Laplacian Eigenmaps and Bayesian Clustering Based Layout Pattern Sampling and Its Applications to Hotspot Detection and OPC

Tetsuaki Matsunawa[1], Bei Yu[2], and David Z. Pan[3]

[1]Toshiba Corp., Semiconductor & Storage Products Company, Yokohama, Japan
[2]CSE Department, The Chinese University of Hong Kong, NT, Hong Kong
[3]ECE Department, University of Texas at Austin, TX, USA
tetsuaki.matsunawa@toshiba.co.jp, byu@cse.cuhk.edu.hk, dpan@ece.utexas.edu

Abstract

Effective layout pattern sampling is a fundamental component for lithography process optimization, hotspot detection, and model calibration. Existing pattern sampling algorithms rely on either vector quantization or heuristic approaches. However, it is difficult to manage these methods due to the heavy demands of prior knowledges, such as high-dimensional layout features and manually tuned hypothetical model parameters. In this paper we present a self-contained layout pattern sampling framework, where no manual parameter tuning is needed. To handle high dimensionality and diverse layout feature types, we propose a nonlinear dimensionality reduction technique with kernel parameter optimization. Furthermore, we develop a Bayesian model based clustering, through which automatic sampling is realized without arbitrary setting of model parameters. The effectiveness of our framework is verified through a sampling benchmark suite and two applications, lithography hotspot detection and optical proximity correction.

I. Introduction

As the feature size of semiconductor transistors continues shrinking, it is more and more important to verify the complicated mask, so that the overall process cost can be reduced and the manufacturing yield can be improved. Machine learning based techniques have been demonstrated to be effective in several IC manufacturing applications, such as mask optimization [1], hotspot detection [2], and lithography verification [3]. The common goal of these methods is to learn a highly accurate prediction model with a small amount of data. Apart from the development of learning algorithm, an effective layout pattern sampling method is also critical to these industrial applications, as the types of training or test data will greatly affect the prediction model performance.

To reduce the training time of mask synthesis and process model calibration, minimum set of test patterns shall be extracted and sampled to reflect key characteristics in real layouts while maintaining high prediction accuracy [4]. For example, in hotspot detection problem, balanced test patterns between non-hotspots and real hotspots are required to prevent the overfitting issue [5]. However, automatically extracting essential components from real layouts tends to be difficult because there are innumerable pattern variations in real layouts and the number of dimensions in layout data is high. This is known as an unsupervised problem in machine learning in which some hidden structures must be determined from the given unlabeled data.

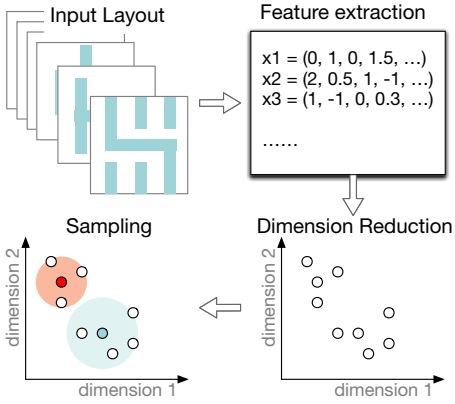

Fig. 1. Example of layout pattern sampling.

Fig. 1 illustrates a typical flow of layout pattern sampling. Given input layout, first the feature extraction transfers the geometric information into a set of high dimensional vectors. Then dimension reduction is to identify the critical features. At last, on the simplified dimension space, clustering is carried out to select the sampling results.

So far, several pattern sampling works have been proposed to acquire a set of test patterns. Some clustering techniques have been proposed for test pattern sampling [6–8]. Those related works contribute to design automation by extracting feature vectors that represent characteristics of layout patterns and training a classification model based on the feature vectors. However, it is difficult to directly apply identical clustering technique to different sampling problems, with the following two reasons. Firstly, a criterion for defining pattern similarity to evaluate essential characteristics in real layouts is unclear. Secondly, most clustering algorithms require several preliminary experiments because there are some parameters that must be tuned in advance such as the total number of clusters.

In this paper, we propose a new pattern sampling framework for creating appropriate test patterns from a given layout effectively. Our key contributions are summarized as follows.

- We develop an efficient feature comparison method with nonlinear dimensionality reduction technique with kernel parameter optimization.

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

- We develop an automated pattern sampling method using Bayesian model based clustering without manual parameter tuning.
- We demonstrate promising test pattern extraction under industrial-strength test chips.

The rest of the paper is organized as follows. Section II gives the problem formulation. Section III introduces the overall flow of layout pattern sampling. Section IV and Section V present details of two key algorithms, dimensionality reduction and clustering. Section VI lists the experimental results, followed by the conclusion in Section VII.

II. Problem Formulation

To quantify the sampling performance and to compare diverse layout feature types, a clustering result evaluation method is needed. In this work we apply **Bayes Error** [9] to evaluate the degree of overlapping clusters based on Bayes' theorem. Bayes error (BE) is defined as follows:

$$BE = \int \min\{1 - P(\omega_k|x)\}P(x)dx \tag{1}$$

where x is a given feature vector, $P(\omega_k|x)$ is a conditional probability in class $\omega_k(i = k, ..., K)$ that indicates a probability of erroneously determining x, K is the total number of classes, and $P(x)$ is a prior probability of x. BE accurately expresses a quality of distributions among clusters.

The problem formulation of layout pattern sampling is given as follows.

Problem 1 (Layout Pattern Sampling)**.** *Given layout data, a classification model is trained to extract representative patterns. The goal is to classify the layout patterns into a set of classes minimizing the Bayes error.*

Layout pattern sampling can be realized by vector quantization, which maps data sets of vector representations to the limited number of representative patterns called *centroids*. The main algorithm to acquire the representative patterns is clustering, which is an unsupervised learning toward a classification model that sorts given data into multiple categories. It should be noted that it is easy to recognize one dimensional layout patterns, but for general two dimensional random layout patterns, sampling is very difficult.

III. Overall Flow

The overall flow of our automated layout pattern sampling is illustrated in Fig. 2, which consists of "Sampling" phase and "Application" phase. In the **sampling** phase, first a set of features are extracted from given layout. Then dimensionality reduction is carried out to simplify the feature space (Section IV). At last, all the features are clustered, and the patterns located in the center of each cluster are used as test patterns (Section V). In the **application** phase, extracted test patterns are used for various purposes, such as prediction model training for lithography hotspot detection, mask optimization or process simulation and so on. Note that the quality of extracted patterns can be measured through several applications on layout level, such as hotspot detection, mask optimization, and wafer inspection.

In the layout feature extraction, different from conventional window based scanning, a DRC based feature point generation is proposed to identify the key windows. Therefore, we can reduce the scanning window number. Besides, our framework is robust enough that all the feature extraction techniques in previous works (e.g., [2]) can be seamlessly integrated.

IV. Dimensionality Reduction

The total number of dimensions in a layout feature can be more than thousands for some complexed feature types (e.g., image feature). In such high-dimensional space, due to the concentration on the sphere issue [10], it is extremely difficult to train a prediction model. Principal component analysis (PCA), which reduces dimensions by transforming data into values of a linearly uncorrelated axis [5], is the most commonly used dimensionality reduction technique. Although PCA allows us to reduce high-dimensional feature vectors into a lower-dimensional space, it has a disadvantage in that the existing cluster structure in original data is not preserved. We will further discuss the disadvantage of linear dimensionality reduction in Section B. To avoid this issue and to handle different types of layout features, we propose an effective nonlinear dimensionality reduction technique.

A. Laplacian Eigenmaps (LE)

Our nonlinear dimensionality reduction technique is based on a Laplacian Eigenmaps (LE) [11] while preserving the existing cluster structure. LE effectively reduces complicated feature structures using a kernel method. The embedded matrix $\Psi = (\psi_{n-1}, \psi_{n-2}, ..., \psi_{n-m})^T$ is calculated by solving the following generalized eigenvalue problem:

$$L\psi = \gamma D\psi \tag{2}$$

where $L = D - W$ is the Laplacian matrix, $D = \text{diag}(\sum_{i'=1}^{n} W_{i,i'})$ is the diagonal matrix, γ is the matrix of the eigenvalues in $(\gamma_1 \geq ... \geq \gamma_n)$, and $W_{i,i'}$ is the kernel representing a similarity matrix for k-nearest neighbors (kNN) defined as 1 if $x_i \in kNN(x_{i'})$ and 0 otherwise. Compared with PCA, LE can effectively map a original data into a lower-dimensional space while maintaining the existing cluster structure. Furthermore, it is advantageous in that it can be applied to any kind of feature vectors because the kernel design provides a lot of flexibility. In contrast, because LE uses a kernel method, characteristics of embedded feature space highly depend on the kernel parameter setting. In this paper, we propose an automatic kernel parameter optimization method based on the difference between input feature vectors and an embedded feature vectors.

B. Kernel Parameter Optimization

Density ratio estimation is a method to directly estimate the density ratio between the two probability distributions without each probability distribution. The kernel parameter can be optimized through the density ratio estimation. We optimize the kernel parameter with the Kullback-Leibler Importance Estimation Procedure ($KLIEP$) because the optimiza-

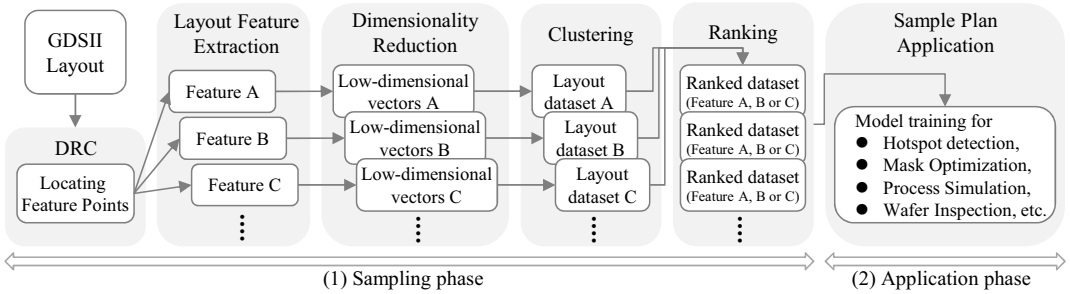

Fig. 2. The overall CAD flow for pattern sampling.

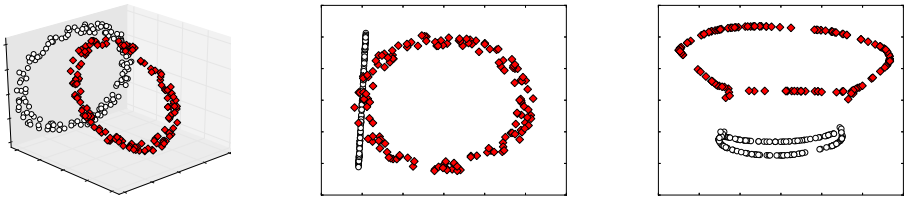

Fig. 3. Comparison of dimensionality reduction: (a) Test feature; (b) PCA result; (c) LE result.

tion problem involved in $KLIEP$ is convex [12]. The density ratio of the probability distribution $P(x)$ and $P'(x)$ of data x is defined as $r(x) = P'(x)/P(x)$. In $KLIEP$, the estimated ratio $\hat{r}$ is defined as the following linear model: $\hat{r}(x) = \sum_j^b w_j \phi_j(x)$, where w are the parameters to be learned from data samples, $\phi_j(x)$ is the similarity kernel and b is the total number of data. The parameters w are determined so as to minimize the Kullback-Leibler (KL) divergence described as $\mathrm{KL}[P'(x)\|\hat{P}'(x)]$. The minimization of the KL divergence is equivalent to maximizing the following: $\int P'(x)\log(\hat{r}(x))dx$, under the following constraint: $\int \hat{r}(x)P(x)dx = 1$. By approximating the expectation with sample average, the following convex optimization problem is derived:

$$\max_{w} \sum_{i=1}^{n'} \log(w^T \phi(x'_i)) \quad \text{subject to} \sum_{i=1}^{n} w^T \phi(x_i) = n \text{ and } w \geq 0 \tag{3}$$

where n' and n are the test input samples and the training input samples in a likelihood cross-validation [13], respectively. Then we can obtain the unique global solution by simply performing gradient ascent and feasibility satisfaction iteratively [12]. Meanwhile, the kernel parameter can be learned using the likelihood cross-validation method by approximating the unbiased estimator of the KL divergence. In this paper, the kernel parameter, the number of k-nearest samples in the similarity matrix W, is optimized by using the given feature vectors as $P(x)$, and the embedded feature vectors as $P'(x)$.

Fig. 3 shows the difference between linear and nonlinear dimensionality reduction methods. The red data in Fig 3(a) indicates a ring-shaped test feature and the gray shows a ring-shaped test feature intersecting the red data in three-dimensional space. Fig. 3(b) and (c) indicate dimensionally reduced data by PCA and dimensionally reduced data by LE, respectively. The figures show that if a data includes complicated nonlinear cluster structures, it is lost in linear dimensionality reduction technique. In our proposed framework, by combining LE and $KLIEP$, dimensionally reduced feature data can be obtained without arbitrary parameter tuning, while also preserving the existing cluster structure.

V. Bayesian Clustering

As mentioned in the introduction, the need for a method of determining the total number of clusters K continues one of the major issues concerning conventional clustering algorithms. Although several K estimation methods have been proposed, it is difficult to manage these methods. For example, the Jain-Dubes method is proposed [14] for K estimation in K-means clustering, which is a well-known and widely used clustering algorithm. However, this method does not work well if the feature space is complicated and consists of nonlinearly distributed clusters, because K-means is known to be a local-minimum solution and assumes that each cluster is a hypersphere.

In order to overcome the above issues, we propose a Bayesian model-based clustering method. For the clustering task, there are many unknown parameters, such as the number of clusters, the cluster labels, the cluster shapes, and the cluster parameters including a mean or a variance. In a Bayesian model (BM) approach, all unknown parameters can be naturally learned from a given data by expressing a parameter distribution as an infinite dimensional discrete distribution. Specifically, we first consider an infinite Gaussian mixture model in which data x is generated.

$$P(x|\alpha, P(\theta)) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}(\mu_k, \sigma_k^2) \tag{4}$$

where $P(\theta)$ is the prior distribution of parameters θ in a Gaussian distribution $\mathcal{N}(\mu_0, \sigma_0^2)$ with the mean μ_0 and the variance σ_0^2, α is the learnable hyper-parameter, and π is the mixing ratio. The BM considers that all data are automatically classified while generating each data x from any of infinite mixture distributions. When a cluster label z_n of data x_n is unknown, the

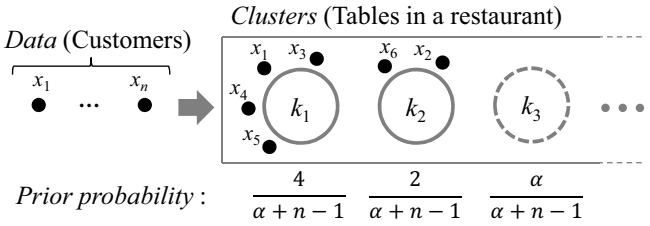

Fig. 4. Overview of the CRP-based cluster selection.

posterior probability of z_n is given based on Bayes' theorem.

$$P(z_n = k \mid x_n, z_1, ..., z_{n-1}) \propto \begin{cases} P(x_n|k)\dfrac{n_k}{\alpha+n-1} & (k = 1 \cdots K) \\ P(x_n|k^{new})\dfrac{\alpha}{\alpha+n-1} & (k = K+1) \end{cases} \quad (5)$$

where n_k is the number of k in $z_1^{n-1} = z_1, ...z_{n-1}$, K is the number of current clusters, the first term is the likelihood of x_n, and the second term is the prior probability of z_n. This procedure is known as a Chinese Restaurant Process (CRP), which is a distribution on partitions obtained by imagining a process in which customers share tables in a Chinese restaurant [15].

Fig. 4 shows an overview of the CRP-based cluster selection. By using (5), when a feature vectors $X = (x_1, ..., x_N)$ are given, a cluster labels of the features $Z = (z_1, ..., z_N)$: $z_n \in 1 \cdots K$ can be solved by using Gibbs sampling [16], as shown in Algorithm 1. The cluster label Z followed by a true distribution $P(Z|X)$ is given by iteratively sampling the hidden variable z_n from a conditional probability $P(z_n|X, Z_{-n})$, where Z_{-n} is Z without z_n. Note that this is an exchangeable process in that the probability does not depend on the order in x_n.

Algorithm 1 Automatic clustering with Gibbs sampling

Require: X, θ
1: **while** not converged **do**
2: **for** n in random permutation $(1, ..., N)$ **do**
3: Remove x_n from cluster z_n and update θ
4: Sample $z_n \backsim P(z_n|X, Z_{-n})$
5: Add x_n into cluster z_n and update θ
6: **end for**
7: **end while**
8: **return** $z_1, ..., z_N$

Though the posterior distribution of the cluster assumes Gaussian distribution, this assumption works as the cluster distribution because the feature vectors are partially pre-classified by our nonlinear dimensionality reduction technique. In addition, the hyper-parameter α can also be determined automatically by k-fold cross-validation [13]. Furthermore, the proposed framework allows us to easily quantify the clustering results because unknown parameters such as the mean or variance of each cluster can be learned directly from given data. Therefore, in our framework, automatic clustering can be expected without manual parameter tuning.

VI. Experimental results

A. Experimental Setup

The proposed methodologies were implemented in C++ and Python on a Linux machine with eight 3.4GHz CPUs and 32GB memory. Calibre [17] was used to perform lithography simulation with wavelength $\lambda = 193$nm and $NA = 1.35$. Two 32nm node industrial layouts, A and B, were applied as benchmark. The areas of the layout A and layout B are $10092.2\mu m^2$ and $12702.3\mu m^2$, respectively.

In layout feature extraction, layout feature is important as it determines how to encode initial geometrical information. We implemented two conventional layout features: density based feature and diffraction based feature. In density based feature, for each layout region, we split them into a set of grids and then the densities in the grids are encoded in a vector [18]. In our implementation, the area of layout region is set to 1000nm, and the grid number in each layout region is set to 10. Diffraction based feature represents pattern information based on a Fourier spectrum [4]. The total feature dimension numbers are 100 for density based feature and 225 for diffraction based feature.

Besides the proposed Laplacian eigenmaps based dimensionality reduction (Sec.IV), we also implemented a conventional dimensionality reduction technique, principal component analysis (PCA). PCA has been applied in several layout analysis works (e.g., [5]).

In the implementation of Bayesian clustering (Sec.V), the hyper-parameter α is determined through 5-fold cross-validation. A Gaussian-Wishart distribution is used as the prior distribution because the mean and the variance in each cluster are unknown. Parameters of the priors, prior mean and prior covariance are set to 0, the covariance of input feature vectors, respectively. The other parameter of the prior, the freedom of Wishart distribution, is also determined with 5-fold likelihood cross-validation. The total iterations of the Gibbs sampling is set to 1000 and burn-in is half of the total iterations. We also implemented K-means algorithm, which is a classical clustering method. In the K-means algorithm, the K value is determined by the Jain-Dubes method [14].

B. Effectiveness of Pattern Sampling

In the first experiment, we verify the effectiveness of the proposed Laplacian eigenmaps and Bayesian clustering. Layout A is used as test layout to extract layout patterns. Table I lists the pattern sampling results with different dimensionality reduction and clustering techniques. "**PCA**" and "**LE**" indicate principal component analysis and Laplacian eigenmaps, respectively. "**Km**" and "**BM**" indicate K-means clustering and Bayesian clustering, respectively. Different combination of dimensionality reduction and clustering are tested. For example, column "PCA+Km" means principal component analysis is applied for dimensionality reduction, while K-means is used for clustering. For each combination, columns "**K**", "**BE**" and "**CPU(s)**" give the number of final clusters, the Bayes error defined in Eqn. (1), and the runtime in seconds.

From Table I we can see that our proposed method ("LE+BM") can achieve the best clusters in terms of BE. Though the combination with LE and K-means is known as a spectral-clustering, defining K remains difficult. It can be

TABLE I
COMPARISON OF PATTERN SAMPLING TECHNIQUES

Layout B	PCA+Km			LE+Km			PCA+BM			LE+BM		
	K	BE	CPU(s)	K	BE	CPU(s)	K	BE	CPU(s)	K	BE	CPU(s)
Density	4	143.4	0.4	5	1198.0	99.8	8	82.7	29.9	11	57.7	130.5
Diffraction	4	230.7	0.7	6	898.3	100.8	13	183.9	39.5	19	117.9	148.0
Average	4	187.1	0.5	5.5	1048.2	100.3	10.5	133.3	34.7	15	87.8	139.2
Ratio	-	**1.0**	-	-	**5.6**	-	-	**0.7**	-	-	**0.5**	-

also seen that K in our method tends to be slightly higher than K-means based methods. It should be noted that there is no correct number of clusters because pattern sampling is unsupervised learning task. We apply obtained samples (test patterns) to the next two applications to evaluate the effectiveness of the sampling framework. It should be also noted although BM is time-consuming, the runtime can be reduced by using variational Bayes inference [10], which is a subject for future work.

C. Effectiveness on Hotspot Detection Application

In the second experiment, we verify our sampling methodologies in hotspot detection application, where hotspots need to be quickly detected withough lithography simulation. Applied in early physical design stage, hotspot detection [2, 18, 19] can effectively reduce the turn-around time (TAT) and the design cost. A hotspot detection model was trained with the test patterns in the layout A obtained in the sampling experiments and the hotspots that are detected by our industry setting verification process. Then the layout B was scanned using the detection model.

Although many effective algorithms have been proposed such as artificial neural network and support vector machine [2, 18], we focus on a specific detection algorithm to evaluate the effect of the samples obtained by our sampling framework since proposal of the optimal detection algorithm for the hotspot detection problem is not the intent of this paper. Furthermore, other algorithms proposed by [2, 18] are not compared in this paper. The reason of such limitation is that it is difficult to measure the performance of hotspot detection for actual full-chip layout because the public benchmarks used in the related works consist only of limited clipped layouts [20]. This paper uses the AdaBoost classifier which has shown relatively better performance compared to other classifiers [3, 21].

Fig. 5 gives the hotspot detection results with different dimensionality reduction and clustering techniques. Two important metrics are used to evaluate the performance of hotspot detection. The first one is the hotspot detection **accuracy** defined as $Hit/(\#of\ hotspots)$, where Hit is the number of correctly detected hotspots. Another one is the **H/E** ratio (false alarm) defined as $Hit/Extra$, where $Extra$ is the number of falsely detected hotspots. From Fig. 5 we can see that the prediction model performance tends to deteriorate according to the increase in BE in terms of false alarm. Moreover, from the point of view of both accuracy and false alarm, the diffraction based feature is fit to the hotspot detection problem. The results also show that the representative training patterns can be obtained by our framework because the results with our proposed method show the highest accuracy and the lowest false alarm.

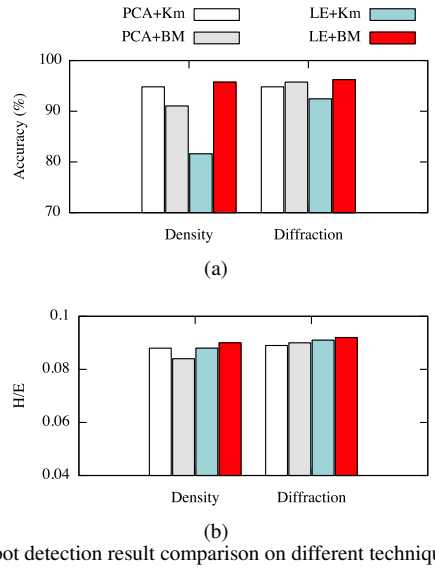

Fig. 5. Hotspot detection result comparison on different techniques. (a) The impact on accuracy; (b) The impact on false alarm.

D. Effectiveness on OPC Application

In the last experiment, we verify our proposed sampling techniques in optical proximity correction (OPC) regression application. OPC is a mask optimization technique to improve image pattern fidelity on a wafer. The most widely used method is model-based OPC in which the displacement amount of fragment movement of a mask pattern is computed based on lithography simulation. Although this method is expected to achieve very high accuracy, it is also known to be extremely time-consuming. To reduce the TAT, linear regression-based OPC is proposed and this showed that it is possible to reduce the iterations in model-based OPC [22]. However, the question of what kind of training patterns should be used for the regression model remains open. To evaluate the performance of OPC regression, we use the root mean square prediction error (RMSPE) defined as follows:

$$RMSPE = \sqrt{(1/N)\sum(y_i - \hat{y})^2} \tag{6}$$

where N is the total number of samples, y_i is the fragment movement determined by model-based OPC, and $\hat{y}$ is the predicted fragment movement.

All displacement amounts of the fragments in the layout A and B are computed by the lithography simulation with our industry setting model-based OPC. Then, the displacements of the patterns obtained in the sampling experiments are used to train a linear regression model defined by

$$y = \sum_{i=0}^{b} w_i x + \epsilon \tag{7}$$

TABLE II
OPC PERFORMANCE COMPARISON

Layout A	PCA+Km		LE+Km		PCA+BM		LE+BM	
	TD#	RMSPE	TD#	RMSPE	TD#	RMSPE	TD#	RMSPE
Density	308	7.8	282	8.4	526	6.7	577	6.6
Diffraction	211	12.2	508	13.3	1048	11.1	1102	9.7
Average	259.5	10.0	395.0	10.9	787.0	8.9	839.5	8.2
Ratio	-	**1.0**	-	**1.1**	-	**0.9**	-	**0.8**

where w are the coefficients, x is the feature vector, b is the dimensions, and ϵ is a random factor. Finally, all displacements of the patterns in the layout B are predicted using the regression model. Note that the primary objective of this paper is to sample an appropriate training patterns as an input of regression model. All the initial patterns are sampled through different layout pattern sampling techniques.

Table II compares the OPC performances under different sampling techniques. Similar to Table I, we enumerate different combinations of dimensionality reduction and clustering techniques. For each combination, columns "**TD#**" and "**RMSPE**" give the number of trainng data and RMSPE value through Eqn. (6), respectively. We can see that the model trained with our test patterns achieved the best prediction accuracy. From Table II we can also see that the number of training data can be dramatically reduced by using our method while maintaining high prediction accuracy. Although the "LE+Km" method achieves the best reduction ratio, the corresponding RMSPE value is the worst. It should be noted that compared with the result of the "PCA+BM" method, our method achieved a better RMSPE even though the number of training data is very similar to the case of "PCA+BM". This indicates that the test patterns obtained by our method include sufficient characteristics of whole chip layout even for different types of layout features.

VII. CONCLUSION

In this paper we proposed a novel layout pattern sampling framework for IC manufacturing design. By applying our nonlinear dimensionality reduction technique with kernel parameter optimization, dimensionality- and type-independent layout feature can be used in accordance with applications. The Bayesian model based clustering technique is able to classify layout data without manual parameter tuning. The experimental results show that our proposed method can effectively identify the key layout patterns that represent characteristics of whole chip, thus it promises to dramatically reduce both the manufacturing cost and the cost of process optimization. In the future, we expect to extend our framework to more IC manufacturing design applications.

REFERENCES

[1] N. Jia and E. Y. Lam, "Machine learning for inverse lithography: using stochastic gradient descent for robust photomask synthesis," *Journal of Optics*, vol. 12, no. 4, pp. 045 601:1–045 601:9, 2010.

[2] D. Ding, B. Yu, J. Ghosh, and D. Z. Pan, "EPIC: Efficient prediction of IC manufacturing hotspots with a unified meta-classification formulation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2012, pp. 263–270.

[3] Z. Xiao, Y. Du, H. Tian, M. D. F. Wong, H. Yi, H.-S. P. Wong, and H. Zhang, "Directed self-assembly (DSA) template pattern verification," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 55:1–55:6.

[4] S. Shim, W. Chung, and Y. Shin, "Synthesis of lithography test patterns through topology-oriented pattern extraction and classification," in *Proceedings of SPIE*, vol. 9053, 2014.

[5] B. Yu, J.-R. Gao, D. Ding, X. Zeng, and D. Z. Pan, "Accurate lithography hotspot detection based on principal component analysis-support vector machine classifier with hierarchical data clustering," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 14, no. 1, p. 011003, 2015.

[6] J. Oberschmidt, A. Abdo, T. Desouky, M. Al-Imam, A. Krasnoperova, and R. Viswanathan, "Automation of sample plan creation for process model calibration," in *Proceedings of SPIE*, vol. 7640, 2010.

[7] W. C. Tam, O. Poku, and R. D. Blanton, "Systematic defect identification through layout snippet clustering," in *IEEE International Test Conference (ITC)*, 2010, pp. 1–10.

[8] W. Zhang, X. Li, S. Saxena, A. Strojwas, and R. Rutenbar, "Automatic clustering of wafer spatial signatures," in *ACM/IEEE Design Automation Conference (DAC)*, 2013, pp. 71:1–71:6.

[9] K. Tumer and J. Ghosh, "Estimating the bayes error rate through classifier combining," in *IEEE International Conference on Pattern Recognition (ICPR)*, vol. 2, 1996, pp. 695–699.

[10] C. M. Bishop *et al.*, *Pattern Recognition and Machine Learning*. Springer New York, 2006, vol. 4, no. 4.

[11] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Conference on Neural Information Processing Systems (NIPS)*, vol. 14, 2001, pp. 585–591.

[12] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Conference on Neural Information Processing Systems (NIPS)*, 2008, pp. 1433–1440.

[13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2012.

[14] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988.

[15] D. Blackwell and J. B. MacQueen, "Ferguson distributions via pólya urn schemes," *Annals of Statistics*, pp. 353–355, 1973.

[16] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[17] Mentor Graphics, "Calibre verification user's manual," 2008.

[18] S.-Y. Lin, J.-Y. Chen, J.-C. Li, W.-Y. Wen, and S.-C. Chang, "A novel fuzzy matching model for lithography hotspot detection," in *ACM/IEEE Design Automation Conference (DAC)*, 2013, pp. 681–686.

[19] J.-Y. Wuu, F. G. Pikus, A. Torres, and M. Marek-Sadowska, "Rapid layout pattern classification," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2011, pp. 781–786.

[20] J. A. Torres, "ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2012, pp. 349–350.

[21] T. Matsunawa, J.-R. Gao, B. Yu, and D. Z. Pan, "A new lithography hotspot detection framework based on AdaBoost classifier and simplified feature extraction," in *Proceedings of SPIE*, 2015.

[22] A. Gu and A. Zakhor, "Optical proximity correction with linear regression," *IEEE Transactions on Semiconductor Manufacturing (TSM)*, vol. 21, no. 2, pp. 263–271, 2008.