Model-based Tabular Structure Detection and Recognition in Noisy Handwritten Documents

Jin Chen, Daniel Lopresti
Computer Science & Engineering
Lehigh University
{jic207, lopresti}@cse.lehigh.edu
Introduction

- For **table detection**, Hu et al.\[1\] introduced a medium-independent algorithm based on inside-space.
- For **table recognition**, Vajda et al.\[2\] recently used Hough transform to detect the pre-printed table substrate for their transcription system.
- The target documents may contain MP/HW text, pre-printed rulings, hand-drawn rulings, signatures, logos, etc.
- Table rulings may not present, and some table cells can be missing.

Workflow

**Processing**
- Detect HW/MP Word
- Detect HW/MP Ruling
- Detect “Key Point”
- Extract Key Point Feature
- Train Graph Model
- Predict on Testing Set
- Compute Table Row/Column
- Evaluate Performance

**Data**
- Word Polygon
- Ruling Bounding Box
- Key Point Spec. (XML)
- Key Point Feature (TXT)
- Model Parameters (TXT)
- Predicted Key Point Label
- 2-D Table Spec. (XLS)
- Performance Statistics
To focus on the table analysis, we chose to manually label handwritten words using oriented rectangles because:

- There are many existing techniques to segment handwritten text lines and words.
- Pre-printed ruling lines may be detected reliably using a model-based method\cite{1}.

Document components labeled using GEDI\cite{2}:

- pre-printed rulings, hand-drawn rulings
- machine-printed/handwritten text
- other writing: e.g., signatures
- text lines (groups of HW/MP)
- table rows/columns (groups of HW/MP)

\cite{1}. J. Chen and D. Lopresti. A model-based ruling line detection algorithm for noisy handwritten documents. ICDAR 2011.
\cite{2}. D. Doermann, E. Zotkina, and H. Li. GEDI—a ground truthing environment for document images. DAS 2010.
To focus on the table analysis, we chose to manually label handwritten words using oriented rectangles because:

• There is a large amount of existing techniques to segment handwritten text lines and words.
• Pre-printed ruling lines may be detected using model-based approaches.
To handle tables with/without table rulings, we define *key points* to be an overlapping region of horizontal and vertical spacing between adjacent text.

As in Hu et al.’s algorithm.

A key point is indeed a region, but is shown as a cross-sign in our figures.
How to decide the minimum spacing required for a key point?

- Small spacing introduces false-alarms between plain text lines.
- Large spacing neglects legitimate key points in table structures.

We estimate the threshold spacing $W$ by assuming the inter-word spacing $Z$ follows a bimodal distribution:

$$p(z) = \pi_1 \mathcal{N}(\mu_1, \Sigma_1) + \pi_2 \mathcal{N}(\mu_2, \Sigma_2)$$

where $\pi_i$ is the weight between Gaussians and is set to be 1.

Next, we use Expectation-Maximization (EM) to find the maximum-likelihood estimates of the mean and the covariance matrices.
Using K-Means to initialize the model, we compute the probability of a sample belonging to Gaussian $i$:

$$\lambda_{ik} = \frac{\mathcal{N}_i(z; \mu_i, \Sigma_i)}{\sum_{j=1}^{2} \mathcal{N}_j(z; \mu_j, \Sigma_j)}$$

Then at the M-step, the model is updated using computed probabilities from the E-step:

$$\mu_i = \frac{\sum_{j=1}^{n} \lambda_{ik} z_k}{\lambda_{ik}}$$

$$\Sigma_i = \frac{\sum_{j=1}^{n} \lambda_{ik} (z_k - \mu_i)(z_k - \mu_i)^T}{\sum_{j=1}^{n} \lambda_{ik}}$$

Considering the number of samples for the EM estimation, we set the covariance matrix to be $\gamma_i \mathbf{I}$, an identity matrix with a scaling factor. Finally, we set the threshold spacing to be $\mathcal{W} = \mu_1 + 2\sqrt{\gamma_1}$.
**Key Point Detection**

Detected *preliminary key points*, shown in cross-signs.

<table>
<thead>
<tr>
<th>Delivered and Filtration Efficiencies of Charcoal Cigarettes with Wet and Dry BPL Carbon</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compound</strong></td>
</tr>
<tr>
<td>Methane</td>
</tr>
<tr>
<td>Ethylene</td>
</tr>
<tr>
<td>Ethanol</td>
</tr>
<tr>
<td>Propylene</td>
</tr>
</tbody>
</table>
Impact of Hand-drawn Rulings

Hand-drawn rulings usually indicate separation of plain text lines and tables, or connection between table rows. We observed the following rules classifying such rulings reliably:

- Separating rulings have T-junctions only while connecting ones should have at least one crossing.
- If multiple horizontal ruling segments are present, they serve as connecting rulings.
Impact of Hand-drawn Rulings

If separating and connecting rulings present, we use them to validate the key points:

- Vertical rulings validate key points with small widths.
- Connecting rulings validate key points with heights larger than the average spacing between text lines.
- Separating rulings invalidate key points even if they have reasonable widths.

For clarity, the rulings are not shown.
Key Point Grid Generation

- First, cluster key points based on their x-positions using Basic Sequential Algorithmic Schemes (BSAS).
- Next within each cluster, build horizontal projection profiles (HPPs) of key points to find columns of key points.
- Traverse each row of key points to add imaginary key points, or split them into two if they span multiple table columns.

For clarity, the rulings are not shown.
Denote $X = \{x_i\}$ to be the observed features from each node, and $Y = \{y_i\}$ to be random variables over corresponding labels, i.e., 1 means true key points in tables and 0 means otherwise. Then the joint distribution has the following form:

$$p(y_i | x_i) \propto \exp(pA(y_i, X) + q \sum_{\{(i),(j)\} \in E} I(y_i, y_j, X))$$

where $A(\cdot)$ is the association potential describing how likely the node has a specific label, and $I(\cdot)$ is the interaction potential measuring influences of neighboring nodes.

In our work, we define these two potentials as follows:

$$A(y_i, X) = \sum_{j \in CN} \frac{\text{ColFeat}(j)}{\exp(-R(CN))} + \sum_{j \in RN} \frac{\text{RowFeat}(j)}{\exp(-1)}$$

$$I(y_i, y_j, X) = \exp(y_i \times y_j)$$
Currently, the row and column features are simple binary indicators.

\[
\begin{align*}
\text{ColFeat}(KR) &= \begin{cases} 
0 & \text{if } w < \overline{w} \\
1 & \text{otherwise}
\end{cases} \\
\text{RowFeat}(\cdot) &= \begin{cases} 
1 & \text{if any HW words exist} \\
0 & \text{otherwise}
\end{cases} \\
\text{ColFeat}(KI) &= \begin{cases} 
-1 & \text{if } w < \overline{w} \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

To facilitate the graph labeling, we define source capacity to be the association potential \( A(y_i, X) \), and the sink capacity to be \( 1 - A(y_i, X) \).
Graph Models for Tables

In addition, due to the scale of the dataset available for now, we manually set the weights to be equal. When more data becomes available, we plan to learn the weights from labeled key points.

On the left, detected key points and imagined ones are fed to the max-flow/min-cut algorithm, which labels true key points in tables on the right.
Table Analysis Results

The output of our table analysis algorithm is a 2-D arrangement of table cells, as can be drawn in a spreadsheet.

<table>
<thead>
<tr>
<th>Component</th>
<th>32307 T.S.</th>
<th>BPL + 26%</th>
<th>H2O Included</th>
<th>BPL, Dried</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methane</td>
<td>1143</td>
<td>1141</td>
<td></td>
<td>1090</td>
</tr>
<tr>
<td>Ethylene</td>
<td>216</td>
<td>250</td>
<td></td>
<td>230</td>
</tr>
<tr>
<td>Ethane</td>
<td>399</td>
<td>404</td>
<td></td>
<td>363</td>
</tr>
<tr>
<td>Propylene</td>
<td>214</td>
<td>200</td>
<td>6.5</td>
<td>90</td>
</tr>
</tbody>
</table>
Data Preparation:
• Collected MP table documents from Tobacco800.
• Asked college students to copy the content to paper using their own choice of pen and sheets. Students also decided where to break lines and the spacing between words and table columns.
• To mimic a private handwriting dataset, we added/removed rulings in the template tables and left certain cells empty.
• Each handwritten document was scanned at 600 DPI into PDF files, using an HP copier with a bitonal setting in plain text mode.
• The typical size of extracted TIFF images is 5100w * 6600h.

As an ongoing event, we have collected 20 pages from two students, and aim for a dataset with 200 pages or more.
We have labeled 584 table cells (369 key points) from the 20 pages with 22 handwritten tables.

The cell precision is 100% and the cell recall is 93%.

\[
\text{cell precision} = \frac{\text{number of segmented true table cells}}{\text{number of segmented table cells}}
\]

\[
\text{cell recall} = \frac{\text{number of segmented true table cells}}{\text{total number of true table cells}}
\]

More errors occurred when no hand-drawn rulings presented and some table rows are isolated from the others.

Currently, the evaluation is still preliminary and eventually we would like to quantify how much effort a human user needs to correct the errors that an algorithm makes.
Experimental Evaluation

- Number of segmented true table cells = 12.
- Number of segmented table cells = 15.
- Total number of true table cells = 24.
- Cell precision = 80%.
- Cell recall = 50%.

*Hard cases include ruling-free tables, irregular row/column spacing.*
More Results

Cell precision = 100%.
Cell recall = 100%.

Cell precision = 100%.
Cell recall = 100%.
Conclusions

• Introduced a model-based framework for detecting and recognizing handwritten tables.
• The cell precision is 100% and the cell recall is 93%.
• More errors occurred when no hand-drawn rulings presented and some table rows are isolated from the others.

For future work:
• Add automatic modules to find rulings and text components.
• Relax key point detection to not rely on text lines as input, but rather work on text words based on proximity analysis.
• Label detected key points to enable supervised graph learning.
• Quantify human effort on correcting algorithmic errors for performance evaluation.
Thank You!