Optical Proximity Correction with Hierarchical Bayes Model

Tetsuaki Matsunawa\textsuperscript{1}, Bei Yu\textsuperscript{2} and David Z. Pan\textsuperscript{2}

\textsuperscript{1}Toshiba Corporation
\textsuperscript{2}The University of Texas at Austin
Outline

• Background
• OPC with Linear Regression
• Hierarchical Bayes Model (HBM)
• Markov Chain Monte Carlo (MCMC)
• Experimental results
• Conclusion
Optical Proximity Correction

- Issue: Conventional OPC consumes very long time
- Goal: High accurate correction in short runtime

Mask image  | Printed image
---|---
Iteration: 0  | F | F
Iteration: 3  | F | F
Iteration: 5  | F | F

Drastic reduction of OPC time is required!

Ratio of model-based OPC time (normalized by 40nm node)

Model-based OPC is the most widely used technique
Related works

- **Inverse lithography**

  ![Inverse lithography images]

  D. S. Abrams, et al., SPIE, 2006

- **Linear regression**

  ![Linear regression diagram]

  $y = \beta x + \epsilon$

  Allan Gu, et al., IEEE, 2008

- **Nonlinear regression**

  ![Nonlinear regression diagram]

Regression-based OPC

Learning phase

- Training layout
- Edge fragmentation
- Model-based OPC
- Feature extraction
- Model training

Testing phase

- Testing layout
- Edge fragmentation
- Feature extraction
- Prediction
OPC with Linear Regression

Layout feature

Concentric square sampling (CSS)

Linear model

\[ y(x, w) = w_0 + \sum_{j=1}^{D} w_j x_j \]

Least squares solution

\[ w = (X^T X)^{-1} X^T y \]

Output

\[ x = (0, 1, 0, 0, 1, \ldots, v_j)^T \]

\( V_j \): pixel value of j-th dimension

\( y \): predicted edge displacement
Pros and Cons

- Fast runtime
- Reasonable prediction accuracy
- Easy to understand

- Over-fitting, especially even with small data
- Required a large amount of training data
- Cannot model complex phenomena
OPC with Hierarchical Bayes Model (HBM)

• Fast runtime
• Reasonable prediction accuracy
• Easy to understand

• Prevent over-fitting
• Model complicated phenomena with small amount of data
Outline

• Background
• OPC with Linear Regression
• Hierarchical Bayes Model (HBM)
• Markov Chain Monte Carlo (MCMC)
• Experimental results
• Conclusion
Regression example

\[ y = (w_{F_0} + w_{R_0}) + (w_{F_1} + w_{R_1})x \]

\[ y = w_0 + w_1x \]

Edge displacement differs depending on the edge type.
GLM vs GLMM

Generalized Linear Model

\[ y(x, w) = \sum_{j=0}^{D} w_j x_j \]

#dimensions

Parameter

Feature vector

Generalized Linear Mixed Model

\[ y(x, w_F, w_R) = \sum_{j=0}^{D} \left( w_{F_j} + w_{R_j} \right) x_j \]

Fixed effects

Random effects

Solution: Maximum likelihood estimation
Concept of Hierarchical Bayes Model (HBM)

**Generalized Linear Model**

\[ y(x, w) = \sum_{j=1}^{D} w_j x_j \]

**Introduction of hidden variables**

\[ y(x, w_F, w_R) = \sum_{j=1}^{D} \left( w_{Fj} + w_{Rj} \right) x_j \]

**Solution:** Maximum likelihood estimation

**Solution:** MCMC
Hierarchical Bayes Model (HBM)

**Model**
\[ p(y|x, \theta) = \mathcal{N}(y(x, \theta), \sigma_y) \]

**Prior distributions**
- \( w_{f_j} \sim \mathcal{N}(0, \sigma_f) \)
- \( w_{r_j} \sim \mathcal{N}(0, \sigma_r) \)
- \( \sigma_y \sim \mathcal{U}(0, 10^4) \)

**Hyper-prior distributions**
- \( \sigma_f \sim \mathcal{U}(0, 10^4) \)
- \( \sigma_r \sim \mathcal{U}(0, 10^4) \)

**Posterior distribution**
\[
p(\theta|y) \propto \prod_{i=1}^{N} \prod_{j=0}^{D} p(y_i|\theta_j) \cdot p(w_{f_j}|\sigma_f) \cdot p(w_{r_j}|\sigma_r) \cdot p(\sigma_y) \cdot p(\sigma_f) \cdot p(\sigma_r)
\]
Hierarchical Bayes

Bayes’ theorem

\[ p(\theta|y) \propto p(y|\theta) \ p(\theta) \]

Posterior \hspace{1cm} Likelihood \hspace{1cm} Prior

Posterior distribution for OPC model

\[ p(\theta|y) \propto \prod_{i=1}^{N} \prod_{j=0}^{D} p(y_i|\theta_j) p\left(w_{f_j}|\sigma_f\right) p\left(w_{r_j}|\sigma_r\right) p(\sigma_y)p(\sigma_f)p(\sigma_r) \]
Outline

• Background
• OPC with Linear Regression
• Hierarchical Bayes Model (HBM)
• Markov Chain Monte Carlo (MCMC)
• Experimental results
• Conclusion
Markov Chain Monte Carlo (MCMC)

- **Markov Chain**
  - Next state is defined based on the previous state

- **Monte Carlo**
  - Stochastic algorithm
Markov Chain Monte Carlo (MCMC)

- All parameters are fixed except sampling parameter

\[
p(\theta | y) \propto \prod_{i=1}^{N} \prod_{j=0}^{D} p(y_i | \theta_j) p(w_{F_j} | \sigma_F) p(w_{R_j} | \sigma_R) p(\sigma_y) p(\sigma_F) p(\sigma_R)
\]

\[
p(\sigma_F | \ldots) \propto \prod_{i=1}^{N} \prod_{j=0}^{D} p(y_i | \theta_j) p(w_{F_j} | \sigma_F) p(\sigma_F)
\]

\[
p(\sigma_R | \ldots) \propto \prod_{i=1}^{N} \prod_{j=0}^{D} p(y_i | \theta_j) p(w_{R_j} | \sigma_R) p(\sigma_R)
\]
Outline

• Background
• OPC with Linear Regression
• Hierarchical Bayes Model (HBM)
• Markov Chain Monte Carlo (MCMC)
• Experimental results
• Conclusion
Experiments

• Model training
  – Motif: Layout A (training layout)

• Comparison with linear/nonlinear regression
  – Motif: Layout A (training), Layout B (testing)
  – Algorithms: linear regression (LR), support vector regression (SVR)

• Comparison with model-based OPC
  – Motif: Layout B (testing)
  – Optical model: 32nm node (λ=193nm, NA=1.35)
  – Model-based OPC: Calibre nmOPC
Sampling results of hidden parameters

Clear difference among edge types
Comparison with linear/nonlinear regression

RMSE (nm)
(Root mean square error)

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>3.2</td>
<td>3.66</td>
</tr>
<tr>
<td>SVR</td>
<td>0.5</td>
<td>8.88</td>
</tr>
<tr>
<td>HBM</td>
<td>3.2</td>
<td>3.49</td>
</tr>
</tbody>
</table>

Prediction error
(RMSE @ Test layout)

- Over-fitting

#tr data: 33127
10000
1000

© 2015 Toshiba Corporation
Comparison with model-based OPC (1/2)

![Graph showing Edge Placement Error (nm)]
Comparison with model-based OPC (2/2)

The number of iterations of model-based OPC can be reduced 2.
Conclusion

• Toshiba and UTDA developed a new Regression-based OPC using hierarchical Bayes model (HBM).

• The experimental results showed:
  – High accurate prediction model can be achieved with small amount of data while preventing over-fitting.
  – The number of iterations in model-based OPC can be reduced 2.