Learning smooth models of nonsmooth functions via convex optimization

Fabien Lauer, van Luong Le, Gérard Bloch

2012 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 23–26, 2012, SANTANDER, SPAIN

LEARNING SMOOTH MODELS OF NONSMOOTH FUNCTIONS VIA CONVEX OPTIMIZATION

F. Lauer

Université de Lorraine, LORIA, UMR 7503
CNRS
Inria

V.L. Le, G. Bloch

Université de Lorraine, CRAN, UMR 7039
CNRS

ABSTRACT

This paper proposes a learning framework and a set of algorithms for nonsmooth regression, i.e., for learning piecewise smooth target functions with discontinuities in the function itself or the derivatives at unknown locations. In the proposed approach, the model belongs to a class of smooth functions. Though constrained to be globally smooth, the trained model can have very large derivatives at particular locations to approximate the nonsmoothness of the target function. This is obtained through the definition of new regularization terms which penalize the derivatives in a location-dependent manner and training algorithms in the form of convex optimization problems. Examples of application to hybrid dynamical system identification and image reconstruction are provided.

1. INTRODUCTION

This paper proposes a learning framework and a set of algorithms dedicated to nonsmooth function regression. While a number of efficient machine learning tools exist to learn smooth functions with high accuracy from a finite data sample, the accuracy of these approaches becomes less satisfactory for nonsmooth target functions. More precisely, and since learning without minimal smoothness assumptions is not possible, we consider piecewise smooth (PWS) target functions. Such functions have discontinuities in the function itself or the derivatives at particular locations or boundaries between regions in which the function is smooth. In particular, we focus on the case where these boundaries are unknown, since with known boundaries the problem simply amounts to independently solving a classical smooth regression subproblem in each region. Such behaviors are typically observed in physical systems subject to saturations or switches and in images where the intensity can vary smoothly within an object and jump at edges.

Two major approaches can be applied to learn nonsmooth functions: general methods for smooth regression and methods dedicated to nonsmooth regression. The first class of methods includes regularized kernel methods such as Support Vector Machines (SVM) and Kernel Ridge Regression (KRR) [1], for which the training algorithms amount to solving a convex optimization problem. However, the regularization typically considered for such methods explicitly draws the solution away from nonsmooth functions. On the other hand, methods designed for nonsmooth regression typically consider a collection of local (and smooth) submodels. The mixtures of experts (ME) [2] are a widely known example where a gating network controls the mixing of the experts for a particular input. However, such approaches rely on nonconvex optimization, which implies that despite the existence of efficient algorithms such as expectation-maximization, these algorithms are only guaranteed to converge to a local solution and are sensitive to their initialization. Other works in control theory consider switching regression, either for linear [3]) or nonlinear [4] submodels, and suffer from similar issues. Except for [5], the situation is often worse since these methods do not constrain the submodels to be active in different regions of input space, which creates local solutions that are not consistent with piecewise models.

In this paper, we derive a new learning approach for piecewise smooth functions in the spirit of regularized kernel regression. This means that the model belongs to a class of smooth functions and that training amounts to solving a convex optimization problem, thus avoiding local minima issues. Though constrained to be globally smooth, the trained model can have very large derivatives at particular locations to approximate the nonsmoothness of the target function. This is obtained through the definition of new regularization terms which penalize the derivatives in a location-dependent manner. We show how these can be chosen to obtain algorithms in the form of convex optimization problems leading to the desired properties for the model.

Though clearly rooted in machine learning, the proposed approach is also inspired by works from the signal and image processing literature. In particular, the new regularization schemes are related to the variational methods for denoising which involve the total variation (TV) of the sought function (see [6] for a general overview). TV-based methods typically

978-1-4673-1026-0/12/$31.00

estimate a piecewise constant version of the signal, but extensions to piecewise affine reconstructions were also proposed (see [7] and references therein). In comparison, the proposed approach aims at solving a regression problem rather than a denoising one, i.e., the output of the algorithm is a predictive model of the data rather than a finite set of function values. In addition, the method can deal with irregular samplings of continuous input spaces of arbitrary dimensions and with piecewise smooth models with nonlinear pieces.

2. PRELIMINARIES

2.1. Notations and definitions

Vectors and matrices are written in boldface, e.g., $\boldsymbol{x}_i \in \mathbb{R}^d$ is a vector of index i, whereas $x_i \in \mathbb{R}$ denotes the ith entry in $\boldsymbol{x}$. $\delta_{i,j}$ denotes the Kronecker delta which is 1 iff $i = j$. We use $\odot$ and $\otimes$ to denote the Hadamard (entrywise) and Kronecker products, respectively. For $n \in \mathbb{N}$, we recursively define the differential operator of order n in dimension d, $D^{(n)}$, by

$$D^{(n)} = \nabla \otimes D^{(n-1)} = \begin{pmatrix} \frac{\partial}{\partial x_1} D^{(n-1)} & \dots & \frac{\partial}{\partial x_d} D^{(n-1)} \end{pmatrix}^T, \tag{1}$$

where $D^{(0)}$ is the identity operator. In particular, if the function $f : \mathbb{R}^d \to \mathbb{R}$ is class C^n, then $D^{(1)} f = \nabla f$ is the gradient of f, $D^{(2)} f = \mathrm{vec}(Hf)$ is the vector representation of the Hessian and $D^{(n)} f$ is a function of $\mathbb{R}^d$ to $\mathbb{R}^{d^n}$ that computes all the nth order derivatives of f. We will denote by $\mathcal{C}^n$ the set of functions of class C^n.

2.2. Learning in RKHS

Let K be a real-valued positive type function [8] on $\mathcal{X}^2$ and $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ the corresponding reproducing kernel Hilbert space (RKHS), i.e., K is the reproducing kernel of $\mathcal{H}$. Assume we are given a training set of N pairs $(\boldsymbol{x}_i, y_i) \in (\mathcal{X} \subset \mathbb{R}^d) \times (\mathcal{Y} \subset \mathbb{R})$, $i = 1, \dots, N$, with the general goal of learning a function $f \in \mathcal{H}$ such that this function minimizes a regularized functional representing a trade-off between the fit to the data and some regularity conditions on f:

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), y_i) + \lambda \mathcal{R}(f), \tag{2}$$

where the data term is defined through a loss function ℓ of $\mathbb{R}^2$ to $\mathbb{R}^+$, $\mathcal{R}(f)$ is a general regularization term and $\lambda \geq 0$ tunes the trade-off between the two terms. A typical choice for $\mathcal{R}(f)$ is $\|f\|^2_{\mathcal{H}}$, for which the representer theorem [9] provides solutions in the form of kernel expansions over the training set. Such a regularizer based on the induced norm in RKHS is a global measure of the function smoothness and is particularly suitable for cases without prior information on the shape of the target function in order to avoid overfitting. However, these global regularizers (in the sense that minimizing $\|f\|_{\mathcal{H}}$ influences the shape of f over the entire input space $\mathcal{X}$) are not always suitable. For instance, $\|f\|^2_{\mathcal{H}}$ similarly penalizes a noisy function with many oscillations and a very smooth function with large jumps at a few locations. Thus, in the case of piecewise smooth target functions, minimizing $\|f\|^2_{\mathcal{H}}$ yields globally smoother functions and discards optimal solutions without distinguishing them from noisy functions.

3. PROPOSED LEARNING FRAMEWORK

The following introduces local regularization terms which can distinguish between globally nonsmooth functions and smooth functions with large derivatives at sparse locations. Then, Sect. 3.2 presents the learning algorithm based on the proposed regularizers, while the different choices for the hyperparameters are discussed in Sect. 3.3 and 3.4.

3.1. Learning with local regularization of higher order

We define a local regularization functional of order (n, p) as

$$\forall f \in \mathcal{C}^n, \ \forall \boldsymbol{x} \in \mathcal{X}, \quad R_{n,p}(\boldsymbol{x}, f) = \|D^{(n)} f(\boldsymbol{x})\|_p, \tag{3}$$

where $D^{(n)}$ is a differential operator of order n as defined in (1) and p is a parameter that selects a particular norm. Given a function class $\mathcal{H} \subseteq \mathcal{C}^n$, the learning problem with a local regularizer as in (3) reads

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i), y_i) + \lambda \sum_{i=1}^{M} R_{n,p}(\boldsymbol{z}_i, f), \tag{4}$$

where local regularization terms are minimized for M sample points $\boldsymbol{z}_i$ in order to globally regularize f over $\mathcal{X}$. Various sampling strategies can be considered here. The $\boldsymbol{z}_i$'s can be chosen on a grid in order to obtain a sufficient coverage of $\mathcal{X}$ or equal to the training points, $\boldsymbol{x}_i$, in order to obtain a representative sample of the data distribution (the latter is used in all experiments below). Another choice combining the two features is to sample $\boldsymbol{z}_i$ as perturbed versions of the $\boldsymbol{x}_i$.

3.2. Learning algorithms for nonsmooth functions

To be more specific, we now consider the square loss function, $\ell(f(\boldsymbol{x}_i), y_i) = (y_i - f(\boldsymbol{x}_i))^2/2$, and the Gaussian RBF kernel, $K(\boldsymbol{x}', \boldsymbol{x}) = \exp(-\|\boldsymbol{x}' - \boldsymbol{x}\|_2^2 / 2\sigma^2)$. We further restrain ourselves to models f built as kernel expansions over the training set, i.e., f is in the subspace of $\mathcal{H}$ spanned by $\{K(\boldsymbol{x}_i, \cdot)\}_{i=1}^N$:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}), \tag{5}$$

where $[\alpha_1, \dots, \alpha_N]^T = \boldsymbol{\alpha}$ is the vector of parameters to estimate. The form of f in (5) indeed constraints the problem since the representer theorem [9] does not apply to (4).

By defining the kernel matrix $\boldsymbol{K}$ of elements $(\boldsymbol{K})_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the target vector $\boldsymbol{y} = [y_1, \ldots, y_N]^T$, the data term can be written as $\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2$. Since f is a linear combination of kernel functions with weights α_i, any derivative of f is linear wrt. $\boldsymbol{\alpha}$ and we can rewrite $D^{(n)}f(\boldsymbol{x}_i)$ as

$$D^{(n)}f(\boldsymbol{z}_i) = [\boldsymbol{d}_{i1}, \ \ldots, \ \boldsymbol{d}_{id^n}]^T\boldsymbol{\alpha} = \boldsymbol{D}_i\boldsymbol{\alpha}. \tag{6}$$

This yields the finite-dimensional optimization problem

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda\sum_{i=1}^{M}\|\boldsymbol{D}_i\boldsymbol{\alpha}\|_p, \tag{7}$$

where convexity of the data term is obvious from the choice of a convex loss function and where the regularization term is a sum of norms of linear functions, hence convex. Therefore any local solution of (7) is a global solution. However, the regularization term makes the cost function nonsmooth.

In (7), we are looking for a sparse solution in terms of the vector $\boldsymbol{r} = \left[\|D^{(n)}f(\boldsymbol{z}_1)\|_p, \ \ldots, \ \|D^{(n)}f(\boldsymbol{z}_M)\|_p\right]^T$ by minimizing its ℓ_1-norm, since $\lambda\sum_{i=1}^{M}\|D^{(n)}f(\boldsymbol{z}_i)\|_p = \lambda\|\boldsymbol{r}\|_1$. Therefore, the vector of derivatives, $\boldsymbol{D}_i\boldsymbol{\alpha}$, will be drawn to zero at points where its norm is small, while large derivatives will be left at few points. Note that it is crucial here not to use *squared* ℓ_p-norms in (7). For instance, using *squared* ℓ_2-norms amounts to minimizing $\|\boldsymbol{r}\|_2^2$, which would lead to a smooth optimization problem, but not to sparse solutions.

3.3. Choice of the ℓ_p-norm

We now describe the algorithms obtained by the choice of p.

ℓ_2-norm ($p = 2$). The most natural choice is $p = 2$, which yields the nonsmooth convex optimization program

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2 + \lambda\sum_{i=1}^{M}\sqrt{\boldsymbol{\alpha}^T\boldsymbol{D}_i^T\boldsymbol{D}_i\boldsymbol{\alpha}}. \tag{8}$$

While practical algorithms have been proposed to solve such problems, we instead eliminate the nonsmoothness of the cost function by considering a constrained (and convex) formulation. This yields the Second-Order Cone Program (SOCP):

$$\begin{aligned}\min_{\boldsymbol{\alpha}\in\mathbb{R}^N,\xi\in\mathbb{R},\boldsymbol{t}\in\mathbb{R}^M} \ & 2\xi + \lambda\sum_{i=1}^{M} t_i \\ \text{s.t.} \ & \left\|\begin{matrix}1-\xi \\ \boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{y}\end{matrix}\right\|_2 \leq 1 + \xi, \\ & \|\boldsymbol{D}_i\boldsymbol{\alpha}\|_2 \leq t_i, \quad \forall i \in [\![1, M]\!].\end{aligned} \tag{9}$$

ℓ_∞-norm ($p = \infty$). Another norm which can be employed here is the ℓ_∞-norm, i.e.,

$$\begin{aligned}R_{n,\infty}(\boldsymbol{z}_i, f) &= \|D^{(n)}f(\boldsymbol{z}_i)\|_\infty \\ &= \max_{\{k_1,\ldots,k_n\}\in\{1,\ldots,d\}^n}\left|\frac{\partial^n f(\boldsymbol{z}_i)}{\partial z_{k_1}\ldots\partial z_{k_n}}\right|.\end{aligned}$$

In this case, the resulting optimization problem can be written as a quadratic program (QP) with $2Mm$ linear constraints, where m is the number of partial derivatives of order n:

$$\begin{aligned}\min_{\boldsymbol{\alpha}\in\mathbb{R}^N,\boldsymbol{t}\in\mathbb{R}^M} \ & \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{y}^T\boldsymbol{K}\boldsymbol{\alpha} + \lambda\sum_{i=1}^{M} t_i \\ \text{s.t.} \ & -t_i \leq \boldsymbol{d}_{ik}^T\boldsymbol{\alpha} \leq t_i, \quad \forall (i,k) \in [\![1, M]\!]\times[\![1, m]\!].\end{aligned} \tag{10}$$

Note that by the symmetry of the mixed derivatives, in practice when $d > 1$ and $n > 1$, we have $m < d^n$.

3.4. Choice of the regularization order n

The following discusses the choice n in order to gradually deal with piecewise constant, affine and nonlinear functions.

Piecewise constant (PWC) functions. By choosing $n = 1$, we have $R_{n,p}(f, \boldsymbol{z}_i) = \|\nabla f(\boldsymbol{z}_i)\|_p$, where, for the Gaussian RBF kernel,

$$\nabla f(\boldsymbol{z}_i) = \frac{1}{\sigma^2}\sum_{j=1}^{N}\alpha_j K(\boldsymbol{x}_j, \boldsymbol{z}_i)(\boldsymbol{x}_j - \boldsymbol{z}_i),$$

i.e., in (7), we set $\boldsymbol{D}_i = \frac{1}{\sigma^2}(\boldsymbol{X} - \boldsymbol{1}\boldsymbol{z}_i^T)^T\boldsymbol{K}_i$, where $\boldsymbol{K}_i$ is a diagonal matrix with $(\boldsymbol{K}_i)_{jj} = K(\boldsymbol{x}_j, \boldsymbol{z}_i)$.

With $p = 2$, the regularization term of Problem (8) then becomes closely related to the TV regularization term used in denoising and known to yield piecewise constant solutions due to the well-known staircasing effect [6].

The *top row* of Fig. 1 shows an example where the aim is to learn a PWC target function over $\mathcal{X} = [-10, 10]$ from a noisy data set (left). We first applied kernel ridge regression (KRR) [1] with a Gaussian RBF kernel ($\sigma = 0.2$), but failed to find a satisfactory tuning of the regularization constant, as expected. Indeed, the middle plot of Fig. 1 shows that the value $\lambda = 1$ is already too large to allow the model to correctly estimate the large variations of the function, while being also too small to counter the effect of the noise. On the contrary, the proposed method with $\lambda = 0.005$ (right plot) yields a good model in terms of both noise removal and accuracy of the nonsmoothness approximation. In addition, regularization constants in the range $0.002 \leq \lambda \leq 0.01$ yield satisfactory solutions.

Piecewise affine (PWA) functions. PWA functions have piecewise constant gradient almost everywhere, which can be obtained by applying the regularizer defined for PWC functions to the gradient vector field, i.e., by minimizing $\|(\nabla \otimes \nabla f)(\boldsymbol{z}_i)\|_p$. This amounts to setting $n = 2$ in (6)-(7), and to penalizing the Frobenius norm of the Hessian matrix, $\boldsymbol{H}_i$, for $p = 2$ as $R_{2,2}(\boldsymbol{z}_i, f) = \|Hf(\boldsymbol{z}_i)\|_F = \|\boldsymbol{H}_i\|_F$, or its max-norm with $R_{2,\infty}(\boldsymbol{z}_i, f) = \|\boldsymbol{H}_i\|_{\max}$. In such cases,

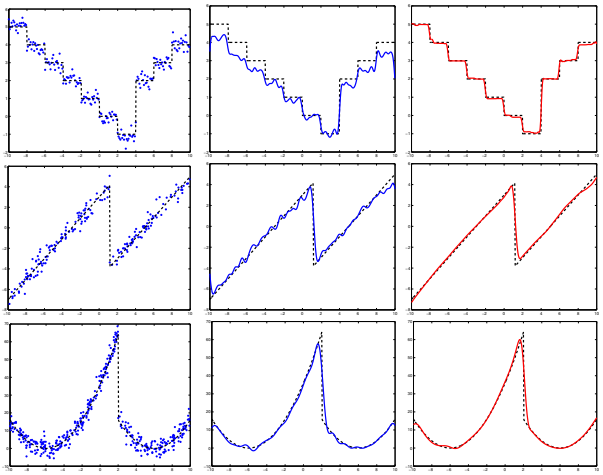

Fig. 1. From top to bottom: examples of piecewise constant (PWC), affine (PWA) and quadratic (PWQ) function approximation. Dash lines: nonsmooth target functions. Left: data points. Middle: KRR. Right: proposed method with $n = 1$ for PWC, $n = 2$ for PWA and $n = 3$ for PWS.

the elements of the Hessian at point $\boldsymbol{z}_i$, are given by

$$(\boldsymbol{H}_i)_{kl} = \frac{1}{\sigma^4} \sum_{j=1}^{N} \alpha_j K(\boldsymbol{x}_j, \boldsymbol{z}_i) \left[(x_{jk} - z_{ik})(x_{jl} - z_{il}) - \delta_{k,l}\sigma^2 \right],$$

and we set the (kl)-th row of $\boldsymbol{D}_i$ to

$$\boldsymbol{d}_{i(kl)}^T = \frac{1}{\sigma^4} \left[(\boldsymbol{X}_k - z_{ik}\mathbf{1}) \odot (\boldsymbol{X}_l - z_{il}\mathbf{1}) - \delta_{k,l}\sigma^2 \mathbf{1} \right]^T \boldsymbol{K}_i.$$

The example in the *middle row* of Fig. 1 considers learning a PWA target function over $\mathcal{X} = [-10, 10]$ with $\sigma = 0.5$. The proposed method (right plot) with $\lambda = 0.01$ can accurately estimate the large central jump while preserving smoothness of the function at all other points. On the other hand, KRR tuned in order to correctly approximate the jump yields a very perturbed model (middle plot). Here, applying KRR with a larger regularization constant would reduce the effect of noise, but also dramatically decrease the accuracy near the nonsmoothness of the target function.

General piecewise smooth (PWS) functions. Regularization of higher order derivatives can be considered in order to learn PWS functions with nonlinear pieces. Here, we present results with third order derivatives particularly suitable for piecewise quadratic (PWQ) functions, while the extension to higher orders is straightforward.

For f defined as in (5) and a Gaussian RBF kernel, we have $\partial^3 f(\boldsymbol{z}_i)/\partial z_k \partial z_l \partial z_m = \boldsymbol{d}_{i(klm)}^T \boldsymbol{\alpha}$, with

$$\begin{aligned}\boldsymbol{d}_{i(klm)}^T = &\frac{1}{\sigma^6} \left[\left((\boldsymbol{X}_k - z_{ik}\mathbf{1}) \odot (\boldsymbol{X}_l - z_{il}\mathbf{1}) \right.\right. \\ &\left.\left. - \sigma^2(\delta_{k,l} + \delta_{k,m} + \delta_{l,m})\mathbf{1} \right) \odot (\boldsymbol{X}_m - z_{im}\mathbf{1}) \right]^T \boldsymbol{K}_i.\end{aligned}$$

In the above, we used the symmetry of the mixed derivatives and the convention that higher orders are computed first, e.g., $\partial^3 f(\boldsymbol{z}_i)/\partial z_1 \partial z_2^2 = \partial^3 f(\boldsymbol{z}_i)/\partial z_2^2 \partial z_1$ and is computed by using $k = l = 2$ and $m = 1$. The last row of Fig. 1 shows an example of such a procedure with $\sigma = 0.5$ and $\lambda = 0.05$.

4. OBTAINING PIECEWISE MODELS

Piecewise smooth functions can be modeled by a collection of s smooth submodels, $\{f_j\}_{j=1}^s$, with a partition of the input space determining which submodel is used to compute the output for a particular input $\boldsymbol{x}$. For a partition $\mathcal{X} = \cup_{j=1}^s \mathcal{S}_j$, PWS models are written as

$$\forall \boldsymbol{x} \in \mathcal{S}_j, \quad f(\boldsymbol{x}) = f_j(\boldsymbol{x}), \qquad j = 1, \ldots, s. \tag{11}$$

For some applications, such models can be easier to handle. In this case, the following simple procedure can be used to transform the solution of (7) into (11).

1. Solve (7) with $\{\boldsymbol{z}_i\}_{i=1}^M = \{\boldsymbol{x}_i\}_{i=1}^N$.
2. Build a new data set $\overline{D}_{\boldsymbol{x}}$ by excluding the points close to the boundaries of the regions $\mathcal{S}_j$ from the training set, i.e., $\overline{D}_{\boldsymbol{x}} = \{\boldsymbol{x}_i \,:\, i \in [\![1, N]\!],\ \|D^n f(\boldsymbol{x}_i)\|_p < \tau\}$.
3. Map these points to a feature space with $\forall \boldsymbol{x}_i \in \overline{D}_{\boldsymbol{x}}$, $\boldsymbol{x}_i \mapsto \varphi(\boldsymbol{x}_i) = [\boldsymbol{x}_i^T,\ D^{(n-1)} f(\boldsymbol{x}_i)^T]^T$.
4. Apply a clustering algorithm, e.g., k-means, in that feature space to estimate the labels q_i for all $\boldsymbol{x}_i \in \overline{D}_{\boldsymbol{x}}$.
5. Build s local submodels, f_j, from the s data subsets, $\mathcal{D}_j = \{\boldsymbol{x}_i \in \overline{D}_{\boldsymbol{x}} : q_i = j\}$, with a classical regression method suitable for the choice of n.
6. Build a classifier h on the data set $\{(\boldsymbol{x}_i, \tilde{q}_i)\}_{i=1}^N$, labeled by $\tilde{q}_i = \arg\min_{j=1,\ldots,s} |y_i - f_j(\boldsymbol{x}_i)|$.

Step 2 above detects points close to the boundaries by thresholding the norm of derivatives used in (7), since the smoothness assumptions on f in (7) and f_j in (11) imply a zero norm inside each region $\mathcal{S}_j$. Note that, after solving (7) through the optimization of (9) or (10), the values of the norm at all points are directly given by the slack variables t_i. Steps 3 and 4 assume that the partition $\cup_{j=1}^s \mathcal{S}_j$ can be represented by a Voronoi diagram. However, more complex partitions can still be represented by increasing the number of regions s. The classifier h estimating the partition in Step 6 can be directly given by the clustering algorithm of Step 4, e.g, k-means yields the centers of Voronoi cells which can be used to estimate labels of new data points. Alternatively, we can train a new classifier with a supervised algorithm to correct potential errors of the clustering algorithm. Predictions for test points $\boldsymbol{x}$ are then given by $f_q(\boldsymbol{x})$, where $q = h(\boldsymbol{x})$.

Figure 2 shows the recovery of the partition of the input space and the submodels by the procedure above on two examples: a PWA target function (the 'ML' shape) and a PWQ function (the 'cursive $\mathcal{M}$L' shape) with 6 pieces each.

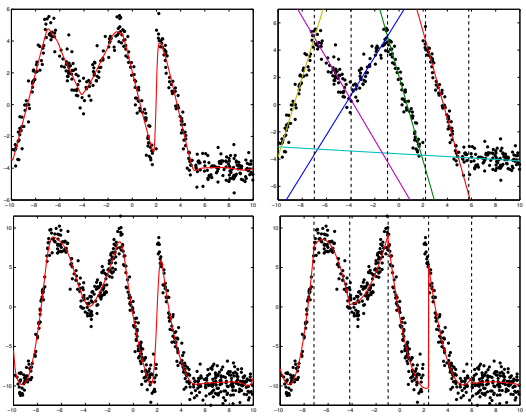

Fig. 2. Learning a PWA (top) and a PWQ (bottom) model. Left: smooth model obtained after Step 1 with $n = 2$ (top) and $n = 3$ (bottom). Right: affine (top) and quadratic (bottom) submodels with the partitions of the input space (dash lines).

Table 1. Comparison of one-step-ahead MSE.

Method	Ref. [5]	SOCP (9)	PWA (Sect. 4)
MSE	4.31 ± 4.01	1.06 ± 0.41	0.81 ± 0.43

5. EXAMPLES AND APPLICATIONS

5.1. Piecewise smooth dynamical system identification

PWS dynamical systems offer a convenient framework to model systems involving both continuous dynamics and discrete events. Such systems can be described by a PWS function f as $y_i = f(\boldsymbol{x}_i)$, where $\boldsymbol{x}_i$ is built from past inputs u_i and outputs y_i of the system [3]. We consider the PWA system studied in the Example 1 of [10], where $\boldsymbol{x}_i = [y_{i-1}, u_{i-1}]^T$. In each of the following 100 experiments, we generate $N = 400$ points with this system and split them in two subsets, a training set and a validation set, of 200 samples each. Another test set with $N_t = 500$ data points is used to computed the one-step-ahead mean square error, MSE $= 1/N_t \sum_{i=1}^{N_t} (y_i - f(\boldsymbol{x}_i))^2$. Problem (9) is solved on the training data while tuning of $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.05, 0.1\}$ is performed on the validation set. We use an RBF kernel with $\sigma = 0.5$. Solving (9) yields a smooth approximation of the PWA system, which is then used in the procedure of Sect. 4 to learn a PWA model with $s = 3$ (in this case, λ is tuned wrt. the validation error of the PWA model). We compare with the PWA system identification method of [5] for a parameter c tuned in the range $\{6, 7, 8, 9, 10, 12, 15\}$. Table 1 shows that the proposed procedures outperform the method of [5] on average. The latter also leads to a large standard deviation of the MSE due to large errors in about 20% of the experiments.

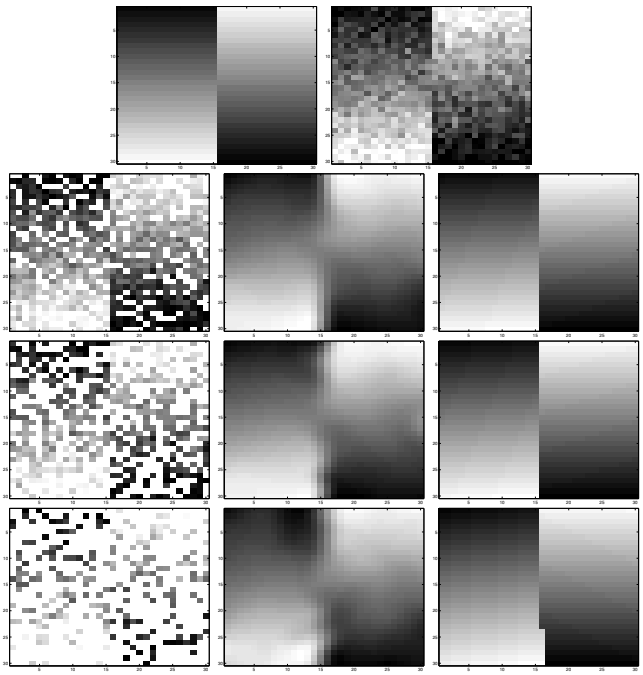

Fig. 3. Top row: original and complete noisy images. Next rows show results obtained with an increasing percentage of missing pixels (25%, 50%, 75%): noisy image with holes (left), smooth model (5) solution to (9) (middle) and PWA model given by the algorithm of Sect. 4 (right).

5.2. Image denoising with missing pixels

Image denoising aims at recovering an original image I from a noisy image, $I_n = I + E$, while preserving edges. Image denoising with missing pixels considers the slightly more difficult task where only parts of a noisy image are available. The proposed framework particularly fits such cases, where reconstructing the entire image simply amounts to computing predictions with the trained model at all pixels, i.e., for an N_x-by-N_y image, for all $\boldsymbol{x} \in [\![1, N_x]\!] \times [\![1, N_y]\!]$. Figure 3 shows the results of such experiments for an original PWA image and Fig. 4 for a piecewise quadratic (PWQ) image. Note that since the proposed approach is a general regression method rather than a dedicated image processing technique, the aim is not to compare the performance with other denoising methods but rather to illustrate potential applications. Figures 3 and 4 also show the results of the procedure of Sect. 4 to obtain piecewise models (11). Note that these PWA and PWQ models are mostly applicable when the optimal target model is truly (or close to) PWA or PWQ. However, the smooth models trained by solving (9) or (10) are always applicable as shown by Fig. 5, where the illuminated peach image is piecewise smooth, but not piecewise quadratic.

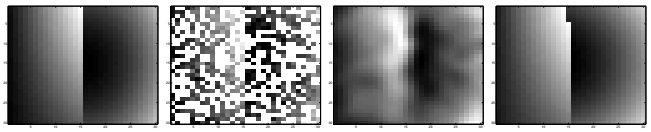

Fig. 4. Left to right: original image, noisy image with 50% of missing pixels, smooth model (5), and PWQ model (11).

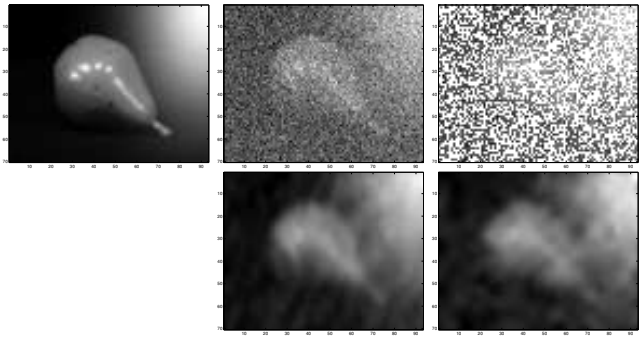

Fig. 5. Top: original and noisy images (0 and 50% of missing pixels). Bottom: smooth models trained by (9) with $n=3$.

6. DISCUSSIONS

When $\mathcal{X}$ is a discrete input space and is completely represented in the data set, the regularization term with $R_{1,2}(\boldsymbol{x}_i, f) = \|\nabla f(\boldsymbol{x}_i)\|_2$ is the total variation of f and, with the square loss, the learning problem (4) is equivalent to the classical ROF model [11] for TV-based denoising. For continuous input spaces, an implicit discretization is given by the sampling of the data set. Note that this constitutes the only discretization in comparison with TV-based methods where f is also typically discretized and only defined through its values at the sampled points. Therefore, the setting is rather different: TV-based methods typically operate on data sets covering the entire discrete input domain, whereas the proposed framework aims at problems in sparsely sampled continuous spaces. Compared with denoising approaches, the proposed method thus offers a predictive model with generalization capabilities. In practice, this model allows the derivatives to be computed explicitly and not through discrete approximations such as finite differences. This also means that dealing with an irregular sampling of the input space or missing data points is straightforward (as with classical learning approaches).

Straightforward extensions to other loss functions may be considered depending on the desired properties of the model (sparsity, robustness to outliers…). For instance, if the ℓ_1-loss or ε-insensitive loss are used with $p = 2$, the first conic constraint in (9) simply becomes a linear constraint; with $p = \infty$, this yields linear programs instead of the QP (10).

Directly solving the constrained programs (9) and (10) can become prohibitive for very large data sets. Future work will consider faster algorithms for the unconstrained optimization of a smoothed version of the cost function (7) and investigate the extension of existing strategies for minimizing the TV functional to the proposed framework. Another research direction with practical consequences concerns the derivation of the full solution path wrt. λ.

Casting the problem as the convex optimization of a regularized risk also paves the way for further analysis of the consistency or error bounds in a classical learning framework.

7. REFERENCES

[1] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *ICML*, 1998, pp. 515–521.

[2] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, and G.E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[3] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems: a tutorial," *European Journal of Control*, vol. 13, no. 2-3, pp. 242–262, 2007.

[4] V.L. Le, G. Bloch, and F. Lauer, "Reduced-size kernel models for nonlinear hybrid system identification," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2398–2405, 2011.

[5] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.

[6] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," *Mathematical Models of Computer Vision*, vol. 17, 2005.

[7] J. Yuan, C. Schnörr, and G. Steidl, "Total-variation based piecewise affine regularization," in *SSVM*, 2009, vol. 5567 of *LNCS*, pp. 552–564.

[8] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Kluwer Academic Publishers, Boston, 2004.

[9] B. Schölkopf, R. Herbrich, and A.J. Smola, "A generalized representer theorem," in *COLT/EuroCOLT*, 2001, vol. 2111 of *LNAI*, pp. 416–426.

[10] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Transactions on Automatic Control*, vol. 50, no. 10, pp. 1567–1580, 2005.

[11] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.