Multilevel Lasso Applied to Virtual Metrology in Semiconductor Manufacturing

Simone Pampuri$^{(1)}$, Andrea Schirru$^{(1)}$, Giuseppe Fazio$^{(2)}$, Giuseppe De Nicolao$^{(1)}$
$^{(1)}$ University of Pavia, Italy $^{(2)}$ Micron Technologies
{*simone.pampuri, andrea.schirru, giuseppe.denicolao*}*@unipv.it*
giuseppe.fazio@micron.com

***Abstract*— In semiconductor manufacturing, the state of the art for wafer quality control is based on product monitoring and feedback control. The required metrology operations, that usually involve scanning electron microscopes, are cost-intensive and time-consuming. For this reason, it is not possible to measure every wafer: a small subset of a lot (one to three wafers) is measured at the metrology station, and these measurements are designated to represent the whole lot. Virtual Metrology (VM) methodologies aim to obtain reliable estimates of metrology data without actually performing measurement operations. This goal is usually achieved by means of statistical models, linking easily collectible process data to target measurements. In this paper, we tackle two of the most important issues in VM: (i) regression in high dimensional spaces with few meaningful variables (ii) data heterogeneity caused by inhomogeneous production and equipment logistics. We propose a hierarchical framework based on ℓ_1-penalized machine learning techniques and solved by means of multitask learning strategies and multilevel statistical models. The proposed methodology is validated on actual process and measurement data from semiconductor manufacturers.**

INTRODUCTION

In semiconductor manufacturing, metrology operations (usually performed by means of scanning electron microscopes) are so expensive and time-consuming that only a relatively small sample of the production is actually evaluated. Virtual sensors that rely on process data to predict metrology results take the name of Virtual Metrology (VM) tools. A reliable VM tools is expected to increase the amount and readiness of metrology data. The interaction between metrology-related applications (such as Run-to-Run controllers and sampling tools) and such VM tool allows to reduce actual metrology costs while increasing production quality [**?**]. In the design of a VM tool, the main goal is to find and exploit a relationship between easily collectible data (such as sensor readings and equipment setpoints) and metrology results. Such relationship can derive either from physical laws or statistical inference; either way, the core of VM is a mathematical model linking measurements (outputs) to a set of process data (inputs). When the model is established, it can predict metrology results for new wafers at process time, and at no cost. Given such premises, VM tools are unsurprisingly receiving a great deal of attention from semiconductor manufacturers; research directions include algorithm development, interaction between VM and control systems and performance assessment. Two of the most important issue in VM are **(i)** regression in high dimensional spaces with very few meaningful variables and **(ii)** inhomogeneous (and possibly small) datasets due to sampling strategy and inter-chamber variability. In a statistical framework, the high number of sensor readings compared to the small size of metrology datasets leads easily to ill-conditioned problems and calls for techniques able to simultaneously handle input selection and model estimation. Furthermore, production processes usually involve multichamber equipments performing the same step in parallel; in order to obtain reliable estimates, it is necessary to explicitly handle concurrent sources. A real example is depicted in tree form in Figure 1: a three-chamber CVD (Chemical Vapor Deposition) equipment is mainly involved in two production processes. Since every chamber is split in two subchambers, twelve different logistic paths are possible. Intuitively, ignoring the intrinsic differences between chambers would yield suboptimal estimates, while focusing on every different logistic path would produce extremely small datasets.

In this paper, issue **(i)** is tackled by means of ℓ_1-penalized machine learning techniques, while a novel hierarchical framework is proposed in order to deal with issue **(ii)**. The paper is organized as follows:

- Section I reviews basic concepts of machine learning with specific focus on methods able to yield sparse solutions
- Section II defines the proposed hierarchical methodology with reference to the VM theme
- Section III validates the proposed methodology by means of numerical simulations and real datasets provided by semiconductor manufacturers

I. BASICS OF MACHINE LEARNING AND ℓ_1-PENALIZED METHODS

This section provides an introduction to machine learning techniques in reference to the VM theme. Machine learning methodologies are assume that a reliable model can be learned from data: given a training set of n examples $\{x_i, y_i,\ i = 1, \ldots, n\}$ with $x_i \in \mathbb{R}^{1\times p}$ and $y_i \in \mathbb{R}$, let

This work has been developed as a part of European project EU-IMPROVE, dedicated at improving and researching new techniques in quality process control in semiconductor industry. Our best thanks go to Giuseppe Fazio, Carlo Bevilacqua, Andrea Marchelli (Micron Italy), for the constant support and proposal of challenging problems.

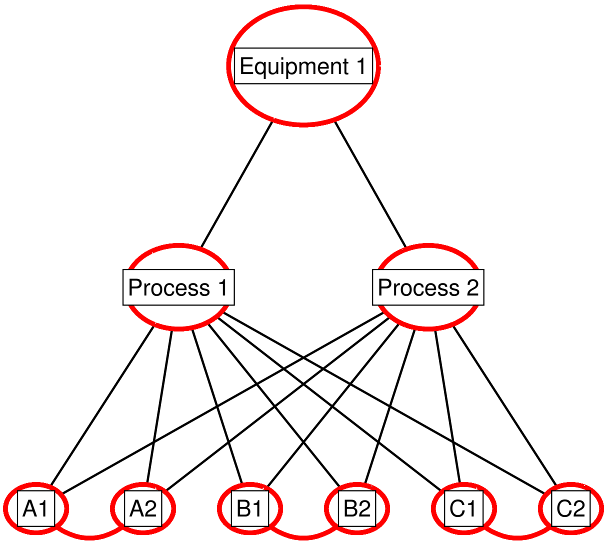

Fig. 1. Tree representation of a CVD equipment with three chambers (A, B, C) with two subchambers each (1 and 2), involved in two processes (Process1 and Process 2)

$X \in \mathbb{R}^{n\times p}$ be the matrix of p-variate inputs obtained by vertical juxtaposition of the x_i's; furthermore, let $Y \in \mathbb{R}^n$ be the associated real-valued output vector. In the motivating example presented in this paper, x_i would be the set of process variables of the i-th wafer, and y_i would be the associated metrology measurement (for instance, layer thickness or critical dimension). The target is then to estimate a function $f : \mathbb{R}^p \to \mathbb{R}$ such that, given a new observation $\{x_{new}, y_{new}\}$, a suitable norm of the differences between $f(x_{new})$ and y_{new} will be small. The most popular choice of $f(x)$ is given by the linear model

$$f(x_i) = \sum_{j=1}^{p} x_{i,j} w_j \tag{1}$$

in which w_j's are unknown coefficients. Typically, the estimation problem of $w \in \mathbb{R}^p$ is solved by the OLS - Ordinary Least Squares method, in which the vector w minimizing the Residual Sum of Squares (RSS):

$$\text{RSS}(w) := \frac{1}{2}\|Y - Xw\|^2 = \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i w)^2 \tag{2}$$

From a statistical point of view, this maximizes the conditional probability $p(Y|X)$ when assuming $Y|X \sim N(Xw, \sigma^2 I)$ or, equivalently, $Y = Xw + \epsilon$ with $\epsilon \sim N(0, \sigma^2 I)$ (i.i.d. Gaussian noise). The optimal coefficient vector w^{OLS} is

$$w^{OLS} = (X'X)^{-1}X'Y \tag{3}$$

and, remarkably, does not depend on σ^2. This very popular approach suffers from two main drawbacks: **(i)** when few observations are available ($n \simeq p$), the estimated $f(x)$ may overfit or even interpolate the training examples, and **(ii)** the matrix $X'X$ may be ill-conditioned or even singular, leading to an unstable solution. In order to overcome these drawbacks, *regularization* techniques have been developed throughout the last century: such methodologies make additional assumptions on the a priori probability of w aiming to reduce the variance of prediction error for new observations. Ridge Regression is perhaps the most popular regularized machine learning algorithm: a linear estimator is obtained by minimizing the loss function

$$J_{RR}(w) := \frac{1}{2}||Y - Xw||^2 + \frac{\lambda}{2}w'w = \text{RSS}(w) + \frac{\lambda}{2}w'w \tag{4}$$

where $\lambda \in \mathbb{R}^+$ is a regularization (hyper)parameter. Under a Bayesian framework, J_{RR} is a logposterior distribution, and the term $\frac{\lambda}{2}w'w$ in (4) is related to the prior distribution of w, $p(w)$, assuming $w \sim N(0, \lambda^{-1}I)$. The larger λ, the smaller the variance of the estimator, at the cost of introducing some bias; in practical applications, λ is often used as a "tuning knob" controlling the bias/variance tradeoff, which is typically tuned either via crossvalidation or other statistical criteria. The optimal coefficient vector w^{RR} and the estimator $f_{RR}(x)$ are

$$\begin{aligned} w^{RR} &= (X'X + \lambda I)^{-1}X'Y && (5) \\ f_{RR}(x) &= x(X'X + \lambda I)^{-1}X'Y && (6) \end{aligned}$$

The numerical stability problems of equation (3) are now avoided, because $(X'X + \lambda I)$ has full rank for any $\lambda > 0$. This approach, however, suffers from the so-called "curse of dimensionality": the number of selected regressors (that is, the variables that are included in the model) grows almost linearly with the number of *candidate* regressors: when dealing with high dimensional spaces, this easily leads to overparametrized models. In order to overcome this issue, it is possible to penalize w under a ℓ_1 norm. The most popular techniques employing such a penalty is the LASSO, that is obtained by solving the following

Problem 1: find

$$w = \arg\min \text{RSS}(w) \tag{7}$$

under the constraint

$$\sum_{j=1}^{p} |w_j| \leq \lambda \tag{8}$$

Remark: an alternative formulation for the LASSO problem can be achieved by using Lagrange multipliers:

$$J_{LASSO} := \frac{1}{2}||Y - Xw||^2 + \frac{\lambda^*}{2}\sum_{j=1}^{p}|w_j| \tag{9}$$

it can be proved that there is a bijective correspondance between the two formulations; in the following sections,

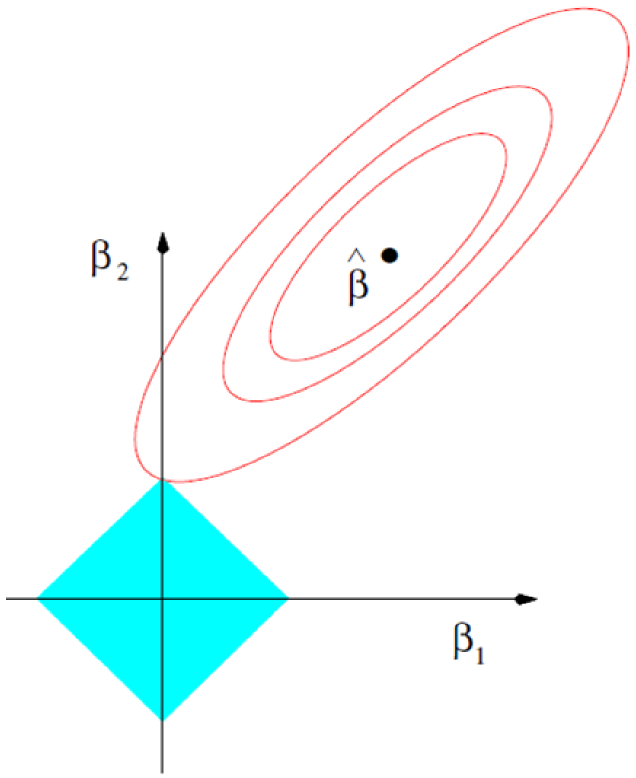

Fig. 2. 2-D graphical example of the sparsity of the LASSO

the formulation from Problem 1 will be consistently used. A probabilistic interpretation of the LASSO is obtained by defining a prior for w as uniform prior over (8). This formulation allows to obtain a sparse solution for w (that is, some entries of the selected w are 0: Figure 2): a full proof is presented in. This extremely convenient property of the LASSO allows for the creation of low-order models even when the input space has high dimension. The hyperparameter λ acts again as a tuning knob: by lowering λ, models of more and more reduced order will be selected. Problem 1 has no closed-form solution: it is necessary to resort to optimization techniques to find LASSO estimates.

In the next section, the properties of LASSO are extended to a hierarchical framework, leading to the definition of the Multilevel LASSO.

II. Multilevel Lasso for Virtual Metrology

In order to extend the properties of ℓ_1 penalization to a hierarchical framework, it is necessary to introduce some definitions: with reference to the graph of Figure 1, let a growing cardinal number $\mathcal{L}$ be conventionally assigned to the η nodes, moving from the root (node $\#0$) to the leaves and from the left to the right; the last node is numbered $\#\eta{-}1$ (Figure 3). Furthermore, let every observation $\{x_i, y_i\}$ follow a logistic path $\mathcal{P}_i = \{\mathcal{L}_0, \mathcal{L}_1, \dots\}$. For instance, if the i-th wafer undergoes "Process 1" and is processed by chamber "A1", $\mathcal{P}_i = \{0, 1, 3\}$. This notation will be used to incorporate logistic informations into the model. It

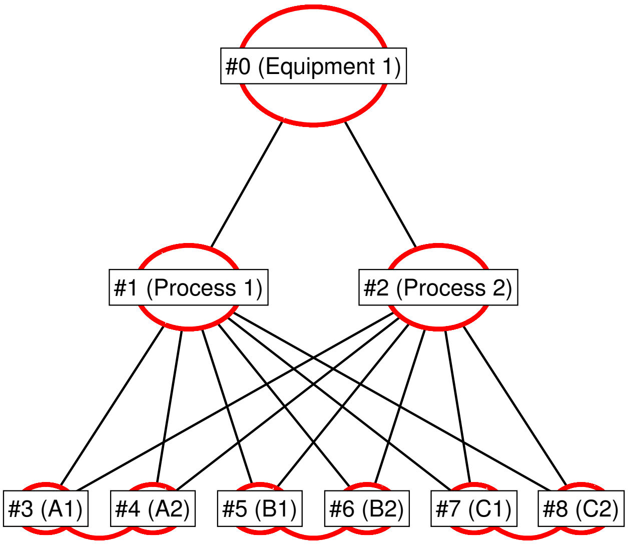

Fig. 3. Numbered representation of the tree in Figure 1

is worth nothing that, in most cases, the functions $f_j(x.)$ will not depend on the entire vector $x.$, but only on a subset of variables here named $x._{,j} \in \mathbb{R}^{1\times p_j}$. For example, only process-related variables within $x.$ will be assigned to "Process" nodes ($\#1$ and $\#2$), while only equipment-related variables will be assigned to "Chamber" ($\#3 - \#8$) and "Equipment" ($\#0$) nodes.

A. Multilevel LASSO

The proposed estimator of a generic $x.$, $f(x., \mathcal{P}.)$, follows a generalized additive structure:

$$f(x.) = \sum_{j\in\mathcal{P}.} f_j(x._{,j}) \tag{10}$$

with

$$f_j(x._{,j}) = x._{,j} w_j \tag{11}$$

where $w_j = [w_{j,1}, w_{j,2}, \dots, w_{j,p_j}]'$ is the coefficient vector associated to the j-th node. It is then necessary to estimate η functions f_j simultaneously. Before introducing the proposed methodology, we focus briefly on tuning parameters. While a single regularization parameter, λ suffices the needs of "single level" LASSO, it is convenient to define a set of regularization parameters for the multilevel case. By defining $\mathcal{G}_k = \{\mathcal{L}_{k,0}, \mathcal{L}_{k,1}, \dots\}$ as a subset of the nodes of the tree, it is possible to insert a new constraint through a regularization parameter $\lambda_k \in \mathbb{R}^+$:

$$\sum_{j\in\mathcal{G}_k} \sum_{z=1}^{p_j} |w_{j,z}| \leq \lambda_k \tag{12}$$

that is, the sum of the absolute values of every coefficient associated to the nodes included in $\mathcal{G}_k$ must be lower than λ_k. For the sake of readability, we will refer to equation (12) as $\mathcal{C}_k$. Given n_λ constraints, the multilevel LASSO problem is defined as

Problem 2: find

$$[w_1, w_2, \dots, w_\eta] = \arg\min \sum_{i=1}^{n} \frac{1}{2} \left(y_i - \sum_{j\in\mathcal{P}.} x_{i,j} w_j \right)^2$$

subject to $\mathcal{C}_1 \cap \mathcal{C}_2 \cap \cdots \cap \mathcal{C}_{n_\lambda}$.

Remark: the admissible region of Problem 2 is convex since it arises as an intersection of convex regions. Furthermore, if $n_\lambda = 1$ and $\mathcal{G}_1$ includes $\mathcal{L}_j$ $\forall j$, the admissible region of Problem 2 is equivalent to the admissible region of Problem 1 with $\lambda_1 = \lambda$. Furthermore, let

$$\overline{\mathcal{C}} = \mathcal{C}_1 \cap \mathcal{C}_2 \cap \cdots \cap \mathcal{C}_{n_\lambda} \qquad (13)$$

The array of regularization parameters

$$\overline{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_{n_\lambda}] \qquad (14)$$

allows to introduce flexible constraints in the proposed model: for instance, with reference to the case discussed in the present paper, n_λ could be 3 with a general λ for all the nodes, a λ_{pr} devoted to control the variability of the ”process” nodes (#1 and #2 in Figure 1), while a λ_{eq} regularizes the variability of ”equipment” nodes (#4 to #9 in Figure 1). This way, the proposed parametrization allows to include in the model prior knowledge about the underlying process.

It is useful to write Problem 2 in matrix form: let

$$\overline{p} = \sum_{j=1}^{\eta} p_j \qquad (15)$$

furthermore, let $Y \in \mathbb{R}^n$ be the observation vector, and let $X \in \mathbb{R}^{n\times\overline{p}}$ be the extended input matrix. The i-th row of X, x_i, is

$$x_i = [\overline{x}_{i,1} \ \overline{x}_{i,2} \ \dots \ \overline{x}_{i,\eta}] \qquad (16)$$

where

$$\overline{x}_{i,j} = \begin{cases} x_{i,j} & j \in \mathcal{P}_i \\ 0_{1\times p_j} & j \notin \mathcal{P}_i \end{cases} \qquad (17)$$

By defining

$$\overline{w} = [w_1' \ w_2' \ \dots w_\eta']' \qquad (18)$$

as the overall column coefficient vector, Problem 2 is then equivalent to the following

Problem 2b: find

$$\overline{w} = \arg\min \frac{1}{2} ||Y - X\overline{w}||_2^2$$

subject to $\mathcal{C}_1 \cap \mathcal{C}_2 \cap \cdots \cap \mathcal{C}_{n_\lambda}$.

B. Error model

In order to produce a probabilistic output for the VM module, we introduce the following

Problem 3: find

$$\overline{w} = \arg\min ||Y - X\overline{w}||_W^2 \qquad (19)$$

subject to $\mathcal{C}_1 \cap \mathcal{C}_2 \cap \cdots \cap \mathcal{C}_{n_\lambda}$.

where W is a diagonal matrix with positive diagonal entries and $||\cdot||_W^2$ is the squared euclidean norm weighted on W. In a probabilistic interpretation, the weighting matrix W allows the observation to be corrupted by independent (but not identically distributed) Gaussian noise: that is,

$$Y|\overline{w} \sim N(X\overline{w}, W^{-1}) \qquad (20)$$

or, in punctual form,

$$y_i|\overline{w} \sim N(x_i\overline{w}, \epsilon_i) \qquad (21)$$

with $\epsilon_i \sim N(0, W_{ii}^{-1})$ and Cov$[\epsilon_i, \epsilon_j] = 0$ if $i \neq j$. To follow the hierarchical structure of the proposed model, we define the i-th diagonal element of W^{-1} as

$$W_{ii}^{-1} = \sum_{j\in\mathcal{P}_i} \sigma_j^2 \qquad (22)$$

where σ_j^2 is a variance associated to the j-th node. The σ_j^2 and λ parameters can be tuned by means of Generalized Cross Validation (GCV).

C. Model estimation

The aim of this section is to define a technique able to estimate the optimal value of $\overline{w}$ in Problem 3 for a given error covariance matrix W^{-1}. We observe that the Jacobian of (19) is

$$J = X'WX\overline{w} - X'WY \qquad (23)$$

while the Hessian is

$$H = X'WX \qquad (24)$$

If there were no constraints, and since H is positive definite, the optimal (in the sense of least squares) estimate of $\overline{w}$ might be computed by ensuring that $J = 0$. This is achieved by solving the linear equation system

$$H\overline{w}^* = X'WY \qquad (25)$$

In order to find $\overline{w}^*$ under the constraints $\mathcal{C}_1, \dots, \mathcal{C}_{n_\lambda}$, it is necessary to resort to an iterative approach: we propose an SMO (Sequential Minimal Optimization) approach, summarized in the following

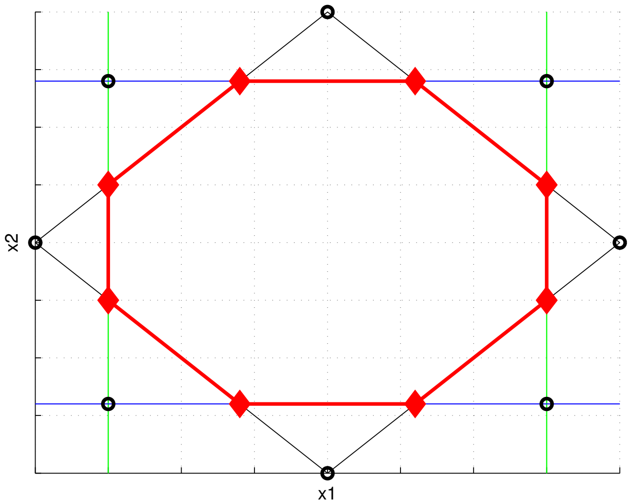

Fig. 4. Borders of the admittable region for a step of Algorithm 1

```
Algorithm 1: solution of Problem 3
  1) Initialize $\overline{w}^* := 0_{\overline{p}\times 1}$
  2) For $i = 1, \ldots,$ maxIter
       a) Compute $J$ and $H$ accordingly to (23) and (25), and let $Z = X'WY$
       b) Select $\mathcal{S}^+$ as the subset of the variables that can increase their absolute value without breaking any constraint
       c) Select $\mathcal{S}^-$ as the subset of the variables that can decrease their absolute value
       d) Select
            $j = \arg\min_{\mathcal{S}^+} J * \mathrm{sign}(\overline{w})$
            $k = \arg\max_{\mathcal{S}^-} J * \mathrm{sign}(\overline{w})$
       e) Solve the linear system
            $\begin{bmatrix} H_{jj} & H_{jk} \\ H_{kj} & H_{kk} \end{bmatrix} \begin{bmatrix} \gamma_j \\ \gamma_k \end{bmatrix} = Z^* = \begin{bmatrix} Z_j \\ Z_k \end{bmatrix}$
       f) If $[\gamma_j \gamma_k]' \in \overline{\mathcal{C}}$,
            $\begin{bmatrix} \overline{w}_j \\ \overline{w}_k \end{bmatrix} = \begin{bmatrix} \gamma_j \\ \gamma_k \end{bmatrix}$
       g) Otherwise, update $[\overline{w}_j \overline{w}_k]'$ performing an SMO step (Algorithm 2).
       h) Break if stop criterion is met
  3) end for
```

Note: in order to implement Algorithm 1, it is necessary to handle correctly the cases of empty $\mathcal{S}^+$ and $\mathcal{S}^-$.

In order to define the SMO step to be performed if (2f)f does not hold, it is necessary to evaluate the score function on the borders of the admittable region. We consider the following

Proposition 1. Given two indexes j and k, the admittable region of $[w_j w_k]'$ is entirely determined by a maximum of 3 conditions $\mathcal{C}_z$.

In the following, let Λ_1 be the least λ parameter that applies to w_j but not to w_k, and let Λ_2 be the least λ parameter that applies to w_k but not to w_j. Furthermore, let Λ_3 be the least λ parameter that applies to both the regressors. If one of these conditions dooe not apply, let the Λ be $+\infty$. It is then possible to define 16 points $(p_1, \ldots, p_{16})$ in the space spanned by w_j and w_k (Table II-C and Figure 4).

	w_j	w_k		w_j	w_k
p_1	0	Λ_3	p_9	Λ_1	$\Lambda_3 - \Lambda_1$
p_2	0	$-\Lambda_3$	p_{10}	$-\Lambda_1$	$\Lambda_3 - \Lambda_1$
p_3	Λ_3	0	p_{11}	Λ_1	$\Lambda_1 - \Lambda_3$
p_4	$-\Lambda_3$	0	p_{12}	$-\Lambda_1$	$\Lambda_1 - \Lambda_3$
p_5	Λ_1	Λ_2	p_{13}	$\Lambda_3 - \Lambda_2$	Λ_2
p_6	$-\Lambda_1$	Λ_2	p_{14}	$\Lambda_2 - \Lambda_3$	$-\Lambda_2$
p_7	Λ_1	$-\Lambda_2$	p_{15}	$\Lambda_3 - \Lambda_2$	Λ_2
p_8	$-\Lambda_1$	$-\Lambda_2$	p_{16}	$\Lambda_2 - \Lambda_3$	$-\Lambda_2$

The set of points defining the boundaries of the admittable region for Problem 3 are chosen as

$$\begin{aligned}
\Lambda_3 \leq \Lambda_2, \Lambda_3 \leq \Lambda_1 &\rightarrow (p_1 \text{ to } p_4)\\
\Lambda_3 \leq \Lambda_2, \Lambda_1 < \Lambda_3 &\rightarrow (p_1, p_2, p_9 \text{ to } p_{12})\\
\Lambda_3 \leq \Lambda_1, \Lambda_2 < \Lambda_3 &\rightarrow (p_3, p_4, p_{13} \text{ to } p_{16})\\
\Lambda_1 + \Lambda_2 \leq \Lambda_3 &\rightarrow (p_5 \text{ to } p_8)\\
\Lambda_1 < \Lambda_3, \Lambda_2 < \Lambda_3, \Lambda_1 + \Lambda_2 > \Lambda_3 &\rightarrow (p_9 \text{ to } p_{16})
\end{aligned}$$

Let p^* be the counterclockwise sorted set of selected points; such sorting of p^* can be obtained by means of the two-argument arctangent function, **atan2**. In order to find the minimum of the score function on the boundaries of the admittable region, the SMO step is performed using the following

```
Algorithm 2: SMO step
1) For $i = 1, \dots, \text{length}(p^*) + 1$
    a) Find the best point on the line connecting $p_i^*$ to $p_{i+1}^*$ as
        $\begin{bmatrix} \gamma_j^{(i)} \\ \gamma_k^{(i)} \end{bmatrix} = p_{i+1}^* + (p_i^* - p_{i+1}^*)\xi_i^*$
       where
        $\xi_i^* = \frac{(Z^* - p_{i+1}^*)' H^* (p_i^* - p_{i+1}^*)}{(p_i^* - p_{i+1}^*)' H^* (p_i^* - p_{i+1}^*)}$
       with
        $(\xi_i^* < 0) \rightarrow (\xi_i^* = 0)$
        $(\xi_i^* > 1) \rightarrow (\xi_i^* = 1)$
2) end for
3) Identify the best point among the $[\gamma_j^{(i)} \gamma_k^{(i)}]'$ and set
    $\begin{bmatrix} \overline{w}_j \\ \overline{w}_k \end{bmatrix} = \begin{bmatrix} \gamma_j^{(i)} \\ \gamma_k^{(i)} \end{bmatrix}$
```

III. Results

In order to validate the proposed methodology, a dataset from the semiconductor industry (courtesy of the Micron Technology facility in Agrate Brianza, Italy) is employed as benchmark. Such dataset consists of 76 wafers with homogeneous recipe coming from 3 different equipments, and includes process data in form of time series and metrology data. The target for prediction is the difference between post-etch and pre-etch Critical Dimensions (ΔCD). The dataset is anonymized and randomly split between training (60 wafers) and test (16 wafers). The hyperparameters λ of the proposed model are tuned by means of Generalized Cross Validation (GCV) on the training dataset, and the Root Mean Squared Error (RMSE) of the predictions upon the test set serves as a comparison criterion for different algorithms. The proposed methodology was compared with two other algorithms:

- Average of the training set: this ”naive” strategy is optimal only if there is no exploitable connection between process data and metrology results, and it is therefore not expected to yield satisfying results. It is, however, a good baseline to determine the improvement of the proposed methodology with respect to a very simple approach.
- Regular (”single-level”) LASSO applied equipment-wise: this approach ignores the commonalities between different equipments; therefore, it is expected to behave worse than the proposed methodology.

Figure 5 shows the prediction capabilities of the proposed methodology on the test set, while Figure 6 shows a boxplot of the prediction error of the three concurrent approaches: as expected, the naive strategy obtains the worst results. Remarkably, the multilevel LASSO is able to outperform the regular LASSO in the majority of the test dataset.

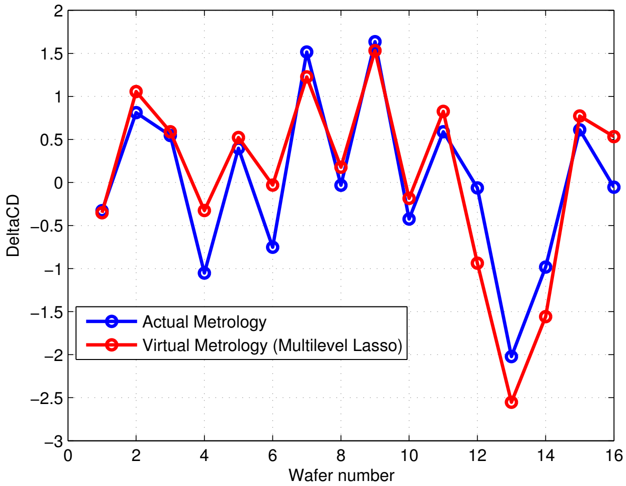

Fig. 5. Predictions of the proposed methodology

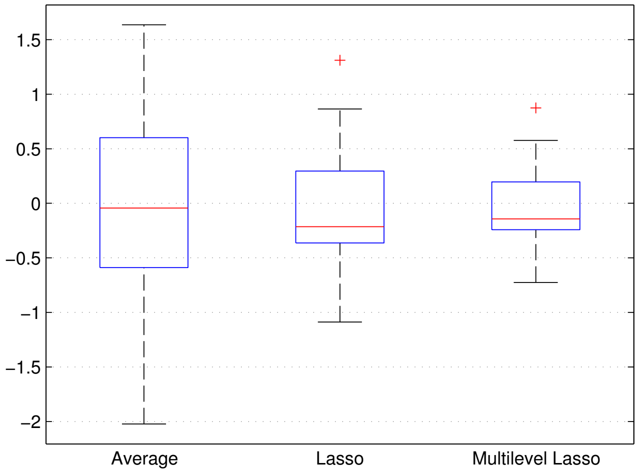

Fig. 6. Boxplot of the prediction error for 3 methodologies

Conclusions

In this paper, a novel approach for Virtual Metrology in semiconductor manufacturing was proposed. The proposed algorithm, namely Multilevel LASSO, is able to obtain models of reduced order by means of a suitable ℓ_1 penalty score. Furthermore, it can handle nested level of variability, exploiting data commonalities to obtain more reliable predictions. Finally, the proposed methodology was tested against data from the semiconductor manufacturing, showing promising performances.