Automatic Information Extraction from 2-Dimensional Plots in Digital Documents

William Brouwer¹, Saurabh Kataria², Sujatha Das², Prasenjit Mitra², C. Lee Giles²

¹Department of Chemistry
²College of Information Sciences and Technology
The Pennsylvania State University, University Park, PA
wjb19@psu.edu, skataria@ist.psu.edu, gud111@ist.psu.edu, pmitra@ist.psu.edu, giles@ist.psu.edu

Funded by NSF Chemistry
Outline

• Motivation
• Related Work
• Our Contribution
• System Components
  – 2D plot identification, segmentation
  – Axis, tic period identification
  – Data series extraction, disambiguation
  – Optical Character Recognition
• Current Work
• Applications
• Conclusions & Future Work
Motivation

- Large amount of results in digital documents are recorded in figures, time series, experimental results (e.g., NMR spectra).
- Extraction is an important task, for purposes of:
  - Further modeling using presented data
  - Indexing, meta-data creation for storage & search on figures
- Current extraction done manually!
- Goal: automate this process
  - There exists a large body of available algorithms and techniques which may be applied directly to the task.
  - These can be combined with some novel approaches and heuristic information to create figure extraction systems tailored to specific figure types, in this case 2D plots.
- Place in XML data file for reuse
- Integrate into a working digital library
  - ChemXSeer [http://chemxseer.ist.psu.edu](http://chemxseer.ist.psu.edu)
  - CiteSeerX [http://citeseerx.ist.psu.edu](http://citeseerx.ist.psu.edu)
Related Work

- Picture and non picture classification (Jia Li et al.)
  - Uses wavelet coefficients detect picture and non picture blocks in images
- Image categorization in digital documents (Xiaonan Lu et al.)
  - graph images & others
- Curve data extraction from 2-D plots (Lu et al.)
  - Uses thinning and chain coding to detect curves in plots.
- Finding Text in Images (Wu 1997 et al.)
  - Uses texture features to identify the text blocks in images
- Describing a graphical pattern corrupted by noise. (Carnevali et al.)
  - Apply simulated annealing, detect known but corrupted shapes
Our contribution

• 2D plot Recognition
• Segmentation
  – Axis & tic labels, legend, axes co-ordinates
• Extraction
  – Line/data point segregation
  – data point co-ordinates
  – overlapping data point disambiguation
• Optical Character Recognition (OCR)
  – layout analysis & disambiguation
• Current Work
  – Rotation invariance for OCR
  – Kernel Density Estimation to improve data point extraction from background of extraneous objects
  – Integrate into existing system
Metadata & data to extract:
2 Dimensional Plot

- y-label
- x-label
- y-axis
- x-axis
- numerical scale(s)
- tics
- data points
- legend

Diagram labels:
- Concentration (M)
- Time (s)
System Overview: Automatic Figure Information Extraction

- **Figure doc**
- **Pre-process, identify segment**
  - **Axes indices**
    - CCL, Fuzzy line filter
      - Labelled Points
        - Shape Analysis, Disambiguation
          - Data Series
  - **Plot data pixel region**
    - Text Filter
      - Text Segmentation, Disambiguation, Optical Character Recognition
    - Tic Period, axis classification
  - **Axes pixel regions**
    - Text Filter
      - Text Segmentation, Disambiguation, Optical Character Recognition
      - Axis labels, tic labels
System Components

- Identify, convert/binarize image, and segment according to features
- Subsequent fields are plot region, x and y axis regions and legend
- Using axis coordinates (indices of data matrix) in conjunction with projection of axis pixels to establish numerical periodicity of tics.
- Apply supervised learning, CCL & fuzzy line filter to segregate lines/curves from points, obtain data series with co-ordinates
- Apply simulated annealing to overlapping/obscured data points
- OCR to extract text from axis labels, with heuristic-based algorithms to improve OCR accuracy
2D plot Identification/Segmentation

- Use approach of Li et al, divide image into blocks, create wavelet coefficients for each as training features for linear SVM
- Augment these features with those germane to 2D plots, eg., orthogonal axes and textual phrases (indicative of 2D plots, eg., 'slope', 'plot', 'range')
- Further supported/validated by profile based heuristics, e.g., axes correspond to maxima in pixel projections in orthogonal directions

![Diagram showing pixel projections along rows and columns](image-url)
Results: Segmentation

<table>
<thead>
<tr>
<th>Features</th>
<th>%Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only IS</td>
<td>85.2</td>
</tr>
<tr>
<td>Only CT</td>
<td>78.3</td>
</tr>
<tr>
<td>IS + CA</td>
<td>85.8</td>
</tr>
<tr>
<td>CT+CA</td>
<td>80.7</td>
</tr>
<tr>
<td>IS+CT</td>
<td>85.6</td>
</tr>
<tr>
<td>All</td>
<td>88.3</td>
</tr>
</tbody>
</table>

IS==Image Segmentation
CT==Caption Text
CA==Co-ordinate Axes
Tic Period Determination

- Pixel regions immediately adjacent to axes give absolute measure to indices of data points extracted from data region, may be contaminated by noise and presence of smaller tics
- Alleviate using FFT & properties of Dirac function comb:

\[ F \left( \sum \delta(t - nT) \right) = \frac{1}{T} \sum \delta \left( n - \frac{k}{T} \right) \]

- Numerical tic period is difference between largest two peaks in twice FFT’d pixel projection data, *pattern may also be used for classification (eg., log, linear etc)*

- Graphic showing tic period determination on linear and log axes with associated pixel projections.
Steps Thus Far…

- After recognition, segmentation and extraction of tic information for numerical scaling of pixel indices, it remains to recognize individual data points and textual content.
- *Supervised ML is robust, prevents objects (data points) from being identified differently to clusters of identical objects.*
- Conventional OCR with some added page layout/analysis works well for text.
Dimension Reduction and Data Normalization

• Use matrix methods to reduce data dimensions and normalize; eg., 5x5 matrix representing a square is reduced to series of normalized eigenvalues

• Formula using Grammian ($A^T A$) provides translation invariance, normalization using trace

```
0 0 0 0 0
0 1 1 1 0
0 1 1 1 0
0 1 1 1 0
0 0 0 0 0
```

"square"\[\rightarrow\]

\[
eig(A^T A) \quad \text{trace}(A^T A) \quad \lambda_1, \lambda_2, \lambda_3,.....
\]
Features for Supervised Learning

- Good discrimination properties, decision regions are fairly distinct
- Use SVM to classify independent, fully resolved data elements

Elements which fail classification overlap -> use *simulated annealing* to disambiguate…
Algorithm for overlapping data points

- Use sums of identified, resolved pixel regions $e_{ij}$, $d_{ij}$, ... to create image $A$, to approximate real image $B$ of overlapping points.
- Optimize indices $i,j$ (offset), weights $b,a$ and total numbers $L,M$ using simulated anneal.

\[
A = \sum_{l} b_l e_{ij} + \sum_{m} a_m d_{ijm} =
\]
Simulated Annealing for Data Point Disambiguation

- Cost function ($E$) is squared difference between generated ($A$) and target ($B$) images.
- Use simulated anneal to create best approximation to $A$; *Metropolis scheme with rapid annealing & re-annealing ensures convergence*
- Absolute numbers of points/images used trimmed successively, co-ordinates/indices randomly walked around available space

\[
E = Tr\left\{ (A - B)^T [(A - B)] \right\}
\]

Boltzmann factor:
\[
P = \exp\left(-\frac{(E_f - E_i)}{T}\right)
\]

Annealing Schedule:
\[
T_f = (1 - \varepsilon) T_i^*;
\]
\[
T_i^* = T_f + \text{mod}(n, \alpha) \Delta E_i
\]
Examples of Machine Learnt Overlapping Points

• Simulated annealing gives excellent results for retrieving both data identity and coordinate information, even in the presence of noise.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Total</th>
<th>#Correct</th>
<th>%Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diamond</td>
<td>72</td>
<td>64</td>
<td>88.9</td>
</tr>
<tr>
<td>Triangle</td>
<td>78</td>
<td>71</td>
<td>91.0</td>
</tr>
</tbody>
</table>
Optical Character Recognition & Issues

- Problem considered ‘solved’ for conventional, well posed text
- Figures provide challenges in that text is located amongst extraneous objects
- Use heuristics to extract eg., axis text labels lie outside axes
- Additional challenges arise from a) overlap between characters and b) arbitrary orientations c) presence of extraneous objects eg., lines
- A custom algorithm has been devised for a)
- Problems b) and c) being treated in current work; once text is appropriately segmented, conventional OCR works very well eg., latest release GOCR and OCRopus
Character Segmentation

- Use profile based heuristic to initially locate text blocks; generally these occur in regular positions eg., labels, legends

- Apply novel algorithm to locate points of contact/overlap
  - Determine average character width $\mu$ and std deviation $\sigma$
  - If text block width is $> \mu+\sigma$, apply further segmentation recursively, looking for points of contact between characters

Minima in pixel projections & local points of inflection
Example: Accurate Character Segmentation

Fails on text combination widths $< \mu + \sigma$

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>83.5%</td>
<td>75%</td>
<td>80.1%</td>
</tr>
</tbody>
</table>
Further Problems....

Hollow/narrow data points

Mixed text orientations
Current Work: Robust line & point segregation, identification

- Data Points and lines overlap, solid points & lines easily segregated since lines ~ axis width, use fuzzy k-means
- *Hollow/thin points more difficult, can be thrown out with lines* -> *use KDE to identify their positions*
- Using KDE may also prevent hollow points being identified differently eg.,

\[ \Delta \quad \triangle \]

are different from machine perspective -> identify using KDE and fill in

\[ \blacktriangle \quad \rightarrow \quad \blacktriangle \]

Classified as identical
Kernel Density Estimation

• Non-parametric and unsupervised method; count modes in $d$-dimensional probability density function (PDF) to determine modes/clusters

• Parzen/Rosenblatt PDF defined as:

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{1}{h} K \left[ \frac{x - X_i}{h} \right] \right)$$

• $K$ is kernel function (eg., Gaussian) and $h$ is bandwidth, to be determined from data, $n$ random samples

• Bandwidth $h$ may be chosen various ways eg., Sheather-Jones, significant speed improvements using Fast Gauss Transform
Example: KDE applied to point/line segregation

- Mode counting of KDE reveals three significant distributions whose pixels correspond to data points
- Distribution of line pixels broad and underlying
Current Work: Rotation Invariant Features

- Plots contain mixtures of text orientation eg., x-axis label and tic labels
- Calculating rotation invariant moments expensive (eg., Zernike), use log-polar coordinates about center of mass instead ie.,
  \[ \eta = \arctan(j/i) \]
  \[ \xi = \log \sqrt{i^2 + j^2} \]
- Projection of polar image is invariant to rotation, however ambiguity arises eg., between characters ‘d’ and ‘p’

Rotate 90 degs ->
Results Summary

- A number of useful elements are **readily available** while a number of others are with **less certainty**; current work is devoted to these.
- With the elements available, a number of tasks may be performed....
Application: Reconstructing Spectra

- Nuclear Magnetic Resonance (NMR) spectroscopy data often presented as contour plots
- Spectra contain information on local bonding arrangements and relative concentrations of chemical species
- *Use CCL and 2D interpolation with methods outlined here to re-create spectral data from image*

\[ ^{45}\text{Sc MQMAS spectra image, Scandium Oxide} \]  

\[ \text{Intensity + frequencies} \]
Structural Refinement

- Data extracted from spectra may be coupled with information mined from text e.g., atomic coordination number, bond lengths, angles
- *Use information + machine learning to solve the inverse problem i.e., structure determination using input spectra*
Application: Chemical Kinetics

- Time series of chemical species concentration; extracted data may be used with Kalman/particle filter etc for modeling and learning factors of relevance in diverse chemical systems.

\[
\frac{dx_i}{dt} = f_i(x) + \sum_j^n g_i^j(x)\xi_j(t)
\]
Conclusions / Future Work

- 2 dimensional figures are largely ignored in digital libraries yet contain valuable information.
- Traditional algorithms with some changes and heuristics combine to form tailored figure extraction systems.
- Information made available includes data for further modeling as well as metadata for indexing etc.
- Points to possibility of machine learning on extracted data to perform complex tasks eg., use of NMR spectra in predicting structure, modeling chemical kinetics in environmentally relevant scenarios.
- System integration within ChemXSeer:
  - XML data generation
  - Open source tool for Lucene/SOLR
- Extension to other figures (3D, ...)
References


