Modelling the Yield Learning Process of Semiconductor pro duct using an ANN

Toly Chen

Department of Industrial Engineering and Systems Management, Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung City, Taiwan
Email:tolychen@ms37.hinet.net

Yao-Lei Wang

Department of Industrial Engineering and Systems Management, Feng Chia University, 100, Wenhwa Road, Seatwen, Taichung City, Taiwan
Email:lsps9150302@yahoo.com.tw

Abstract. Estimating the future yield of a product is an important task to a semiconductor manufacturer. However, the existing methods cannot differentiate the effects of various sources of yield improvement. To address this issue, an innovative approach is proposed in this study to model the yield learning process of a semiconductor product with artificial neural networks, which enables the separation of the effects of various sources of yield learning. A real case is used to illustrate the proposed methodology.

Keywords: yield, learning, semiconductor, artificial neural network

1 Introduction

Semiconductor manufacturing is a highly competitive industry. To survive in this industry and to gain an advantage over the rivals, semiconductor manufacturers adopt various strategies including poaching research and development engineers from the rivals [1], keeping investment on the next-generation technologies or staying in the existing market, moving wafer fabrication factories (wafer fabs) to places near demand, switching to fabless, switching to producing more profitable products, becoming foundry, and others [2]. Regardless of the strategy adopted, in a wafer fab, yield, i.e. [3]. For this reason, all wafer fabs have sought to enhance yield. In addition, it is also necessary to estimate the future yield to avoid the misallocation of efforts and resources on products that turn out to have low yields. If an existing product is to be fabricated in a new fab, estimating the possible yield is also a prerequisite to such a migration [4].

Modelling the improvement in the yield of a product as a learning process is a prevalent approach in this field [5]. The parameters in the learning process are usually derived in a Bayesian manner [6]. The yield learning process is subject to a lot of uncertainty, which was tackled by using fuzzy parameters in some studies [7]. However, the fuzziness of the yield estimate is a problem. To control the fuzziness, Chen and Wang [3] proposed a fuzzy collaborative intelligence approach that formed the fuzzy intersection of several fuzzy yield estimates.

Most of the existing methods fit a yield learning model to estimate the future yield. From a novel perspective in this paper, an artificial neural network (ANN) approach is proposed to model a multi-source yield learning model. The remainder of this paper is organized as follows. In Section 2, a single-input perceptron is used to model a single-source yield learning model. A constrained gradient descent (CGD) algorithm is proposed to train the ANN. Then, some discussions are made and conclusions are given.

2 Modelling a Single-source Yield Learning Model with a Single-input Perceptron

The simplest form of ANNs is a perceptron [8]. Inputs to a perceptron, { x_k }, are converted into the output o as:

$$o = \frac{1}{1 + e^{-(\sum_{k=1}^{K} w_k x_k - \theta)}} \quad (1)$$

where w_k is the weight assigned to the k-th input; θ is the threshold. On the other hand, Gruber's general yield learning model [5] portrays the improvement in yield with time as:

$$Y_t = Y_0 e^{-\frac{b}{t} + r(t)} \quad (2)$$

where Y_t is the yield at time period t; Y_0 is the asymptotic/final yield that is a real-valued function of the point defect density per unit area, chip area, and a set of parameters unique to the specific yield model; $b > 0$ is the learning constant; $r(t)$ is a homoscedastical, serially non-correlated error term. $r(t)$ reflects the uncertainty of the yield learning process, but is usually ignored in practical applications. There have been various models for estimating Y_0, as summarized in Table1.

Table 1. Model for estimating Y_0

Distribution	Yield model	Defect density distribution	Reference
Poisson	$Y_0 = e^{-D_0 A}$	-------	1960
Murphy (Poisson-Triangular mixture)	$Y_0 = \left(\frac{1 - e^{-D_0 A}}{D_0 A}\right)^2$	Trangular	Murghy (1964)
Poisson-Uniform mixture	$\frac{1 - e^{-2D_0 A}}{2D_0 A}$	Uniform	Murghy
Seeds (Poisson-Exponential mixture)	$Y_0 = \frac{1}{1 + D_0 A}$	Exponential	Seeds (1967)
Negative Binomial (Poisson-Gamma mixture)	$Y_0 = \frac{1}{(1 + D_0 A/\alpha)^\alpha}$	Gamma	Stapper (1973)
Half-Gasussian (Poisson-Half-Gasussian mixture)	$Y_0 = exp[(\pi/4)D_0^2 A]erfc\left(\frac{\pi^{\frac{1}{2}} D_0 A}{2}\right)$	Half-Gasussian	Stapper (1991)
Poisson-Weibull mixture	$\sum_{k=0}^{\infty}(-1)^k \frac{(D_0 A)^k}{k!} \frac{\Gamma(1+k/\alpha)}{[\Gamma(1+1/\alpha)]^k}$	Weibull	-------
Poisson-Rayleigh mixture	$Y_0 = 1 - D_0 A exp[(D_0 A)^2/\pi]\left(erfc\left(\frac{D_0 A}{\pi^{1/2}}\right)\right)$	Rayleigh	Raghavachari et al. (1996)
Poisson-Inverse Gaussian mixture	$Y_0 = exp\left[\emptyset\left(1 - (1 + (2D_0 A/\emptyset))^{1/2}\right)\right]$	Inverse Gaussian	Raghavachari et al. (1996)

Theorem 1. The values of b and Y_0 in (2) can be derived in a Bayesian manner as [7]:

$$\hat{b} = -\frac{S_{xy}}{S_{xx}} \quad (3)$$

$$\hat{Y}_0 = e^{\frac{1}{T}\sum_{t=1}^{T} \ln Y_t + \frac{\hat{b}}{T}\sum_{t=1}^{T}\frac{1}{t}} \quad (4)$$

where

$$S_{xx} = \sum_{t=1}^{T} \frac{1}{t^2} - \frac{1}{T}\left(\sum_{t=1}^{T} \frac{1}{t}\right)^2 \quad (5)$$

$$S_{xy} = \sum_{t=1}^{T} \frac{\ln Y_t}{t} - \frac{1}{T}\left(\sum_{t=1}^{T} \frac{1}{t}\right)\left(\sum_{t=1}^{T} \ln Y_t\right) \quad (6)$$

(2) can be rewritten as

$$Y_t = e^{\ln Y_0 - \frac{b}{t}} = e^{-(\frac{b}{t} - \ln Y_0)} \quad (7)$$

Adding 1 to the both sides of (7), then inverting them gives

$$\frac{1}{1 + Y_t} = \frac{1}{1 + e^{-(\frac{b}{t} - \ln Y_0)}} \quad (8)$$

which performs exactly the same operation as a single-input perceptron does, with $o = 1/(1 + Y_t)$ $w_1 = b$, $x_1 = 1/t$, and $\theta = \ln Y_0$, as shown in Fig. 1.

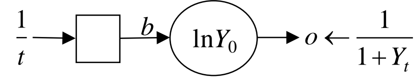

Fig. 1. The single-input perceptron for modelling a single-source yield learning model.

After training,

$$\hat{b} = w_1^* \quad (9)$$

$$\hat{Y}_0 = e^{\theta *} \quad (10)$$

$$\hat{Y}_t = \frac{1}{o} - 1 \quad (11)$$

subject to the requirements that $\hat{b} \geq 0$ and $0 \leq \hat{Y}_0 \leq 1$. In addition, it is easy to see that o has to be between 0.5 and 1 for $\hat{Y}_t$ to be within [0, 1]. Owing to these requirements, the existing training algorithms cannot be directly applied. For this reason, the CGD algorithm is proposed:

1. Specify the learning rate $0 \leq \eta \leq 1$.
2. Specify the initial values of the network parameters.
3. Input the next example $x_1 = 1/t$ to the perceptron, and derive the output o according to (1).
4. Calculate the deviation between the network output and the actual value as

$$\delta = \frac{1}{1 + Y_t} - o \quad (12)$$

5. Calculate the additional modifications that have to be made to the network parameters as

$$\Delta w_1 = -\eta \delta x_1 \quad (13)$$

$$\Delta\theta = -\eta\delta \tag{14}$$

6. If all examples have been learned, go to Step 7; otherwise, return to Step 3.
7. Evaluate the learning performance in terms of the mean squared error (MSE):

$$\text{MSE} = \frac{\sum \delta^2}{n} \tag{15}$$

8. Record the values of the network parameters if
   (a) $w_1 \geq 0$
   (b) $\theta \leq 0$
   (c) $0.5 \leq o \leq 1$
   (d) MSE is less than the smallest MSE that has been recorded.
9. Add the modifications to the corresponding network parameters.
10. If the number of epochs has been reached, or MSE is already less than a threshold, stop; otherwise, return to Step (3).

The complexity of the CGD algorithm is basically a linear function of the number of the training examples.

In addition, since the performance of an ANN is sensitive to the initial values of parameters. To tackle this issue, the initial values of parameters are randomized for a couple of times, from which the one giving the best performance will be chosen.

To illustrate the proposed algorithm, the real case in Chen and Wang [7] is used as an example (see Table 2). Initially, η, w_1, and θ were randomized. The improvement in MSE during the training process is shown in Fig. 2. After 74 epochs of batch learning, the MSE has been less than $4.385{*}10^{-3}$. The fitted yields are summarized in Fig. 3. The results are compared with those obtained using Chen and Wang's method.

Table 2. An illustrative example.

t	Y_t
1	0.7671
2	0.363
3	0.8938
4	0.6678
5	0.9007
6	0.7055
7	0.8836
8	0.6336
9	0.8219
10	0.4589

1. The yield learning process fitted using the proposed methodology is slower than that fitted using Chen and Wang's method. Such a difference is reasonable since the objective functions to be minimized in the two methods are not the same:
   (Chen and Wang's method)

$$\text{Min } \Sigma\left(lnY_t - ln\hat{Y}_t\right)^2 \tag{16}$$

   (the proposed methodology)

$$\text{Min } \sum\left(\frac{1}{1+Y_t} - \frac{1}{1+\hat{Y}_t}\right)^2 \tag{17}$$

2. The asymptotic yields estimated using the two methods are not close. Therefore, the two methods represent different points of view in fitting the yield learning process.
3. The randomization of the initial values of parameters was repeated 100 times, the best initial values $\eta = 0.2$, $w_1 = 0.060916$, $\theta = -0.39029$ and $\hat{Y}_0 = 0.676861$.

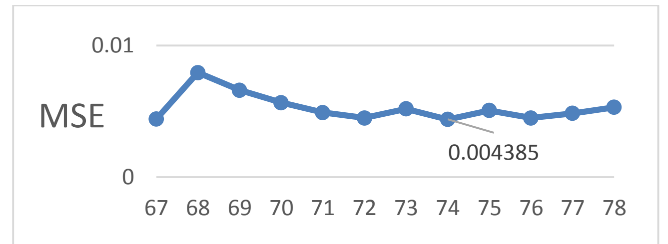

Fig. 2. The improvement of MSE

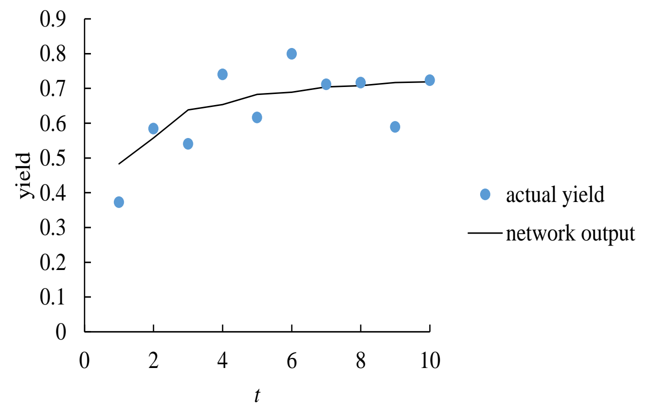

Fig. 3. The estimation results.

3 Conclusions

Estimating the future yield of a product with a learning process is an important task to a semiconductor man ufacturer. Most of the existing methods fit a yield learning process to estimate the future yield. A novel ANN approach is proposed in this study to model a yield learning process. It is also the first attempt to model a yield learning process with a perceptron. In addition, with the

nonlinear approximation capability of an ANN, it is possible to model the complex interactions among the various sources of yield improvement by adding hidden layers. In this way, the ANN becomes a backward propagation network (BPN).

The effectiveness of the proposed methodology needs to be tested with more real cases.

4 Acknowledgment

This study was sponsored by the Ministry of Science and Technology, Taiwan.

References

Chao L.C. and Tong L.I. (2009) Constructing a Wafer Defect Diagnostic System by Integrating Yield Prediction and Defect Pattern Recognition.

EE Times: TSMC Sues Over Trade-secret Leak to Samsung(2015).http://www.eetimes.com/document .ap?doc_id=1325589.

Chen T. (2013) *An Effective Fuzzy Collaborative Forecasting Approach for Predicting the Job Cycle Time in Wafer Fabrication. Computers & Industrial Engineering,* **66**, 834-848.

Chen T. and Wang Y.C. (2014) An Agent-based Fuzzy Collaborative Intelligence Approach for Precise and Accurate Semiconductor Yield Forecasting. *IEEE Transactions on Fuzzy Systems,* **22(1)**, 201-211.

Ahmadi A., Stratigopoulos H.G., Nahar A., Orr B., Pas M., and Makris Y. (2015) Yield Forecasting in Fab-to-fab Production Migration Based on Bayesian Model Fusion. *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design,* 9-14.

Gruber H. (1994) Learning and Strategic Product Innovation: Theory and Evidence for the Semiconductor Industry, *Elsevier Science B. V., The Netherlands*

Mullenix P., Zalnoski J., and Kasten A. J. (1997) Limited Yield Estimation for Visual Defect Sources. *IEEE Transactions on Semiconductor Manufacturing,* **10**, 17-23.

Chen T., and Wang M.J.J. (1999) A Fuzzy Set Approach for Yield Learning Modeling in Wafer Manufacturing. *IEEE Transactions on Semiconductor Manufacturing,* **12(2)**, 252-258.

Freund Y., and Schapire R.E. (1999) Large Margin Classification Using the Perceptron Algorithm. *Machine Learning,* **37(3)**, 277-296.