eudossiana 18, 00184, Rome
{simone.scardapane,danilo.comminiello,
michele.scarpiniti}@uniroma1.it, aurel@ieee.org

Abstract. Transductive learning is the problem of designing learning machines that succesfully generalize only on a given set of input patterns. In this paper we begin the study towards the extension of Extreme Learning Machine (ELM) theory to the transductive setting, focusing on the binary classification case. To this end, we analyze previous work on Transductive Support Vector Machines (TSVM) learning, and introduce the Transductive ELM (TELM) model. Contrary to TSVM, we show that the optimization of TELM results in a purely combinatorial search over the unknown labels. Some preliminary results on an artifical dataset show substained improvements with respect to a standard ELM model.

Keywords: Transductive learning, extreme learning machine, semi-supervised learning.

1 Introduction

In the classical Machine Learning setting [1], starting from a limited set of data sampled from an unknown stochastic process, the goal is to infer a general predictive rule for the overall system. Vapnik [2] was the first to argue that in some situations, this target may be unnecessarily complex with respect to the actual requirements. In particular, if we are interested on predictions limited to a given set of input patterns, then a learning system tuned to this specific set should outperform a general predictive one. In Vapnik words, the advice is that, "*when solving a problem of interest, do not solve a more general problem as an intermediate step*" [2]. Vapnik also coined a term for this setting, which he called *Transductive Learning* (TL).

In [2] he studied extensively the theoretical properties of TL, and his insights led him to propose an extension to the standard Support Vector Machine (SVM) algorithm, namely the Tranductive SVM (TSVM). While SVM learning results in a quadratic optimization problem, TSVM learning is partly combinatorial, making it a difficult non-convex optimization procedure. However, a number of interesting algorithms have been proposed for its efficient solution. The interested reader can find a comprehensive review of them in Chapelle et al. [3].

By drawing theoretical and practical ideas from TSVMs, in this paper we extend *Extreme Learning Machine* (ELM) theory [4] to the transductive setting. ELM models

S. Bassis et al. (eds.), *Recent Advances of Neural Network Models and Applications*,
Smart Innovation, Systems and Technologies 26,
DOI: 10.1007/978-3-319-04129-2_3, © Springer International Publishing Switzerland 2014

have gained some attention as a conceptual unifying framework for several families of learning algorithms, and possess interesting properties of speed and efficiency. An ELM is a two-layer feed-forward network, where the input is initially projected to an highly dimensional feature space, on which a linear model is subsequently applied. Differently from other algorithms, the feature space is fully fixed before observing the data, thus learning is equivalent to finding the optimal output weights for our data. We show that, in the binary classification case, Transductive ELM (TELM) learning results in a purely combinatorial search over a set of binary variables, thus it can be solved more efficiently with respect to TSVM. In this preliminary work we use a simple Genetic Algorithm (GA) [5] as a global optimizer and test the resulting algorithm on an artificial dataset. Results show promising increase in performance for different sizes of the datasets.

Transductive learning has been throughly studied lately due to the interest in Semi-Supervised Learning (SSL) [6]. In SSL, additional unlabelled data is provided to the algorithm (as in TL), but the goal is to infer a general predictive rule as in classical inductive learning. In this respect, unlabelled data is seen as additional information that the algorithm can use to deduce general properties about the geometry of input patterns. Despite TL and SSL have different objectives, their inner workings are in some respects similar, and many TL and SSL algorithms can be used interchangeably in the two situations. In particular, TSVMs are known as Semi-Supervised SVM (S3VM) [3] in the SSL community. Hence, our work on TELM may be of interest as a first step towards the use of ELM models in a SSL setting.

The rest of this paper is organized as follows: in Section 2 we introduce some basic concepts on TL, and detail the TSVM optimization procedure. Section 3 summarizes the main theory of ELM. Section 4, the main contribution of this work, extends ELM theory using concepts from Section 2. Section 5 shows some preliminary results on an artificial dataset. Although we provide a working algorithm, two fundamental questions remain open, and we confront with them in Section 6. Finally, we make some final remarks in Section 7.

2 Transductive Learning

2.1 Inductive Learning and Support Vector Machines

Consider an unknown stochastic process described by the joint probability function $p(\mathbf{x},y) = p(\mathbf{x})p(y|\mathbf{x}), \mathbf{x} \in X, y \in Y$, where X and Y are known as the *input* and *output* spaces respectively. In this work we restrict ourselves to the binary classification case, i.e., $Y = \{-1,+1\}$. Given a loss function $L(\mathbf{x},y,\hat{y}) : X \times Y \times Y \to \mathbb{R}$ that measures the loss we incur by estimating $\hat{y} = f(\mathbf{x})$ instead of the true y, and a set of possible models H, the goal of inductive learning is to find a function that minimizes the *expected risk*:

$$I[f] = \int_{X \times Y} L(\mathbf{x},y,f(\mathbf{x}))p(\mathbf{x},y)d\mathbf{x}dy \tag{1}$$

We are given only a limited dataset of N samplings from the process $S = (\mathbf{x}_i, y_i)_{i=1}^N$, that we call the *training set*. The *empirical risk* is defined as:

$$I_{emp}[f;S] = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i, y_i, f(\mathbf{x}_i)) \tag{2}$$

Vapnik [2] derived several bounds, known as *VC bounds*, on the relation between (1) and (2) for limited datasets. All bounds are in the following form and are valid with probability $1-\eta$:

$$I[f] \leq I_{emp}[f;S] + \Phi(h, N, \eta) \tag{3}$$

where h is the *VC-dimension* of the set H, and $\Phi(h,N,\eta)$ is known as a *capacity* term. In general, such term is directly proportional to h. Thus, for two functions $f_1, f_2 \in H$ with the same error on the dataset, the one with lower VC-dimension is preferable. Practically, this observation can be implemented in the Support Vector Machine (SVM) algorithm, as we describe below.

Consider a generic *Reproducing Kernel Hilbert Space H* as set of models. There is a direct relationship [7] between h and the inverse of $\|f\|_H$, $f \in H$, where $\|f\|_H$ is the norm of f in H. Thus, the optimal function is the one minimizing the error on the dataset and of minimum norm. When using the *hinge loss* $L(\mathbf{x}_i, y_i, f(\mathbf{x})) = \max\{0, 1 - y_i f(\mathbf{x}_i)\}$ as loss function, we obtain the SVM for classification [8]. It can be shown that learning corresponds to a quadratic optimization problem:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \frac{1}{2}\|f\|_H^2 + C_s \sum_{i=1}^{N} \zeta_i \\ \text{subject to} \quad & y_i f(\mathbf{x}_i) \geq 1 - \zeta_i,\ \zeta_i \geq 0,\ i = 1, \ldots, N. \end{aligned} \tag{4}$$

where ζ_i are a set of *slack variables* that measures the error between predicted and desired output and C_s is a regularization parameter set by the user. Solution to (4) is of the form $f(\mathbf{x}) = \sum_{i=1}^{N} a_i k(\mathbf{x}, \mathbf{x}_i)$, where $k(\cdot,\cdot)$ is the *reproducing kernel* associated to H.

2.2 Transductive Learning and Transductive SVM

In *Transductive learning* (TL) we are given an additional set[1] $U = (\mathbf{x}_i)_{i=N+1}^{N+M}$, called the *testing set*, and we aim at minimizing $I_{emp}[f;U]$. An extension of the theory described above [2] leads to minimizing the error on both S and U.

By denoting with $\mathbf{y}^* = [y_{N+1}^*, \cdots, y_{N+M}^*]^T$ a possible labelling of the elements in U, this results in the following (partly combinatorial) optimization problem, known as the *Transductive SVM* (TSVM):

$$\begin{aligned} \underset{f,\mathbf{y}^*}{\text{minimize}} \quad & \frac{1}{2}\|f\|_H^2 + C_s \sum_{i=1}^{N} \zeta_i + C_u \sum_{i=N+1}^{N+M} \zeta_i \\ \text{subject to} \quad & y_i f(\mathbf{x}_i) \geq 1 - \zeta_i,\ \zeta_i \geq 0,\ i = 1, \ldots, N. \\ & y_i^* f(\mathbf{x}_i) \geq 1 - \zeta_i,\ \zeta_i \geq 0,\ i = N+1, \ldots, N+M \end{aligned} \tag{5}$$

[1] Note the peculiar numbering on the dataset.

where we introduce an additional regularization term C_u. In particular, equation (5) is combinatorial over $\mathbf{y}^*$, since each label is constrained to be binary. This makes the overall problem highly non-convex and difficult to optimize in general. Some of the algorithms designed to efficiently solve it are presented in [3].

Typically, we also try to enforce an additional constraint on the proportion of labellings over $\mathbf{U}$, of the form:

$$\rho = \frac{1}{M}\sum_{i=1}^{M} y_i^*$$

where ρ is set *a-priori* by the user. This avoids unbalanced solutions in which all patterns are assigned to the same class.

3 Extreme Learning Machine

An Extreme Learning Machine (ELM) [9,4] is a linear combination of an L-dimensional feature mapping of the original input:

$$f(\mathbf{x}) = \sum_{i=1}^{L} h_i(\mathbf{x})\beta_i = \mathbf{h}(\mathbf{x})^T\beta \qquad (6)$$

where $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]^T$ is called the *ELM feature vector* and β is the vector of expansion coefficients. The feature mapping is considered fixed, so the problem is that of estimating the optimal β. Starting from a known function $g(\mathbf{x}, \theta)$, where θ is a vector of parameters, it is possible to obtain an ELM feature mapping by drawing parameters θ at random from an uniform probability distribution, and repeating the operation L times. Huang et al. [4] showed that almost any non-linear function can be used in this way, and the resulting network will continue to be an universal approximator. Moreover, they proposed the following regularized optimization problem, where we aim at finding the weight vector that minimizes the error on S and is of minimum norm:

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \frac{1}{2}\|\beta\|_2^2 + \frac{C_s}{2}\sum_{i=1}^{N}\zeta_i^2 \\ \text{subject to} \quad & \mathbf{h}^T(\mathbf{x}_i)\beta = y_i - \zeta_i, \; i = 1, \ldots, N. \end{aligned} \qquad (7)$$

As for SVM, C_s is a regularization parameter that can be adjusted by the user, and $\zeta_i, i = 1, \ldots, N$ measure the error between desired and predicted output. The problem is similar to (4), but has a solution in closed form. In particular, a possible solution to (7) is given by [4]:

$$\beta = \mathbf{H}^T(\frac{1}{C_s}\mathbf{I}_{N\times N} + \mathbf{H}\mathbf{H}^T)^{-1}\mathbf{y} \qquad (8)$$

where $\mathbf{I}_{N\times N}$ is the $N \times N$ identity matrix, and we defined the hidden matrix $\mathbf{H} = [\mathbf{h}(\mathbf{x}_1), \ldots, \mathbf{h}(\mathbf{x}_N)]$ and the output vector $\mathbf{y} = [y_1, \ldots, y_N]^T$. When using ELM for classification, a decision function can be easily computed as:

$$f'(\mathbf{x}) = sign(f(\mathbf{x}))$$

4 Transductive ELM

Remember that in the TL setting we are given an additional dataset $U = (\mathbf{x}_i)_{i=N+1}^{N+M}$ over which we desire to minimize the error. To this end, similarly to the case of TSVM, we consider the following modified optimization problem:

$$\begin{aligned} \underset{\beta, y^*}{\text{minimize}} \quad & \frac{1}{2}\|\beta\|_2^2 + \frac{C_s}{2}\sum_{i=1}^{N}\zeta_i^2 + \frac{C_u}{2}\sum_{i=N+1}^{N+M}\zeta_i^2 \\ \text{subject to} \quad & \mathbf{h}^T(x_i)\beta = y_i - \zeta_i,\ i = 1,\ldots,N. \\ & \mathbf{h}^T(x_i)\beta = y_i^* - \zeta_i,\ i = N+1,\ldots,N+M. \end{aligned} \tag{9}$$

We call (9) the *Transductive ELM* (TELM). At first sight, this may seems partly combinatorial as in the case of TSVM. However, for any possible choice of the labelling $\mathbf{y}^*$, the optimal β is given by (8), or more precisely, by a slightly modified version to take into account different parameters for C_s and C_u:

$$\beta = \mathbf{H}^T(\mathbf{C}^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^T)^{-1}\begin{bmatrix}\mathbf{y} \\ \mathbf{y}^*\end{bmatrix} \tag{10}$$

Where $\mathbf{C}$ is a diagonal matrix with the first N elements equal to C_s and the last M elements equal to C_u, and the hidden matrix is computed over all $N+M$ input patterns:

$$\mathbf{H} = [\mathbf{h}(\mathbf{x}_1),\ldots,\mathbf{h}(\mathbf{x}_N),\mathbf{h}(\mathbf{x}_{N+1}),\ldots,\mathbf{h}(\mathbf{x}_{N+M})]$$

Back-substituting (10) into (9), we obtain a fully combinatorial search problem over $\mathbf{y}^*$. This can be further simplified by considering:

$$\hat{\mathbf{H}} = \mathbf{H}^T(\mathbf{C}^{-1}\mathbf{I} + \mathbf{H}\mathbf{H}^T)^{-1} = \begin{bmatrix}\hat{\mathbf{H}}_1 & \hat{\mathbf{H}}_2\end{bmatrix} \tag{11}$$

Where $\hat{\mathbf{H}}_1$ is the submatrix containing the first N columns of $\hat{\mathbf{H}}$, and the other block follow. Equation (10) can be rewritten as:

$$\beta = \hat{\mathbf{H}}_1\mathbf{y} + \hat{\mathbf{H}}_2\mathbf{y}^* \tag{12}$$

Where the vector $\hat{\mathbf{H}}_1\mathbf{y}$ and the matrix $\hat{\mathbf{H}}_2$ are fixed for any choice of the labeling of U. Any known algorithm for combinatorial optimization [5] can be used to train a TELM model, and form (12) is particularly convenient for computations. We do not try to enforce a specific proportion of positive labels (although this would be relatively easy) since in our experiments the additional constraint never improved performance.

5 Results

The TELM algorithm was tested on an artificial dataset known in literature as *the two moons*, a sample of which is shown in Fig. 1. Two points, one for each class, are shown in red and blue respectively. All simulations were performed by MATLAB 2012a, on an Intel i3 3.07 GHz processor at 64 bit, with 4 GB of RAM available, and each result is averaged over 100 runs. The TELM is solved using a standard *Genetic*

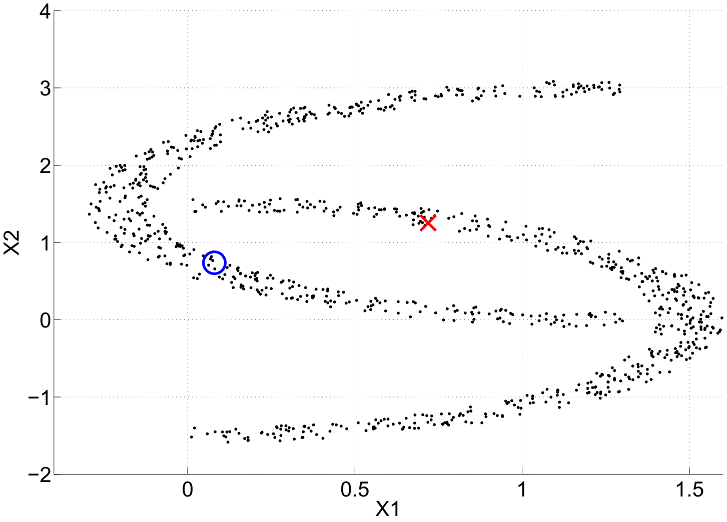

Fig. 1. Sample of the dataset

Algorithm [5]. For comparison, we implemented as baseline a standard ELM model and a binary SVM.

Sigmoid additive activation functions are used to construct the ELM feature space:

$$g(\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{ax}+b)}} \tag{13}$$

Using standard default choices for the parameters, we consider 40 hidden nodes, and set $C = 1$. Parameters $\mathbf{a}$ and b of equation (13) were generated according to an uniform probability distribution. The SVM uses the Gaussian kernel:

$$k(\mathbf{x},\mathbf{y}) = \exp\{-\gamma\|\mathbf{x}-\mathbf{y}\|_2^2\} \tag{14}$$

Parameter γ in (14) was also set to 1 in all the experiments. Algorithms were tested using five different sizes of the datasets. For the first four experiments, a total of 100 samples was considered, and the training size was gradually increased. In the last experiment, instead, we considered two datasets of 100 elements each. For each method we present the classification accuracy in Table 1, where the highest accuracy in each row is highlighted in boldface.

As can be seen, TELM outperforms both methods for every combination we considered. In particular, it gives a small improvement when trained using very small training datasets (first two rows), very large increments with datasets of medium size (third and fourth row), and is able to reach 100% classification accuracy with sufficient samples (fifth row).

Table 1. Experimental results: classification accuracy

	SVM	ELM	TELM
$N = 4$, $M = 98$	0.77	0.75	**0.79**
$N = 10$, $M = 90$	0.81	0.75	**0.86**
$N = 40$, $M = 60$	0.85	0.80	**0.93**
$N = 60$, $M = 40$	0.85	0.81	**0.97**
$N = 100$, $M = 100$	0.93	0.95	**1**

6 Open Questions

Two main questions remain to be answered for an effective implementation of the TELM algorithm. We detail them briefly in this Section.

1. Our formulation suffers from a major drawback which is encountered also on TSVMs. In particular, it cannot be easily extended to the regression case. It is easy to show that any minimizer β of the first two terms of equation (10) automatically minimizes the third with the trivial choice $y_i^* = h(\mathbf{x}_i)^T\beta$. Thus, some modifications are needed, for example following [10].
2. The genetic algorithm imposes a strong computational effort in minimizing (10). This can be addressed by developing specialized solvers able to take into consideration the specific nature of the problem. To this end, we imagine that many of the algorithms used for TSVMs can be readily extended to our context.

7 Conclusions

In this work we presented an initial study for the extension of ELM theory to the transductive learning framework. We showed that this results in a fully combinatorial optimization problem. In our experiments, we solved it using a standard GA. Results are highly promising in the dataset we considered. However, there is the need of further optimizing the learning algorithm before a successful real-world application.

References

1. Cherkassky, V., Mulier, F.: Learning from data: concepts, theory, and methods (2007)
2. Vapnik, V.: The nature of statistical learning theory, 2nd edn., vol. 8. Springer (January 1999)
3. Chapelle, O., Sindhwani, V., Keerthi, S.: Optimization techniques for semi-supervised support vector machines. Journal of Machine Learning Research 9, 203–233 (2008)
4. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Transactions on Systems, Man, and Cybernetics 42(2), 513–529 (2012)
5. Luke, S.: Essentials of metaheuristics (2009)

6. Chapelle, O., Schölkopf, B., Zien, A.: Semi-supervised learning (2006)
7. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. Advances in Computational Mathematics 13, 1–50 (2000)
8. Steinwart, I., Christmann, A.: Support vector machines, 1st edn. (2008)
9. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: Theory and applications. Neurocomputing 70(1-3), 489–501 (2006)
10. Cortes, C., Mohri, M.: On transductive regression. In: Advances in Neural Information Processing Systems (2007)