Multi-scale Support Vector Regression

Stefano Ferrari, Francesco Bellocchio, Vincenzo Piuri, and N. Alberto Borghese

***Abstract*—A multi-kernel Support Vector Machine model, called Hierarchical Support Vector Regression (HSVR), is proposed here. This is a self-organizing (by growing) multiscale version of a Support Vector Regression (SVR) model. It is constituted of hierarchical layers, each containing a standard SVR with Gaussian kernel, at decreasing scales. HSVR have been applied to a noisy synthetic dataset. The results illustrate their power in denoising the original data, obtaining an effective multiscale reconstruction of better quality than that obtained by standard SVR. Furthermore with this approach the well known problem of tuning the SVR parameters is strongly simplified.**

***Index Terms*—Support Vector Machine, Support Vector Regression, Kernel functions**

I. Introduction

Support Vector Machines (SVM) have been introduced as a powerful method for classification [1][2]. They are based on setting the classification boundary such that the distance between the data points with different labels that are closest to the boundary is maximized. The boundary between classes is a hyperplane defined by a linear combination of a subset of the data points to be classified, called Support Vectors (SV). To determine the solution to the problem, that is to identify the SVs and their associated coefficients, the problem is reformulated as a quadratic optimization problem that, being convex, guarantees the uniqueness and the optimality of the solution. Moreover, standard optimization algorithms can be used to find the solution.

It was soon recognized that, as such, the method was able only to classify the data that exhibit local linear separability property, that was not sufficient for many applications. For this reason, a mapping machinery that transforms the classification problem inside the native space into a classification problem inside a higher dimensional space, called feature space, was developed. Goal of this mapping is to obtain local linear separability in this higher dimensional space. This machinery makes use of kernel functions to represent the internal product of two data vectors projected into this higher dimensional space, that is, the inner product of the projected data. As such, the kernel output gives a measure of similarity of the data pairs.

One of the most used kernel is the Gaussian function that realizes a mapping into an infinite dimensional space. This is an obstacle for the computation of the solution, but the optimization procedure does not need the explicit mapping values to compute the SVM solution. Instead, the optimization procedure requires only the internal product of the mapped data pairs; therefore the mapping is only implicitly computed (this advantageous scheme is known as "kernel trick").

The SVM approach has been more recently extended to regression problems [3], domain in which this approach has been named Support Vector Regression (SVR).

In the standard approach to SVR, a single kernel function is used, with one single shape and set of parameters. Like the other methods based on kernels, the quality of the regression depends on the proper choice of the kernel function and of its parameters, that must be suitable to represent the data. Generally, this choice, also known as kernel selection [4], is a difficult task: the function is often chosen by trial and error, through genetic optimization or by resorting to the experience of the people designing the software for the particular regression (or classification) task.

Besides this, the choice of a single kernel function can be questioned. In fact when the data are characterized by a different frequency content over the input domain, the use of a single kernel is not able to produce satisfying results and multiple kernel approaches have been recently investigated [5][6]. In these approaches the kernel effectively used in the solution is computed during the optimization phase as a linear combination of a given set of kernels. However, even in this case, all the SVs of the solution will feature the same kernel, although its shape was the result of optimization.

The problem of using a single kernel is highlighted in the examples reported in Fig. 1. The data points have been sampled on the curve $h(\cdot)$

$$h(x) = \sin(2\pi x^4) + x \tag{1}$$

whose local frequency content increases with x. The sampling step is decreased with the local frequency according to $\frac{1}{120x}$. The regression computed with a large kernel fails in reconstructing the details as shown in Fig. 1b. On the other side, using a kernel with a small scale, such as the one used in Fig. 1c, the regression will be prone to overfitting and lack of generalization in scarcely sampled regions.

We present here a novel approach that allows adapting automatically the kernels parameters to the local frequency content of the data. This approach is based on a multi-layer structure, where each layer contains SVs that have the same kernel function with the same parameters, but the different layers feature SVs still having the same shape but with different parameters. The output of the whole architecture is the sum of the output produced by each layer and we have therefore termed this approach Hierarchical Support Vector Regression (HSVR). Being the different layers composed of kernels that are different, this approach can be in fact considered a multi-kernel approach.

N.A. Borghese is with the Department of Computer Science, Università degli Studi di Milano, Italy, e-mail: borghese@dsi.unimi.it

F. Bellocchio, S. Ferrari and V. Piuri are with the Department of Information Technology, Università degli Studi di Milano, Italy e-mail {francesco.bellocchio,stefano.ferrari,vincenzo.piuri}@unimi.it

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

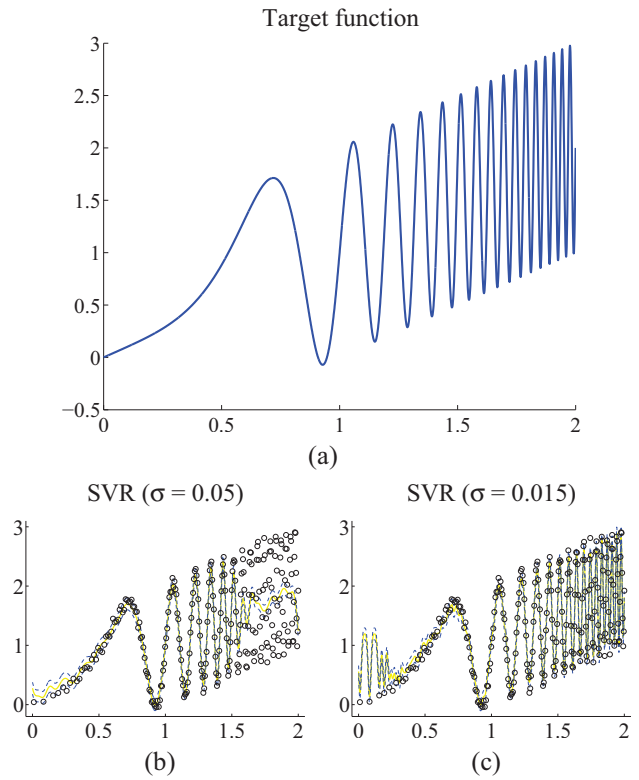

Fig. 1. (a) A function with non-stationary frequency content, and (b)–(c) some single kernel SVR with two different scale parameters, σ. (b) A large scale kernel provides smooth regression, but is unable to reconstruct the details, while (c) a small scale kernel suffers of overfitting providing poor generalization.

II. Support Vector Regression

Let $S = \{(x_1, z_1), ..., (x_n, z_n)\}$ be the set of n examples that constitute the training set, where $x_i\,(1 \leq i \leq n)$ is a vector belonging to an input space $X \subseteq \mathbb{R}^D$ and $z_i \in \mathbb{R}$. The vectors x_i are called examples or instances and z_i labels. Aim of a SVR is to find a regression function, $f : X \to \mathbb{R}$ of this type:

$$f(x) = \omega^T \phi(x) + b, \tag{2}$$

where ω is the weight vector of the features space, $\phi(x)$ is a suitable mapping of data point x in the features space, and $b \in \mathbb{R}$ is a threshold constant. ω and b can be found solving the optimization problem:

$$\begin{aligned} \min_{\omega,b} \quad & \tfrac{1}{2}\omega^T\omega + C\sum_{i=1}^n \xi_i^+ + C\sum_{i=1}^n \xi_i^- \\ \text{s.t.} \quad & z_i - \omega^T\phi(x_i) - b \leq \epsilon + \xi_i^+ \\ & \omega^T\phi(x_i) + b - z_i \leq \epsilon + \xi_i^- \\ & \xi_i^+, \xi_i^- \geq 0, i = 1, \ldots, n \end{aligned} \tag{3}$$

where $\epsilon \geq 0$ determines a "tube" around the regression curve inside which the points do not contribute to the cost function (3) (ϵ-insensitive loss function). Therefore ϵ controls the accuracy of the regression curve. The parameter C adjusts the trade off between the regression error and the regularization on f. $\xi^+ = \{\xi_1^+, \ldots, \xi_n^+\} \in \mathbb{R}^l$ and $\xi^- = \{\xi_1^-, \ldots, \xi_n^-\} \in \mathbb{R}^l$ are slack variables for relaxing the approximation constraints, and measure the distance of each data point from the ϵ tube.

Introducing Lagrange multipliers α_i^+ on constraints corresponding to ξ_i^+ and α_i^- on constraints corresponding to ξ_i^-, the dual problem of (3) can be written as:

$$\begin{aligned} \max_{\alpha^+,\alpha^-} \quad & -\tfrac{1}{2}(\alpha^+ - \alpha^-)^T K(\alpha^+ - \alpha^-) \\ & -\epsilon\textstyle\sum_{i=1}^n(\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^n z_i(\alpha_i^+ - \alpha_i^-) \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^n(\alpha_i^+ - \alpha_i^-) = 0 \\ & \alpha_i^+, \alpha_i^- \in [0, C], i = 1, \ldots, n \end{aligned} \tag{4}$$

where $\alpha^+ = \{\alpha_1^+, \ldots, \alpha_n^+\} \in \mathbb{R}^n$ and $\alpha^- = \{\alpha_1^-, \ldots, \alpha_n^-\} \in \mathbb{R}^n$ are the dual variables, and $K \in \mathbb{R}^{n\times n}$ is the kernel matrix evaluated from a kernel function $k : X \times X \to \mathbb{R}$, $K_{i,j} = k(x_i, x_j)$. Solving α^+, α^-, and b using the Karush-Kuhn-Tucker (KKT) conditions in (4), and introducing $\beta_i = \alpha_i^+ - \alpha_i^-$, the regression function of (2) becomes:

$$f(x) = \sum_{i=1}^{n} \beta_i\, k(x,\, x_i) + b \tag{5}$$

where $f(\cdot)$ is expressed as a linear combination of replicas of the kernel function, $k(\cdot, \cdot)$ (instead of the mapping function, ϕ used in (2)), centered in each training sample. For the KKT conditions the β_i coefficients have to satisfy the following:

$$|\beta_i| = \begin{cases} 0, & \|y_i - f(x_i)\| < \epsilon \\ [0,\, C], & \|y_i - f(x_i)\| = \epsilon \\ C, & \|y_i - f(x_i)\| > \epsilon \end{cases} \tag{6}$$

The points that have non-zero β coefficients are called support vectors (SV); the SVs that lie outside the ϵ-tube are called "bounded" and the absolute value of their associated β is set equal to C. As a result, the linear combination in (5) can be limited only to the SVs.

Among others, the most commonly used kernel for regression is the Gaussian kernel, whose width is characterized by its scale parameter, σ:

$$k(x,\, x_i) = G(||x - x_i||; \sigma) = \exp\left(-\frac{||x - x_i||^2}{\sigma^2}\right) \tag{7}$$

The parameter σ affects the extension of the influence of each support vector, x_i, in its neighborhood, and hence it is related to the resolution of the regression. Using Gaussians of very small scale would allow reconstructing the finest details, while a large scale kernel will provide a rough approximation. However, the use of a single scale kernel may not be the best choice when the dataset is sampled from a non stationary source, which presents slow varying regions alternated with rapid variation (Fig. 1a). In fact, when operating with a small scale kernel, the reconstruction of flat regions may induce, in the best case, a waste of computational resources, but, when the training samples are too spaced with respect to the scale parameter, the resulting regression will provide a poor generalization. On the other

hand, when a large scale kernel is employed, detailed regions could be reconstructed only using a very large number of SVs.

An approximation scheme that allows to adapt, for each SV, the scale of the kernel to the frequency content of the region in which the SV is situated, represents a better solution. In the next Section, we describe the Hierarchical Support Vector Regression (HSVR) approach presented here. It is based on a hierarchical scheme to achieve a multi-scale approximation using a pool of SVMs which operate at a different scale.

III. The HSVR Model

The output of the HSVR model is the sum of the output of a pool of single-kernel SVRs, $\{a_l\}$, organized as a hierarchy of layers, each of which is characterized by a different scale:

$$S(x) = \sum_{l=1}^{L} a_l(x, \sigma_l) \tag{8}$$

where L is the number of layers and σ_l determines the scale of the kernel of the l-th layer. The scale decreases increasing the layer number, that is $\sigma_l \geq \sigma_{l+1}$ holds.

When the kernels are Gaussian functions, the output of each layer can be written as:

$$a_l(x; \sigma_l) = \sum_{k=1}^{M_l} \beta_{l,k}\, G(||x - x_{l,k}||; \sigma_l) + b_l \tag{9}$$

where M_l is the number of SVs, $\beta_{l,k}$ is the coefficient of the k-th SV and b_l is the bias of the l-th layer.

Each SVM layer, l, realizes a reconstruction of the target function up to a certain scale, determined by σ_l. The training of the hierarchical structure is obtained by adding and configuring one layer at a time, proceeding from the layer featuring the largest scale to that featuring the smallest one.

The first layer is trained such that the distance between the regression curve produced by the layer and the data is minimized (3). All the other layers are trained to approximate the residual, that is the difference between the original data and the output of the HSVR model produced by the already configured layers. For each layer, the residual, r_l, is computed for each data point as:

$$r_l(x_i) = r_{l-1}(x_i) - a_l(x_i) \tag{10}$$

where $r_0(x_i) = z_i$ is assumed.

The value of the scale parameter of the first layer, σ_1, is somehow arbitrary. For instance it can be chosen proportional to the size of the input region (e.g., the length of the diagonal of the input data bounding box). New layers are added during training until a given stopping criterion is satisfied (e.g., when the validation error does not decrease anymore).

Two other parameters are defined for each layer: C_l, the trade-off between the regression error and the regressor smoothness, and ϵ, that controls the accuracy of the regressor itself.

The parameter C_l is computed, for each layer, as J times the range of the residuals used to configure that layer:

$$C_l = J\left(\max_i r_{l-1}(x_i) - \min_i r_{l-1}(x_i)\right) \tag{11}$$

where J has been experimentally set to 5. Moreover, we notice that as C_l is the value assumed by the Lagrange multipliers associated to the SVs of the l-th layer (4), its value represents the maximum weight that can be associated to each Gaussian kernel. For the regions of the input space where the SVs have no significant overlap (this depends on the Gaussian scale parameter), the value of C_l is approximately the maximum value that the regression curve can assume in those regions, given that the Gaussian kernel maximum amplitude is equal to one. For this reason, C_l should be large enough to allow the regression curve reaching the maximum or minimum value of the data points inside the whole input domain. On the other hand, a too large value of C_l could favor overfitting. The value set in (11) represents a trade-off between these two requirements.

Hence, the only parameter that cannot be estimated from the data set is the parameter ϵ. This should be proportional to the accuracy required for the regression, as its optimal value is linearly related to noise amplitude [7].

A. Training set selection

In general, the regression curve obtained with HSVR is of better quality with respect to standard SVR. The drawback of this scheme is the total number of SVs used, that is significantly higher than in standard SVR.

Moreover, in HSVR the layers with a large value of σ have a number of SVs similar to the layers with a small σ. This appears in contrast with common sense, as fewer units should be required to realize a reconstruction at a large scale than those necessary to realize a reconstruction with a fine level of detail. This is due to the fact that all the data points distant from the regression curve by more than ϵ are selected as SVs (6). Hence, in the first layer, when the regression curve has a low frequency content, many data points will result distant from the curve and will be selected as SVs, thus leading to an unnecessary high number of SVs in the first layers that features the largest scale.

To avoid this, after the configuration of each layer, a selection step is carried out on the training data as described in the following. We first notice that the distance of the training points from the regression curve measures the suitability of the curve to describe the information conveyed by the data points. In this sense, the points the are too distant from the regression curve cannot be "explained" by the regression curve computed up to that layer and their utility can be questioned. This intuition has been confirmed experimentally as we observed that the quality of the regression at a given scale does not degrade by considering only the points close to the estimated regression curve.

The configuration phase of each layer is then structured in two training steps: the first one, carried out considering the residual of all the training points, provides the best regression curve at the considered scale, while the second one, carried

out only on the subset of the data that have been selected, realizes an efficient approximation of the same regression curve (the actual output of the layer).

To this aim, after the first step, the distance of the data points from the current regression curve is analyzed and only those points that are sufficiently close to the curve are selected for the second step in which this subset of data is used for the final configuration of the layer. To further reduce the number of SVs used and the computational time, we observe that the SVs inside the ϵ-tube do not contribute to the regression [3]. At the same time, the points that are very distant from the ϵ-tube can be considered as outliers and rarely give meaningful contribution to the estimate of the curve. For these reasons, an acceptable approximation of the regression can be obtained using only those points that lie close to the border of the ϵ-tube.

We explicitly remark that, similarly to [8][9] any reconstruction error due to the reduction of the training set, will increase the residual that is used to configure the next layer of the architecture. Therefore such error is not critical, as it will be taken care by the next layer.

IV. Results

We explored the HSVR model with the space-varying function, $h : \mathbb{R} \rightarrow \mathbb{R}$, defined in (1) and plotted in Fig. 1a.

In order to simulate the effect of measurement noise, the training dataset has been obtained sampling 252 points from the function $\hat{h}(\cdot)$, computed from (1) as:

$$\hat{h}(x) = h(x) + u_{[-0.1,\,0.1]} \tag{12}$$

where $u_{[-0.1,\,0.1]}$ is a random variable uniformly distributed in $[-0.1,\,0.1]$. The performance has been evaluated using a test set and a validation set, each composed of 500 points sampled from $h(\cdot)$ using a uniform distribution.

The model has been evaluated in terms of accuracy of the regression through the root mean square error (RMSE), the mean absolute error ($\mathrm{Err}_{\mathrm{mean}}$) and its standard deviation ($\mathrm{Err}_{\mathrm{std}}$) computed over the test set, the number of SVs, and the computational time. We have also compared the performances of the hierarchical model (with and without training set reduction) with the standard SVR.

The optimization problem in (4), that arises for both the hierarchical and the standard SVR model, was solved through LibCVM Toolkit Version 2.2 [10]. This software has shown the same accuracy of SVM^{light} [11] (that is one of the most used software packages for SVM) with a substantial saving of computational time.

To decide when to stop the learning procedure, the validation error at the end of the configuration phase of each layer is monitored. When it does not decrease for two consecutive layers, learning is stopped; the last two layers in which validation error increased are then discarded as they are supposed to produce overfitting.

For sake of comparison we have chosen the best HSVR model and the best SVR model, in terms of final validation error. To this aim, we analyzed the results produced by a

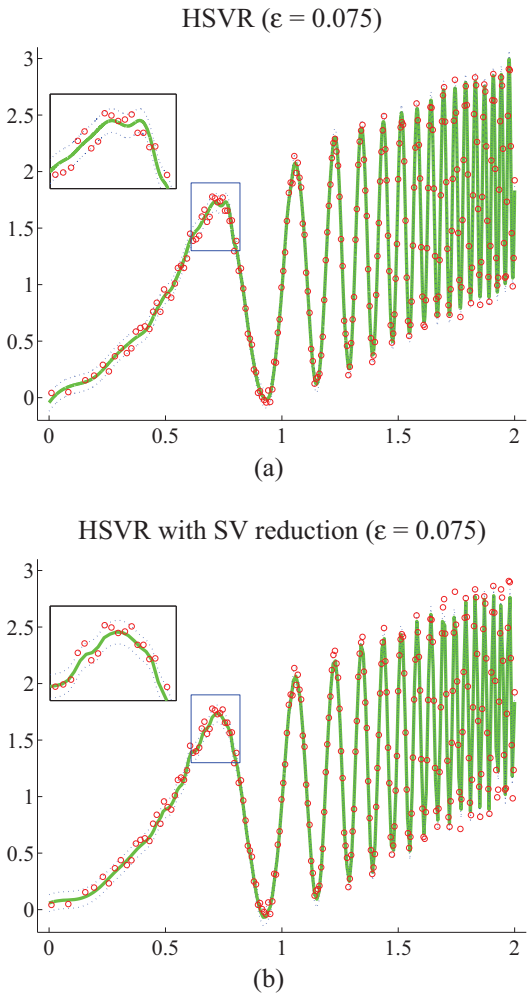

Fig. 2. Reconstruction provided by HSRV (a) and HSRV with SV reduction (b) using $\epsilon = 0.075$. The dashed lines limit the ϵ-insensitive region (i.e., the data points that lie in this region do not increase the cost function (3)).

TABLE I
Performance

	$\mathrm{Err}_{\mathrm{mean}}$	$\mathrm{Err}_{\mathrm{std}}$	RMSE	#SVs	time [s]
HSVR	0.0254	0.0239	0.0349	1462	0.475
HSVR (red.)	0.0228	0.0212	0.0311	206	0.644
SVR	0.0979	0.171	0.197	163	0.595

SVR with all the possible combinations of the following parameters, ϵ, σ, and C:

$$\epsilon \in \{0.0, 0.01, 0.025, 0.05, 0.075, 0.1, 0.2\} \tag{13}$$

$$\sigma \in \{0.015, 0.0313, 0.0625, 0.125, 0.25, 0.5, 0.75, 1, 1.5, 2\} \tag{14}$$

$$C \in \{0.5, 1, 1.5, 2, 5, 7, 10, 20\} \tag{15}$$

and consider the model that produced the lowest validation error. HSVR was configured only for the different possible values of ϵ, considering the value of C set by (11) and σ set equal to the size of the input domain.

Table I reports the accuracy of the different models and the associated reconstruction is shown in Figs. 2a-b, and 3.

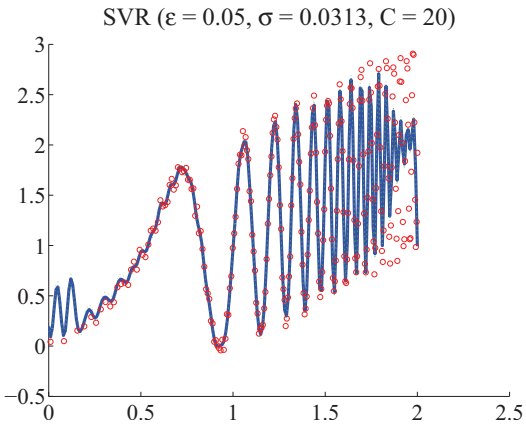

Fig. 3. Reconstruction provided by SRV ($\epsilon = 0.05$, $\sigma = 0.0313$, $C = 20$).

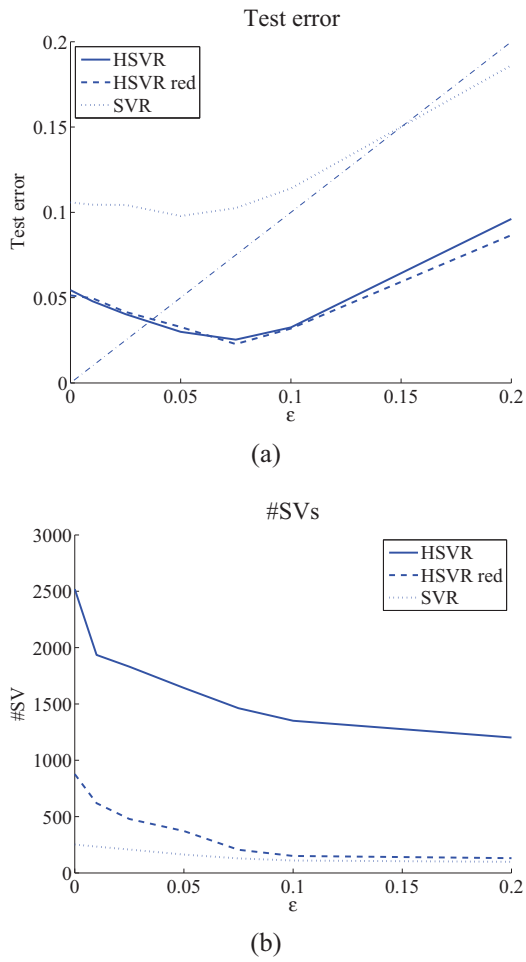

Fig. 4. Comparison of the performance of the SVR and HSVR models, in terms of (a) mean test error, and (b) number of SVs used with respect to the value of ϵ. For reference, the value of ϵ has been reported as a dot-dashed line.

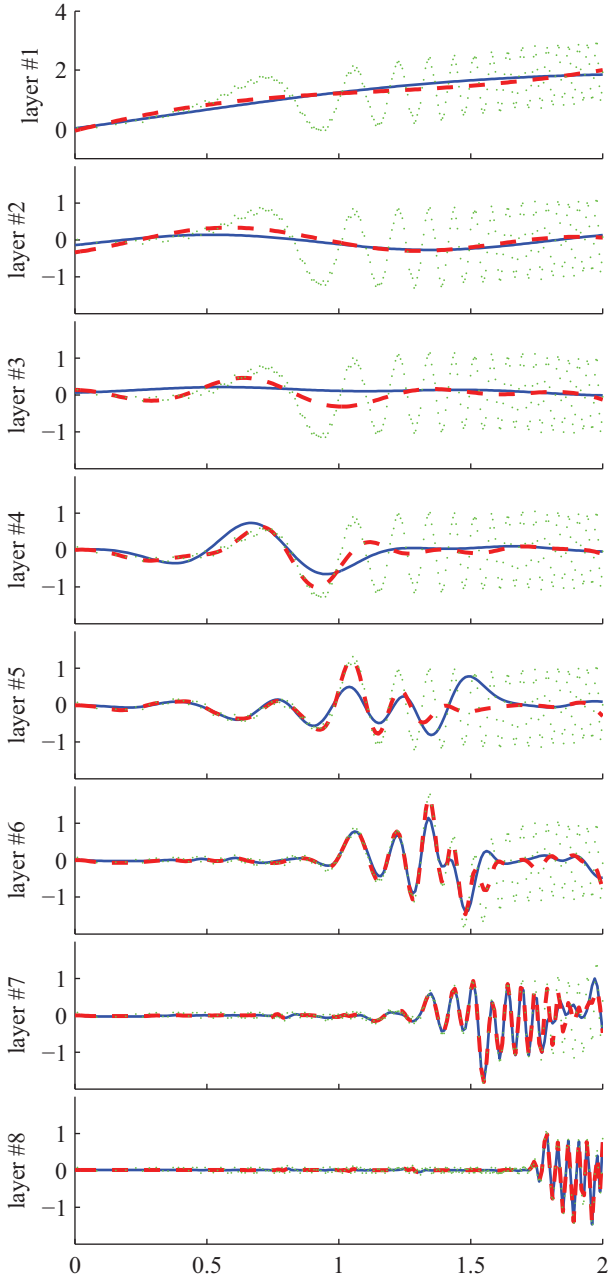

Fig. 5. Reconstruction operated by the different layers of the HSVR with reduction as obtained after the first (dashed line) and second (continuous line) configuration steps. The points of the residual (i.e., the training points for the layer) are reported as dots. The SV reduction has the effect of smoothing the regression in the first layers, but it tends to disappear in the last layer, where the two curves are almost coincident.

The accuracy and the number of SVs as a function of ϵ are shown in Fig 4. Notice that the $\mathrm{Err}_{\mathrm{mean}}$ is smaller than the ϵ value; this means that the data points, on average, lie inside the ϵ-tube around the curve.

To further investigate the behavior of the hierarchical models, we analyzed the reconstruction operated by each layer and reported in Table II. The regressor obtained at the different layers is plotted in Fig. 5. Lastly, the impact of the reduction of the SVs number in each layer is also

TABLE II
DETAILS OF THE HSVR MODELS ($\epsilon = 0.075$)

	HSVR				HSVR red.			
# Layer	Err_{mean}	Err_{std}	RMSE	#SVs (tot.)	Err_{mean}	Err_{std}	RMSE	#SVs (tot.)
1	0.463	0.341	0.575	237 (237)	0.475	0.363	0.597	4 (4)
2	0.455	0.339	0.567	239 (476)	0.454	0.346	0.570	5 (9)
3	0.405	0.375	0.552	231 (707)	0.463	0.339	0.573	6 (15)
4	0.352	0.367	0.508	216 (923)	0.408	0.367	0.549	9 (24)
5	0.284	0.366	0.463	195 (1118)	0.340	0.423	0.543	14 (38)
6	0.197	0.321	0.377	160 (1278)	0.233	0.356	0.426	35 (73)
7	0.0759	0.179	0.194	115 (1393)	0.100	0.256	0.275	63 (136)
8	0.0254	0.0239	0.0349	69 (1462)	0.0228	0.0212	0.0311	70 (206)

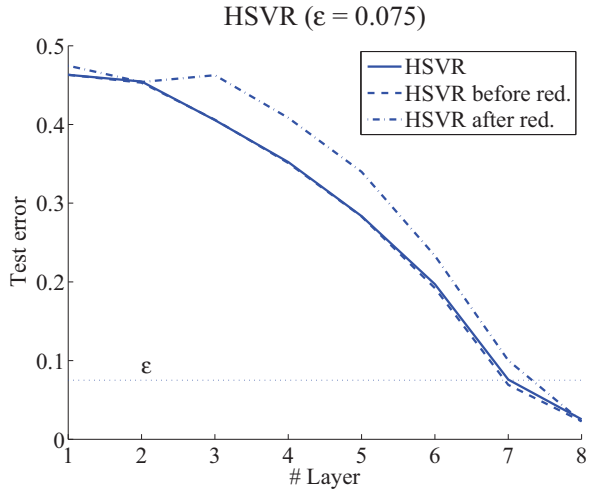

Fig. 6. Evolution of the test error achieved by the HSVR models as the layers are inserted.

shown in Fig. 5, where the reconstruction before and after the reduction is reported. The test error in the two cases as a function of the number of layers is reported in Fig. 6.

V. DISCUSSION

The actual dataset used to explore the model has been chosen because it is very difficult to obtain a good generalization using a single scale function as the basis for the approximation space. In fact, as shown Table I, the best regression curve obtained with the best combination of C, σ and ϵ provided by the single scale SVR achieves a test error of 0.0979, while the HSVR models reach 0.0228 and 0.0254 (with and without SVs reduction, respectively) with an accuracy improvement of 429% with respect to the standard SVR.

The source of the higher error of SVR is evident comparing Fig. 3a, which reports the best regression curve obtained by SVR, with Figs. 2b, which report the best regression curve obtained by the hierarchical models. It can be clearly noticed that the SVR fails in generalizing the behavior of the dataset both in the smooth (but scarcely sampled) and in the highly variable regions, as the scale of its kernel ($\sigma = 0.0313$) causes overfitting in the smooth region, while it is unable of approximating the points when the frequency content of the dataset increases. On the other hand, the two HSVR models provide very similar regression curves, as it can be noticed comparing Figs. 2a and 2b. Besides, as the average test error is below the ϵ value used in the configuration phase ($\epsilon = 0.075$), most of the training points are enclosed into the ϵ-tube. Namely, they are 183 and 181 (without and with SV reduction, respectively) while in the SVR model, only 89 are contained inside it. Similar results are obtained also for the test set, where 478 for both HSVR models lie into the ϵ-tube.

As shown in Fig. 4a, the test error of the HSVR models is well below the error of the SVR for every value of ϵ tested: the accuracy of the HSVR model is clearly higher at all the values of ϵ, even considering the best combination of C and σ for the SVR for each value of ϵ.

Fig. 4b depicts the efficiency of SVs reduction. In fact, the number of SVs employed by the HSVR model drops of about 1/7-th of the number of SVs used when SVs reduction is not applied. The saving in the number of SVs, is paid with an increase in the computational time. However, as shown in Table I, the increase of 36% in the configuration time is worth a 610% saving in the number of SVs. Moreover, both the number of SVs and the computational time compares well with the corresponding figures of the traditional SVR (Table I) when SVRs reduction is applied. These considerations are confirmed by the data reported in Table II. The layer-by-layer average test errors of the HSVR with and without reduction are similar and do not exceed the value of ϵ. The number of SVs of all but the last layer is much larger when SVs reduction is not applied. Hence, it is clear that most of the SVs employed in the first layers, when no reduction is applied, is wasted in the vain attempt of approximating details with a kernel that operates at a too large scale.

On the contrary, as can be noticed by observing Fig. 5, the reduction of the SVs smooths the reconstruction operated by each layer. However, the difference between the regression curve obtained with and without reduction is added to the residual, which is passed to the next layer and it is therefore recovered by higher layers. This difference becomes smaller and smaller as new layers are added, and, in the last layer, the two curves are almost coincident.

These qualitative considerations are confirmed by the plot

in Fig. 6. Here, with a continuous line, the test error obtained by the HSVR, in which all the layers are configured without the SVs reduction is used as a reference. This accuracy is very similar to that obtained by applying SVs reduction in the previous layers but not in the current one (dashed line). This accuracy is higher in the intermediate layers, than that obtained by reducing layer by layer the SVs.

In this respect, the number of layers may be critical. Different strategies can be used to stop the learning procedure. If no a-priori information is available, validation error guarantees that good generalization capability is obtained. Otherwise, we can stop learning when the error on the residual drops below the given threshold: this can be for instance associated to measurement noise [9].

The value of C has been set according to the range of the regression curve in (11). The factor J has been experimentally set analyzing different data set. Although its optimal value depends on the particular data set, it has been verified that results are robust with respect to variations of J, while its value is much more critical for standard SVR.

We are currently investigating the applicability of HSVR to real-data models both of dense data points like those available from [12] and sparse multi-variate data points like those contained in [13]. Preliminary results show that these observations still hold. The HSVR model outperforms the SVR model, when the data has different frequency contents in different regions. In this situation, exploiting multiscale regression results into a large saving in resources, better approximation , and large saving in computational time as it avoids the exploration of the parameter space. Otherwise, the use of our model still allows to save computational time, while accuracy and number of SVs is comparable.

In principle, the learning schema can work with kernels other than the Gaussians. However, the Gaussian kernel has two main properties: the scale parameter, σ, allows shaping the kernel such that the SVs are sensitive to different frequency ranges, and the non orthogonality, which allows to recover in the next layers the possible reconstruction error left by the previous layers. Besides this, most optimization engines like LibCVM, used here, suppose Gaussian kernels. Although in principle other kernels that enjoy the above mentioned properties could be used, adequate optimization engines should be developed, that goes beyond the aim of this work. The use of different kernels will be investigated in future works.

VI. Conclusion

A Hierarchical Support Vector Regression model is here presented and applied for generalizing a noisy synthetic dataset. The proposed hierarchical model allows a more accurate reconstruction thanks the use of a set of kernels at different scale. Moreover, the HSVR can be enriched by a SVs reduction procedure which allows for obtaining the desired regression curve without substantially increasing the computational resources required by a standard SVR model. Besides with the presented model the problem of choosing the parameter of the SVR is heavily reduced.

References

[1] V. Vapnik, *Statistical Learning Theory*. Wiley, 1998.

[2] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[3] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 2004.

[4] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semi-definite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[5] S. Qiu and T. Lane, "Multiple kernel learning for support vector regression," Computer Science Department, The University of New Mexico, Albuquerque, NM, USA, Tech. Rep., 2005.

[6] Z. Wang, S. Chen, and T. Sun, "MultiK-MHKS: A novel multiple kernel learning algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 348–353, 2008.

[7] A. Smola, N. Murata, B. Schölkopf, and K.-R. Müller, "Asymptotically optimal choice of ϵ-loss for support vector machines," in *Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, pages 105 – 110*. Springer Verlag, 1998, pp. 105–110.

[8] S. Ferrari, M. Maggioni, and N. A. Borghese, "Multi-scale approximation with hierarchical radial basis functions networks," *IEEE Trans. on Neural Networks*, vol. 15, no. 1, pp. 178–188, Jan. 2004.

[9] S. Ferrari, F. Bellocchio, V. Piuri, and N. Borghese, "A hierarchical RBF online learning algorithm for real-time 3-D scanner," *IEEE Transactions on Neural Networks*, vol. 21, no. 2, pp. 275–285, feb 2008.

[10] I. Tsang, J. Kwok, and P.-M. Cheung, "Core vector machines: Fast svm training on very large data sets," *Journal of Machine Learning Research*, vol. 6, pp. 363–392, 2005.

[11] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 11, pp. 169–184.

[12] The Stanford 3D Scanning Repository. [Online]. Available: http://graphics.stanford.edu/data/3Dscanrep/

[13] Delve Repository — Data for Evaluating Learning in Valid Experiments. [Online]. Available: http://www.cs.toronto.edu/ delve/