2018 IEEE Computer Society Annual Symposium on VLSI

Sparse VLSI Layout Feature Extraction: A Dictionary Learning Approach
(Invited Paper)

Hao Geng[1], Haoyu Yang[1], Bei Yu[1], Xingquan Li[2], and Xuan Zeng[3]

[1]Department of Computer Science & Engineering, The Chinese University of Hong Kong, NT, Hong Kong
[2]Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, China
[3]State Key Laboratory of ASIC & Systems, Microelectronics Department, Fudan University, China
{hgeng,hyyang,byu}@cse.cuhk.edu.hk, n130320024@fzu.edu.cn, xzeng@fudan.edu.cn

***Abstract*—Recently, in VLSI design for manufacturability (DFM), capturing and representing the intrinsic characteristics of a layout is of great importance. Especially, there has been revival of interest in applying machine learning techniques into DFM field. Feature extraction of layout patterns is imperative before feeding into learning models so that feature representation directly affects performance of machine learning model. In this paper, a literature review of recent progress on VLSI layout feature extraction is firstly conducted. Then, for the first time, we propose a dictionary learning approach wrapped in an online learning model in applications of VLSI layout such as sub-resolution assist feature (SRAF) generation and hotspot detection. With mapping original features into a sparse and low-dimension space, dictionary learning model is benefit to calibrate a machine learning model. The experimental results show that our method not only improves the accuracy of hotspot detection but also boosts F_1 score in machine learning model-based SRAF generation with less time overhead.**

***Index Terms*—VLSI layout; Hotspot detection; SRAF generation; Feature extraction; Dictionary learning**

I. Introduction

In modern VLSI design for manufacturability (DFM), measuring the similarity among different layout designs is extremely crucial and meanwhile involved in almost all applications in the field [1], [2]. Capturing and representing the intrinsic characteristics such as topological information of a layout is the kernel to addressing the problem. Since pattern intuitively describes and summarizes two-dimensional polygon configurations in a layout design, pattern-based scheme is widely used in layout design. For example, design rule check (DRC) Plus exploiting a library of patterns to identify problematic 2D patterns, has been proven to be effective [3]. However, as integrated circuit feature sizes continue to decrease, patterning technology may have poor process margin [4]. In addition, the number of patterns increases dramatically, which brings about challenges in identifying, organizing, and carrying forward the learning of each pattern from test layout designs to mature products. On the other hand, recently, machine learning technologies have been heavily introduced into DFM. To a machine learning model, the fed features directly affect the performance of regression and prediction. Therefore, the problem how to extract characteristics from numerous patterns properly demands prompt solution. In the paper, we will exemplify two applications in computational lithography domain to go in depth on feature extraction of layouts.

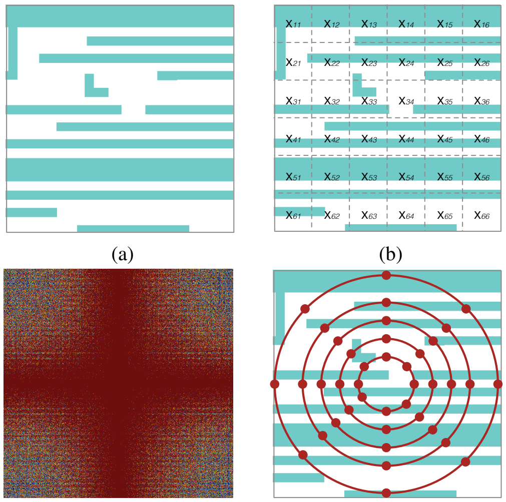
With feature size of semiconductors entering the nanometer era, lithographic process variations emerge more commonly in manufacturing process. It will lead to manufacturing defects and decrease yield. Although lithographic simulation is able to generate fabrication result accurately, it suffers from great runtime consumption. To address these problems, different kinds of approaches are proposed. One is mask optimization through various resolution enhancement techniques (RETs) [5]–[7]. In modern DFM, this strategy plays an important role in patterning and litho-friendly layout design [8], which can improve the yield of semiconductors. Sub-resolution assist feature (SRAF) [9] is a representative strategy of numerous RET techniques. Via transferring light to the positions of target patterns, small SRAF patterns can improve the robustness of target patterns to lithographic variations. There are many algorithms to generate SRAFs such as rule-based [10], model-based [11] including machine learning model-based approach [12]. Rule-based SRAF generation method is very fast, but it is hard to define and extract rules from model based SRAFs. Hence, the performance can not be guaranteed. Although model-based method is more accurate, it is time-consuming. In addition, it is hard for conventional algorithms to generate consistent SRAF patterns and may require too much engineering efforts. However, the machine learning model-based scheme can faster and more precisely obtain consistent SRAF patterns.

Another way to alleviate lithographic process variations, especially for some sensitive layout patterns, is so-called hotspot detection. Many methods such as pattern matching-based [13], [14], machine learning model-based [15], [16] and recently convolutional neural network (CNN) model-based [17] hotspot detection algorithms are proposed. Pattern matching provides speedup in comparison with lithographic simulation. However, it is only applicable to detect already known or similar patterns and has poor hotspot recognition rate on unknown patterns. The approaches based on machine learning, especially deep learning techniques have been able to achieve reasonable good result for hotspot detection with less time consumption.

With a remarkable success on some DFM applications [18]–[20], machine learning has been known as emerging and promissing technique applied in SRAF generation and hotspot detection. By computing the lithographic objective function, a mathematical model is calibrated based on the training data set. Then calibrated model can predict some values such as hotspot or non-hotspot, inserting SRAF or not for the testing data set. In a machine learning flow, before feeding into the learning machine engine, raw data should be preprocessed in feature extraction stage. Feature representation of original data directly affects prediction performance. In other words, with more representative and generalized features, the model has better performance of approximation and prediction. Besides, the better-selected features can avoid overfitting to some extents. In this paper, we propose a dictionary learning based approach wrapped in an online learning model to extract features. To the best of our knowledge, there is no previous art in applying dictionary learning method into the applications of VLSI DFM. Our main contributions are listed as follows.

This work is supported in part by The Research Grants Council of Hong Kong SAR (Project No. CUHK24209017), National Key Research and Development Program of China 2016YFB0201304, and National Natural Science Foundation of China (NSFC) research projects 61574046 and 61774045.

2159-3477/18/$31.00 ©2018 IEEE
DOI 10.1109/ISVLSI.2018.00094



(a) (b) (c) (d)

Fig. 1: (a) Original layout; (b) Density-based feature; (c) Spectrum-based feature; (d) CCAS feature.

- Dictionary learning concept is firstly introduced into DFM domain of VLSI. Within an online learning model, it can handle a large amount of layout patterns.
- Our proposed framework are applied into two issues, SRAF generation and hotspot detection. The revised features are more sparse and representative.
- The experimental results show that our method not only improves the accuracy of hotspot detection but also boosts the F_1 score of model-based SRAF generation.

The rest of this paper is organized as following. Section II gives a quick survey of recent progress on VLSI layout feature extraction. Section III introduces the concept of dictionary learning and online learning model. Section IV illustrates the experiment results, followed by conclusion in Section V.

II. Previous Feature Extraction Approaches

In DFM, layout feature extraction plays an important role. In the case of VLSI layouts, a set of feature vectors will be extracted to represent layout pattern information. To represent layout accurately, numerous kinds of feature extraction methods have been proposed in previous work.

A. Density-based Feature

The density-based feature [13], [21] summarizes local pattern density of a layout within given grids. This feature is reasonable to measure the mask printability since layouts with high pattern density have a higher risk of suffering defects. The core scheme is first divide encoding area of each layout into some square grids, and then calculate the ratio between the pattern area in one grid and the corresponding grid area. Compared with other features, the density-based feature vectors are prone to be separated in low-dimension space. It benefits training and inference of a machine learning model. However, this rough idea will lead to global information loss and degenerate of machine learning model in high-dimension. In some cases, the extraction method even extract the same feature vectors from different patterns. Hence, a modified version of density-based feature extracted by local grid density differential (LGDD) method is proposed in [22]. Traditional density-based feature just calculates the density value of a gird. However, the new scheme sets triangles locating at 4 corners of a grid as sampling area and concatenate density values from different sampling areas to form a feature vector. But the longer feature dimensionality may increase the risk of overfitting. To alleviate the overfitting problem, some extended approaches are investigated. In [15], the parameters such as grid size and window size of density-based feature are optimized by maximizing the Mahalanobis [23] distance between non-hotspot and hotspot features. The principal component analysis (PCA) method [24] is exploited to reduce the dimensionality of feature vectors. However, information loss is inevitable since non-principal components are ignored. In addition, there also exists a problem that how many principal components need to be kept. To give a direct understanding, the density feature is shown in Fig. 1(b).

B. Spectrum-based Feature

As illustrated in Fig. 1(c), spectrum-based feature applies frequency domain transforms such as discrete Fourier transform (DFT), discrete cosine transform (DCT). In [21], [25], a feature vector consists of the coefficients of Fourier transform of a layout pattern. Since the feature reflects an effect due to projection optics, it is expected to benefit a machine learning model with highly accurate prediction. In addition, in [25], the feature has made a remarkable success on reflected and shifted patterns.

After achieving a success on wafer clustering tasks [26], [27], DCT coefficients are also exploited as the input of convolutional neural network (CNN) [17]. Compared with raw layouts as inputs, CNN with DCT as inputs can achieve a higher accuracy. The success of DCT is that the extracted features will be easier to obtain high sparsity and global representations than raw images.

C. Concentric Circle Area Sampling Feature

Recently proposed concentric circle area sampling (CCAS) [28] is developed from concentric square sampling (CSS) [29]. It takes advantages of layout properties and lithography process, thus has made considerable improvements on hotspot detection accuracy. Meanwhile, because of reflecting light diffraction effects, CCAS layout features are also exerted to generate SRAFs in [12]. With considering concentric propagation of diffracted light from mask patterns, the core method of CCAS is sub-sampling on concentric circles. However, since adjacent circles contain similar information, the CCAS features have much redundant information. The redundancy will result in some problems such as hindering fitting of a machine learning model. As a result, concentric circle sampling (CCS) method which exploits the mutual information to select import circles of CCAS is proposed in [16]. The objective of circle selection is maximizing the dependency of selected circles on the corresponding classification variable. Because of reducing redundancy of CCAS, CCS is benefit to calibrate machine learning model. Fig. 1(d) shows the CCAS feature extraction method.

D. Other Successful Features

Besides the above features, there are many kinds of other successful features such as modified transitive closure graph representation [14], fragmentation-based context characterization feature [30], [31], histogram of oriented light propagation (HOLP) [32], improved tangent space-based characterization [33] and so on. In [14], authors modified the transitive closure graph (TCG) [34] method to extract critical topological features within a pattern. Meanwhile, in [33], the improved tangent space representation which reflects the radius and angle of a polygon in a layout clip has been proposed. Considering the geometric shape of a layout pattern and combined impact of its neighboring patterns, Yu *et al.* investigated a fragmentation-based feature [30] consisting of important information such as pattern shapes, the distance between patterns and corner information (convex or concave). Recently, inspired by histogram of oriented gradient (HOG) [35], HOLP [32] feature which reflects light propagation in the exposure process has been presented. This feature is robust to small shifts of layout patterns.

III. Dictionary Learning based Feature Extraction

A. Dictionary Learning Approach

Dictionary learning and sparse representation are the two related topics in terms of data decomposition [36]. Recently, these two models are coupled in a self-adaptive dictionary learning model, aiming at decomposing the signal with sparse nature over a learned dictionary. Specifically, the mechanism of dictionary learning and sparse representation is to select only a few atoms from a well-trained dictionary and obtain their linear combination to approximate the data sparsely and accurately. The joint objective function of dictionary learning model is following:

$$\min_{\mathbf{x},\mathbf{D}} \frac{1}{N}\sum_{i=1}^{N}\{\frac{1}{2}\left\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\right\|_2^2 + \lambda\left\|\mathbf{x}_i\right\|_p\}, \tag{1}$$

where $\mathbf{y}_i \in \mathbb{R}^n$ is the input data vector, and $\mathbf{D} = \{\mathbf{d}_j\}_{j=1}^{s}, \mathbf{d}_j \in \mathbb{R}^n$ refers to the dictionary, $\mathbf{x}_i \in \mathbb{R}^s$ indicates sparse decomposition coefficients and p denotes the type of norm. Meanwhile, N refers to the total number of training data vectors in memory.

Since the joint optimization of both dictionary and sparse coefficients is non-convex, but sub-problem with one variable fixed is convex. Two stages, sparse coding and dictionary constructing, are alternatively performed in a dictionary model.

The objective function for sparse coding of i-th training data vector in memory is showed in Equation (2):

$$\mathbf{x}_i \triangleq \arg\min_{\mathbf{x}} \frac{1}{2}\left\|\mathbf{y}_i - \mathbf{D}\mathbf{x}\right\|_2^2 + \lambda\|\mathbf{x}\|_p. \tag{2}$$

According to the format of penalty term, the method of sparse coding can be simply divided into two categories. If $p = 0$ in (1), this is an NP-hard problem. Some greedy solutions are proposed. Matching pursuit (MP) [37] and orthogonal matching pursuit (OMP) [38] are the representatives among them. Convex relaxation algorithms use convex replacement for non-convex l_0-norm. If $p = 1$ in (1), this is the famous problem about least absolute shrinkage and selection operator (LASSO) [39]. Least angle regression (LARS) [40], coordinate descent [41] with its generalizations and fast iterative shrinkage-threshold algorithm (FISTA) [42] are classical and efficient schemes to solve LASSO problem.

The objective function for dictionary construction is:

$$\mathbf{D} \triangleq \arg\min_{\mathbf{D}} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\right\|_2^2 + \lambda\|\mathbf{x}_i\|_p. \tag{3}$$

To construct dictionary, different classes of methodologies are proposed. Fourier transformations, wavelet [43], curvelet [44] and contourlet transform [45] are deduced based on the analytical solutions which have the fixed mathematical formats. These methods belong to analytical dictionary, in which the signal is decomposed over pre-defined atoms. However, these schemes do not explore the characters of the data such as structure or texture. In addition, the bases of analytical dictionary should be orthogonal, which restricts their application in VLSI. Another class of methods is called non-analytical dictionary, or self-adaptive dictionary learning method such as K-SVD [46], recursive least square dictionary learning method (RLS) [47], online dictionary learning method (ODL) [48] and so forth. These more flexible methods aim at obtaining an over-complete dictionary, i.e. the number of atoms is far larger than the dimension of one atom. The atoms are learnt from samples so that they can explore the data with complicated structure. The common model of self-adaptive dictionary learning is summarized in Algorithm 1.

Algorithm 1 Dictionary Learning Approach

Require: Original features $\mathbf{Y} \leftarrow \{\mathbf{y}_i\}_{i=1}^{N}, \mathbf{y}_i \in \mathbb{R}^n$.
Ensure: Dictionary $\mathbf{D} \leftarrow \{\mathbf{d}_j\}_{j=1}^{s}, \mathbf{d}_j \in \mathbb{R}^n$, sparse features $\mathbf{X} \leftarrow \{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^s$.
1: **Initialization**: Initial dictionary $\mathbf{D}_0$,
2: **while** not convergence **do**
3: Sparse coding; ▷ Equation (2)
4: Dictionary construction; ▷ Equation (3)
5: **end while**

K-SVD creativity uses singular value decomposition (SVD) to obtain the atoms consequentially and use indices matrix before SVD to keep the sparsity constraint. However, K-SVD is a batch learning method which means that the whole dataset should be loaded into memory at the beginning of computation. In the stage of sparse coding, K-SVD solves l_0-norm problem via OMP method which is very time-consuming. In addition, SVD also has a high computation complexity based on the size of matrix. Therefore, K-SVD cannot easily handle large dataset. To deal with large data set, RLS, ODL and its modified version that double online dictionary learning (DODL) [49], are proposed. RLS dictionary learning method involves some recursive-decomposition and searching computations. It is time-consuming when dealing with large data set. DODL uses the sub-sampling matrix to further shrink the dimensionality of input medical data. However, this scheme of randomly choosing the dimension of a feature vector may lead to unnecessary information loss.

In fact, with integrated circuits entering ultra-large-scale era, VLSI layouts are more and more complex. Traditional dictionary learning techniques, like K-SVD, cannot satisfy the requirements of handling a layout with numerous patterns. The simplified online framework is suit and good enough for dictionary learning in VLSI.

B. Online Learning Algorithm

Based on the review, we utilize dictionary learning approach wrapped within online framework. The model is in a stochastic fashion as there is an assumption in online learning that the number of input samples can be infinite and i.i.d. input $\mathbf{y}_i$ is drawn from an unknown probability distribution.

Sparse coding and dictionary construction are still performed alternatively in iterations. In one iteration, one sample (or a mini-batch) is loaded into memory and processed at a time. In the stage of sparse coding, decomposition only performs on current sample. Therefore, it can be seen that Equation (2) is in an incremental sense. Since l_1 norm can be solved efficiently, we adopt it as the penalty term in Equation (2). Coordinate descent algorithm is exploited to address the sub-problem, and time overhead can be dramatically reduced. In the stage of dictionary construction, two auxiliary variables which carry the past information from sparse coefficients and input data, are introduced to help compute the dictionary. They

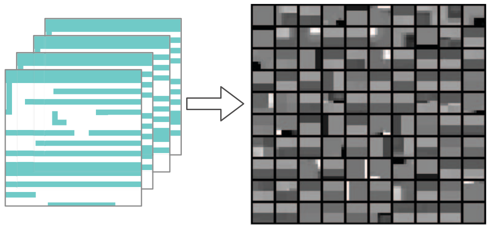


Fig. 2: Visualization of a dictionary trained by ODL algorithm.

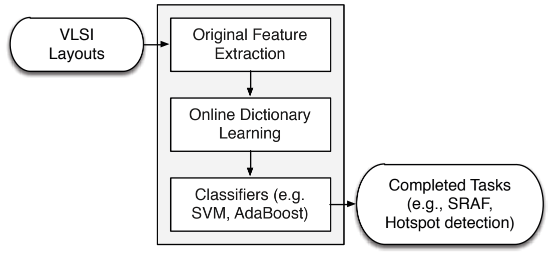


Fig. 3: The whole algorithm flow.

play an important role in updating atoms, specifically when the block coordinate descent method with warm start is applied as the optimization scheme. The updating rules for auxiliary variables are showed in Equation (4) and Equation (5), respectively.

With the help from auxiliary variables, the rules for sequentially updating atoms are summarized in Equations (6) and (7). In Equation (6), $\mathbf{D}_{t-1}$ is selected as the warm start of $\mathbf{D}$. $\mathbf{b}_j$ indicates the j-th column of $\mathbf{B}_t$, while $\mathbf{c}_j$ is the j-th column of $\mathbf{C}_t$. $\mathbf{C}[j,j]$ denotes the j-th element on diagonal of $\mathbf{C}_t$. Equation (7) is an l_2 norm constraint on atoms to avoid atoms becoming arbitrarily large (which may lead to arbitrarily small sparse coefficients). [50] proves that in the stage of constructing dictionary, the convex optimization problem allowing separable constraints in the updated blocks (columns) will guarantee the global convergence. The online framework for constructing dictionary in t-th iteration is illustrated in Algorithm 2.

$$\mathbf{B}_t \leftarrow \left(1-\frac{1}{t}\right)\mathbf{B}_{t-1} + \frac{1}{t}\mathbf{y}_t\mathbf{x}_t^\top. \tag{4}$$

$$\mathbf{C}_t \leftarrow \left(1-\frac{1}{t}\right)\mathbf{C}_{t-1} + \frac{1}{t}\mathbf{x}_t\mathbf{x}_t^\top. \tag{5}$$

$$\mathbf{u}_j \leftarrow \frac{1}{\mathbf{C}[j,j]}\left(\mathbf{b}_j - \mathbf{D}\mathbf{c}_j\right) + \mathbf{d}_j. \tag{6}$$

$$\mathbf{d}_j \leftarrow \frac{1}{\max\left(\|\mathbf{u}_j\|_2, 1\right)}\mathbf{u}_j. \tag{7}$$

Algorithm 2 Online Dictionary Construction

Require: $\mathbf{D}_{t-1} \leftarrow \{\mathbf{d}_j\}_{j=1}^s, \mathbf{d}_j \in \mathbb{R}^n$,
$\mathbf{B}_{t-1} \leftarrow \{\mathbf{b}_j\}_{j=1}^s, \mathbf{b}_j \in \mathbb{R}^n$,
$\mathbf{C}_{t-1} \leftarrow \{\mathbf{c}_j\}_{j=1}^s, \mathbf{c}_j \in \mathbb{R}^s$.
Ensure: Dictionary $\mathbf{D}_t \leftarrow \{\mathbf{d}_j\}_{j=1}^s, \mathbf{d}_j \in \mathbb{R}^n$.
1: Update two auxiliary variables $\mathbf{B}_t$, $\mathbf{C}_t$; ▷ Equations (4) and (5)
2: **for** $j = 1$ to s **do**
3: Update the j-th atom $\mathbf{d}_j$; ▷ Equations (6) and (7)
4: **end for**

An original layout is showed in Fig. 1(a). With many original layouts as input samples, we visualize the dictionary trained by dictionary learning method in online framework. From the visualization showed in Fig. 2, we can explore that some basic texture characteristics of the layouts have been obtained. Some redundant information exists in the dictionary since it is over-complete.

IV. Experimental Results

A. Overall Flow

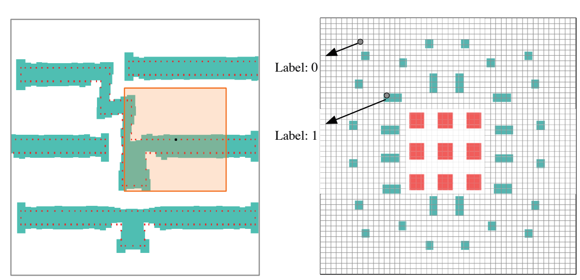
The whole working flow of our model is shown in Fig. 3. The step for online dictionary learning is after completing original feature extraction. Our well-trained dictionary is made up of atoms which are representatives of original features. The original features are decomposed over the well-trained dictionary and represented as the linear combinations of atoms. The new features, i.e., sparse decomposition coefficients, are expected to be benefit to avoid overfitting. In next stage, the classifier is fed and calibrated by the new features. In our proposed model, online dictionary learning plays an important role in initial pattern sampling and mapping original features into a sparse and low-dimension domain. As a transforming method of feature space, it can cooperate with many feature extraction methods for different purposes in DFM.

Proposed online dictionary learning framework is applied into two applications of VLSI domain: hotspot detection and SRAF generation. To verify the performance, the framework is embedded in the advanced model of [16] to predict hotspots and [12] to generate SRAFs. Our method is implemented via `Python` with the `Scikit-learn` library [51] on a 8-core 3.7GHz Intel platform.

B. Case Study: Hotspot Detection

In the application of hotspot detection, the original dataset, ICCAD-2012 benchmark [52] is not big enough. Therefore, we adopt three larger and more complicated industry cases, `Industry1-Industry3`, as our benchmark set. In Fig. 4(a), the small dots denote the sampling points to generate layout clips of benchmarks for hot-spot detection. Hence, this benchmark set of VLSI layout clips is sampled densely. As a result, the training dataset consisting of many similar instances is good for training dictionary.

TABLE I summarizes our results compared with the results by the prior art [16]. Column "Benchmarks" lists all the test datasets of layouts. Columns "Accu", "FA" and "CPU" refer to the evaluation metrics in terms of the accuracy [52], false alarms [52] and the total runtime. Column "ICCAD'16" indicates the experiment results by [16], while Column "Ours" is corresponding to the results of our dictionary learning model within online framework. Note that for fair comparison, in "Ours", the same classifier as [16] is utilized.



(a) (b)

Fig. 4: Dense sampling in (a) hotspot detection; (b) SRAF generation.

TABLE I: Comparison with [16] on Hotspot Detection.

Benchmarks	ICCAD'16 [16]			Ours		
	Accu(%)	FA#	CPU(s)	Accu(%)	FA#	CPU(s)
`Industry1`	89.9	1136	192.9	98.1	7600	203.7
`Industry2`	88.4	7402	249.9	93.2	22049	253.5
`Industry3`	82.3	8609	288.9	94.2	23468	302.4
Average	86.9	5716	243.9	95.2	17706	253.2
Ratio	1.00	**1.00**	**1.00**	**1.10**	3.09	1.04

To fit the consequence classifier, we truncate the numerical value of sparse coefficients. In fact, it is reasonable since the absolute values of sparse coefficients are concentrated in the range from 0 to 200. It can be seen from the table that with slightly longer running time, the accuracy has been improved by 10% in average, while the false alarms are dramatically increased. The reason may be that the benchmarks are quite imbalanced.

C. Case Study: SRAF Generation

In SRAF generation issue, the original benchmark is kept since it is really large. Fig. 4(b) illustrates the feature extraction of machine learning model based SRAF generation. The layout which is put on 2-D grid plane and sampled at each grid will generate many similar feature vectors. So the training process of dictionary is benefit from dense sampling. To demonstrate the performance of proposed model, we exploit the same benchmark set as applied in [12], which consists of 8 dense layouts and 10 sparse layouts. The spacing between adjacent contacts for dense and sparse layouts are different, which are set to $70nm$ and $\geq 70nm$ respectively.

TABLE II compares our method with a state-of-the-art machine learning based SRAF generation tool [12]. Column "Benchmarks' lists adopted test layouts. Columns "F_1 score" [53] and "CPU" are the evaluation metrics in terms of the learning model performance and the total runtime. Column "ISPD'16" refers to the experiment results by [12]. One thing worth to mention is in "Ours", the follow-up classifier is also the same with [12]. The results from TABLE II prove that in average, 5.8% improvement on F_1 score with about 20% speed-up on running time is achieved.

TABLE II: Comparison with [12] on SRAF Generation.

Benchmarks	ISPD'16 [12]		Ours	
	F_1 score(%)	CPU(s)	F_1 score(%)	CPU(s)
`DenseClip1`	95.37	1.44	**97.24**	**1.04**
`DenseClip2`	94.73	1.29	**97.06**	**1.02**
`DenseClip3`	94.00	1.20	**96.86**	**0.94**
`DenseClip4`	93.89	1.51	**96.46**	**1.12**
`DenseClip5`	94.34	1.40	**96.52**	**1.10**
`DenseClip6`	93.44	1.11	**96.81**	**0.89**
`DenseClip7`	94.20	1.34	**97.10**	**1.06**
`DenseClip8`	93.43	1.32	**97.21**	**1.07**
`SparseClip1`	90.56	2.47	**93.62**	**1.88**
`SparseClip2`	87.65	6.65	**94.11**	**4.91**
`SparseClip3`	86.21	13.33	**93.51**	**10.19**
`SparseClip4`	86.54	22.50	**93.34**	**17.56**
`SparseClip5`	85.55	27.66	**93.23**	**22.20**
`SparseClip6`	85.32	39.3	**93.21**	**30.78**
`SparseClip7`	84.94	53.96	**93.01**	**41.87**
`SparseClip8`	84.28	70.98	**92.72**	**55.61**
`SparseClip9`	85.02	90.39	**90.21**	**70.41**
`SparseClip10`	83.96	100.86	**92.60**	**80.13**
Average	89.64	24.37	**94.85**	**19.16**
Ratio	1.000	1.000	**1.058**	**0.786**

V. Conclusion

In this paper, we propose an online dictionary learning based approach to extract sparse VLSI feature. We apply the framework into the two famous issues: hotspot detection and SRAF generation. To verify the performance, our framework has been embedded into two fancy models of above issues. The experiment results show that the accuracy of hotspot detection on complicated and large VLSI layouts has been improved and the F_1 score of machine learning model in SRAF generation has also been boosted with less time overhead. Although the results of false alarms are high and some metrics of SRAF issue like process variation band (PV band) area and edge placement error (EPE) are still need to be verified by lithography simulation, the framework of dictionary learning is promising. On the other hand, the dictionary learning algorithms introduced in this paper are unsupervised learning methods. In other words, the unsupervised dictionary learning do not exploit the label information. Our future work will focus on introducing supervised dictionary learning into VLSI DFM. With the transistor size shrinking rapidly and the layouts becoming more and more complex, we expect to apply dictionary learning framework into more DFM applications.

Acknowledgment

The authors would like to thank Jing Su, Chenxi Lin, and Yi Zou from ASML for helpful comments on hotspot detection. The authors would also like to thank Xiaoqing Xu from ARM and David Z. Pan from UT Austin for granting the access to SRAF generation framework [12].

References

[1] J. Xu, K. N. Krishnamoorthy, E. Teoh, V. Dai, L. Capodieci, J. Sweis, and Y.-C. Lai, "Design layout analysis and dfm optimization using topological patterns," in *Proceedings of SPIE*, vol. 9427, 2015.

[2] L. Capodieci, "Data analytics and machine learning for continued semiconductor scaling," in *SPIE News*, 2016.

[3] E. Teoh, V. Dai, L. Capodieci, Y.-C. Lai, and F. Gennari, "Systematic data mining using a pattern database to accelerate yield ramp," in *Proceedings of SPIE*, vol. 9053, 2014, p. 905306.

[4] J. P. Cain, M. Fakhry, P. Pathak, J. Sweis, F. E. Gennari, and Y.-C. Lai, "Pattern-based analytics to estimate and track yield risk of designs down to 7nm," in *Proceedings of SPIE*, vol. 10148, 2017, p. 1014805.

[5] J.-R. Gao, X. Xu, B. Yu, and D. Z. Pan, "MOSAIC: Mask optimizing solution with process window aware inverse correction," in *ACM/IEEE Design Automation Conference (DAC)*, 2014, pp. 52:1–52:6.

[6] Y. Ma, J.-R. Gao, J. Kuang, J. Miao, and B. Yu, "A unified framework for simultaneous layout decomposition and mask optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 81–88.

[7] H. Yang, S. Li, Y. Ma, B. Yu, and E. F. Young, "GAN-OPC: Mask optimization with lithography-guided generative adversarial nets," in *ACM/IEEE Design Automation Conference (DAC)*, 2018.

[8] S. Shim, S. Choi, and Y. Shin, "Light interference map: A prescriptive optimization of lithography-friendly layout," *IEEE Transactions on Semiconductor Manufacturing (TSM)*, vol. 29, no. 1, pp. 44–49, 2016.

[9] C. H. Wallace, P. A. Nyhus, and S. S. Sivakumar, "Sub-resolution assist features," Dec. 15 2009.

[10] J. Jun, M. Park, C. Park, H. Yang, D. Yim, M. Do, D. Lee, T. Kim, J. Choi, G. Luk-Pat *et al.*, "Layout optimization with assist features placement by model based rule tables for 2x node random contact," in *Proceedings of SPIE*, vol. 9427, 2015.
[11] S. D. Shang, L. Swallow, and Y. Granik, "Model-based sraf insertion," Oct. 11 2011.
[12] 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 *ACM International Symposium on Physical Design (ISPD)*, 2016, pp. 161–168.
[13] W.-Y. Wen, J.-C. Li, S.-Y. Lin, J.-Y. Chen, and S.-C. Chang, "A fuzzy-matching model with grid reduction for lithography hotspot detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 33, no. 11, pp. 1671–1680, 2014.
[14] Y.-T. Yu, Y.-C. Chan, S. Sinha, I. H.-R. Jiang, and C. Chiang, "Accurate process-hotspot detection using critical design rule extraction," in *ACM/IEEE Design Automation Conference (DAC)*, 2012, pp. 1167–1172.
[15] 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*, vol. 9427, 2015.
[16] H. Zhang, B. Yu, and E. F. Y. Young, "Enabling online learning in lithography hotspot detection with information-theoretic feature optimization," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 47:1–47:8.
[17] H. Yang, J. Su, Y. Zou, B. Yu, and E. F. Y. Young, "Layout hotspot detection with feature tensor generation and deep biased learning," in *ACM/IEEE Design Automation Conference (DAC)*, 2017, pp. 62:1–62:6.
[18] 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.
[19] B. Yu, D. Z. Pan, T. Matsunawa, and X. Zeng, "Machine learning and pattern matching in physical design," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2015, pp. 286–293.
[20] S. Choi, S. Shim, and Y. Shin, "Machine learning (ML)-guided OPC using basis functions of polar fourier transform," in *Proceedings of SPIE*, vol. 9780, 2016.
[21] T. Matsunawa, B. Yu, and D. Z. Pan, "Laplacian eigenmaps-and bayesian clustering-based layout pattern sampling and its applications to hotspot detection and optical proximity correction," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 15, no. 4, pp. 043 504–043 504, 2016.
[22] H. Zhang, H. Yang, B. Yu, and E. F. Y. Young, "VLSI layout hotspot detection based on discriminative feature extraction," in *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2016, pp. 542–545.
[23] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.
[24] I. Jolliffe, *Principal Component Analysis*. Wiley Online Library, 2005.
[25] S. Shim and Y. Shin, "Topology-oriented pattern extraction and classification for synthesizing lithography test patterns," *Journal of Micro/Nanolithography, MEMS, and MOEMS (JM3)*, vol. 14, no. 1, pp. 013 503–013 503, 2015.
[26] 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.
[27] 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.
[28] T. Matsunawa, B. Yu, and D. Z. Pan, "Optical proximity correction with hierarchical bayes model," in *Proceedings of SPIE*, vol. 9426, 2015.
[29] 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.
[30] 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.
[31] 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.
[32] Y. Tomioka, T. Matsunawa, C. Kodama, and S. Nojima, "Lithography hotspot detection by two-stage cascade classifier using histogram of oriented light propagation," in *IEEE/ACM Asia and South Pacific Design Automation Conference (ASPDAC)*, 2017, pp. 81–86.
[33] J. Guo, F. Yang, S. Sinha, C. Chiang, and X. Zeng, "Improved tangent space based distance metric for accurate lithographic hotspot classification," in *ACM/IEEE Design Automation Conference (DAC)*, 2012, pp. 1173–1178.
[34] J.-M. Lin and Y.-W. Chang, "TCG: A transitive closure graph-based representation for non-slicing floorplans," in *ACM/IEEE Design Automation Conference (DAC)*, 2001, pp. 764–769.
[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.
[36] M. J. Gangeh, A. K. Farahat, A. Ghodsi, and M. S. Kamel, "Supervised dictionary learning and sparse representation-a review," *arXiv preprint arXiv:1502.05928*, 2015.
[37] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
[38] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems, and Computers*. IEEE, 1993, pp. 40–44.
[39] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B*, vol. 58, pp. 267–288, 1996.
[40] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, and others., "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.
[41] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, p. 1, 2010.
[42] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences (SIIMS)*, vol. 2, no. 1, pp. 183–202, 2009.
[43] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory (TIT)*, vol. 36, no. 5, pp. 961–1005, 1990.
[44] E. Candes, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale Modeling & Simulation*, vol. 5, no. 3, pp. 861–899, 2006.
[45] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.
[46] M. Aharon, M. Elad, and A. Bruckstein, "k-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
[47] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.
[48] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *International Conference on Machine Learning (ICML)*, 2009, pp. 689–696.
[49] A. Mensch, J. Mairal, B. Thirion, and G. Varoquaux, "Dictionary learning for massive matrix factorization," in *International Conference on Machine Learning (ICML)*, 2016, pp. 1737–1746.
[50] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.
[51] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct., pp. 2825–2830, 2011.
[52] A. J. 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.
[53] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation," in *European Conference on Information Retrieval (ECIR)*. Springer, 2005, pp. 345–359.