

**Classification algorithms for  
virtual metrology**

S. Tilouche, S. Bassetto,  
V. Partovi Nia

G-2014-84

November 2014

---

Les textes publiés dans la série des rapports de recherche *Les Cahiers du GERAD* n'engagent que la responsabilité de leurs auteurs.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2014.

The authors are exclusively responsible for the content of their research papers published in the series *Les Cahiers du GERAD*.

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2014.



# Classification algorithms for virtual metrology

**Shaima Tilouche**<sup>a</sup>

**Samuel Bassetto**<sup>b</sup>

**Vahid Partovi Nia**<sup>a</sup>

<sup>a</sup> GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7

<sup>b</sup> Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec) Canada, H3C 3A7

shaima-2.tilouche@polymtl.ca  
samuel-jean.bassetto@polymtl.ca  
vahid.partovinia@polymtl.ca

**November 2014**

**Les Cahiers du GERAD  
G–2014–84**

Copyright © 2014 GERAD

**Abstract:** Virtual metrology in quality control deals with drifts in product quality that occur during non-sampling periods. This approach enables a hundred percent control and improves the precision of statistical control, specially while there is no sampling activity in manufacturing process. The main challenge in virtual metrology is inaccurate predictions. As such, the choice of an appropriate algorithm for prediction is crucial. We compare several algorithms that can be used for prediction in virtual metrology. The comparison over different prediction algorithms is made on a simulated data inspired from virtual metrology application.

## 1 Introduction

Semiconductor manufacturing production line includes many steps and requires a high level of accuracy in each step. Several sensors are placed in different locations in order to detect the defects to stop following the manufacturing process for already defected items. Information collected from these sensors are, then, used in quality control. The control is performed in many levels to ensure the stability and the performance of the production process. [2] enumerates some levels that should be taken into account during the control process. These levels are mentioned as layers of control from tools to product. At the product level, electric tests verify proper functionality of chips. These tests are performed at the end of the manufacturing stage [13] [18]. Some other integration tests are applied during the fabrication to verify the properties of technological modules. Tests like verifying that transistor's shape is correctly processed, or examining that electrical properties of several layers of materials are in the desired specifications. After each step, usually it is possible to perform some tests to monitor any abnormal variations of tools that operate a particular process. Finally, at the tool level, numerous sensors are employed to regulate and to monitor the process.

All the tests and the monitoring steps described above are based on few samples of products, except for the electrical wafer sort being the final product. As a consequence, a drift in production can occur in the non-sampling period and this drift is hardly detected [6]. Virtual metrology (VM) algorithms have been suggested as an alternative to 100% wafer measurement, in order to support wafer-to-wafer control [12]. VM can increase metrology data availability, reduce send-ahead wafers, improve quality guarantee levels, and reduce cycle time. The main challenge in using VM is the poor prediction accuracy. Thus, the choice of the algorithm used in VM method is of a great importance. VM refers to classification of products (correct or defected) using some auxiliary information collected during the manufacturing process. We briefly review and compare several classification algorithms that can be used in VM. Section 2 presents a literature review on VM. Section 3 describes the data simulation setup, used to compare the classification algorithms, Section 4 introduces classification algorithms and Section 5 discusses the numerical comparison.

## 2 Virtual metrology

Virtual metrology has been introduced to employ mathematical models on accessible measurements from an operating process with the aim of predicting some variables of interest. This methodology allows to predict relevant variables using equipment measurement, without physically conducting quality measurement [21] [14]. [6] observed a strong correlation between the tool history and the wafer measurement. They found the coefficient of determination  $R^2 > 0.97$  can be achieved with more than 500 wafers on deposition as the response and equipments as the predicting variables. This strong correlation suggests that a wafer-to-wafer control can be quickly enabled using an existing lot-to-lot control system. This indirect technique, called virtual metrology, provides an efficient and economical alternative to wafer-to-wafer control.

Several algorithms have been proposed in the VM literature. [14] suggested a linear regression model to predict the state of the wafer in real time. They proposed to find the coefficients using least squares. The univariate output of their model was the wafer measurement and the inputs were the equipments' measurement. The model was updated as new output measurement was available. [3] studied a virtual metrology model using partial least squares. This model predicts chemical vapor deposition oxide thickness for an *Inter Metal Dielectric* deposition process.

Many VM algorithms have been developed based on neural networks [22]. Neural network is an implicit nonlinear model fitting. Different versions of neural networks have been considered in the VM literature. [15] established a VM model using radial basis network. The effectiveness of the proposed VM system was tested on chemical vapor deposition processes in a practical semiconductor manufacturing. Their result confirmed that neural network can be used effectively to construct a predictive model. [14] adopted a system using back propagation neural network for establishing a model for the etching process in semiconductor manufacturing. [7] compared the performance of the radial basis network and the back propagation neural network on the thin-film transistor liquid crystal display industry. The radial basis function network and the back propagation neural network produced quite similar results. Some other versions of neural models are

proposed to detect wafer anomaly such as polynomial neural network, piecewise linear neural network, and fuzzy neural network, see for instance [5], [4], [11], and [20].

Kernel approaches, specifically support vector machines, are a powerful tool to predict wafer drifts based on equipment measurements [16]. [7] reports that the support vector machines approach give a better prediction accuracy compared with the radial basis function network and also compared with the back-propagation approach, see also [1].

Genetic algorithm is a powerful optimization tool for model fitting [8]. A kernel adjustment is proposed to deal with overfitting problems. [7] combines the support vector machines and the genetic algorithm to construct a virtual metrology system for the chemical vapor deposition process. [23] suggests principal components axes to reduce the dimensionality of plasma dimensions after etching process. It is well-known that uncorrelated features improve the estimation and the prediction of statistical models. The principal components are linear combinations of the inputs and are mutually uncorrelated.

Existence of a large list of different classification algorithms makes the choice of an appropriate algorithm difficult. We aim to compare different classification algorithms proposed in the literature on a simple simulated example motivated from a VM problem.

### 3 Data simulation

We tried to simulate the data according realistic conditions appearing in VM. The inputs, say  $x$ , represent the equipment measurements. In practice, the inputs can be power, pressure, temperature, etc. Some of such inputs are intercorrelated and some others are independent from each other. We simulated the total of 10 inputs. The output variable, say  $y$ , is a binary variable that represents the final product's state, being correct or defected. The following matrix describes the data structure

$$\left( \begin{array}{cccc|c} x_{11} & x_{12} & \cdots & x_{110} & y_1 \\ x_{21} & x_{22} & \cdots & x_{210} & y_2 \\ \vdots & \vdots & & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{n10} & y_n \end{array} \right),$$

where each row corresponds to a measurement of on a wafer. Each column  $\mathbf{x}_j$  is a continuous value of an input variable, say equipment  $j$ , and the last column  $\mathbf{y}$  shows the binary output. The number of rows,  $n$ , is the total number of wafers. The matrix entries  $x_{ij}$  represents the measurement of wafer  $i$  on equipment  $j$  and  $y_i$  is the final state of wafer  $i$ .

We simulated the data with the following structure. Input variables  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are a block of intercorrelated variables; another block of correlated variables contains  $\mathbf{x}_3$ ,  $\mathbf{x}_4$ , and  $\mathbf{x}_5$ . The other inputs  $\mathbf{x}_6$ ,  $\mathbf{x}_7$ ,  $\mathbf{x}_8$ ,  $\mathbf{x}_9$  and  $\mathbf{x}_{10}$  are generated independently, all irrelevant to classify the output. The latter block does not affect the output variable, but they contribute in the classification error generated by the measurement system. Table 1 illustrates the dependence structure of the simulated input variables in which  $\mathcal{N}_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  denotes a  $p$ -variate Gaussian distribution with mean  $\boldsymbol{\mu}$  and variance-covariance  $\boldsymbol{\Sigma}$ .

First, we simulated a binary output  $y_i$  being generated as a function of only three input variables  $x_{i1}$ ,  $x_{i3}$ , and  $x_{i4}$

$$\begin{cases} y_i = 1 & \text{if } \mathbf{a}'\mathbf{z}_i \geq 0, \\ y_i = 0 & \text{if } \mathbf{a}'\mathbf{z}_i < 0, \end{cases} \quad (1)$$

where  $\mathbf{a} = (a_1, a_2, a_3)$  and  $\mathbf{z}_i = (x_{i1}, x_{i3}, x_{i4})$ . This model produces observations that only  $\mathbf{x}_1$ ,  $\mathbf{x}_3$ , and  $\mathbf{x}_4$  are useful for classification, and the other variables are noise. Second, we simulated a binary output using a quadratic function of  $\mathbf{x}_1$ ,  $\mathbf{x}_2$ , and  $\mathbf{x}_4$  is generated

$$\begin{cases} y_i = 1 & \text{if } \mathbf{a}'\mathbf{z}_i + \mathbf{z}_i'\mathbf{A}\mathbf{z}_i \geq 0, \\ y_i = 0 & \text{if } \mathbf{a}'\mathbf{z}_i + \mathbf{z}_i'\mathbf{A}\mathbf{z}_i < 0. \end{cases} \quad (2)$$

**Table 1:** The generated input data structure. Three blocks of input variables are generated of size  $n = 100$  observations.

block	Inputs	Correlation	Distribution
1	$(\mathbf{x}_1, \mathbf{x}_2)$	yes	$\mathcal{N}_2 \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} \right]$
2	$(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5)$	yes	$\mathcal{N}_3 \left[ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.9 & 0.9 \\ 0.9 & 1 & 0.9 \\ 0.9 & 0.9 & 1 \end{pmatrix} \right]$
3	$\mathbf{x}_6, \dots, \mathbf{x}_{10}$	no	$\mathcal{N}_1(0, 1)$

The elements of the vector  $\mathbf{a}$  are sampled independently and uniformly from  $\{-6, -3, 3, 6\}$ , and the elements of the symmetric matrix  $\mathbf{A}$  are sampled from  $\{-6, -3, 0, 3, 6\}$ . We generated  $n = 100$  observations as the training set. A dataset of the same size is generated as the validation set. The model is fitted on the training set, and the precision of the resulting classification is evaluated on the validation set. The total of 20 Monte Carlo simulations have been run.

## 4 Classification algorithms

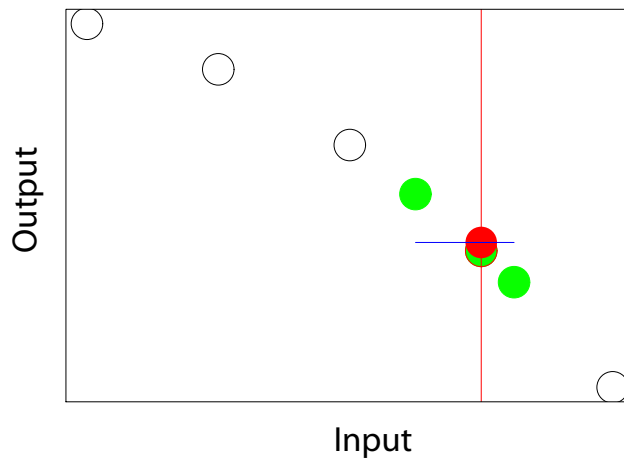
Several classification algorithms listed below are used to predict the output as a function of the inputs.

### 4.1 k-nearest-neighbours

The  $k$ -nearest-neighbours is a model-free algorithm that predicts the output based on its  $k$  nearest neighbours. The nearest neighbours are found using a distance, often the Euclidean distance, computed over the corresponding input variables. Suppose  $N(\mathbf{x})$  is the neighbourhood at point  $\mathbf{x}$  where the  $k$  data fall into, then

$$\hat{y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N(\mathbf{x})} y_i. \quad (3)$$

This technique gives a step function approximation to the classification function, see Figure 1. The tuning parameter  $k$  is chosen manually or is estimated using cross-validation.



**Figure 1:** An illustrative example of a 3-nearest-neighbours algorithm. The circles are observations from an unknown function. The three green blobs are the data that fall in the neighborhood of  $\mathbf{x}$ . The vertical red line represents  $\mathbf{x}$  and the horizontal blue line shows the neighbourhood of size 3, denoted by  $N(\mathbf{x})$  in (3). The output is predicted by the average of the closest 3 points, denoted by the red blob.

## 4.2 Logistic regression

The logistic regression is a generalization of the linear regression where the output variable is binary. This technique is used to predict a binary outcome based on one or more continuous predictor variables. The logistic regression estimates the coefficients of a linear classifier using the conditional distribution of  $y_i | \mathbf{x}_i$ . Since the logistic regression uses a probabilistic model to estimate the classification function, the probability of  $(y = 1 | \mathbf{x})$  can be extracted after the fitting for a given  $\mathbf{x}$ . In order to produce a binary predict, this estimated probability is cut at a certain point, usually 0.5. The probability of  $y_i | \mathbf{x}_i$  is expressed as

$$\Pr(y_i = 1 | x_i) = \frac{\exp(\beta_0 + \mathbf{x}_i' \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i' \boldsymbol{\beta})},$$

where  $\mathbf{x}_i' = (x_{i1}, x_{i2}, \dots, x_{i10})$  and  $\boldsymbol{\beta} = (\beta_1, \dots, \beta_{10})'$ . The regression coefficients are estimated using maximum likelihood. The log likelihood function of the Bernoulli distribution is maximized using *iterative reweighted least squares*. The Bernoulli log likelihood, say  $\ell(\boldsymbol{\beta})$ , is expressed as

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^n \log \left[ \left\{ \frac{\exp(\beta_0 + \mathbf{x}_i' \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i' \boldsymbol{\beta})} \right\}^{y_i} \left\{ 1 - \frac{\exp(\beta_0 + \mathbf{x}_i' \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}_i' \boldsymbol{\beta})} \right\}^{1-y_i} \right]$$

Like linear regression, logistic regression suffers from overfitting and produces unstable estimation of coefficients while a many of noise variables is added in the model. As a remedy the penalized logistic regression is fitted. The penalty term, penalizes large absolute values of model coefficients. The maximizing function is

$$\ell(\boldsymbol{\beta}) - \lambda \|\boldsymbol{\beta}\|_2^2$$

where  $\|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{10} \beta_j^2$  is the squared Euclidean norm of the regression coefficients. Here  $\lambda$  is a positive tuning parameter, usually estimated by cross-validation.

## 4.3 Neural network

A neural network is a set of simple but highly interconnected processing elements, called neurons, to fit a highly nonlinear model, see Figure 2. Neural network has been evaluated for different number of hidden layers with different weights. The most predictive number of layers is chosen. This approach help regularizing this algorithm and avoids overfitting.

## 4.4 Linear discriminant

Linear discriminant analysis separates data into different classes (two classes for a binary output) using a linear hyperplane, see Figure 3 (top left panel). However, linear discriminant coefficients are sensitive to the correlation between the input variables. In order to improve the classification performance, it is proposed to perform the classification on the principal components of data [17]. We applied linear discriminant on four principal components.

An alternative to improve the performance in the presence of correlated variables is penalization. A penalized discriminant analysis is suggested in [9]. Absolute norm penalty, also called the lasso penalty, is applied to the discriminant vectors to encourage variable selection simultaneously.

## 4.5 Quadratic discriminant

Quadratic discriminant analysis, as its name indicates, proposes quadratic boundaries to separate data. This algorithm is similar to the linear discriminant, except it allows for quadratic coefficients as well, see Figure 3 (top right panel). We applied this algorithm on the principal components of data also.



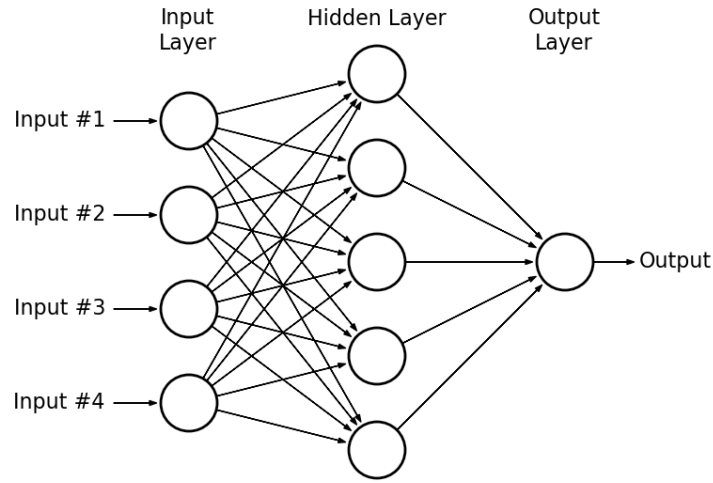


Figure 2: Schematic representation of a neural network model.

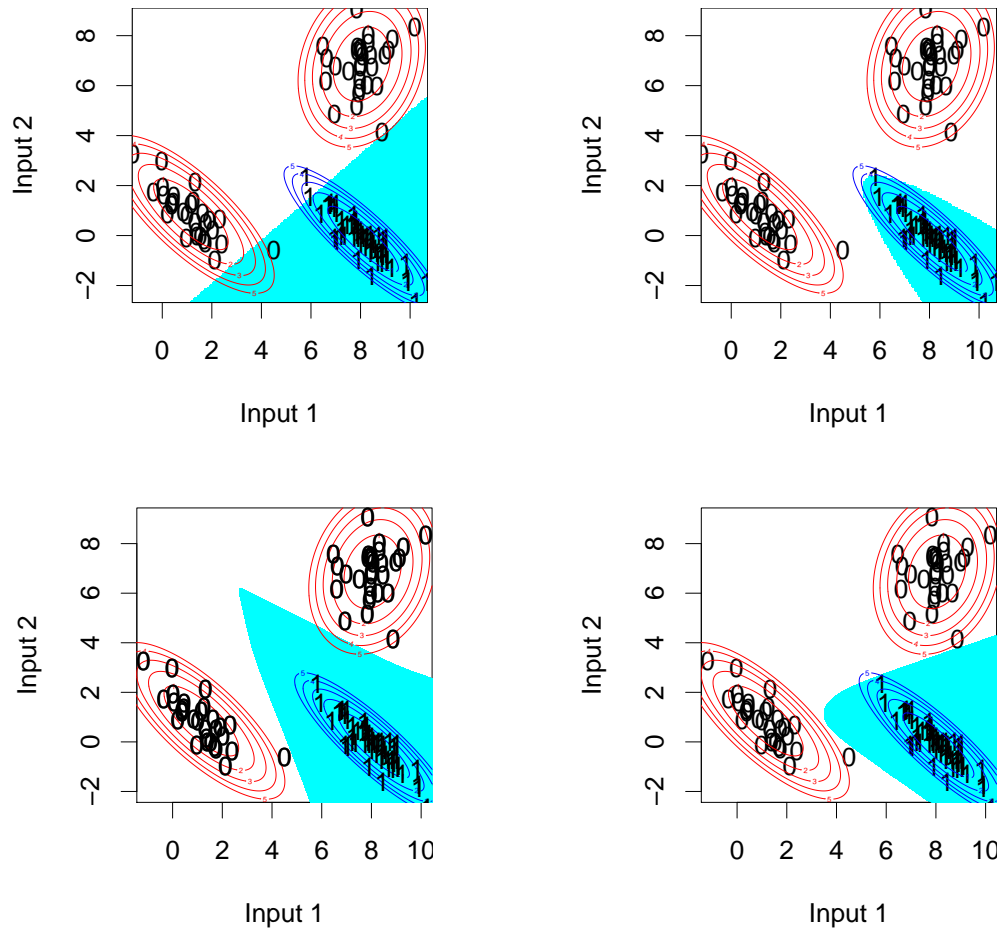


Figure 3: Scatter plot of the quadratic simulated data over Input 1 and Input 2, see also Table 1. Linear discriminant (top left panel), quadratic discriminant (top right panel), mixture discriminant (bottom left panel) and neural networks (bottom right panel) are used to find the decision boundaries.

## 4.6 Mixture discriminant

Polynomial boundaries such as linear and quadratic functions are too restrictive for complex data. Mixture of discriminant functions covers a flexible class of classification functions. This algorithm is called mixture discriminant analysis [10]. A Gaussian mixture model for the  $k$ th class has density

$$\Pr(X|G = k) = \sum_{i=1}^{R_k} \pi_{kr} \phi(X, \mu_{kr}, \Sigma_{kr}),$$

where the mixing proportions  $\pi_{kr}$  sum to one. This has  $R_k$  prototypes for the  $k$ th class, and in our specification, the covariance matrix  $\Sigma_{kr}$  is used as the metric throughout. Given such a model for each class, the class posterior probabilities are given by

$$\Pr(X|G = k) = \frac{\sum_{r=1}^{R_k} \pi_{kr} \phi(X, \mu_{kr}, \Sigma_{kr}) \pi_k}{\sum_{l=1}^K \sum_{r=1}^{R_k} \pi_{lr} \phi(X, \mu_{lr}, \Sigma_{lr}) \pi_l},$$

where  $\pi_l$  represents the class prior probabilities. The parameters of mixture discriminant are estimated using maximum likelihood. The classification obtained through Mixture discriminant is compared with linear and quadratic discriminant and also with neural networks as shown in Figure 3. We can conclude that mixture discriminant gives better results than linear and quadratic discriminant and basically, as good results as neural networks. This good classification result is due to the flexibility of mixture discriminant boundaries.

## 4.7 Kernelized support vector machines

A support vector machine constructs a hyperplane in a high dimensional space. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class, so-called functional margins, see Figure 4. Like the other methods, after computing the hyperplane, the data are categorized into 2 classes. Instead of the linear support vector machines, we tested a more flexible version called kernelized support vector machines. The kernel function transforms the classification problem into a new space defined by the kernel (inner product) on the input variables. We used the radial basis kernel also called Gaussian kernel.

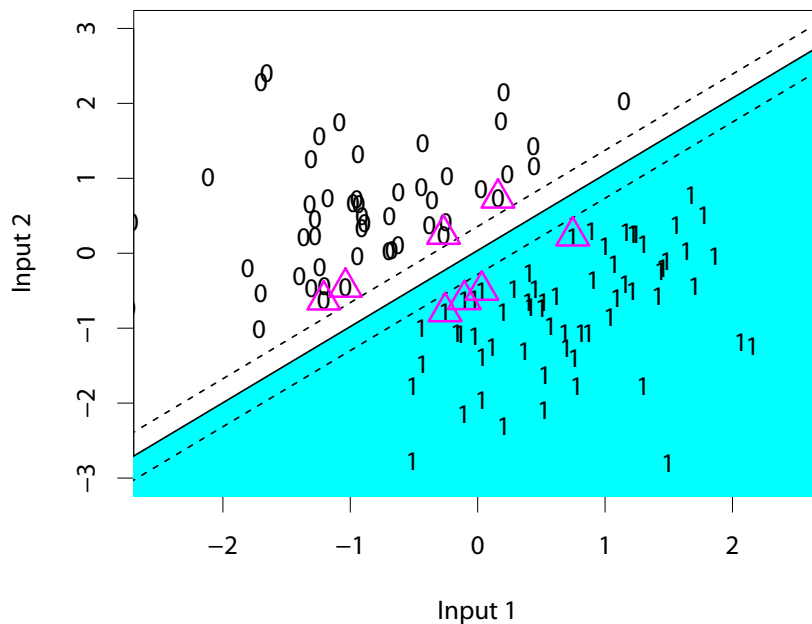


Figure 4: Linear support vector machines shown on a separable illustrative example. The dashed lines show the margins and the data that fall on the margin, shown by triangles, are called the support vectors.

## 5 Numerical results

A Monte Carlo simulation study over the linear and the quadratic models (1) and (2) is summarized in Table 2. The simulation codes are written the statistical programming language R [19]. Simulations are performed using a 2.30 GHz Intel core i5-2410m processor and 6.00 Go RAM, taking around 2 minutes to run all algorithms. Datasets and the R codes are available and will be provided upon request. The correct classification rates of the output variable is summarized in Table 2.

Table 2: The estimated correct classification rates for different algorithms in percentages,  $\hat{p}$ , and their respective 95% confidence lower and upper bounds,  $\hat{p}_L$  and  $\hat{p}_U$ . The results are demonstrated once for the linear simulated data (left) and once for the quadratic simulated data (right).

Algorithm	Linear			Quadratic		
	$\hat{p}_L$	$\hat{p}$	$\hat{p}_U$	$\hat{p}_L$	$\hat{p}$	$\hat{p}_U$
Neural Network	94	94	94	83	84	84
Kernel SVM	88	88	88	82	82	83
MDA-PCA	86	86	87	81	82	82
QDA-PCA	86	87	87	81	82	82
KNN	85	86	85	82	82	82
LDA-PCA	87	87	88	75	76	77
Penalized LDA	87	88	88	76	76	77
Penalized LR	90	91	91	66	67	68
LR	88	89	89	63	64	64

Neural network outperforms all other algorithms for both linear and quadratic data. The logistic regression (LR) and the penalized logistic regression are, also, good classifiers for the linear data. However, they give significantly inferior correct classification rates for the quadratic data. The penalized logistic regression (Penalized LR) improves the rate of correct classification compared to the logistic regression, particularly for the quadratic output. It achieves an increase of 3% (from 64% to 67%). The quadratic discriminant combined with the principal components (QDA-PCA) shows better results than the linear discriminant (LDA-PCA). The QDA-PC on the quadratic output shows 6% increase of the correct classification rate compared to LDA. The mixture discriminant method (MDA) gives results similar to the quadratic discriminant, but better than the LDA-PCA and the Penalized LDA, for the linear output. The kernelized support vector machines (Kernel SVM) gives accurate predictions for both linear and quadratic outputs.

## 6 Conclusion

We briefly reviewed the existing literature on quality control with an emphasis on virtual metrology. We insist that the choice of a proper classification algorithm is of great importance in this area. Therefore, we studied several algorithms that could be used for VM on some simulated data. These algorithms perform differently depending on the outputs (linear or quadratic). However, neural network outperforms all others in both cases. This suggests to keep neural network method as a strong potential candidate for modelling in VM.

## References

- [1] R. Baly and H. Hajj, Wafer classification using support vector machines.semiconductor manufacturing, IEEE Transactions, 25(3), 373-383, 2012.
- [2] S. Bassetto and A. Siadat, Operational methods for improving manufacturing control plans: case study in a semiconductor industry, *Journal of Intelligent Manufacturing*, 20(1), 55-65, 2009.
- [3] J. Besnard, D. Gleispach, H. Gris, A. Ferreira, A. Roussy, C. Kernaflen, and G. Hayderer, Virtual metrology modeling for cvd film thickness, *International Journal of Control Science and Engineering*, 2(3), 26-33, 2012.
- [4] S. Bhatikar and A. Siadat, Operational methods for improving manufacturing control plans: Case study in a semiconductor industry, *Journal of Intelligent Manufacturing*, 20(1), 55-65, 2009.

- [5] Y.J. Chang, Y. Kang, C.L. Hsu, C.T. Chang, and T.Y. Chan, Virtual metrology technique for semiconductor manufacturing, in *Neural Networks, 2006. IJCNN '06. International Joint Conference on*. IEEE, 2006, pp. 5289–5293.
- [6] Y.T. Chen, H.C. Yang, and F.T. Cheng, Multivariate simulation assessment for virtual metrology, in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1048–1053.
- [7] P.H. Chou, M.J. Wu, and K.K. Chen, Integrating support vector machine and genetic algorithm to implement dynamic wafer quality prediction system, *Expert Systems with Applications*, 37(6), 4413–4424, 2010.
- [8] D.E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley Reading Menlo Park, 1989, vol. 412.
- [9] T. Hastie, A. Buja, and R. Tibshirani, Penalized discriminant analysis, *The Annals of Statistics*, 23(1), 73–102, 1995.
- [10] T. Hastie and R. Tibshirani, Discriminant analysis by gaussian mixtures, *Journal of the Royal Statistical Society, Series B*, 58(1), 155–176, 1996.
- [11] K.L. Hsieh and L.I. Tong, Optimization of multiple quality responses involving qualitative and quantitative characteristics in ic manufacturing using neural networks, *Computers in Industry*, 46(1), 1–12, 2001.
- [12] A.A. Khan, J.R. Moyne, and D.M. Tilbury, Virtual metrology and feedback control for semiconductor manufacturing processes using recursive partial least squares. *Journal of Process Control*, 18(10), 961–974, 2008.
- [13] W. Kuo and T. Kim, An overview of manufacturing yield and reliability modeling for semiconductor products. *Proceedings of IEEE*, 87(8), 1329–1344, 1999.
- [14] T.H. Lin, F.T. Cheng, W.M. Wu, C.A. Kao, A.J. Ye, and F.C. Chang, Nn-based key-variable selection method for enhancing virtual metrology accuracy, *Semiconductor Manufacturing, IEEE Transactions on*, 22(1), 204–211, 2009.
- [15] T.H. Lin, M.H. Hung, R.C. Lin, and F.T. Cheng, Virtual metrology scheme for predicting cvd thickness in semiconductor manufacturing, in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 1054–1059.
- [16] K. Mao, Feature subset selection for support vector machines through discriminative function pruning analysis, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(1), 60–67, 2004.
- [17] G. L. Marcialis and F. Roli, Fusion of LDA and PCA for face verification, in *Biometric Authentication*. Springer, 2002, pp. 30–37.
- [18] J. Moyne, E. Del Castillo, and A.M. Hurwitz, *Run-to-run control in semiconductor manufacturing*. CRC Press, 2010.
- [19] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: <http://www.R-project.org/>
- [20] D. Stokes and G. May, Real-time control of reactive ion etching using neural networks, *Semiconductor Manufacturing, IEEE Transactions on*, 13(4), 469–480, 2000.
- [21] G.A. Susto, A. Beghi, and C. De Luca, A virtual metrology system for predicting cvd thickness with equipment variables and qualitative clustering, in *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on*. IEEE, 2011, pp. 1–4.
- [22] J.C. Yung-Cheng and F.T. Cheng, Application development of virtual metrology in semiconductor industry, in *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*. IEEE, 2005.
- [23] D. Zeng and C.J. Spanos, Virtual metrology modeling for plasma etch operations, *Semiconductor Manufacturing, IEEE Transactions on*, 22(4), 419–431, 2009.