IDENTIFYING DEFECT PATTERNS IN HARD DISK DRIVE MAGNETIC MEDIA MANUFACTURING PROCESSES USING REAL AND SYNTHETIC DATA

NVIDIA GPU TECHNOLOGY CONFERENCE

Nicholas Propes | Seagate Analytics
San Jose, CA
March 29, 2018
Outline

• Seagate Technology
• Magnetic Media, Scanned Data and Defect Patterns
• Manual Feature Extraction
• Automated Feature Extraction
• Architecture / Implementation
• Results
Seagate’s Global Presence

- Beaverton, OR, USA
- Fremont, CA, USA
- Cupertino, CA, USA
- Valencia, CA, USA
- Longmont, CO, USA
- Colorado Springs, CO, USA
- Oklahoma City, OK, USA
- Shakopee, MN, USA
- Bloomington, MN, USA
- Rochester, MN, USA
- Houston, TX, USA
- Round Rock, TX, USA
- Guadalajara, Mexico
- São Paulo, Brazil
- Danderyd, Sweden
- Dublin, Ireland
- Springfield, N. Ireland
- Paris, France
- Havant, UK
- Maidenhead, UK
- Munich, Germany
- Amsterdam, Netherlands
- Moscow, Russia
- Korat, Thailand
- Teparuk, Thailand
- Johor, Malaysia
- Penang, Malaysia
- Shugart, Singapore
- Woodlands, Singapore
- Sydney, Australia
- New Delhi, India
- Mumbai, India
- Pune, India
- Bangalore, India
- Tokyo, Japan
- Taipei, Taiwan
- Hong Kong, China
- Wuxi, China
- Shenzhen, China
- Chengdu, China
- Shanghai, China
- Tianjin, China
- Beijing, China

HQs, Admin/Sales | Design | Manufacturing | Customer Support
Hard Drive / Magnetic Media

- Complex System
- > 300,000 tracks per inch
- Read/write head fly height < 20 angstroms
- Rotation speed 4500-15000 RPM
- Control of read/write head
- Lots of testing for different parameters
- HAMR area density (2 TB / sq in)
Objective: Classify defect patterns that occur on scanned magnetic media for the purpose of identifying issues in manufacturing line.
Scanning Magnetic Media Defects

Manufacturing Processes
- Washing
- Buffing / Polishing
- Sputtering
- Inspection
- etc.
## Data

### Defect Point Locations on Magnetic Media

<table>
<thead>
<tr>
<th>ID</th>
<th>SIDE</th>
<th>Radius</th>
<th>Angle (Deg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1234</td>
<td>A</td>
<td>35000</td>
<td>20</td>
</tr>
<tr>
<td>A1234</td>
<td>A</td>
<td>64301</td>
<td>50</td>
</tr>
<tr>
<td>A1234</td>
<td>A</td>
<td>45000</td>
<td>185</td>
</tr>
<tr>
<td>A1234</td>
<td>A</td>
<td>21443</td>
<td>354</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>C3212</td>
<td>B</td>
<td>54531</td>
<td>124</td>
</tr>
<tr>
<td>C3212</td>
<td>B</td>
<td>34222</td>
<td>342</td>
</tr>
<tr>
<td>C3212</td>
<td>B</td>
<td>18888</td>
<td>351</td>
</tr>
</tbody>
</table>
Defect Patterns

Pattern A

Pattern B

Pattern C

Pattern D

Pattern E

Pattern F

Pattern G

Pattern H
Method 1: Manual Feature Engineering

Clustering

{variance, number of points, etc.}

{variance, number of points, etc.}

{variance, number of points, etc.}

Feature Extraction

{variance, number of points, etc.}

{variance, number of points, etc.}

{variance, number of points, etc.}

Classification

Pattern A

Pattern B

Pattern C

etc.
Method 1: Manual Feature Engineering

Clustering Algorithms

- Spatial Grouping
- KDClus
- Tessellation
- Band-pass Filtering / Downsampling Images
- Density-based Scan (DBSCAN)
- etc.
Method 1: Manual Feature Engineering

Feature Extraction

- cluster defect counts
- cluster lengths
- cluster widths
- cluster variances
- entropy
- etc.
Method 1: Manual Feature Engineering

Classifiers

- decision trees
- fuzzy logic
- logistic regression
Method 1: Manual Feature Engineering
Method 1: Manual Feature Engineering

Classification Scheme

Pattern A Classifier
- Pattern A / Not Pattern A (and points associated)

Pattern B Classifier
- Pattern B / Not Pattern B (and points associated)

Pattern H Classifier
- Pattern H / Not Pattern H (and points associated)
Method 1: Manual Feature Engineering

Issues

• Noisy patterns
• Density changes for defect patterns
• Overlapping patterns

Makes clustering difficult to perform reliably!
Method 2: Automatic Feature Engineering

- Multiple Image Processing Layers
- Image Processing Functions are Learned from Data

- Basic Neural Net Classifier
- Parameters are Learned from Data

Band (0.9)
Heavy Galaxy (0.8)
S_Circ_MD (0.8)
S_Circ_OD (0.7)
Circ_Scratch (0.1)
…
U-Net Image Segmentation

Image Segmentation

Pattern D
U-Net Classifier

defect type
output NN layer
upsample
conv.
maxpool
conv.
conv.
conv.
conv.
conv.
Synthetic Data Generation

```xml
<xml version="1.0"/>

<file_prefix> FILE </file_prefix>
<file_start_index> 0 </file_start_index>
<number_of_files> 1000 </number_of_files>
<number_of_discs_per_file> 100 </number_of_discs_per_file>
<max_pattern_per_disc> 3 </max_pattern_per_disc>

<pattern>
  <pattern_name>
  <pattern_type>
  <param> min_num_defects : 21 </param>
  </pattern_type>
  </pattern_name>
</pattern>

<pattern>
  <pattern_name>
  <pattern_type>
  <param> min_num_defects : 21 </param>
  </pattern_type>
  </pattern_name>
</pattern>

<pattern>
  <pattern_name>
  <pattern_type>
  <param> min_num_defects : 21 </param>
  </pattern_type>
  </pattern_name>
</pattern>
```
Method 1: Manual Feature Engineering

Classification Scheme

- Pattern A Classifier
- Pattern B Classifier
- Pattern H Classifier

Pattern A CNN Image Segmentation
Pattern B CNN Image Segmentation
Pattern H CNN Image Segmentation

Pattern A / Not Pattern A (and points associated)
Pattern B / Not Pattern B (and points associated)
Pattern H / Not Pattern H (and points associated)
Method 2: Manual Feature Engineering

Classification Scheme

- **Pattern A**
  - CNN Image Segmentation
  - Pattern A / Not Pattern A (and points associated)

- **Pattern B**
  - CNN Image Segmentation
  - Pattern B / Not Pattern B (and points associated)

- **Pattern H**
  - CNN Image Segmentation
  - Pattern H / Not Pattern H (and points associated)
Pattern trained image segmentation

Input Data to CNN

Ground truth (region)

CNN output

Pattern Exist Cases

No Pattern Exist Cases
Method 2: Automatic Feature Engineering

• CNN trained with synthetic data (100K images)
• Validated with real and synthetic Data
• Simple to create models and maintain (just add/replace with new model)
• Improved accuracy with CNN
• Needs GPU or High Power CPU to perform calculations quickly
Hybrid Solution

Pattern A (Method 1)
Pattern B (Method 2)
Pattern C (Method 1)
pattern Z (Method 2)

Pattern A / Not Pattern A (and points associated)
Pattern B / Not Pattern B (and points associated)
Pattern C / Not Pattern C (and points associated)
Pattern Z / Not Pattern Z (and points associated)
Hardware

GPU Computer

• 2x NVIDIA Titan X Pascal GPUs (12 GB memory & 3584 cores each)
• 32 GB DDR4 3000 RAM
• 30 TB Hard Drive Space
• Intel Core i7-7700K 4.2 CPU
• 1000W Power Supply
Software
On Ubuntu 16.04

- **Keras**
- **TensorFlow**
- **Python 2.7.x or 3.5**
- **NVIDIA CUDA TOOLKIT and cuDNN Library**
Implementation Details

Requests to compute over network

Data

GPU Server

Python Main Application

Python Thread

...}

GPU Resource

Keras / Tensorflow

GPU
Results

Real Data Validation Results

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAND</td>
<td>97.76%</td>
</tr>
<tr>
<td>CIRCUMFERENTIAL</td>
<td>98.13%</td>
</tr>
<tr>
<td>CLUSTER</td>
<td>95.26%</td>
</tr>
<tr>
<td>HALF GALAXY</td>
<td>96.76%</td>
</tr>
<tr>
<td>NONE</td>
<td>99.50%</td>
</tr>
<tr>
<td>RADIAL</td>
<td>90.77%</td>
</tr>
<tr>
<td>SKEW</td>
<td>94.89%</td>
</tr>
<tr>
<td>GALAXY</td>
<td>85.91%</td>
</tr>
<tr>
<td>CIRCUMFERENTIAL</td>
<td>78.43%</td>
</tr>
</tbody>
</table>

- Synthetic data didn’t work well for some defect pattern classes
- Method is suitable for new defect pattern classes
- Management of models: tradeoff between memory/storage and retraining
- Some defect pattern classes may not be suitable for CNN when higher resolution scans are possible
- Future work:
  - Grouping defect patterns in different models
  - Reducing size of models
  - Improve synthetic data generation for some defect patterns
Questions?