Machine Learning (ML)-Based Lithography Optimizations

Seongbo Shim[§‡], Suhyeong Choi[§], and Youngsoo Shin[§]
§ School of Electrical Engineering, KAIST, Daejeon 34141, Korea
‡ Samsung Electronics, Hwasung 18448, Korea

***Abstract*—Recent lithography optimizations demand higher accuracy and cause longer runtime. Optical proximity correction (OPC) and sub-resolution assist feature (SRAF) insertion, for example, take a few days due to lengthy lithography simulations and high pattern density. Etch proximity correction (EPC) is another example of intensive optimization due to a complex physical model of etching process. Machine learning has recently been applied to these lithography optimizations with some success. In this paper, we introduce basic algorithms of machine learning technique, e.g. support vector machine (SVM) and neural networks, and how they are applied to lithography optimization problems. Discussion on learning parameters, preparation of compact learning data set, technique to avoid over-fitting are also provided.**

I. Introduction

As feature size scales down, lithography optimizations such as OPC, EPC, and SRAF insertion become more challenging. Traditional model-based OPC (MB-OPC), which iterates lithography simulation and mask image correction [1], becomes more time consuming due to lengthy simulation and a lot of iterations [2]. Similarly, model-based SRAF (MBAF) [3], [4], which relies on repeated SRAF placement and its evaluation through lithography simulations, is very time consuming. Conventional EPC methods are not satisfactory in terms of accuracy due to too simple rules and empirical models [5]–[8].

Machine learning (ML) has recently been adopted to these lithography optimizations for higher speed and better accuracy. ML technique consists of two phases: training and actual testing as shown in Fig. 1. In training phase, ML model receives a vector of parameters ($\mathbf{x}$), such as local densities, extracted from a layout pattern and outputs a predicted value ($f(\mathbf{x})$), such as a mask bias for OPC application. ML model is optimized so that the difference between predicted value and desired value (y) is minimized for all training patterns. In testing phase, parameters extracted from unknown pattern of interest are provided to trained ML model, which then outputs predicted value.

In this paper, we address support vector machine (SVM) and artificial neural network (ANN) in Section II as basic algorithms of ML technique. In Section III, we discuss how ML algorithms are applied to a few lithography optimizations including OPC, EPC, and SRAF insertion. Model parameters and over-fitting are important issues, which we discuss in Section IV. We summarize the paper in Section V.

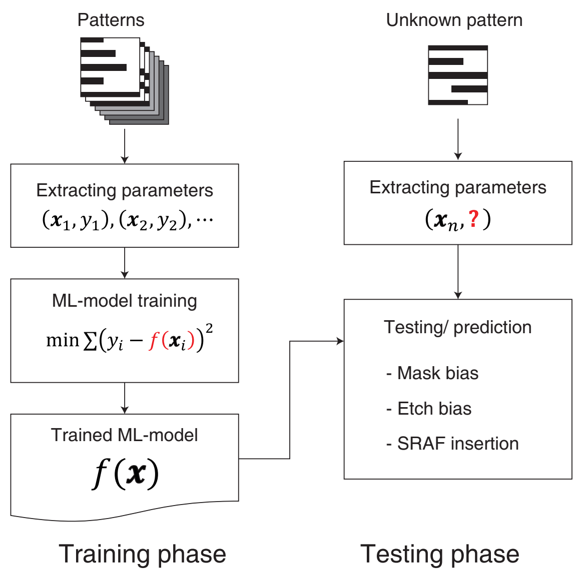

Fig. 1. Typical procedure of machine learning.

II. Machine Learning Algorithms

A. Support Vector Machine (SVM)

SVM is a popular binary classification technique. In SVM, data is parameterized and mapped to corresponding vectors ($\mathbf{x}_i$) in n-dimensional space with binary marks ($y_i = -1$ and $y_i = 1$) as shown in Fig. 2(a). The goal of SVM is to identify a hyper-plane ($h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b$) that separates the two different marks in n-dimensional space with maximizing a margin ($\frac{2}{|\mathbf{w}|}$); the vectors deciding the margin are called support vectors. In case when data are not completely separated by a hyper-plane, a soft margin is used to find a hyper-plane that separates many but not all data. The corresponding problem can be formulated as:

$$\min\ |\mathbf{w}| + W \sum_i \eta_i, \tag{1}$$

subject to:

$$y_i h(\mathbf{x}_i) \geq 1 - \eta_i, \qquad \forall\ i, \tag{2}$$
$$\eta_i \geq 0, \qquad \forall\ i, \tag{3}$$

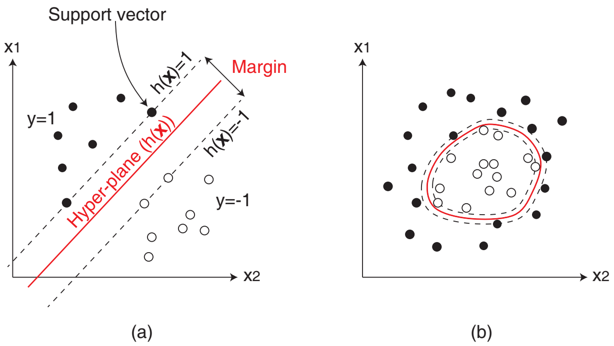

Fig. 2. SVM in (a) linear parameter space and (b) non-linear parameter space.

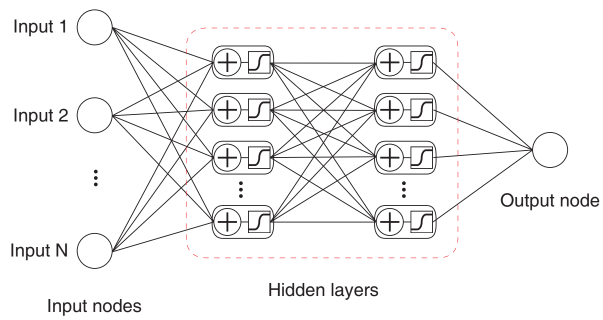

Fig. 3. Neural network.

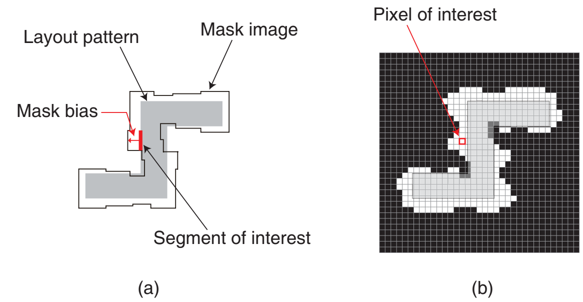

Fig. 4. (a) Edge-based OPC and (b) pixel-based OPC.

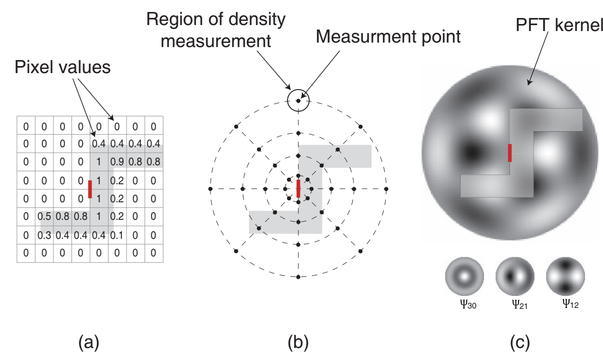

Fig. 5. Extracting parameters from a layout pattern: (a) pixel values, (b) local densities, and (c) kernel signals.

where η_i are slack variables, and W is very large constant used as a penalty parameter; this can be solved by using Lagrange multiplier.

In many practical problems, however, the distribution of data vectors is not linear to the value of parameters as shown in Fig. 2(b). For these problems, we can employ kernel trick, which is to map the vectors to the higher dimensional space using radial basis function (RBF), e.g. a Gaussian function; in the higher dimensional space, nearly all simplicity of a separating hyper-plane can be retained.

B. Artificial Neural Network (ANN)

ANN is a model inspired by biological neural network, which consists of neurons (corresponding to nodes in Fig. 3) and synapses (corresponding to edges). Input nodes receive some parameters extracted from a layout. Their values are propagated to every hidden nodes in the first hidden layer; each edge has a weight, which is multiplied by a propagated signal value. Weighted signals that are received by a hidden node are summed up; if the sum is larger than a threshold, hidden node outputs 1 and it outputs 0 otherwise. Thresholding is often approximated by using a sigmoid function, which outputs a floating value between 0 and 1. This process is repeated until an output node is reached, which finally yields a predicted output.

ANN is trained and optimized (to determine edge weight and threshold value) so that the predicted output matches desired value as closely as possible. The resulting ANN is used as our ML model for lithography optimizations.

III. Applications to Lithography Optimizations

A. OPC

OPC tries to find the mask image that produces the desired litho image on a wafer. Depending on how the mask image is represented, there are two types of OPC: edge-based and pixel-based.

In edge-based OPC, a mask image is corrected in a way that each edge segment of a layout is moved inward or outward as shown in Fig. 4(a); the extent of the correction is called a mask bias. ML-OPC in this context is to predict the mask bias of each segment through ML-technique instead of repeated simulation and correction. To construct an ML model, a large test layout and its mask image that is corrected by MB-OPC are prepared; a segment of interest in the layout is represented by some parameters, e.g. pixel values, local densities, or optical kernel signals as shown in Fig. 5; an ML model receives the parameters and outputs a predicted mask bias of the segment; the ML model is trained for all (or some) sampled segments in the layout such that the predicted mask bias can be as close as possible to the mask bias from the MB-OPC. For a new layout, each segment is parameterized similarly, and the trained ML model receives the parameters and then outputs its predicted mask bias; this is performed for every segment in the new layout, and a corrected mask

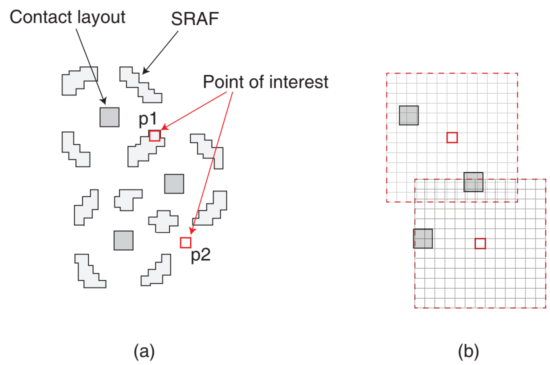

Fig. 6. (a) MBAF result and (b) parameter extraction from a point of interest for ML-SRAF.

is then obtained by applying the predicted mask biases to all segments at once.

In pixel-based OPC, on the other hand, a mask image is represented by an array of pixels, whose values are binary as shown in Fig. 4(b); the mask image is corrected by modifying the binary value of each pixel. In ML-OPC for the pixel-based mask image, parameters are extracted from a pixel of interest (and its surroundings) in a layout and inputted to an ML model, which then outputs a predicted binary value of the pixel. This process is repeatedly performed for every pixel in a mask image; since a mask image is sometimes too complex to manufacture, so some subsequent processes (such as MRC check and simplification) should follow for mask manufacturability.

ML-OPC leaves some errors on a mask image, which may cause some edge placement errors (EPEs) on a wafer. So, it is suggested to perform ML-OPC for a good initial mask, which is then submitted to subsequent MB-OPC with small iterations for fine optimization [9]. In our experiment, OPC time is reduced by 67% compared to MB-OPC being applied alone.

B. SRAF Insertion

SRAFs are small patterns placed close to main patterns (e.g. contacts) to improve manufacturability of the main patterns as shown in Fig. 6(a). The position of SRAF is determined in a way that lights from the SRAF and the main pattern constructively interfere [10]. There are two conventional approaches of SRAF placements such as rule-based SRAF (RBAF) and model-based SRAF (MBAF). In RBAF [11], [12], a few empirical rules are established and are applied, manually or automatically; this is only limited to simple assist features. MBAF [3], [4] relies on repeated SRAF placement and its evaluation through lithography simulations; it is accurate but very time consuming.

In ML-SRAF, some MBAF results are prepared as shown in Fig. 6(a). Local contact layout near an interesting point is extracted and pixelated as shown in Fig. 6(b). The pixel values are used as parameters and inputted to ML model, which then outputs a binary value. If there is a SRAF at an interesting point in the MBAF result (see p1 in Fig. 6(a)), the ML model should outputs 1, and otherwise 0 (e.g. p2). The ML model is trained for many sample points from the MBAF results. The trained model is applied to every points in a new layout, and the resulted SRAFs are simplified for manufacturability.

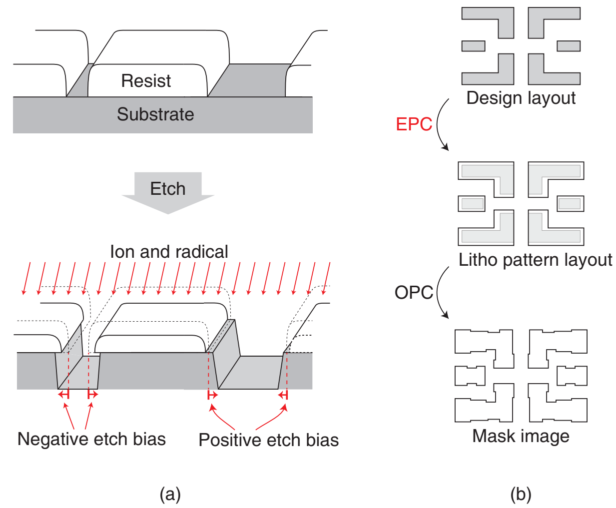

Fig. 7. (a) Etch proximity effect and (b) etch proximity correction (EPC).

In state-of-the-art of ML-SRAF [13], parameters are extracted by using the constrained concentric circle with area sampling (CCCAS), which is to measure local pattern densities at some sample points along many concentric circles as shown in Fig. 5(b). A decision tree and logistic regression are adopted as a binary ML model to handle a lot of training points. About 10X speed up and comparable performance in terms of edge placement error (EPE) and process variation band (PVB) are observed.

C. EPC

Etch process may yield over-etched substrate due to lateral erosion or under-etched substrate due to redeposition as shown in Fig. 7(a) [14]. This phenomenon is called etch proximity effect and the extent of over- or under-etch is called etch bias. A design layout should be modified through etch proximity correction (EPC) to compensate for etch bias before it is submitted to OPC as illustrated in Fig. 7(b).

Etch bias is affected by the amount of particles (e.g. ions or radicals) that collide with substrate surface together with their incidence angle and direction [15]. EPC often relies on a few geometric rules or empirical kernels [7], [8], but these conventional approaches cannot accurately correct various patterns [5], [6].

ANN has been employed for better accuracy of etch bias prediction [16]. Similar with ML-OPC, parameters are extracted from an interesting segment and its surroundings in a layout pattern; these parameters become the inputs of ANN, which outputs predicted etch bias of a segment. The accuracy of output etch bias from the network is determined by the chosen parameters. A local pattern density is measured at a few positions along concentric circles as shown in Fig. 5(b); the density value corresponds to the amount of etching particles, and the measurement position is associated with the incidence angle and direction of the particles. Optical kernel signals (Fig. 5(c)) are also included as a parameter to take account of a side wall angle of the resist, which affects etch bias [14]. The local densities and optical signals model etching phenomena better than the parameters in conventional model-based EPC in physical point of view. Demonstration on 20nm node DRAM gate layout indicate that prediction accuracy of etch bias increases by 23%, which eventually results in better OCV of gate size with 2.4% smaller 3-sigma [16].

IV. Considerations on Efficiency

A. Model Parameters

The accuracy of ML model and runtime are affected by chosen parameters for training. Pixelating a layout pattern (Fig. 5(a)) is most popular way, but too many parameters are required to cover a layout within the optical influence range. CCCAS (Fig. 5(b)) extracts less parameters, and circular propagation of diffracted light from a mask pattern is represented well [9], [13], [16]; but, the number of parameters is still more than some hundreds. Alternatively, optical kernel signal has been proposed [2], [16] to represent a layout with much smaller parameters. Optical kernels, e.g. basis functions of polar Fourier transform and SOCS kernels, are widely used to model optical diffraction and interference. Since OPC, EPC, and SRAF insertion are all affected by layout's optical characteristic, it is natural to use the optical kernel signals as parameters for ML-based lithography optimizations.

B. Over-fitting

Since many variables are involved in ML model training, e.g. variables of a hyper-plane in SVM and weight and threshold values in ANN, the ML model may become too specific to the training data, which may degrade prediction accuracy for unknown data; this is called over-fitting.

In SVM, the width of the kernel (i.e. the sigma of a Gaussian function) determines the specificity of a hyper-plane to the training data, and too narrow kernel may cause over-fitting. In ANN, as the number of layers and nodes increase, the ANN is more likely to be specific to training data. To avoid this, we should carefully determine the width of kernel in SVM and the complexity of ANN. We can adjust them through k-fold cross-validation. The set of data is randomly divided into k subsets of equal size: $k-1$ of these sets are used to construct an ML model while the remaining set is used to assess the proportion of correctly predicted data. The accuracy is averaged over k iterations. This adjustment procedure is embedded into an optimization that determines the kernel width and the number of layers and nodes so that the highest accuracy can be achieved.

V. Summary

Machine learning technique has recently been applied to lithography optimizations for higher speed and better accuracy. ML-OPC builds a model that predicts a mask bias for OPC process; some remaining errors can be corrected by employing MB-OPC with a few iterations. In ML-SRAF, a binary ML model is constructed to determine whether SRAF is necessary for every position near a given layout. Another application of ML is EPC, in which a model is built to predict etch bias. Model parameters and over-fitting are important issues in ML-based optimizations, which have been discussed.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2015R1A2A2A01008037)

References

[1] N. Cobb, "Fast optical and process proximity correction algorithm for integrated circuit manufacturing," Ph.D. dissertation, University of California at Berkeley, Feb. 1998.

[2] S. Choi, S. Shim, and Y. Shin, "Machine learning (ML)-guided OPC using basis functions of polar Fourier transform," in *Proc. SPIE Advanced Lithography*, Mar. 2016, pp. 1–8.

[3] S. Jayaram, P. LaCour, J. Word, and A. Tritchkov, "Effective model-based sraf placement for full chip 2d layouts," in *Proc. SPIE Advanced Lithography*, Mar. 2013, pp. 1–9.

[4] B. Yenikaya, O. Alexandrov, Y. Kwon, A. Liu, A. Mokhberi, and A. Sezginer, "Model-based assist features," in *Proc. SPIE Advanced Lithography*, Mar. 2009, pp. 1–6.

[5] M. Salama and A. Hamouda, "Efficient etch bias compensation techniques for accurate on-wafer patterning," in *Proc. SPIE Advanced Lithography*, Mar. 2015, pp. 1–7.

[6] J. Park, S. Kim, S. Shim, S. Suh, and H. Oh, "The effective etch process proximity correction methodology for improving on chip cd variation on 20 nm node DRAM gate," in *Proc. SPIE Advanced Lithography*, Mar. 2011, pp. 1–10.

[7] S. Shang, Y. Granik, and M. Niehoff, "Etch proximity correction by integrated model-based retargeting and OPC flow," in *Proc. SPIE Advanced Lithography*, Mar. 2007, pp. 225–232.

[8] Q. Liu, R. Cheng, and L. Zhang, "Study of model based etch bias retarget for opc," in *Proc. SPIE Advanced Lithography*, Mar. 2010, pp. 1–6.

[9] T. Matsunawa, B. Yu, and D. Z. Pan, "Optical proximity correction with hierarchical Bayes model," in *SPIE Advanced Lithography*, Mar 2015, pp. 1–10.

[10] M. Mukherjee, S. Mansfield, L. Liebmann, A. Lvov, E. Papadapoulou, M. Lavin, and Z. Zhao, "The problem of optimal placement of sub-resolution assist features (SRAF)," in *Proc. SPIE Advanced Lithography*, May 2005, pp. 1417–1429.

[11] S. Jayaram and J. Word, "Automatic SRAF size optimization during OPC," in *Proc. SPIE Advanced Lithography*, Mar. 2009, pp. 1–10.

[12] J. Kuang, W. Chow, and E. F. T. Young, "A robust approach for process variation aware mask optimization," in *Proc. Des. Autom. Test Europe Conf. Exh. (DATE)*, Mar. 2015, pp. 1591–1594.

[13] X. Xu, T. Matsunawa, S. Nojima, C. Kodama, T. Kotani, and D. Z. Pan, "A machine learning based framework for sub-resolution assist feature generation," in *Proc. Int. Symp. Phys. Des.*, Apr. 2016, pp. 161–168.

[14] C. Wu, W. Yang, L. Luan, and H. Song, "Photoresist 3D profile related etch process simulation and its application to full chip etch compact modeling," in *Proc. SPIE Advanced Lithography*, Mar. 2015, pp. 1–8.

[15] Y. Granik, "Correction for etch proximity: new models and applications," in *Proc. SPIE Advanced Lithography*, Mar. 2001, pp. 98–112.

[16] S. Shim and Y. Shin, "Etch proximity correction through machine-learning-driven etch bias model," in *Proc. SPIE Advanced Lithography*, Mar. 2016, pp. 1–10.