Gray-Box Identification for High-Dimensional Manifold Constrained Regression

Henrik Ohlsson, Lennart Ljung

Division of Automatic Control

E-mail: ohlsson@isy.liu.se, ljung@isy.liu.se

8th April 2009

Report no.: LiTH-ISY-R-2896

Accepted for publication in the 15th IFAC Symposium on System Identification, SYSID 2009

Address:

Department of Electrical Engineering

Linköpings universitet

SE-581 83 Linköping, Sweden

WWW: <http://www.control.isy.liu.se>

AUTOMATIC CONTROL
REGLERTEKNIK
LINKÖPINGS UNIVERSITET

Abstract:

High-dimensional gray-box identification is a fairly unexplored part of system identification. Nevertheless, system identification problems tend to be more high-dimensional nowadays. In this paper we deal with high-dimensional regression with regressors constrained to some manifold. A recent technique in this class is weight determination by manifold regularization (WDMR). WDMR, however, is a black-box identification method. We show how WDMR can be extended to a gray-box method and illustrate the scheme with some examples.

Keywords: Non-parametric identification; System identification; Regularization; Non-parametric regression; Identification algorithms

Gray-Box Identification for High-Dimensional Manifold Constrained Regression

Henrik Ohlsson* Lennart Ljung*

* Division of Automatic Control, Department of Electrical Engineering,
Linköping University, Sweden (e-mail: {ohlsson,ljung}@isy.liu.se)

Abstract: High-dimensional gray-box identification is a fairly unexplored part of system identification. Nevertheless, system identification problems tend to be more high-dimensional nowadays. In this paper we deal with high-dimensional regression with regressors constrained to some manifold. A recent technique in this class is weight determination by manifold regularization (WDMR). WDMR, however, is a black-box identification method. We show how WDMR can be extended to a gray-box method and illustrate the scheme with some examples.

Keywords: Non-parametric identification; System identification; Regularization; Non-parametric regression; Identification algorithms

---

The trend today is to use many inexpensive sensors instead of a few expensive ones, since the same accuracy can generally be obtained by fusing several dependent measurements. It also follows that the robustness against failing sensors is improved. As a result, the need for high-dimensional regression techniques is increasing. High-dimensional regression has previously been discussed by *e.g.* Tibshirani [1996], Hastie et al. [2001].

If measurements are dependent, the regressors will be constrained to some manifold. There is then a representation of the regressors, of the same dimension as the manifold, containing all predictive information. As the manifold is commonly unknown, this representation has to be estimated using data. For this, manifold learning can be utilized. Having found a representation of the manifold constrained regressors, this low-dimensional representation can be used in an ordinary regression algorithm to find a prediction of the output. This has further been developed in the *Weight Determination by Manifold Regularization* (WDMR, Ohlsson et al. [2008]) approach.

In most regression problems, prior information can improve prediction results. This is also true for high-dimensional regression problems. Research to include physical prior knowledge in high-dimensional regression *i.e.*, gray-box high-dimensional regression, has been rather limited, however. Quite recently Rahimi et al. [2007] presented an extension to the *Support Vector Regression* (SVR, Vapnik [1995]) framework to incorporate prior knowledge. In this paper, we explore the possibilities to include prior knowledge in high-dimensional manifold constrained regression by the means of regularization. The result will be called *gray-box WDMR*. In gray-box WDMR we have the possibility to restrict ourselves to predictions which are physically plausible. This is done by incorporating dynamical models for how the regressors evolve on the manifold or for the dynamics of the output.

This paper is organized as follows: The problem is formulated in Section 1. Section 2 introduces the main idea of manifold learning and the manifold learning method *locally linear embedding*. Locally linear embedding gives us a way to express the regressors in a coordinatization of the same dimension as the manifold. These new regressors can be used as regressors in a regression algorithm to predict unknown outputs. However, there may be other coordinatizations more suitable for predicting outputs. The idea of finding a more suitable coordinatization is explored in the so called *Weight Determination by Manifold Regularization* (WDMR), described in Section 3. In Section 4 we take one step further and develop *gray-box WDMR*. In gray-box WDMR we have the possibility to include prior knowledge concerning the observed system's evolution on the manifold. We could for example limit ourselves to (in time) continuous systems or even assume a dynamical model for how the regressors evolve on the manifold. We finish with some examples and a conclusion in Section 5 and 6.

A comprehensive version of the of the theory presented in this paper is given in Ohlsson [2008].

1. PROBLEM FORMULATION

We are interested in finding f , giving the relation between regressors $x \in \mathbb{R}^{n_x}$ and outputs $y \in \mathbb{R}^{n_y}$. If we assume that the measurements of $f(x)$ are distorted by some additive noise e we can write

$$y = f(x) + e. \quad (1)$$

Let us further assume that the regressors are constrained to some manifold $\Omega \subset \mathbb{R}^{n_x}$.

To aid us in our search for f , we are given a set samples $\{x_t, y_t\}_{t \in \mathcal{L}}$ generated from (1) and with $\{x_t\}_{t \in \mathcal{L}} \in \Omega$. Given a new regressor $x_{t_0} \in \Omega$, $t_0 \notin \mathcal{L}$ we would like to be able to give an estimate of the corresponding y produced

by (1). Since our estimate of $y(x_{t_0})$ will be based on the observations, the prediction can be written as

$$\hat{y}(x_{t_0}) = \hat{f}(x_{t_0}, \{x_t, y_t\}_{t \in \mathcal{L}}).$$

With regressors constrained to a manifold, it is many times an advantage to incorporate this information in the predictor

$$\hat{y}(x_{t_0}) = \hat{f}(x_{t_0}, \{x_t, y_t\}_{t \in \mathcal{L}}, \Omega).$$

One reason is that the output commonly varies smoothly along the manifold. This is referred to as the *semi-supervised smoothness assumption* [Chapelle et al., 2006]. As Ω is unknown, it has to be estimated using regressors

$$\hat{\Omega} = \hat{\Omega}(x_{t_0}, \{x_t, y_t\}_{t \in \mathcal{L}}).$$

2. MANIFOLD LEARNING

Manifold learning is a fairly new research area aiming at finding, as the name suggests, descriptions of data on manifolds. The area has its roots in machine learning, and is a special form of nonlinear dimensionality reduction. Some of the most known algorithms are *isomap* [Tenenbaum et al., 2000], *Locally Linear Embedding* (LLE, Roweis and Saul [2000]), *Laplacian eigenmaps* [Belkin and Niyogi, 2003] and *Hessian eigenmaps* (HLE, Donoho and Grimes [2003]).

All manifold learning algorithms take as input a set of points sampled from some unknown manifold. A low-dimensional description of the points (a set of points of the same dimension as the manifold) is then computed by searching for a set of new points preserving certain properties of the data. For example, Laplacian eigenmaps tries to preserve the Euclidean distance between neighboring points. Isomap tries to preserve the geodesic distances *i.e.*, the distance along the manifold, between points and locally linear embedding, and Hessian eigenmaps make assumptions about local linearity which is aimed to be preserved. Most manifold learning algorithms will therefore not give an explicit expression for the map between high-dimensional points and their new low-dimensional representation.

Any manifold learning method could be used, some better suited than others and often depending on the problem, to find a coordinatization of the manifold to which the regressors are constrained. We will here focus on locally linear embedding.

2.1 Locally Linear Embedding

Locally Linear Embedding (LLE, Roweis and Saul [2000]) is a manifold learning technique which aims at preserving neighbors. For a given set of regressors $\{x_1, \dots, x_N\}$ residing on some n_z -dimensional manifold in $\mathbb{R}^{n_x}$, LLE finds a new set of regressors $\{z_1, \dots, z_N\}$, $z_i \in \mathbb{R}^{n_z}$, satisfying the same neighbor-relations as the original regressors. The LLE algorithm can be divided into two-steps:

Step 1: Define the w_{ij} :s

Given data consisting of N real-valued regressors x_i of dimension n_x , the first step minimizes the cost function

$$\varepsilon(\mathbf{w}) = \sum_{i=1}^N \left\| x_i - \sum_{j=1}^N w_{ij} x_j \right\|^2 \quad (2a)$$

under the constraints

$$\begin{cases} \sum_{j=1}^N w_{ij} = 1, \\ w_{ij} = 0 \text{ if } \|x_i - x_j\| > C_i(K) \text{ or if } i = j. \end{cases} \quad (2b)$$

Here, $C_i(K)$ is chosen so that only K weights w_{ij} become nonzero for every i . In the basic formulation of LLE, the number K and the choice of lower dimension $n_z \leq n_x$ are the only design parameters, but it is also common to add a regularization

$$F_r(\mathbf{w}) \triangleq \frac{r}{K} \sum_{i=1}^N [w_{i1}, \dots, w_{iN}] \begin{bmatrix} w_{i1} \\ \vdots \\ w_{iN} \end{bmatrix} \sum_{j: w_{ij} \neq 0} \|x_j - x_i\|^2$$

to (2a), see Roweis and Saul [2000].

Step 2: Define the z_{ij} :s

In the second step, let z_i be of dimension n_z and minimize

$$\Phi(\mathbf{z}) = \sum_{i=1}^N \left\| z_i - \sum_{j=1}^N w_{ij} z_j \right\|^2 \quad (3a)$$

with respect to $\mathbf{z} = [z_1, \dots, z_N]$, and subject to

$$\frac{1}{N} \sum_{i=1}^N z_i z_i^T = I \quad (3b)$$

using the weights w_{ij} computed in the first step. The solution $\mathbf{z}$ to this optimization problem is the desired new low-dimensional representation of the regressors. By expanding the squares we can rewrite $\Phi(\mathbf{z})$ as

$$\begin{aligned} \Phi(\mathbf{z}) &= \sum_{i,j} (\delta_{ij} - w_{ij} - w_{ji} + \sum_l w_{li} w_{lj}) z_i^T z_j \\ &\triangleq \sum_{i,j} M_{ij} z_i^T z_j = \sum_k \sum_{i,j} M_{ij} z_{ki} z_{kj} = \text{Tr}(\mathbf{z} \mathbf{M} \mathbf{z}^T) \end{aligned} \quad (4)$$

with $\mathbf{M}$ a symmetric $N \times N$ matrix with the ij th element M_{ij} . The solution to (3) is obtained by using Rayleigh-Ritz theorem [Horn and Johnson, 1990]. With ν_i the unit length eigenvector of $\mathbf{M}$ associated with the i th smallest eigenvalue,

$$[\nu_1, \dots, \nu_{n_z}]^T = \arg \min_{\mathbf{z}} \Phi(\mathbf{z}) \quad \text{s.t. } \mathbf{z} \mathbf{z}^T = N \mathbf{I}.$$

LLE is an unsupervised method that will find a new representation of the regressors without using any knowledge about the outputs $\{y_t\}_{t \in \mathcal{L}}$. However, since our purpose is to use the representation as new regressors in a regression algorithm, there might be better coordinatizations of the manifold.

3. WEIGHT DETERMINATION BY MANIFOLD REGULARIZATION (WDMR)

In this section we examine the possibilities to find a better suited coordinatization of the manifold by combining the coordinatization and the regression step. The result will be

a regression method adjusted to the manifold constrained regressors. The method was earlier introduced in Ohlsson et al. [2008] and named *Weight Determination by Manifold Regularization* (WDMR).

We can at this point assume that the regressors are ordered such that the labeled estimation regressors are the N_e first regressors $\{x_t\}_{t=1}^{N_e}$, 'e' for *estimation*. The last N_{eu} regressors $\{x_t\}_{t=N_e+1}^{N_e+N_{eu}}$ are then the unlabeled regressors which we seek a prediction for, 'eu' for *end user*. For notational purpose we define $\mathbf{x}_e \triangleq [x_1, x_2, \dots, x_{N_e}]$ and similarly $\mathbf{x}_{eu}$, $\mathbf{y}_e$ and $\mathbf{y}_{eu}$.

First apply the first step of the LLE algorithm (2) to all regressors $\{x_t\}_{t=1}^{N_e+N_{eu}}$ to obtain $\mathbf{M}$ (see (4)). Secondly, to avoid poor coordinatizations, we modify the optimization problem (3) in the second step of the LLE algorithm into

$$\min_{\mathbf{z}_e, \mathbf{z}_{eu}} \lambda Tr([\mathbf{z}_e \ \mathbf{z}_{eu}] \mathbf{M} \begin{bmatrix} \mathbf{z}_e^T \\ \mathbf{z}_{eu}^T \end{bmatrix}) + (1 - \lambda) \|\mathbf{y}_e - f_2(\mathbf{z}_e)\|_F^2 \quad (5)$$

subject to $[\mathbf{z}_e \ \mathbf{z}_{eu}] [\mathbf{z}_e \ \mathbf{z}_{eu}]^T = (N_e + N_{eu}) \mathbf{I}$.

Here, $\|\cdot\|_F$ is the Frobenius norm and f_2 is a function mapping from the coordinatization, z , to the output, y . The parameter λ is a design parameter which can be set to values between 0 and 1. $\lambda = 1$ gives the same coordinatization as LLE and $\lambda = 0$ gives a coordinatization equal to a normalized version of the, possibly noisy, measurements y_e . The function f_2 can be:

- Chosen beforehand.
- Computed by for example alternating between minimizing (5) w.r.t. z and f_2 . However, it is unclear if optimizing over f_2 would improve the results or if there is enough flexibility with a fixed f_2 .

We choose to fix $f_2(z) = z$ and force the coordinatization z to adapt to this.

Remark 1. With $f_2(z) = z$, the coordinatization of the manifold z is assumed to be of the same dimension as the output y . When this is not motivated, we use $f_2(z) = [I_{n_y \times n_y} \ 0_{n_y \times n_z - n_y}] z$. This modification does not introduce any major difficulties. However, for simplicity the derivation in the sequel is done using $f_2(z) = z$.

With a fixed f_2 , the constraint on z can be relaxed since the second term of (5) $(1 - \lambda) \|\mathbf{y}_e - \mathbf{z}_e\|_F^2$ will prevent z from becoming identically zero. The problem is then simplified considerably while many of the properties are still preserved.

The coordinatization z now acts as an estimate of the outputs y . We therefore define $\hat{\mathbf{y}}$ as the coordinatization z that minimizes (5). We then write

$$\begin{bmatrix} \hat{\mathbf{y}}(\mathbf{x})_e^T \\ \hat{\mathbf{y}}(\mathbf{x})_{eu}^T \end{bmatrix} = \arg \min_{\mathbf{z}_e, \mathbf{z}_{eu}} \lambda Tr([\mathbf{z}_e \ \mathbf{z}_{eu}] \mathbf{M} \begin{bmatrix} \mathbf{z}_e^T \\ \mathbf{z}_{eu}^T \end{bmatrix}) + (1 - \lambda) \|\mathbf{y}_e - \mathbf{z}_e\|_F^2 \quad (6)$$

This expression is quadratic in $\mathbf{z}$ and has therefore a solution

$$\begin{bmatrix} \hat{\mathbf{y}}(\mathbf{x})_e^T \\ \hat{\mathbf{y}}(\mathbf{x})_{eu}^T \end{bmatrix} = (1 - \lambda) (\lambda \mathbf{M} + (1 - \lambda) \mathbf{J})^{-1} \begin{bmatrix} \mathbf{y}_e^T \\ 0_{N_{eu} \times n_y} \end{bmatrix}$$

with

$$\mathbf{J} \triangleq \begin{bmatrix} I_{N_e \times N_e} & 0_{N_e \times N_{eu}} \\ 0_{N_{eu} \times N_e} & 0_{N_{eu} \times N_{eu}} \end{bmatrix}.$$

Notice that we get an estimate of the unknown outputs along with the filtered estimation outputs.

The algorithm is an algorithm for computing a weighting-kernel defined by

$$(1 - \lambda) (\lambda \mathbf{M} + (1 - \lambda) \mathbf{J})^{-1}.$$

The kernel accounts for the restriction of data to a manifold and is well consistent with the semi-supervised smoothness assumption. We summarize the WDMR regression algorithm in Algorithm 1. Related formulations have also been developed by Yang et al. [2006].

Algorithm 1 WDMR

Let x_t be the t th element in $[\mathbf{x}_e, \mathbf{x}_{eu}]$, N_e the number of estimation regressors and N_{eu} the number of regressors for which a prediction is searched. For a chosen K , r and λ ,

- (1) Find the weights w_{ij} minimizing

$$\sum_{i=1}^{N_e+N_{eu}} \left\| x_i - \sum_{j=1}^{N_e+N_{eu}} w_{ij} x_j \right\|^2 + F_r(\mathbf{w}),$$

$$\text{subject to } \begin{cases} \sum_{j=1}^{N_e+N_{eu}} w_{ij} = 1, \\ w_{ij} = 0 \text{ if } |x_i - x_j| > C_i(K) \text{ or if } i=j. \end{cases}$$

- (2) With $M_{ij} = \delta_{ij} - w_{ij} - w_{ji} + \sum_k^{N_e+N_{eu}} w_{ki} w_{kj}$ the estimated output is given by

$$\begin{bmatrix} \hat{\mathbf{y}}(\mathbf{x})_e^T \\ \hat{\mathbf{y}}(\mathbf{x})_{eu}^T \end{bmatrix} = (1 - \lambda) \left(\lambda \mathbf{M} + (1 - \lambda) \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{y}_e^T \\ 0_{N_{eu} \times n_y} \end{bmatrix}.$$

---

4. GRAY-BOX WDMR

In this section we present a way to include prior knowledge into the WDMR framework. The result will be a new regression algorithm that we will call *gray-box WDMR regression*.

Since we will include dynamical models in our WDMR framework, we have to introduce the concept of time. We will hence see the set of regressors $\{x_t\}_{t=1}^{N_e+N_{eu}}$ as an ordered sequence of regressors measured at time $t = 1, \dots, N_e + N_{eu}$.

The physical knowledge will now be included in WDMR regression by a regularization term $G(\mathbf{z})$, added to the second step of WDMR, i.e., (6). With the $N_e + N_{eu} \times N_e + N_{eu}$ matrix $\mathbf{M}$ computed using LLE (see (4)) on the set of regressors $\{x_t\}_{t=1}^{N_e+N_{eu}}$, the second step now takes the form

$$\min_{\mathbf{z}} \lambda \sum_{i,j}^{N_e+N_{eu}} M_{ij} z_i^T z_j + (1 - \lambda) \sum_{r \in \mathcal{L}} \|y_r - z_r\|^2 + G(\mathbf{z}).$$

The regularization term $G(\mathbf{z})$ should result in a high cost if the regressors expressed in the coordinatization of the manifold, z , not behave according to the assumed physical model. We could for example assume that there is a linear state space model summarizing our physical knowledge. We write this as

$$s_{t+1} = \mathbf{A} s_t + e_t, \quad e_t = \mathcal{N}(0, \sigma^2), \quad (7a)$$

$$z_t = \mathbf{C}_s s_t, \quad \mathbf{C}_s \text{ a } n_z \times n_s \text{ matrix.} \quad (7b)$$

$s_t \in \mathbb{R}^{n_s}$ is here the state vector of the model. The matrix $\mathbf{A}$ describes how the state evolves into the state of the

next time instance. σ is taken to be a diagonal $n_s \times n_s$ covariance matrix. $\mathbf{C}_s$ specifies how the state s_t relates to our coordinatization of the manifold z_t .

$\mathbf{A}$, σ and $\mathbf{C}_s$ have to be specified in advance. They define our prior knowledge about the data generating system and the desired connection between y and z . We can now define the regularization term $G(\mathbf{z})$ as

$$G(\mathbf{z}) = \min_{\mathbf{s}} \sum_{t=1}^{N_e+N_{eu}} \lambda_A \|s_{t+1} - \mathbf{A}s_t\|_{\sigma^{-2}}^2 + \lambda_S \|z_t - \mathbf{C}_s s_t\|^2.$$

λ_A and λ_S are two design parameters. The norm $\|s_{t+1} - \mathbf{A}s_t\|_{\sigma^{-2}}$ is defined as

$$\|s_{t+1} - \mathbf{A}s_t\|_{\sigma^{-2}}^2 \triangleq (s_{t+1} - \mathbf{A}s_t)^T \sigma^{-2} (s_{t+1} - \mathbf{A}s_t).$$

The first term of $G(\mathbf{z})$ *i.e.*, the term multiplied by λ_A , makes s_t behave according to the assumed dynamics given in (7a). The second term of $G(\mathbf{z})$ *i.e.*, the term multiplied by λ_S enforces the coordinatization z_t to evolve according to the assumed dynamics.

Now with $\mathbf{C}_s$, $\mathbf{A}$ and σ defined by prior knowledge, $\mathbf{M}$ given from the regressors and with λ_S , λ_A and λ design parameters, the resulting minimization takes the form

$$\begin{aligned} \min_{\mathbf{z}, \mathbf{s}} \lambda \sum_{i,j}^{N_e+N_{eu}} M_{ij} z_i^T z_j + (1-\lambda) \sum_{t \in \mathcal{L}} \|y_r - z_r\|^2 \\ + \sum_{t=1}^{N_e+N_{eu}} \lambda_A \|s_{t+1} - \mathbf{A}s_t\|_{\sigma^{-2}}^2 + \lambda_S \|z_t - \mathbf{C}_s s_t\|^2. \end{aligned} \quad (8)$$

Notice that the minimization is now also over the states of the state space model. We hence try to find a coordinatization z which can be well described by the assumed state space model and at the same time fit well with the manifold assumption and the measured outputs.

(8) is quadratic in s_t and z_t . A solution can hence be obtained by setting the gradient equal to zero and then solve the obtained linear equation system for s_t and z_t .

Remark 2. To handle dynamical systems, regressors are typically constructed from delayed inputs and outputs. If the input is high-dimensional, the dimension of the regressor space can grow overwhelmingly fast. Gray-box WDMR avoids increasing the dimensionality of the regressor space by the introduction of the dynamical system (7). The price that has to be paid is that we also have to determine s_t . With a high-dimensional regressor x , this is however often to be preferred.

Remark 3. If the output and the coordinatization of the manifold have the same dimensions, a model (7) can be estimated by computing an *Autoregressive* (AR, see *e.g.* Ljung [1999]) model using the outputs of the estimation data.

Remark 4. If the output and the coordinatization of the manifold do not have the same dimensions, the term $(1-\lambda) \sum_{t \in \mathcal{L}} \|y_r - z_r\|^2$ of (8) should be modified into $(1-\lambda) \sum_{t \in \mathcal{L}} \|y_r - [I_{n_y \times n_y} \mathbf{0}_{n_y \times n_z - n_y}] z_r\|^2$. See also Remark 1.

This work has been inspired by Rahimi et al. [2007] who discusses similar extensions for SVR.

5. EXAMPLES

5.1 Regressors on a Spiral

Consider the system

$$x_{1,t} = \frac{8}{30000} p_t \cos \frac{8}{30000} p_t, \quad (9a)$$

$$x_{2,t} = \frac{8}{30000} p_t \sin \frac{8}{30000} p_t, \quad (9b)$$

$$y_t = \sqrt{x_{1,t}^2 + x_{2,t}^2} + \bar{e}_t = \frac{8}{30000} p_t + \bar{e}_t, \quad (9c)$$

where the output was distorted by some noise $\bar{e}_t \sim \mathcal{N}(0, 0.005)$. Let 225 regressor-output pairs be generated by feeding the system (9) with 225 p -values generated from

$$s_{t+1} = \begin{bmatrix} 1 & 0.1 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} s_t + e_t, \quad p_t = [1 \ 0 \ 0] s_t \quad (10)$$

with $s_{t=0} = [0 \ 0 \ 0]^T$ and $e_t \sim \mathcal{N}(0, 25I)$.

Assume now that of the 225 regressor-output pairs, 13 pairs were picked-out and used as estimation data. The 13 chosen pairs give

$$\mathcal{L} = \{1, 18, 35, 52, 69, 86, 103, 120, 137, 154, 171, 188, 205\}.$$

Figure 1 shows the estimation data. The 212 remaining

Fig. 1. Estimation data used in Example 5.1. * marks the 13 estimation data points. Solid line shows the underlying manifold.

regressors were used as validation regressors. Let us now assume that we know the true dynamic model *i.e.*, (10). We should then use

$$\mathbf{A} = \begin{bmatrix} 1 & 0.1 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0 & 1 \end{bmatrix},$$

$\sigma = 5I_{3 \times 3}$ and, since the manifold has the same dimension as the output (and we therefore use $f_2(z) = z$), $\mathbf{C}_s = [8/30000 \ 0 \ 0]$ in (8). The estimates shown in Figure 2 were obtained by the use of gray-box WDMR regression ($K = 9$, $\lambda = 0.9$, $r = 0.7$, $\lambda_S = 1$, $\lambda_A = 10000$). To compare with, the performance for WDMR regression ($K = 25$, $\lambda = 0.99$, $r = 2$) and LapRLS [Belkin et al., 2006] ($K = 20$, $\gamma_I = 100$, $\gamma_A = 5 \times 10^{-8}$, see (33) in [Belkin et al., 2006]) are shown in Figure 3 respective Figure 4. LapRLS is as WDMR, a regression method for regressors constrained to manifolds.

Fig. 2. Gray-box WDMR: Predicted output using gray-box WDMR regression in Example 5.1. * marks the 13 estimation data points, dashed line the true noise free output and solid line the 225 gray-box WDMR regression predictions (connected with lines).

Fig. 3. WDMR: Predicted output using WDMR in Example 5.1. * marks the 13 estimation data points, dashed line the true noise free output and solid line the 225 WDMR regression predictions (connected with lines).

Fig. 4. LapRLS: Predicted output using LapRLS in Example 5.1. * marks the 13 estimation data points, dashed line shows the true noise free output and solid line the 225 LapRLS predictions (connected with lines).

It may not be that likely to have the exact knowledge of the dynamic system. One could of course estimate an AR model from the estimation outputs. However, as we will see in the next example, a very accurate model is not necessary to get a good result.

5.2 Tracking of Robot

In this example we see images as regressors. The sequence of images used shows an industrial robot and is taken with an ordinary video camera. The video camera took 4 images every second and a total of 111 images. The images consist of 120×160 pixels and the gray-tone in a pixel is represented with a scalar value. By vectorizing the 120×160 matrix which defines an image, we obtain a 19200-dimensional representation associated with each of the 111 images. The 111 images can hence be seen as 111 points in $\mathbb{R}^{19200}$. As the robot has 6 degrees of freedom the 111 points will be constrained to a 6-dimensional manifold in $\mathbb{R}^{19200}$. There is then a 6-dimensional coordinatization of the manifold that could be used to represent each of the 111 images.

Let us now consider the problem of tracking one of the joints of the robot. Since it is tedious to by hand pick out the position of the joint, we would rather not do this in more then a few images. Let's say that we accept to mark out the position of the joint in 6 of the 111 images. We could see this as a regression problem consisting of 6 19200-dimensional labeled regressors. The output is the two-dimensional positions in the image of the joint. And the task is to predict the outputs of the remaining $111 - 6$ regressors.

Since the regressors are constrained to a 6-dimensional manifold the WDMR framework is suitable for this task. We are only interested in a 2-dimensional output and choose therefore to aim for an incomplete 2-dimensional coordinatization of the manifold.

Assume now that we do not have that much information concerning the industrial robot. The same state space model as in Example 5.1 was therefore scaled to fit the dimensions of the tracking problem,

$$s_{t+1} = \begin{bmatrix} I_{2 \times 2} & 0.1I_{2 \times 2} & 0_{2 \times 2} \\ 0_{2 \times 2} & I_{2 \times 2} & 0.5I_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} & I_{2 \times 2} \end{bmatrix} s_t + e_t. \quad (11)$$

C_s was changed to $[I_{2 \times 2} \ 0_{2 \times 2} \ 0_{2 \times 2}]$ to better scale with the movements of the robot (the output of the robot is in the order of tens compared to the output of Example 5.1 which varies between -0.6 and zero). σ was scaled to fit the dimensions of the new state space model (11) and was hence set to $\sigma = I_{6 \times 6}$.

The tracked path of the joint, close to the tool of the industrial robot, is shown in Figure 5. For both WDMR and gray-box WDMR regression $K = 4$, $\lambda = 0.9$, and $r = 0.4$. For gray-box WDMR regression $\lambda_A = 0.1$ and $\lambda_S = 10$.

The paths from the gray-box and ordinary WDMR regression are quite similar even though the path traced by the WDMR tends to be more piecewise linear than a smooth curve. However, what is not seen in Figure 5 is that the path traced out by WDMR regression is temporally not

smooth. This is visualized in Figure 6. Figure 6 shows the difference in predicted joint position between two successive images. WDMR regression has some quite abrupt changes in predicted position (probably not physically possible). Gray-box WDMR regression, however, produces a much more, in time, smooth prediction. So the conclusion is that a fairly simple model is often enough to improve predictions considerably.

Fig. 5. Tracked path of the joint close to the tool of the industrial robot. Dashed line: Gray-box WDMR regression. Solid line: WDMR regression.

Fig. 6. The distance between two successive predicted joint positions. Dashed line: Gray-box WDMR regression. Solid line: WDMR regression.

6. CONCLUSIONS

We have seen how WDMR, a regression method for manifold-constrained regressors, can be extended to incorporate prior assumptions concerning the system to be modeled. The adjustment was shown to improve the prediction performance in two examples. In the first example the true dynamic model was assumed to be known. Predictions were then shown to be almost perfect. The second example gave a more challenging problem. Images of an industrial robot were there seen as regressors and the position of one of its joints, the output to be predicted. With no model of an industrial robot available, the model used in the first example was slightly modified and used to impose continuity. As a result, the predicted position of the joint's movement made more physical sense.

ACKNOWLEDGEMENTS

This work was supported by the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF.

REFERENCES

- Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. ISSN 1533-7928.
- O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, September 2006. ISBN 0262033585.
- David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States of America*, 100(10):5591–5596, 2003.
- T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, August 2001. ISBN 0387952845.
- R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990. Cambridge.
- Lennart Ljung. *System Identification – Theory For the User*. PTR Prentice Hall, Upper Saddle River, N.J., 2 edition, 1999. ISBN 0-13-656695-2.
- Henrik Ohlsson. *Regression on Manifolds with Implications for System Identification*. Licentiate thesis no. 1382, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, November 2008.
- Henrik Ohlsson, Jacob Roll, and Lennart Ljung. Manifold-constrained regressors in system identification. In *Proc. 47th IEEE Conference on Decision and Control*, pages 1364–1369, December 2008. doi: 10.1109/CDC.2008.4739302.
- Ali Rahimi, Ben Recht, and Trevor Darrell. Learning to transform time series with a few examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1759–1775, 2007. ISSN 0162-8828.
- S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by local linear embedding. *Science*, 290(5500):2323–2326, 2000.
- J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society B*, 58:267–288, 1996.
- Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.
- Xin Yang, Haoying Fu, Hongyuan Zha, and Jesse Barlow. Semi-supervised nonlinear dimensionality reduction. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 1065–1072, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.