Bayesian hierarchical modeling for the forensic evaluation of handwritten documents

Amy Crawford

Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/etd

Recommended Citation
Crawford, Amy, "Bayesian hierarchical modeling for the forensic evaluation of handwritten documents" (2020). Graduate Theses and Dissertations. 18078.
https://lib.dr.iastate.edu/etd/18078

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Bayesian hierarchical modeling for the forensic evaluation of handwritten documents

by

Amy M. Crawford

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Statistics

Program of Study Committee:
Alicia Carriquiry, Co-major Professor
Danica Ommen, Co-major Professor
Jarad Niemi
Daniel Nordman
Vivekananda Roy

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2020
Copyright © Amy M. Crawford, 2020. All rights reserved.
DEDICATION

To my family – Mom, Dad, Brady, and Katie. Thank you for instilling the value of hard work and always leading by example. I will be forever grateful for the strong foundation that you provide; it has supported all of the growth and learning in my life.
TABLE OF CONTENTS

LIST OF TABLES ............................................................... v

LIST OF FIGURES ............................................................ vi

ACKNOWLEDGMENTS .......................................................... x

ABSTRACT ........................................................................ xi

CHAPTER 1. INTRODUCTION ................................................ 1
  1.1 A Brief U.S. History and Background of Forensic Handwriting Analysis .... 1
  1.2 Need for Statistical Foundations in the Discipline .................................. 3
  1.3 Other Systematic and Statistical Approaches ........................................ 5
  1.4 Outline of Methods and Results .......................................................... 6
  1.5 References ........................................................................ 9

CHAPTER 2. K-MEANS CLUSTERING AS A FEATURE EXTRACTION TECHNIQUE
  FOR HANDWRITTEN DOCUMENTS ........................................ 12
  2.1 Abstract ........................................................................ 12
  2.2 Introduction ..................................................................... 13
    2.2.1 A Motivating Example .................................................. 15
    2.2.2 Paper Organization ..................................................... 16
  2.3 Document Processing ........................................................ 16
    2.3.1 Pre-Processing ............................................................. 16
    2.3.2 Data Extraction .......................................................... 18
    2.3.3 Deterministic Adjacency Grouping .................................. 19
  2.4 Clustering ........................................................................ 20
    2.4.1 Edit Distances ............................................................. 21
    2.4.2 Distance Measure for Graphs ......................................... 21
    2.4.3 Weighted Mean of Graphs .............................................. 27
    2.4.4 K–means Type Algorithm ............................................. 30
  2.5 Application ...................................................................... 32
    2.5.1 Cluster Assignments ..................................................... 33
    2.5.2 Writer Identification ...................................................... 37
  2.6 Discussion ...................................................................... 44
  2.7 References ...................................................................... 45
CHAPTER 3. A STATISTICAL ANALYSIS OF FORENSIC HANDWRITING WITH INSIGHTS REGARDING WRITER STYLES ................................................. 48
  3.1 Abstract ................................................................. 48
    3.1.1 Significance Statement ......................................... 48
  3.2 Introduction ............................................................ 49
  3.3 Handwriting as Data .................................................. 51
    3.3.1 Feature Extraction with K-Means Clustering ............... 52
  3.4 Methods ................................................................. 55
    3.4.1 Bayesian Hierarchical Mixture Model ....................... 56
    3.4.2 Posterior Estimation ............................................ 57
    3.4.3 Writing Style Insights ......................................... 57
    3.4.4 Posterior Predictive Distributions and Writer Identification Analysis .. 61
    3.4.5 Examining Error ................................................ 61
  3.5 Discussion .............................................................. 64
  3.6 References ............................................................. 65

CHAPTER 4. AN ANGULAR FEATURE OF HANDWRITING COMPONENTS AND EXTENDED BAYESIAN HIERARCHICAL MODELING FOR FORENSIC WRITER IDENTIFICATION ....................................................... 68
  4.1 Abstract ................................................................. 68
  4.2 Introduction ............................................................. 69
  4.3 Part I. Feature Extraction and Extended Modeling ............... 71
    4.3.1 Data ............................................................... 71
    4.3.2 Reference Analysis .............................................. 72
    4.3.3 Centroid-based Slope .......................................... 74
    4.3.4 Rotation Angles through Eigen-Decomposition ............. 78
    4.3.5 Final Model Specification ..................................... 96
    4.3.6 Part I, Other Considerations .................................. 98
  4.4 Part II. Model Testing and Applications .......................... 98
    4.4.1 Application 1 (CSAFE data): Extended testing sets ....... 100
    4.4.2 Application 2 (CSAFE + CVL data): Combining two data sources and an out-of-set evaluation ......................... 104
    4.4.3 Application 3 (IAM data): Test on decreasing number of sentences ...... 108
  4.5 Discussion .............................................................. 112
  4.6 References ............................................................. 114
  4.7 Appendix to Chapter 4: Ad-Hoc Importance Ordering of Clusters ...... 115

CHAPTER 5. DISCUSSION AND FUTURE WORK ..................................... 119
  5.1 Discussion .............................................................. 119
  5.2 Future Work ............................................................ 120
  5.3 References ............................................................. 121
LIST OF TABLES

Table 4.1  Results of the prior sensitivity analysis conducted for the final model specification. The first three lines correspond to varying levels of prior information. The last two lines include parameter settings that affect the concentration of the wrapped Cauchy densities in Equations 4.31 and 4.33. An analysis of the holdout documents was conducted for each set of parameter settings, the percentage of correct assignment is included in the final column with a corresponding 95% credible interval.

Table 4.2  Using the final model from Section 4.3.5, writing samples from the seven prompt/session combinations that have not yet been explored are evaluated. * The eighth result shown here is the test set used in Section 4.3.5, same information is provided in Table 4.1. The median, minimum, and maximum number of graphs extracted from the documents in each test set are provided. The percentage of true writer assignment according to $\bar{p}$ is included in the rightmost column. ** Training data used to estimate model parameters, see Section 4.3.5.

Table 4.3  Four amounts of testing data (in number of sentences) are evaluated. The median, minimum, and maximum number of graphs extracted from each test set are provided. The percentage of true writer assignment according to $\bar{p}$ is included in the rightmost column with a 95% credible set.

Table 4.4  Twenty clusters with the highest $RSS_{adj}$ values. Cluster #s are arbitrary labels from the clustering algorithm. Pop. rank shows the clusters popularity ranking among the training data, Cluster #32 is assigned to the most graphs and Cluster #17 the fewest.

Table 4.5  Predictive results for which only rotation angles of graphs assigned to subsets of clusters were considered. The subset correspond to the rankings of the clusters $RSS_{adj}^k$ defined above. Subsets of size 10 and 20 were considered. “pop20” is the subset of clusters that are the most commonly occurring among graphs in the training data set (the most populated clusters).
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Three graphs whose assignment will contribute to characterizing a writer’s style. D is deterministic group assignment. C is cluster group assignment.</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Partitioning of the cursive word ‘csafe’</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>The letter ‘x’ showing vertices and edge labels from <em>handwriter</em></td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Graphs and their adjacency matrices</td>
<td>20</td>
</tr>
<tr>
<td>2.5</td>
<td>Edge distance measure calculation companion figures</td>
<td>23</td>
</tr>
<tr>
<td>2.6</td>
<td>Weighted mean transition examples for 4 pairs of graphs</td>
<td>29</td>
</tr>
<tr>
<td>2.7</td>
<td>Examples of typical outlier graphs</td>
<td>31</td>
</tr>
<tr>
<td>2.8</td>
<td>Cluster exemplars for CVL handwriting data</td>
<td>34</td>
</tr>
<tr>
<td>2.9</td>
<td>Cluster raw means for CVL handwriting data</td>
<td>35</td>
</tr>
<tr>
<td>2.10</td>
<td>Subset of the outlier cluster for CVL handwriting data</td>
<td>37</td>
</tr>
<tr>
<td>2.11</td>
<td>Densities of posterior samples for multinomial parameter vectors</td>
<td>39</td>
</tr>
<tr>
<td>2.12</td>
<td>Writer identification results for estimated clusters</td>
<td>41</td>
</tr>
<tr>
<td>2.13</td>
<td>Writer identification results for deterministic adjacency groupings</td>
<td>43</td>
</tr>
<tr>
<td>3.1</td>
<td>Connected ink corresponding to the letter ‘f’ that has been parsed and extracted by the <em>handwriter</em> R package. The structure is treated as an attributed graph. Edges of the graph are marked in gray. Nodes are marked by red points, and include edge endpoints and intersections.</td>
<td>51</td>
</tr>
<tr>
<td>3.2</td>
<td>Cluster exemplars (red) and members (stacked in gray behind) for 10 clusters in the final $K$-means iteration of the template creation process.</td>
<td>54</td>
</tr>
<tr>
<td>3.3</td>
<td>Scaled cluster assignment vectors for three documents in the training dataset. The top is a document by Writer 12, the middle by Writer 66, and the bottom by Writer 124. For comparability, the vectors have been scaled so that the cluster assignments sum to one, rather than the number of graphs in each document. The result is a proportional breakdown of each document.</td>
<td>55</td>
</tr>
<tr>
<td>3.4</td>
<td>Posterior density estimates for the $\pi_{w,i}$ elements shown for $w = 1, \ldots, 90$ and $i = 1, \ldots, 10$ (the ten most frequently occurring clusters in the training data). Panes are labelled with the cluster number they correspond to, and include one density for each writer. Each writer’s density is colored by their average posterior $\hat{\rho}_w$ value. Posterior estimates for Writer 66 are highlighted and outlined.</td>
<td>58</td>
</tr>
</tbody>
</table>
Figure 3.5  Average relative frequencies of the ten most frequently occurring clusters in
the training data taken over the 3 training documents. Data are shown for
20 writers that were selected to show the spectrum of \( \hat{\rho}_w \) values. Panes are
labeled by writer number and plots in each pane are colored to reflect the
average posterior \( \hat{\rho}_w \) value estimated for the writer.  . . . . . . . . . . . . . . 59

Figure 3.6  Writing samples taken from the training documents for 10 of the writers
from Figure 3.5, colored by average posterior \( \hat{\rho}_w \) value.  . . . . . . . . . . . . . . 60

Figure 3.7  Breakdown of incorrect (off-diagonal) probability assignment for the 90 hold-
out documents. In each row, the tiles show the breakdown of all off-diagonal
probability assignment. Rows of the plot are independent of one another,
and each sums to one.  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62

Figure 3.8  Results of the posterior predictive analysis defined in Equations 3.7 - 3.9 for
each of the 90 holdout documents. There is one row to summarize writership
results of each questioned document. Ground truth writer is labeled on the
left-hand side. Columns are labeled by known writers. Cells are colored by
elements of their \( \bar{p} \) vector defined in Equation 3.9. Rows are independent and
sum to one, as all of the posterior writership probability is spread across the
writers. Diagonal cells indicate the correct identification for the questioned
documents.  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63

Figure 4.1  Posterior probability of writership as defined in Equations 4.4 - 4.7 for each
of the 90 holdout documents, one row for each. True writers are labeled on
the left-hand side. Columns are labeled by known writers. Cells are colored by
elements of the \( \bar{p} \) vectors for each holdout document. Thus, each row
sums to one and stands alone.  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75

Figure 4.2  The process of finding centroid-based slopes shown for two graphs. Moving
from left to right, the images show a processed graph, the location of graph
centroid, and finally the left and right hand centroids with a line connecting
them. The slope of that line is taken as the centroid-based slope feature.
Slopes for the examples shown here are 0.295 (top) and 1.02 (bottom).  . . . 76

Figure 4.3  Results of the posterior analysis from the model with centroid-based slopes
for each of the 90 holdout documents. One row for each document, ground
truth writers are labeled on the left-hand side. Columns are labeled by
known writers. Cells are colored by elements of the \( \bar{p} \) vectors for each
holdout document. Thus, each row sums to one and stands alone.  . . . . . . . 79

Figure 4.4  Writing samples from the training documents of two writers, six graphs from
each are included. Graphs are accompanied by a unit vector pointing in the
direction of the rotation angle, \( \theta \), for the corresponding graph, given by the
principal component analysis. Graphs are plotted to fill the space in the
figure, so pixels may be shown in different sizes.  . . . . . . . . . . . . . . . . . . . . . 81

Figure 4.5  The cluster exemplars (red) and cluster members (grey) from the final iter-
ation of clustering template creation (see Chapters 2 and 3).  . . . . . . . . . . . 82
Figure 4.6 Left, for all Writer/Cluster combinations: A traditional histogram of the rotation angles with 20 bins, a point for each observation is plotted on the x-axis. Right, for all Writer/Cluster combinations: A Nightingale Rose diagram displaying the rotation angles in 20 ‘petals’. A point for each observed rotation angle is plotted on the outermost ring of the diagram.

Figure 4.7 A plot depicting elements of the data model \( SRA_{w,k,j} \sim \text{Beta}(\alpha_{w,k}, \beta_{w,k}) \) with \( w = 95 \) and \( k = 29 \). Top: values of \( \alpha_{95,29}^{(m)} \) and \( \beta_{95,29}^{(m)} \) for 12 iterations, \( m \), taken from the middle of the chain and used to plot the 12 corresponding beta densities. Bottom: the scaled rotation angles, \( SRA_{w,k,j} \), \( \forall j \), serving as data for parameter estimation.

Figure 4.8 A plot depicting elements of the data model \( SRA_{w,k,j} \sim \text{Beta}(\alpha_{w,k}, \beta_{w,k}) \) with \( w = 1 \) and \( k = 16 \). Top: values of \( \alpha_{1,16}^{(m)} \) and \( \beta_{1,16}^{(m)} \) for 12 iterations, \( m \), taken from the middle of the chain and used to plot the 12 corresponding beta densities. Bottom: the scaled rotation angles, \( SRA_{w,k,j} \), \( \forall j \), serving as data for parameter estimation.

Figure 4.9 The scaled rotation angles for graphs in Cluster #16 extracted from the holdout document by Writer #85.

Figure 4.10 Estimated posterior probabilities \( (\tilde{p}) \) for the holdout document by Writer #85. Values are rounded to two decimal places and labels for values less than 0.02 are omitted.

Figure 4.11 A demonstration of rotation angle behavior (a.1) and fully wrapped angles on the entire polar coordinate system (a.2) and (b). Left (a.2 and b): rose diagram with 20 petals for rotation angles on the upper half plane, as originally measured. Right (a.2 and b): a 20 petal rose diagram for fully wrapped rotation angles. A wrapped Cauchy density is shown in red with parameter values set to the MLEs for the corresponding data.

Figure 4.12 The Beta(4,1) distribution.

Figure 4.13 Rose diagrams depicting wrapped rotation angles from six different writer/cluster combinations. Each includes a corresponding wrapped Cauchy distribution shown in red. These are not estimated posterior distributions, rather, each fit using maximum likelihood estimation of the location and concentration parameters. In the title of each plot is the average value of \( \phi_{w,k} \)

\[
\left( \hat{\phi}_{w,k} = \frac{1}{3000} \sum_{m=1}^{3000} \phi_{w,k}^{(m)} \right),
\]

along with the writer and cluster to which the data were assigned.

Figure 4.14 For each cluster \( k \) shown on the x-axis, a boxplot is constructed using 90 values of \( \hat{\phi}_{w,k} \), one for each writer. The red x shows the Monte Carlo estimate for the median of each \( \theta_k \), from Equation 4.29.

Figure 4.15 Loop measurement examples are shown in blue for graphs from the training documents of Writers #1 and #95.
Figure 4.16  ROC curves for all eight test sets evaluated under the final model estimated for the CSAFE session #1 LND prompts. For a grid of cut-off values in (0,1), a true positive is declared if the value in the index of \( \bar{p} \) corresponding to the ground truth writer is greater than the cut-off. A false positive is declared if the value in that index is less than the cut-off.

Figure 4.17  Results of the posterior analysis under the final model, estimated for 80 CVL and 80 CSAFE writers, one holdout document for each writer is evaluated. True writers of each holdout document are labeled on the rows. Columns are labeled by known writers, and cells are colored by corresponding elements of the \( \bar{p} \) vectors. Vector elements < 0.01 are removed from the plot, showing the gridlines behind. Labels are ordered first by data sources, with CVL writers on the left hand side of the \( x \)-axis, and CSAFE writers on the right, then by writer number. Four holdout documents with the smallest (< 0.5) correct probability assignment are labelled.

Figure 4.18  Results of the posterior analysis under the final model, estimator 80 CVL and 80 CSAFE writers in the closed-set. Fourteen documents generated by writers outside the closed-set are evaluated. True writers of each holdout document are labeled on the rows. Columns are labeled by known writers, and cells are colored by corresponding elements of the \( \bar{p} \) vectors. Vector elements < 0.01 are removed from the plot, showing the gridlines behind. All \( \bar{p} \) elements \( \geq 0.1 \) are labeled.

Figure 4.19  Results of the posterior analysis under the final model, estimated using 16 sentences for each of the 45 writers in the closed-set taken from the IAM database. The four plots are shown on decreasing amounts of test data. True writers of each holdout document are labeled on the rows. Columns are labeled by known writers, and cells are colored by corresponding elements of the \( \bar{p} \) vectors. Vector elements < 0.01 are removed from the plot, showing the gridlines behind. Elements of the \( \bar{p} \) vector corresponding to the ground truth writer are labeled only if they are less than 0.9, and all other elements \( \geq 0.1 \) are labeled.

Figure 4.20  Distributions of the number of graphs in the holdout sentences across all 45 writers in the IAM closed-set. To be eligible for the analysis dataset, sentences were required to contain a minimum of 30 graphs after decomposition by the handwriter R package.

Figure 4.21  The shortest sentence for three of the 45 writers in the closed-set. The number of graphs extracted from each sentence is listed.
ACKNOWLEDGMENTS

This work was possible because of the consistent support and guidance provided by my first major professor, Dr. Alicia Carriquiry. She presented me with many opportunities to share my work and grow as a teacher and a researcher. I thank my other major professor, Dr. Danica Ommen, for the time she spent with me working through details of many difficult statistical problems. I appreciate her hands-on approach to advising and her willingness to sit with me, thinking about the fine points and presentation of this research.

I would like to thank the team at the Center for Statistics and Applications in Forensic Evidence. Your support and generosity provided the opportunity to collaborate with researchers from around the world. In particular, I am grateful to Dr. Heike Hofmann for her mentorship and advice.

I am grateful to my committee members: Dr. Jarad Niemi, Dr. Vivekananda Roy, and Dr. Daniel Nordman. They are wonderful teachers and mentors, and have provided valuable feedback on my research. To Dr. Stephen Vardeman, who advised my Masters research, thank you for teaching me to be patient with difficult content. I am thankful for the strong research foundation that you helped me build. I would also like to acknowledge all of the other teachers who have played a role in my education throughout the years, especially Mike Labine and Lisa Conzemius.

This experience was made much more enjoyable by the friends I have made at Iowa State. In particular, I am grateful for Kiegan Rice. Graduate school would have been a very different experience, and certainly less pleasant, without her friendship and support from the very beginning.

To friends and family who supported me from a distance, thank you for your understanding, patience, and encouragement through the challenges and pressure that come with graduate study. I love you all.

Finally, to Nick Berry, my best friend and biggest supporter... thank you.
ABSTRACT

The analysis of handwritten evidence has been used widely in courts in the United States since the 1930s (Osborn, 1946). Traditional evaluations are conducted by trained forensic examiners. More recently, there has been a movement toward objective and probability-based evaluation of evidence, and a variety of governing bodies have made explicit calls for research to support the scientific underpinnings of the field (National Research Council, 2009; President’s Council of Advisors on Science and Technology (US), 2016; National Institutes of Standards and Technology, 2019). This body of work makes contributions to help satisfy those needs for the evaluation of handwritten documents.

We develop a framework to evaluate a questioned writing sample against a finite set of genuine writing samples from known sources. Our approach is fully automated, reducing the opportunity for cognitive biases to enter the analysis pipeline through regular examiner intervention. Our methods are able to handle all writing styles together, and result in estimated probabilities of writership based on parametric modeling. We contribute open-source datasets, code, and algorithms.

A document is prepared for the evaluation process by first being scanned and stored as an image file. The image is processed and the text within is decomposed into a sequence of disjoint graphical structures. The graphs serve as the smallest unit of writing we will consider, and features extracted from them are used as data for modeling. Chapter 2 describes the image processing steps and introduces a distance measure for the graphs. The distance measure is used in a $K$-means clustering algorithm (Forgy, 1965; Lloyd, 1982; Gan and Ng, 2017), which results in a clustering template with 40 exemplar structures. The primary feature we extract from each graph is a cluster assignment. We do so by comparing each graph to the template and making assignments based on the exemplar to which each graph is most similar in structure. The cluster assignment feature is used for a writer identification exercise using a Bayesian hierarchical model on a small set of
27 writers. In Chapter 3 we incorporate new data sources and a larger number of writers in the clustering algorithm to produce an updated template. A mixture component is added to the hierarchical model and we explore the relationship between a writer’s estimated mixing parameter and their writing style. In Chapter 4 we expand the hierarchical model to include other graph-based features, in addition to cluster assignments. We incorporate an angular feature with support on the polar coordinate system into the hierarchical modeling framework using a circular probability density function. The new model is applied and tested in three applications.
CHAPTER 1. INTRODUCTION

This body of work addresses research questions regarding the evaluation of handwriting in the context of forensic science. We contribute to the statistical foundations of the discipline, and provide groundwork for a document analysis pipeline that is open-source, reproducible, and accessible. In particular, we propose a method to make probabilistic statements regarding the writership of a document of questioned source, with respect to a finite set of genuine handwriting samples from known sources.

1.1 A Brief U.S. History and Background of Forensic Handwriting Analysis

Handwriting analysis became necessary almost as early as the handwritten language was developed, because forgery and fraud followed closely behind (Huber and Headrick, 1999). Genuine handwriting samples were not admitted in any Federal or U.S. States court until 1913, and it was not until 1928 that all 48 of the established states admitted handwriting samples that were not already part of other evidence in the case (Osborn, 1946, Ch. XXXI, Ch. XLVI). Although slowly, the disciplines of handwriting and signature examination were gaining traction during this time. Examiners formed a casual group in 1914 to meet and discuss problems in the field, and the American Society of Questioned Document Examiners was formally organized in 1942 (Hilton, 1979).

In their 1999 book, Huber and Headrick note the unique beginnings of forensic handwriting analysis. Unlike other forensic disciplines, it was “born as a forensic necessity,” rather than being established first in its own right and later utilized in a forensic context. Two famous and high profile FBI cases helped put the discipline on solid footing across the country. In 1932 Charles Lindbergh’s baby boy was kidnapped from his bedroom. The Lindbergh family received thirteen ransom notes from the kidnapper, and handwriting experts unanimously agreed that they were all written by the
same individual. Two years after the kidnapping, a suspect was located by tracking bank notes used in the ransom payment. The case against Bruno Richard Hauptmann was largely circumstantial, and a comparison of handwriting samples was key evidence in his indictment (Federal Bureau of Investigation, Famous Cases & Criminals, a). Then, in the 1950s, one month old Peter Weinberger was kidnapped from his front porch in Long Island, New York. Two handwritten ransom notes were the only pieces of useful physical evidence in the case. FBI handwriting examiners, aided by a team of special agents, combed through almost two million handwriting samples from the Motor Vehicle Bureau, probation offices, schools, and other public paperwork. A similar handwriting sample was discovered in the probation file of Angelo LaMarca, who was located and later confessed to the kidnapping. This case resulted in legislation that reduced the FBI's waiting period for involvement on kidnapping cases from 7 days to 24 hours (Federal Bureau of Investigation, Famous Cases & Criminals, b).

Traditional document examination is conducted by trained Questioned, or Forensic Document Examiners (QDEs or FDEs). These professionals follow a set of steps and guidelines to conduct an investigation of the class and individual characteristics of handwriting. Students usually learn to write using a copy book and as handwriting becomes a natural biomechanical process, individualizing characteristics begin to emerge. Writing characteristics are generally classified as either class characteristics – foundations learned from some institutional or systematic instruction, or individual characteristics – a unique set of tendencies for a particular writer (Huber and Headrick, 1999; Srihari et al., 2002). As computers and internet have become widely available, penmanship taught as a skill necessary to conduct business in the world has become less of a priority (Huber and Headrick, 1999; Expert Working Group for Human Factors in Handwriting Examination, 2020). Still, as we change and adapt, traditional handwriting examination is grounded in the notion that given sufficient quantity and quality of writing, no two writers have the same set of individualizing characteristics (Osborn, 1910; Srihari et al., 2002).
1.2 Need for Statistical Foundations in the Discipline

Most document examiners who testify to the source of a questioned writing sample are compelled to subscribe to independent proficiency testing programs that evaluate their skills. However, on top of many other limiting challenges in handwriting examination work, there is no way around the “reality that the instrument performing the examination and comparison of handwriting in the forensic environment remains the [examiner’s] brain” (Bird and Found, 2016). On cognitive bias, the Expert Working Group for Human Factors 2020 report states that “[a]s long as a human is the main instrument of analysis and interpretation in forensic impression and pattern evidence disciplines, the strengths and limitations of human cognition will be central to forensic casework.” We might never be able to fully express how the human brain aggregates the information it takes in, and from a data science perspective, we consider each examiner as a unique black box with unique perspectives and biases that have been developed over years of formative experiences. See Dror et al. (2011), Dror and Pierce (2019), or Kassin et al. (2013) for further discussion on cognitive and confirmation biases, their impact in forensic sciences, and standards developed to address the issue. Statistical and computer based methods that are developed to “distribut[e] cognition between technology and humans”, if implemented properly, can produce efficient, effective, and quantitative results that mitigate the opportunity for human bias to present itself (Dror and Mnookin, 2010).

A related topic is the matter of error rate estimation. A 2016 report by the President’s Council of Advisors on Science and Technology (PCAST) outlines the need for the quantification of error rates and development of statistical methodologies for objective analyses and feature comparison methods in forensic sciences. It is a difficult task to measure error rates for real case work, and the term “error” is often misunderstood. It is sometimes taken to mean examiner “mistake”, instrumental measurement error, or statistical error. Christensen et al. (2014) conduct a discussion of the concept of error in forensic sciences, citing the 2002 People v Gomez case in which a fingerprint examiner testified to a zero error rate, stating: “And we profess as fingerprint examiners that the rate of error is zero. And the reason we make that bold statement is because we know based on 100 years of research that everybody’s fingerprints are unique, and in nature it is never going to
While the subjective analyses conducted by examiners are generally accurate, a more objective approach would allow for estimation of error. Statistical study design and modeling techniques play a large role in appropriate estimation of error in the forensic sciences.

The 2009 National Research Council report titled “Strengthening Forensic Science in the United States: A Path Forward” was perhaps one of the largest sparks that helped ignite a movement in the forensic science community toward research for more objective and systematic evaluations of evidence. Authors of the report acknowledge areas of forensic science based in chemistry and other well established scientific disciplines as having a solid foundation, and call out the need for stronger scientific foundations for the evaluation of pattern and impression evidence, which encompasses handwriting examination. A recommendation common to several disciplines was the move toward automated systems (National Research Council, 2009). Years later, the 2016 President’s Council of Advisers on Science and Technology (PCAST) report again emphasized shortcomings in feature comparison methods, and called for studies of subjective evidence evaluation techniques to strengthen the scientific foundations of forensic science. The report also recommended development and testing of statistical algorithms for comparing the similarity of features for a variety of evidence types (President’s Council of Advisers on Science and Technology (US), 2016).

The research community has responded, and in an effort to carry out recommendations from these two landmark reports, the Organization of Scientific Area Committees for Forensic Science (OSAC) administered by the National Institute of Standards and Technology (NIST), and other government entities have identified current gaps in knowledge, and published calls for research to close those gaps. In addition, NIST established a Forensic Science Center of Excellence in May 2015. The Center for Statistics and Applications in Forensic Evidence (CSAFE) is led by Iowa State University and “supports NIST’s efforts to advance the utility of probabilistic methods to enhance forensic analysis”. Statisticians at the Center work with the legal and forensic science community to bridge the gap between statistically founded conclusions and presentation of evidence


interpretation. The research conducted at the Center in pattern and digital evidence is open-source and reproducible. The Center’s ongoing project related to the statistical evaluation of handwritten evidence is part of this dissertation.

1.3 Other Systematic and Statistical Approaches

With the shift toward reproducible and systematic evaluation of pattern evidence, a few computer-based systems have been developed to support FDEs in their task of comparing a questioned document to known writing samples. We discuss three widely known systems in the U.S. FDE community here. The first is CEDAR-FOX (Srihari et al., 2007), developed at the University of Buffalo. It is a system that uses a parametric model to evaluate the similarity of samples and maps results to the now withdrawn ASTM E1658-08 standard “Standard Terminology for Expressing Conclusions of Forensic Document Examiners.” The system provides a measure of confidence that two samples are from the same writer, and is interactive to the extent that “user[s] can select which of the macro-features are used” and “which of the characters in the two documents are to be used in making the comparison” (Srihari et al., 2007). The WANDA workbench (Franke et al., 2004) is also an interactive tool that allows for manual annotations and interactive selection of automated measurements. The system combines features via a rank combination scheme. The user can compare a questioned document to a set of known samples, and a ranked “hit list” is provided (Franke et al., 2004). The final system we mention here is the FLASH ID® system (Sciometrics LLC, Chantilly, VA, USA). Similar to the WANDA system, FLASH ID® evaluates the similarity of a questioned document against a closed set of writers and produces a ranked list using a patented algorithm (Miller et al., 2017). The system uses an extensive set of measurements that are extracted from each component of writing. Miller et al. (2017) demonstrate the variety and value of measurements that “are substantially similar to those used in the proprietary product.”

There are a wide variety of approaches for the statistical evaluation of handwritten evidence that have not been encapsulated in an automated system like those mentioned above. Saunders et al. (2011) construct three classifiers using a categorical data vector based on manually seg-
mented writing components corresponding to characters (Miller et al., 2017). The three classifiers developed by Saunders et al. (2011) (Kullback–Leibler distance, Chi-squared distance, and plug-in Naive Bayes) are used to predict the writer of a document within a closed-set. The primary feature extraction technique we use (defined in Chapter 2) is based on the outcomes of a clustering algorithm, and also produces a vector of categorical information for each document. As such, our fully Bayesian approach to closed-set writer identification may be considered similar to the plug-in Naive Bayes method in Saunders et al. (2011). Hepler et al. (2012) explore the use of Score-based approaches for computing likelihood ratios for handwriting evidence. The score-based approaches are distinct from our research because they rely on modeling the score resulting from a comparison of two documents, whereas we directly model features extracted from a single document. Bulacu and Schomaker (2006) discuss the fusion of what they define as texture-level and allograph-level features for writer verification. In similar spirit, our approach uses “document-level” features and “graph-level” features. There are many more left unmentioned, in particular, methods that relate to character recognition or topic modeling have been left out of all discussions.

Srihari et al. (2002) define writer identification as “the task of determining the writer when there are \( n \) candidates.” That is to say, given that the true writer of a questioned sample is contained in a closed set of known writers of size \( n \), the writer identification task entails evaluation of which known writer is the true writer of the questioned sample. Nearly every computer based method we have mentioned in this section is aimed at providing a solution to the writer identification task. The framework we propose in this work is no different. A notable exception is the score based likelihood ratio work of Hepler et al. (2012), in which the similarity of a pair of documents is evaluated. This approach comes with the difficult task of specifying a relevant writer population.

1.4 Outline of Methods and Results

This research, supported by CSafe, began in order to help satisfy the need for statistical and scientific foundations in the field (motivated in Section 1.2). It is a primary mission of the Center to provide open-source datasets, code, and algorithms in the interest of advancing research in the field.
of forensic statistics. Like the FLASH ID® system (Sciometrics LLC, Chantilly, VA, USA), we set out to develop an automated pipeline to evaluate a questioned writing sample against a closed set. This type of approach, while susceptible to cognitive biases of the code developers and statistical bias introduced through methods, removes direct and regular intervention by the examiner.

Our methods are applicable on all writing styles (cursive, print, combination) together, with no need to distinguish between them. The result of our analyses is a probabilistic evaluation of the writer identification task. In contrast to a nonparametric, rank-based “hit list”, we output probability estimates of writership based on parametric modeling of features.

In addition to our methodological research, an experiment to collect handwritten samples was designed and executed. The dataset contains writing samples from 90 participants. Each participant contributed writing at three separate data collection sessions that took place at least three weeks apart. Three prompts were each transcribed three times at each session, resulting in 27 samples written by each participant, and 2430 samples in total. Demographic and session specific information were collected as well. See the Crawford et al. (2019a) article for details regarding data collection procedures, and (Crawford et al., 2019b) for the published dataset that is available for download.

The long-term goal of this project is to create a tool to provide output with a direct probabilisitc interpretation that FDEs can use in their day to day work and, potentially, to support their testimony in court. We have presented our work to the FDE community regularly, and have elicited feedback throughout key developmental stages. The methods have been well received and have gained interest for use cases in which the search of a closed-set is reasonable. An example would be the evaluation of a bomb threat discovered at a school against writing samples from students to obtain probabilities of writership for every student in attendance on the day the threat was written.

In Chapter 2, we introduce the R package handwriter that serves as our document processing tool (Berry, 2019). Using this R package, an image of scanned handwriting is first binarized so that pixels in the image are coded as black or white, isolating the writing from background noise. Then,
the writing is reduced to a single pixel-wide skeleton and the body of writing is decomposed into a series of smaller disjoint skeleton pieces. These smaller pieces are treated as graphical structures with nodes and edges. A novel distance measure is defined to assess the structural similarity between graphs and is used in a $K$−means algorithm (Forgy, 1965; Lloyd, 1982; Gan and Ng, 2017) to dynamically group graphs into 40 clusters. An exemplar from each cluster is calculated, and together make up a template with which we can extract cluster assignments from new documents. Cluster assignment rates for a document provide a 40−dimensional feature vector that serve as data for a simple Bayesian hierarchical model. Chapter 2 concludes with a writer identification analysis on a small set of 27 writers from the Computer Vision Lab (CVL) database (Kleber et al., 2013).

Chapter 3 presents several improvements upon the clustering template algorithm and the simple Bayesian model to increase the accuracy of the writer identification task. An updated clustering template is developed using a larger variety of data sources and a larger number of documents. The cluster assignment feature vector is extracted from documents in the CSAFE database using the new template. We introduce a mixture component into the simple hierarchical model defined in Chapter 2. The estimated mixing parameter for each writer is an indicator of writing style. We explore tendencies of the mixing parameter estimates and conduct an analysis of misplaced probability as it relates to writing style.

The goal of Chapter 4 is to expand the hierarchical model and increase its accuracy. The chapter is presented in two parts. The first explores a number of ways to measure the slant as a feature of a graph. We consider each for inclusion in the hierarchical model and assess predictive performance. At the end of Part I, we select an angular feature with support on the polar coordinate system to be included in the model along with the original cluster assignment feature vector. A circular probability density function is used as the data distribution for the feature. Since graphs with the same cluster assignment will tend to have similar structures, we make comparisons of the angular feature only for graphs within the same cluster assignment. The addition of such a feature aids in discriminating between writers based on the structural tendencies that emerge in their writing.
Part I is concluded with a prior sensitivity analysis. In Part II of the chapter, the new model is applied and tested in three applications. Each application is chosen to stress the model in a particular way so that we can observe the behavior of results in scenarios that are similar to real case work.

1.5 References


President’s Council of Advisors on Science and Technology (US) (2016). *Report to the President, Forensic Science in Criminal Courts: Ensuring Scientific Validity of Feature-comparison Methods*. Executive Office of the President of the United States, President’s Council of Advisors on Science and Technology.


CHAPTER 2. K-MEANS CLUSTERING AS A FEATURE EXTRACTION TECHNIQUE FOR HANDWRITTEN DOCUMENTS

Amy Crawford, Nicholas Berry, and Alicia Carriquiry

2.1 Abstract

We present a method for processing scanned handwritten documents to decompose the writing into small graphical structures, often corresponding to letters. We introduce a measure of distance between two such structures that is inspired by the graph edit distance, and a measure of center for a collection of the graphs. These measurements are the basis for an outlier tolerant $K$-Means algorithm to cluster the graphs based on structural attributes, creating a template for sorting new documents. The template itself serves as a feature extraction tool for handwritten documents. We establish a Bayesian hierarchical model to capture the propensity a writer has for producing graphs that are assigned to certain clusters. We then perform a closed-set writer identification analysis on documents from 27 writers in the Computer Vision Lab dataset. We show results of the identification task under the cluster assignments and compare to the same modeling, but with a less flexible grouping method that is not tolerant of incidental strokes or outliers.

---

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement #70NANB15H176 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, University of California Irvine, and University of Virginia.
2.2 Introduction

Many fields rely on the ability to parse and process handwritten text, and in recent years there has been a shift towards the automation of this processing. At its heart, automatic handwriting processing is the task of converting an image of handwriting into usable data. For us, this data serves as the foundation for an analysis of the handwriting. When done algorithmically, handwriting analysis usually falls into one of two categories. One common analysis objective is to recognize the characters written on a page. Another is a determination of who wrote the sample, which is called writer identification. For writer identification, the goal does not concern investigating what is written, but the way in which it is written.

Within the writer identification framework, it is important to make the distinction between authorship and writership. Authorship analyses often use word choice or punctuation in a document (for example Rosen-Zvi et al. (2004) or Seroussi et al. (2011)) and can involve a mix of the two objectives described above. Writership identification is limited to an analysis of the shapes that a writer emits via their practiced writing style.

One large and active area of writership identification is in forensic statistics. Such analyses can be used to determine the source of a piece of handwritten evidence, e.g. a bank robbery note or a bomb threat. Traditionally, handwriting analysis of this nature is done by trained forensic practitioners. In practice, an examiner comes to a conclusion on a decision scale such as the 9-point scale of ASTM Standard E1658-08 (2008) where terms like “identification,” “probable,” and “elimination,” are used when making a comparison between a document from an unknown writer and one of a known writer. The ASTM standard has since been withdrawn, but similar best practices are in use.

The work presented in this paper is concerned with automating and quantifying portions of the forensic handwriting examination process. In particular, we seek to parse a handwritten document into data, filter those data into clusters with similar characteristics, and explore how a writer’s propensity for creating forms that fall into these clusters can be used in a statistical model.
There is existing software for handwriting analysis of this nature. The proprietary product FLASH ID® (Sciometrics LLC, Chantilly, VA, USA) searches a database for closest writing matches and gives results based on a scoring system. The WANDA workbench (Franke et al., 2004) and CEDAR-FOX (Srihari et al., 2007) systems have tools to facilitate both automated and interactive document examination, as well as database management frameworks.

Our work does not rely on any character recognition techniques. Other work of this nature has been done. For example, in their 2007 paper Bulacu and Schomaker use graphemes that are normalized to a $30 \times 30$ pixel image and overlaid to extract pixel shade differences. A standard clustering algorithm yields a code book, and grapheme distribution across that code book is one of six features used to identify writers in downstream writer verification analysis. Miller et al. (2017), whose methods are “substantially similar to those used in the proprietary product FLASH ID®”, group graphemes in a deterministic fashion, and measurement comparisons are made within those groups.

In this paper, we address grouping graphical structures through a $K$-means clustering algorithm that relies on a distance measure designed specifically for handwritten structures. We develop our own deterministic grouping method, similar to that of Miller et al. (2017), for comparison. Examples of group separation using both deterministic and our proposed dynamic groupings are given in Section 2.2.1.

Our document analysis pipeline begins by processing scanned handwritten documents, and segmenting the writing into small graphical structures which we call graphs. Graphs are the smallest units of writing that we consider, and they often, but not always, correspond to Roman characters. We cluster graphs using measures based on the similarities of their major physical attributes. These groupings are more descriptive and repeatable for writers than deterministic groupings, because of their robustness to small structural differences. This dynamic grouping better templates a writer, and when used as data for a Bayesian hierarchical model, improves writer identification results over those of deterministic groupings.
2.2.1 A Motivating Example

Consider two writers. Suppose that writer A favors formal cursive, so they tend to make loops when forming characters such as ‘l’ and ‘f’. Conversely, writer B uses a more broken writing style and tends to make a single stroke ‘l’ and loop-less segments in an ‘f’.

Ideally, when writer A’s documents are processed their graphs would be assigned to groups that are characterized by a higher rate of cursive style loops. When writer B’s graphs are considered, their distribution over the groups should differ from that of writer A, since writer B will have fewer graphs that are characterized by loops, and more that are assigned to groups embodying simple stroke graphs.

Figure 2.1 shows three example ‘f’s from two writers. The characters in Figures 2.1(b) and 2.1(c) are simple block ‘f’s (like writer B), while Figure 2.1(a) shows a cursive style ‘f’ with a loop near the top (like writer A). Deterministic and clustering based groupings of these three ‘f’s are provided in their captions as D and C, respectively. Notice that when compared to Figure 2.1(c), 2.1(b) is missing an appendage. This results in different deterministic group assignments for the two letters despite other clear structural similarities. This is unfortunate for an algorithm attempting to differentiate writer A from B based on the grouping.

Under our clustering method, both Figures 2.1(b) and 2.1(c) are assigned to the same cluster (C=2). Their overall structure is dominating and the small incidental edge that is missing from 2.1(b) does not force a separation between groups. This results in a partitioning of the ‘f’s that
can be used to help identify each character’s writer. Of course, this is a pathological example with only one character type. In a real example there is a variety of shapes to group and the writer identification is based on an aggregation of small gains across many groups, rather than the hard division described here.

### 2.2.2 Paper Organization

Section 2.3 discusses the document processing pipeline for taking a handwritten document from a scanned image to usable data (graphs). A deterministic grouping method is presented as a direct result of the graphical structures that arise from processing. We introduce our clustering algorithm for graphs in Section 2.4. To address our motivating problem of writer identification, the method is applied to writers from the Computer Vision Lab (CVL) handwriting dataset (Kleber et al., 2013) in Section 4. We compare writership analysis results for the dynamic cluster groupings to results using the same model and writers, but based on deterministic groupings.

### 2.3 Document Processing

The pipeline for document pre-processing and data extraction begins with a scanned handwritten document and results in a set of graphs. Graphs are small pieces of connected ink that serve as individual observations for the analyses that follow. Processing is done using the R package handwriter, available at [https://github.com/CSAFE-ISU/handwriter](https://github.com/CSAFE-ISU/handwriter). This R package, written by Nick Berry for the Center for Statistics and Applications in Forensic Evidence (CSAFE) at Iowa State University, provides a toolkit for handwritten document processing.

#### 2.3.1 Pre-Processing

In the handwriter pipeline, pre-processing begins with binarization to convert an image to pure black and white. Colored images are turned to grayscale using the linear combination $0.2126R + 0.7152G + 0.0722B$ on each pixel. Otsu’s binarization method (Otsu, 1979) is applied to the grayscale
images to assign each pixel to one of two groups in a fashion that maximizes between-class variance. The groups are appropriately normalized to black and white.

Cleaning is done by handwriter using procedures from Stentiford and Mortimer (1983). A set of masks are implemented that isolate and correct spurious pixels, fill white holes likely caused by mistakes in binarization, and widen holes that were likely intentionally made during writing. On the clean binary image we use the Zhang-Suen thinning algorithm (Zhang and Suen, 1984), to reduce the writing to a one pixel wide skeleton structure that maintains the shape and connections of the original. By choosing to work with the writing skeleton we sacrifice all information about the width of lines. However, thinning facilitates our ability to identify structural components in the handwritten images.

Figure 2.2(a) gives an example of a binary image displayed in gray with its thinned, pixel wide counterpart overlaid in black. The overall shape of the written text remains intact and key structural features (like terminal pixels, intersections, and the pixel wide paths connecting them) are easier to detect. Section 2.3.2 expands on the idea of using simple structural elements of the skeleton to extract requisite information from the documents.

![Figure 2.2](image)

**Figure 2.2** The word “csafe” taken from a processed document. (a) Binarized pen strokes in grey, pixel wide skeleton overlaid in black. Red dots denote graph breakpoints of the connected writing. (b)-(g) Each graph shown separately where red dots show nodes (endpoints and intersections).
2.3.2 Data Extraction

After a document has gone through pre-processing, the handwriter package decomposes the skeletonized writing into graphs. Consider the \(i\)th graph of a document as an attributed graph \(G_i\) with a set of vertices \(V_i\) made up of the terminal pixels and locations of intersecting lines of the graph. The set of edges \(E_i\) represents paths in graph \(i\) that connect elements of \(V_i\). For the graph representing an “x”, shown in Figure 2.3, the red dots make up the set \(V_i\), and each of the four paths (individualized by numbers) are elements of the set \(E_i\). These graph structures, similar to those in (Saunders et al., 2011; Miller et al., 2017; Walch and Gantz, 2004; Gantz et al., 2005), will be the basis for evaluating the structural styles of a writer.

We start by breaking the writing down into a set of connected pieces of ink. In cursive writing this generally corresponds to a connected word, and in disjoint print, a single character. Individual connected ink blots are then candidates to be decomposed further into graphs.

handwriter makes the decision on which ink blots to break by looking through each edge in an ink blot and breaking those that satisfy a set of rules designed to estimate the intended character breaks within that ink blot.

![Figure 2.3 The letter ‘X’ processed by handwriter. Numbers show different paths. Red dots (if in color) indicate nodes. Note that the middle node is actually a merging of two close by nodes.](image)

When breaking on an edge, if that edge has a clear trough (lowest point), then it is broken at the base of that trough, and if not then it is broken at its midpoint. Breakpoints of the word “c safe” in Figure 2.2(a) are shown in red.

After breaking, the document has been fully decomposed into a disjoint set of graphs \(\{G_i\}\), each with their respective vertices and edges. Finally, the vertices in each graph are systematically
ordered so that we can compare vertices across graphs in sections that follow. With these ordered vertices and edges between them, an adjacency matrix for each graph is constructed.

It is worth mentioning that we only use handwriter for document processing, but it also has feature extraction capabilities such as finding centroids, slants, and other measurables for each graph. These features are undeniably important for forensic handwriting analysis, but will not be used for the clustering that is of focus here, and are thus omitted.

2.3.3 Deterministic Adjacency Grouping

We previously suggested that by establishing a grouping system for graphs, we could assess the rate at which a writer produces graphs of each group, and use those rates to characterize a writer’s style. The first grouping method, which we will use for baseline comparison with our clustering method, uses only the edge connectivity of a graph. Graphs with identical adjacency matrices are placed together in a group. We call this the adjacency grouping method, and it is readily available after processing by handwriter. This deterministic grouping method is sensitive to small changes in graph structure, because small incidental pen strokes change the adjacency matrix of a graph, and thus the adjacency group assignment. Since this method is so sensitive to small differences between the graphs, the number of resulting groups is very large. Walch and Gantz (2004); Gantz et al. (2005); Saunders et al. (2011); Miller et al. (2017) utilize, in part, a similar method, which they call the “isocode”.

We analyze 162 documents from the CVL database, and handwriter partitions these documents into a total of 52,451 graphs. The number of resulting unique adjacency groups is 1,636. The two most common adjacency groups, shown in Figure 2.4 with graph examples, account for approximately 60% of all graphs in the CVL documents.

Advantages of this grouping method come from the strict structural similarity imposed by the identical adjacency matrices between the members of a group. Within a group, each graph has the same number of edges and vertices, allowing for one-to-one comparisons between structural components of the graphs. Gantz et al. (2005) leverage this advantage. On the other hand, the
required strict similarity means that graphs with minor differences are not grouped together, are never compared, and potentially leave valuable information unused.

![Diagram](image)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2.4 2.4(a) shows an example of the most common graph structure. 2.4(b) shows an example of the next most common. (c) and (d) show the corresponding adjacency matrices.

2.4 Clustering

Rather than assign groups through deterministic means, we propose a dynamic and flexible $K$-means based grouping method that is tolerant to incidental pen strokes. This allows graphs with similar, but slightly different, graphical structures to be captured in the same group.

All $K$-means type clustering algorithms hinge on the ability to calculate two essential quantities: a distance (or discrepancy, or similarity) measure and a measure of center. Neither of these measures are readily or easily defined for graphs. Section 2.4.1 summarizes the notion of an edit distance used in the general graph framework. Then, for graphs, which are a type of attributed graph, we develop a novel distance measure and a mean calculation. Finally, we outline an accompanying clustering algorithm yielding a dynamic graph grouping methodology.
2.4.1 Edit Distances

An existing graph discrepancy measure, which helps to motivate ours, is the error-correcting graph matching of Messmer and Bunke (1998). Their distance measure for two graphs is the cost associated with a sequence of steps necessary to transition from one graph to the other, called the edit distance. This graph edit distance is related to the simpler concept used to quantify the difference in two strings (Levenshtein, 1966). To obtain the string edit distance, operations are sequentially applied to a string $S_1$ until it matches another string $S_2$. The available edit operations are Change, Insert, and Delete. The resulting distance calculation can be posed and solved as a dynamic programming problem (Wagner and Fischer, 1974).

In the graph edit distance calculation, like with strings, the goal is to sequentially apply edit operations to transition one graph to another. The necessary operations for graph editing are conceptually the same as in the string context, except they can be applied to both edges and vertices. Each edit operation is assigned a cost, and the difference between two graphs is the sum of the minimal cost edit sequence between them.

Graphs that are compared using this method have labelled vertices, but generally do not have edge attributes such as lengths or curvatures, or vertex attributes like locations. The graphs that describe handwritten graphs do have these attributes, which will be leveraged in their distance calculations. In the following section, we exploit edge attributes to develop a distance measure for two graphs and a mean calculation for a set of graphs. These measures will emulate aspects of the graph edit distance measure, with costs corresponding to the magnitude of the changes necessary to transition one graph into the other.

2.4.2 Distance Measure for Graphs

There are distinct differences in how distances are approached for graphs, rather than general graphs. First, only edges of the graphs are taken into account. There is no cost directly associated with vertex edits. Every edge has two vertices, so differences in the vertices of the graphs can be
reflected by instead altering edges. Second, the attributes of the edges will be used to determine the cost associated with each change.

Each graph is completely described by its collection of edges, so the task of calculating graph distances can be simplified to calculating edge distances. We present a distance measure for two edges, then provide the mechanism for computing full graph distances by combining edge distances.

### 2.4.2.1 Distance between Two Edges

To develop the edge distance measure we keep in mind that the distance between graphs should be, at least in part, characterized by their structures, while allowing graphs with similar but not identical structures to be grouped together.

For two edges, the edge distance calculation is comprised of three component distances, $d_{loc}$, $d_{sl}$, and $d_{sh}$, capturing the difference in **endpoint locations**, difference in the **straight-line distance** between edge endpoints, and a rough estimation of the difference in edge **shapes** respectively. Each of these components will be addressed in turn in Section 2.4.2.2.

### 2.4.2.2 Edge Distance Calculation

To allow for meaningful location comparisons of edges, we first align graphs on the 2-dimensional coordinate system with their centers of gravity overlapping on the origin. Figures 2.5(a) and 2.5(b) give single edge graphs, each with their centroids anchored at the origin.

Note that the starting and ending points of edges are labeled arbitrarily, so the edge comparison will need to be considered in both traversal directions. When end point ordering is relevant for the three component calculations, a plus sign subscript, $d_+$, indicates a distance component taken in the original assignment order of the paths, and a minus sign subscript, $d_-$, indicates that the order of the second path is considered in reverse. Choosing the correct order of comparison will be addressed in Equation 2.3 after all three of the edge distance components have been established. Only the positive traversal comparisons are pictured in Figure 2.5, as that turns out to be the proper comparison direction for these two edges.
We begin by defining the endpoint location component of the edge distance, $d_{loc}$, which only requires the pixel positions of the end points of the paths. Denote the position of the first terminal pixel, the starting point of an edge, $e$, as $p_1^e$. The second terminal pixel position, the end point of the edge, is $p_2^e$. These locations are recorded as $(x, y)$ points on the coordinate plane. The points connected by red lines in Figure 2.5(c) represent the edge end points.
Let $\| \cdot \|_2$ denote the Euclidean norm. Then, the location component is defined as, $d_{loc}(e_1, e_2) = \{d_{loc+}(e_1, e_2), d_{loc-}(e_1, e_2)\}$, where

$$d_{loc+}(e_1, e_2) = \min \left\{ \|p_{e1}^e - p_{e2}^e\|_2, \|p_{e1}^e - p_{e2}^e\|_2 \right\},$$

$$d_{loc-}(e_1, e_2) = \min \left\{ \|p_{e1}^e - p_{e2}^e\|_2, \|p_{e2}^e - p_{e1}^e\|_2 \right\}.$$ 

In this formula, any two paths that either begin or end at the same pixel location will have a $d_{loc} = 0$ regardless of their shapes, angles, or lengths. Figure 2.5(d) shows $\|p_{e1}^e - p_{e2}^e\|_2 = 1.4$ and $\|p_{e2}^e - p_{e1}^e\|_2 = 14.3$, resulting in $d_{loc+}(e_1, e_2) = 1.4$.

The second and third components will make use of the straight line that connects $p_{e1}$ and $p_{e2}$ in an edge, $e$, which we denote as $\ell^e$. The red line segments in Figure 2.5(c) depict $\ell^{e_1}$ and $\ell^{e_2}$ for the two example single edge graphs.

The second distance component concerns the difference in the lengths of $\ell^{e_1}$ and $\ell^{e_2}$. This component adds a penalty reflecting the difference in the straight line displacement of the endpoints in each edge. Define this straight line distance component $d_{sl}(e_1, e_2)$ as

$$d_{sl}(e_1, e_2) = \left| \|\ell^{e_1}\|_2 - \|\ell^{e_2}\|_2 \right|.$$ 

In this case there is no need to make this comparison in both directions since the specification of the straight line between end points for an edge, $\ell^e$ is unchanged by the ordering of the terminal pixel locations $d_{e1}^e$ and $d_{e2}^e$. The diagonal segment in Figure 2.5(e) depicts this component for our example edges. The solid portion has the same length as the shorter segment, and the dotted portion, with a length of 9.9, represents the difference in the distances of $\ell^{e_1}$ and $\ell^{e_2}$.

For the third component of edge distance, we define a set of points that capture the shape of an edge. Denote the seven points on the edge $e$ that cut it into eight equal length pieces as $q_{1}^e, q_{2}^e, q_{3}^e, \ldots, q_{7}^e$. Similarly, place seven equally spaced points on the straight line that connects the edge endpoints and call them $q_{1}^\ell, q_{2}^\ell, q_{3}^\ell, \ldots, q_{7}^\ell$. Then, subtract each of the line points from each of the corresponding edge points to obtain seven calculated shape points that are a rough representation of the shape of edge $e$. These shape points are $s_{i}^e = q_{i}^e - q_{i}^\ell, \ldots, s_{7}^e = q_{7}^e - q_{7}^\ell$. 


The shape points capture the direction and magnitude of the edge’s deviation from the straight line $\ell^e$. If an edge is relatively straight (think the number one), we do not expect to see much deviation from $\ell^e$. If an edge has more curvature (think the number five) we expect more deviation of the edge from the straight line.

We take $d_{sh}$ as the average difference in the corresponding shape points of $e_1$ and $e_2$. Here, we certainly need to consider both the forward ($d_{sh+}$) and backward ($d_{sh-}$) orderings of the shape points. The shape contribution to the edge distance measure $d_{sh}(e_1, e_2) = \{d_{sh+}(e_1, e_2), d_{sh-}(e_1, e_2)\}$ is

\[
d_{sh+}(e_1, e_2) = \frac{1}{7} \left[ ||s_1^{e_1} - s_1^{e_2}||_2 + \cdots + ||s_7^{e_1} - s_7^{e_2}||_2 \right]
\]

or

\[
d_{sh-}(e_1, e_2) = \frac{1}{7} \left[ ||s_1^{e_1} - s_7^{e_2}||_2 + \cdots + ||s_7^{e_1} - s_1^{e_2}||_2 \right].
\]

In 2.5(f), vectors are drawn from the line points to the edge points, depicting the formulation of the shape points $s_1$ through $s_7$ for each edge. Figure 2.5(g) shows the same vectors, pointing at each of the seven shape points for each edge. The distance between shape points in each panel of Figure 2.5(g) are taken (shown next to the red lines), and the average of those 7 distances make up the shape component of the edge distance. Here,

\[
d_{sh+} = \frac{1}{7} (1.5 + 2.1 + 4.8 + 12.5 + 16.1 + 14.2 + 7.6) = 8.4
\]

Now, with all three edge comparison components ($d_{loc}$, $d_{sld}$, and $d_{sh}$) in-hand, the final step is to combine them. We must take care to scale the edge distances so edges that make up a large proportion of their graphs are weighted more heavily. This gives way to an edge distance calculation that is robust to differences in small edges, but demands that large edges be similar. This is a favorable property since larger edges generally dominate the structure of a graph, and to find similar graphs, we seek similar dominating structures. The small edge distance down weighting also provides tolerance by preventing small edges from drastically influencing the distance measure. The amount that an edge distance is down weighted is based on the average proportion of the two graphs that the edges $e_1$ and $e_2$ make up.
The distance measure between two edges $d(e_1, e_2)$ is the final culmination of the three comparison components and weighting results. It is given as

$$d(e_1, e_2) = \frac{1}{2} \left[ \frac{|e_1|}{\sum_{i=1}^{\vert E_1 \vert} |e^E_i|} + \frac{|e_2|}{\sum_{i=1}^{\vert E_2 \vert} |e^E_i|} \right] \times \min\{d_{loc} + d_{sl} + d_{sh}, d_{loc} - d_{sl} + d_{sh} \}, \quad (2.3)$$

where we let $e^E_i$ denote the $i^{th}$ edge in $E_j$, the set of edges comprising graph $j$. The leading fraction in $(2.3)$ is the weighting component, where $|e_i|$ denotes the length of path $i$ and $|E_j|$ is the total number of edges in graph $j$. The minimum statement of Equation 2.3 is the mechanism for choosing the direction of edge comparison that yields the smallest distance measure.

The edge distance calculation relates to fundamentals of the graph edit distance. The endpoint location component $d_{loc}$ emulates an edit operation that shifts an edge in space. The cost of this edit is equivalent to the distance that an edge must travel to align itself with its counterpart’s nearest endpoint. The straight-line distance component $d_{sl}$ measures the difference in displacement of the edge endpoints. This is comparable to stretching or compressing edges as an edit operation. The final component $d_{sh}$ represents the amount that two paths have to be straightened, curved, or twisted in order to match each other.

### 2.4.2.3 Graph Distance Measure from Edge Calculations

In Figure 2.5 the edge distance is equal to the graph distance since each graph has only one edge. For more complicated graphs, combining edge distances to form a complete graph distance happens in two phases. First, we handle potential differences in edge counts between two graphs. Then, the optimal matching between the edge sets is chosen via evaluation of all available edge distance comparisons. The edge distances for the resulting matches are summed to form the final graph distance measure.

To equalize the number of edges between two graphs, dummy edges are added to the graph with smaller $|E|$ until the number of edges is equal. These dummy paths do not have any physical structure, and thus cannot be compared with real edges in the usual way (using endpoints, shape
points, etc.). Instead, the distance between a real path and a dummy path is assigned solely based on the length of the real path. For a real edge \( e_r \) and a dummy edge \( e_d \), we define the distance as

\[
d(e_r, e_d) = \frac{1}{2} \left[ \frac{|e_r|}{\sum_{i=1}^{\left|\mathcal{E}_r\right|} |e_i|} + \frac{0}{\sum_{i=1}^{\left|\mathcal{E}_d\right|} |e_i|} \right] |e_r|^2
\]

\[
= \frac{1}{2} \left[ \frac{|e_r|^3}{\sum_{i=1}^{\left|\mathcal{E}_r\right|} |e_i|} \right].
\] (2.4)

This calculation uses the same weighting factor as in Equation 2.3 with \( |e_2| = 0 \) and the distance component set to the squared length of the real path, \( e_r \).

Including the dummy edge comparisons, we make \( \max(|\mathcal{E}_1|, |\mathcal{E}_2|) \) unique pairwise edge distance calculations for \( G_1 \) and \( G_2 \). We match the edges in \( G_1 \) with edges in \( G_2 \) in a one-to-one fashion such that the total edge distance is as small as possible. This task is a constrained minimization optimization problem, which is solved via linear programming.

We now give a formal expression for the complete graph distance measure. Define \( I^* = \left\{ i_1^*, \ldots, i_{\max(|\mathcal{E}_1|, |\mathcal{E}_2|)}^* \right\} \) to be the set of indices that reorder the edges of \( G_2 \) to reflect the optimal minimum distance matching with respect to the original ordering of \( G_1 \). The distance between the two graphs is

\[
D(G_1, G_2) = \sum_{i=1}^{\max(|\mathcal{E}_1|, |\mathcal{E}_2|)} d(e_i^{\mathcal{E}_1}, e_{i^*}^{\mathcal{E}_2}).
\] (2.5)

### 2.4.3 Weighted Mean of Graphs

We begin by defining a weighted mean of two graphs. Once this procedure has been established, it can be applied in sequence to produce the mean of a larger set of graphs by iteratively taking weighted means with decreasing weight on each newly introduced graph. Like the graph distance measure relied on combining individual edge distances, the graph mean calculation will rely on combining individually calculated edge means. This procedure requires that the edges of two graphs have been matched such that they provide a minimum graph distance calculation as described in Section 2.4.2.3.

With the exception of dummy edges, an edge is characterized by just its end points, shape points, and length, so the weighted mean of two edges is constructed as the weighted mean of each
of these components. The heavier the weight on the first letter, the closer all of the mean edge components will be to that letters end points, shape points, and length.

Figures 2.6(a)-2.6(f) demonstrate a weighted mean calculation in the simple situation in which both graphs have one edge. Notice that as the weight on the blue graph grows, the endpoints, seven shape points, and length of the red mean are pulled toward the blue graph.

When one of the edges in the mean calculation is a dummy edge, the mean must be calculated slightly differently. A real edge that is matched with a dummy edge lends its spatial components to the mean edge, so the resulting mean has the same end points and shape points as the real edge. However, the length of the resulting edge is a weighted average of the length of the real edge and 0, down weighting that edges importance in the means.

Figures 2.6(g)-2.6(l) show the weighted mean of two graphs that have a different number of edges. In this example, the crossing on the green ‘f’ is matched with the crossing edge on the right side of the blue ‘f’ as result of the linear programming routine. The dummy edge appears in Figure 2.6(h) when the blue graph begins to accumulate some weight, where it is matched with the remaining real edge. Although the images do not show it, the length of that edge is down-weighted between the length of the blue edge it represents and 0 (the length of the dummy edge).

Due to repeated down weighting of an edge across a set of graphs, it is possible that the length of certain edges gets very small. We delete any edge from the mean graph that is shorter than 1 pixel long in the sense of weighted length. This prevents the number of edges in the mean graph from being equal to the largest graph in the set. Figures 2.6(m)-2.6(x) give a look at a more complex comparison where dummy edges and pruning both play a role as the weightings progress towards $p = 1$.

With this process we can calculate the mean of two graphs for any value of the weight, $p$. Now consider a larger set of graphs, say $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$. Define a function to calculate the weighted mean $w'(\mathcal{G}_1, \mathcal{G}_2, p)$ where $p$ is the $[0, 1]$ weight put on $\mathcal{G}_1$. Then, the mean of the first two elements of $\mathcal{G}$ is $m_2 = w'(\mathcal{G}_1, \mathcal{G}_2, 0.5)$. To incorporate the third element into the mean, we must take care that the included graphs are each weighted evenly in $m_3$, so $m_3 = w'(m_2, \mathcal{G}_3, \frac{2}{3})$. For each subsequent
Figure 2.6  Figures showing weighted means of two letters. The red line represents the weighted mean, and the weighting component, \( p \), indicates the amount of weight placed on the blue graph in each panel. For example, Figure 2.6(a) shows the mean weighted completely on the ‘n’, then left to right transitions from ‘n’ to ‘v’, and finally 2.6(f) shows the mean weighted completely on the ‘v’.
mean calculation the update formula is $m_i = w'(m_{i-1}, G_i, i-1)$. Applying this update formula iteratively across a set of graphs provides a formula for calculating a set mean.

While this set mean does a good job of summarizing a set of similar graphs, it has some properties that are sub-optimal when encountering a set of widely varying graphs. The most egregious is that the set mean depends on the order in which the graphs are introduced into the update formula. This is due to the path matching step of the algorithm, and is unavoidable. We address this shortcoming in the coming section by modifying the clustering measure of center.

2.4.4 $K$–means Type Algorithm

Prior to implementing the $K$-means framework, we consider what an outlier may look like in our data format. The presence of outliers can have undesirable effects on cluster groupings (Tseng and Wong, 2005). There will be certainly be graphs that are very different from all of the others, crossed out words, for example. Additionally, there may be sets of similar graphs that do not occur frequently enough to form their own cluster, but are not close to any of the other cluster centers. Such observations are considered outliers in the setting.

Figure 2.7 shows two examples of unconventional graph types that will be far from every cluster mean. Figure 2.7(a) is a large complex graph that is not repeatable, and Figure 2.7(b) is a simple, but uncommon structure in handwriting. When $K$-means encounters an observation very far from any cluster it may either pull the center of a cluster toward it, thereby summarizing the rest of the cluster poorly, or it may create its own cluster with only that point. Neither of these options are desirable.

To accommodate outliers, we implement the method of Gan and Ng (2017). A cutoff distance is derived from the average distance from each observation to its cluster mean, and used to decide which are too far from any center. Such observations are grouped as a set of outliers, meaning they will not contribute to any cluster center calculations. The number of outliers is capped by a parameter, but the number of classified outliers is learned based on the data. In the case of no outliers the modified algorithm simplifies to standard $K$-means.
The clustering framework of Forgy (1965) and Lloyd (1982), the simplest formulation of \( K \)-means, starts with a set of initial clusters and iterates through two steps until convergence. First, each point in a dataset is assigned to the cluster whose center is closest to it, then the cluster centers are recalculated with all of the points that were just assigned to that cluster. Rather than use the cluster means calculated as in Section 2.4.3, we increase the stability of the clustering algorithm by using the cluster \emph{exemplar} as the center for the group reassignment step. A cluster exemplar is simply the data point closest to the calculated cluster mean. The complete algorithm is as follows.

**Outlier Tolerant \( K \)-means Algorithm for Graphs:**

Consider a set of observed graphs \( X = \{X_1, \ldots, X_n\} \), a pre-specified number of clusters \( K \), a set of initial cluster exemplars \( C = \{C_1, \ldots, C_K\} \), a maximum number of outliers \( n_o \), and a parameter \( \gamma \) which is part of the calculation that controls the allowable distance from a center to an observation before it is classified as an outlier. Also define the variable \( T_o \) to be the distance threshold by which outliers are determined. \( T_o \) will be learned and adjusted within each iteration of the algorithm, but the initial value of \( T_o = \infty \). The cluster assignment of observation \( i \) is \( \varphi_i \), where \( \varphi_i = -1 \) denotes that the \( i^{th} \) observation is an outlier.

Iterate through the following steps until the cluster assignments do not change:
1. Assign each graph $X_i \in X$ to the cluster whose exemplar is nearest to $X_i$ with respect to the distance measure in Section 2.4.2.

$$\varphi_i = \arg\min_{j=1, \ldots, K} \{D(X_i, C_j)\}$$

2. Remove the cluster assignment of $X_i$ and call it an outlier (set $\varphi_i = -1$) if its distance to its cluster exemplar, $D(X_i, C_{\varphi_i})$, is larger than $T_o$. If more than $n_o$ observations would be outliers, only reassign the observations with the $n_o$ largest distances.

3. Calculate the mean of the graphs in each cluster as in Section 2.4.3. Update $C_1, \ldots, C_K$ to be the $X_i$ closest to each calculated mean (exemplars). Outlier graphs do not contribute the mean calculations.

$$C_j = \arg\min_{X_i | \varphi_i = j} D(X_i, \text{Mean}(\{X_m | \varphi_m = j\}))$$

4. Update $T_o$ with

$$T_o = \frac{\gamma}{n - \sum_{i=1}^{n} I(\varphi_i = -1)} \sum_{i=1}^{n} I(\varphi_i \neq -1) D(X_i, C_{\varphi_i})$$

When the algorithm has converged, each graph is assigned to the cluster with the nearest center, with the exception of the outlier cluster. Like all versions of K-means, only a local optimum is guaranteed. Running the algorithm with different starting values is suggested to try to reach the global optimum.

The input parameters $n_o$ and $\gamma$ are set with the default values of $n_o = .25n$ and $\gamma = 3$, which allows for up to 25% of the observations to be called outliers and requires a distance 3 times the average within cluster distance to call an observation an outlier. For this algorithm, $K$ is fixed before running, and throughout the remainder of the paper, $K$ will be assumed to be known, so no $K$ selection criteria are discussed.

### 2.5 Application

We demonstrate the use of our clustering method in a forensic handwriting analysis context using a subset of the publicly available CVL handwriting database (Kleber et al., 2013) to perform
a writership analysis on a set of documents with known origins. The data set is a collection of 162 writing samples by 27 writers, and 5 prompts for each writer. Immediately, we set aside the fourth writing sample of each writer to create a holdout set of documents. These will serve as questioned documents in the writer identification exercise to come. The documents are cropped to the smallest bounding box containing writing and down sampled to contain a maximum of 1750000 pixels without changing the aspect ratio of the document. Then, handwriter is used to parse each of the documents into graphs.

2.5.1 Cluster Assignments

For computational purposes, we use the graphs extracted from one document for each writer to perform clustering, effectively creating a cluster template from a diverse collection of writing. After the clusters are created, the graphs that were not used during clustering are sorted into groups based on their distances to the established cluster exemplars.

The training graphs are grouped into $K = 40$ clusters. We consider more clusters than letters in the alphabet because there are a variety of forms for many letters that often correspond to graphs. Supplementary materials provide results for analyses given other reasonable values of $K$.

Figure 2.8 shows the 40 exemplars that result from the clustering, ordered based on the number of observed graphs that fall into each cluster, and Figure 2.9 shows the calculated means for the clusters in the same order. While the exemplars are used for functional purposes, the combination of these two measures of center provides an interesting point of view to investigate behaviors of the clusters built from handwritten documents. For brevity, we will not discuss each of the 40 clusters individually. Instead, we note some general trends and focus on specific clusters of interest.

Most of the highly populated clusters are very compact, and the exemplars are good descriptors for the rest of the graphs. Many of these clusters describe very simple graphs like vertical segments and single path graphs, such as those representing ‘c’s or the stem of an ‘i’.

Clusters that are comprised of more complicated graphs have more natural variability both within and between writers. This can be seen in the less populated clusters like 2.8(ab) and
Figure 2.8 Each subplot represents 1 of the 40 clusters obtained for the 162 CVL documents. The black background figures are the graphs which were grouped into the cluster. The exemplars are shown in red on top of the graphs.
Figure 2.9  These plots show the raw calculated mean for each of the clusters. The panes are ordered to match the exemplars in Figure 2.8. These means indicate generally important characteristics of each cluster.
Looking at the mean of the clusters, rather than the exemplar, can reveal patterns in the clusters of complex graphs. Figure 2.9(al), for example, reveals that in the complex cluster shown in Figure 2.8(al), the common structural component is the loop at the bottom of the graph. In the displayed mean, the dots above the loop signify that there are more edges in the graph, but they are not important in grouping graphs into that cluster (recall that small edges come with very small weights in the distance measure). Figure 2.9(an) reveals a similar characteristic for the cluster in 2.8(an).

The complex clusters are examples of how this graph grouping method is tolerant to small deviations from an overall dominating structure. Many of the graphs in those complex clusters do not have the same adjacency matrix (or even number of edges) and would never be considered in the same group under the deterministic grouping method.

A large number of the clusters align directly with known letter types. Cluster 2.8(m) is largely ‘f’s, cluster 2.8(r) contains ‘n’s, clusters 2.8(s), 2.8(w), and 2.8(aa) are groups of ‘e’s, clusters 2.8(ac) and 2.8(aj) contain ‘o’s, and so on. If each occurrence of a character type were grouped into a single cluster, then we would be counting the frequency in which letters appear in the writing prompts, rather than capturing a characteristic of handwriting forms. As a mini identification example, temporarily ignoring all the other graphs groups, the rate at which the writer emits ‘e’ graphs into 2.8(s), 2.8(w), or 2.8(aa) is information that sets them apart from other writers.

We expected that large, non-repeatable, complex graphs would be called outliers. This is true to an extent, but many of the complicated graphs got a cluster to accommodate them. The cluster shown in Figure 2.8(ab) represents this unexpected, but not unwelcome cluster type. From Figure 2.9(ab) we observe that the mean is essentially a large set of points scattered around the character space, which absorbs the graphs with a large number of edges, making it very diverse. A large proportion of the outliers are actually simple graphs, pictured in Figure 2.10, that do not line up with any of the existing cluster types.

Only certain types of writers are likely to produce the large simple graphs that comprise the outlier cluster. This makes the frequency with which outlier graphs are observed a useful piece of
37

| 6 | 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Figure 2.10  This set of graphs represent the observations categorized as outliers. Some of the outliers are very complex graphs, while more are large, simple graphs. The general trend for outliers is that they are large, simple graphs.

information for writer identification. Thus, we include it as the 41st cluster in the analysis that follows.

2.5.2 Writer Identification

There are 135 documents in the training data set (5 prompts for each of the 27 writers), and 27 documents in the holdout dataset (1 prompt for each of the 27 writers). We will index model elements from 1 to 27 for the sake of visuals and tables, but we note that the writer identification names from CVL are not consecutive and span the range of 1 to 50.

For each document, we tally the number of graphs that fall into each of the 41 clusters. A Bayesian hierarchical model for this count data is developed. The notion of gathering handwriting information in “bins” has been used before (Bhardwaj et al. (2010) use such data with a Latent Dirichlet Allocation model and Saunders et al. (2011) with a variety of classifiers). The desired result of our model is a probabilistic conclusion about writer identification for each of the held out documents. We compare model results from our dynamic cluster groupings with results using the same model, but with the deterministic adjacency grouping method discussed in Section 2.3.3.

2.5.2.1 Model Formulation

Let $Y_{w(d),c}$ be the number of graphs assigned to cluster $c$ for training document $d$ nested in writer $w$ where,
\[ c = \{1, \ldots, K + 1\}, \] where \( K + 1 \) arises from the fixed number of clusters plus the group of outliers, and the cluster ordering is the same as in Figures 2.8 and 2.9,

\[ w = 1, \ldots, 27, \] and

\[ d = 1, \ldots, 5. \]

Then, \( Y_{w(d)} = \{Y_{w(d),1}, \ldots, Y_{w(d),K}, Y_{w(d),K+1}\} \) characterizes the number of graphs assigned to each cluster for document \( w(d) \). We consider these cluster assignments as samples from a multinomial distribution with a writer specific parameter vector, \( \pi_w \), that captures rate at which a writer emits graphs to each cluster. Let \( \pi_w = \{\pi_{w,1}, \ldots, \pi_{w,K+1}\} \) denote the simplex of size \( K + 1 \) for writer \( w \) from a Dirichlet distribution. The \( K + 1 \) hyperparameters of the Dirichlet distribution, denoted as \( \alpha = \{\alpha_1, \ldots, \alpha_{K+1}\} \), are assigned independent gamma priors. The mathematical formulation of the model is

\[
\begin{align*}
Y_{w(d)} \overset{\text{ind}}{\sim} & \ Multi(\pi_w), \\
\pi_w \overset{\text{ind}}{\sim} & \ Dirichlet(\alpha), \\
\alpha_c \overset{iid}{\sim} & \ Gamma(a = 3, b = 0.25). 
\end{align*}
\] (2.6)

Markov chain Monte Carlo (MCMC) estimates were obtained using the rstan R package (Stan Development Team, 2018). In the analysis that follows, 3000 samples were collected after a burn-in period of 2000. Denote these samples as \( \alpha^{(m)} \) and \( \pi_w^{(m)} \), for \( m = 1, \ldots, M (= 3000) \).

Figure 2.11 shows posterior densities of the \( \pi_{w,c} \) components for all writers and seven of the \( K + 1 \) clusters. With overlap in every pane, there is no cluster that can completely separate the writers. However, considering all of the \( \pi_w \) simultaneously, a writer’s style can be sufficiently captured and characterized by the rate in which their graphs are assigned to various clusters. The bottom pane of Figure 2.11 shows the posterior samples for the probability of observing outlier graphs for each writer. Overlap of the densities in the outlier pane resemble those in the traditional cluster panes. The separation of writers observed in the outlier densities is again a suggestion that there is identifying information carried in the frequency of outliers.
Figure 2.11  This plot shows densities of the posterior samples for $\pi$. Within a pane there is a density included for each writer. The top three panes of Figure 2.11 present the cluster proportion estimates for the three largest clusters. That is, the posterior distributions for $\pi_w,1$, $\pi_w,2$, and $\pi_w,3$, $\forall w = 1, \ldots, 27$. The next panes, labelled 10, 11, 12, and Outliers, show the same for less frequently occurring clusters.
2.5.2.2 Prediction

We use the model defined in Equation 2.6 to evaluate a holdout document written by unknown writer, $w^*$. The holdout document is processed by the `handwriter` R package to extract graphs. Every graph is assigned to a cluster, and the number of graphs in each cluster is tallied to obtain a multinomial response vector for the new document $Y_{w^*} = \{Y_{w^*,1}, \ldots Y_{w^*,K+1}\}$. We drop the document nested within writer notation, $w(d)$, here since we consider only one questioned document at a time.

We use the posterior predictive distribution to calculate posterior probabilities of writership for each of the 27 writers in the training data. Let $\pi^{(m)}_{w'}$ be the $m^{th}$ MCMC sample of the multinomial parameter vector for a particular writer in the training data set, $w'$. Then the posterior probability of writership for writer $w'$ on the new document $Y_{w^*}$ at MCMC iteration $m$ is

$$p_{w'}^{(m)} = \frac{\text{Multi}(Y_{w^*}; \pi^{(m)}_{w'})}{\sum_{w_i=1}^{27} \text{Multi}(Y_{w^*}; \pi^{(m)}_{w_i})}. \tag{2.7}$$

Performing this calculation for every known training writer yields the probability vector

$$p^{(m)} = \{p_{w_1}^{(m)}, \ldots, p_{w_{27}}^{(m)}\}, \tag{2.8}$$

and we summarize the probability of writership for writers over all MCMC samples as

$$\bar{p} = \{\bar{p}_{w_1}, \ldots, \bar{p}_{w_{27}}\}, \tag{2.9}$$

where $\bar{p}_{w_i} = \frac{1}{M} \sum_{m=1}^{M} p_{w_i}^{(m)}$. In addition to the mean, we calculate lower and upper 90% credible bounds for the $p_{w_i}$. This posterior predictive procedure was used to evaluate each of the 27 holdout documents.

Figure 2.12 provides a graphical summary of results for the holdout documents using our graph clustering method. Each row presents the summary of the probabilities assigned to each of the 27 possible writers for a given questioned document. The diagonal, where the known writer of the questioned document matches the training set writer, indicates a correct identification. We place more than 0.5 of the average posterior probability on the true writer for all but one of the questioned documents. The holdout document written by writer 42 only places 0.161 average probability on
Figure 2.12 Results of the posterior predictive analysis defined in Equations (3.7) - (3.9) for each of the 27 holdout documents in which graphs were grouped by our clustering algorithm. Columns represent known writers. There is one row to summarize each questioned document analysis with the true writer labeled on the left-hand side. Cells in the row are colored and labeled by elements of their $\bar{p}$ vector defined in Equation (3.9). Rows are independent and sum to one, as all of the posterior writership probability is spread across the writers. Diagonal cells indicate the correct identification for the questioned documents. A point for the average probability, and 90% credible interval band is plotted below the cell label. Cell contents are omitted for any cell whose average probability rounds to 0 in three decimal places.
the true author, while writer 23 gets the rest. Despite this, the credible interval bands in this row are wide for both highlighted cells, indicating the model acknowledges the uncertainty associated with the questioned document for writer 42. The supplementary material includes figures similar to 2.12, but with $K = 30$ and $K = 60$ to cluster instead of 40. Results are similar.

The deterministic adjacency groupings of Section 2.3.3 are used in the same modeling structure. The rigid nature of these groupings are such that approximately 60% of training graphs fall into two of 1636 adjacency groups. This pile-up renders some of the graph group assignment information less valuable, since it tends to happen for most writers. In addition, many groups include only one or two graphs in them. In order to make the deterministic grouping results comparable to that of cluster grouping, we take the sparsely populated adjacency groups and collapse them into two new groups, making 40 total adjacency groups. We call the two new groups “moderate” and “rare” to describe their membership populations.

Figure 2.13 reflects similar information as Figure 2.12, but the modeling results are products of the adjacency grouping method described in Section 2.3.3. Compared to the cluster grouping results of Figure 2.12, it is clear that there is more posterior probability assigned off-diagonal when the deterministic grouping is used.

Under the adjacency grouping method the model places more than 0.5 of the average probabilities on the true writer for only 23 of the 27 holdout documents. We should note that in a forensic statistics context, one would never use the cutoff of 0.5 probability for actual evidence as corresponding conclusions would be fraught with false positives. We use it here to make simple comparisons between the grouping method results. Many of the on-diagonal cells in the deterministic grouping results reflect probabilities between 0.5–0.8, in which case we would probably not be comfortable making an identification. The adjacency grouping handles the questioned document for writer 42 poorly, which is the document that the clustering-based method also struggles to identify.
Figure 2.13  Results of the posterior predictive analysis defined in Equations (3.7) - (3.9) for each of the 27 holdout documents using the deterministic adjacency grouping of graphs. Columns represent known writers. There is one row to summarize each questioned document analysis with true writer labeled on the left-hand side. Cells in the row are colored and labeled by elements of their $\bar{p}$ vector defined in Equation (3.9). Rows are independent and sum to one, as all of the posterior writership probability is spread across the writers. Diagonal cells indicate the correct identification for the questioned documents. A point for the average probability, and 90% credible interval band is plotted below the cell label. Cell contents are omitted for any cell whose average probability rounds to 0 in three decimal places.
2.6 Discussion

In this paper, we create a clustering algorithm for grouping small pieces of handwriting, and use an individuals propensity for creating graphs that belong to particular clusters to identify the writer of questioned source documents. Development of the clustering algorithm hinges on defining measures of distance and center for handwritten graphs. Our graph distance measure emulates a graph edit distance measure while leveraging additional attributes that the graphs possess. On the CVL handwriting dataset we create clusters for graphs derived from handwriting samples and show that writers can be characterized and identified by examining the frequency with which they emit graphs to the various clusters. The graph assignments serve as observations for a multinomial-dirichlet hierarchical model, and posterior predictive of writership for a set of questioned documents is discussed. We compare these identification results to those obtained using deterministic grouping of the graphs, showing that our cluster assignments improve upon the easily available, but volatile, deterministic assignments and proffers an improvement in model results.

In addition to the clustering algorithm, we describe the procedure by which a document is processed and turned to data. The handwriter R package reads, processes, and parses a document into graphs, the observations we work with during clustering. This processing takes the extremely complex data source, and makes usable information readily available in the form of attributed graphs. The clustering and writership analysis as we did it would not be possible without this extraction.

In the handwriting application there are times when the clustering method produces surprising results. In particular, the mean calculation for diverse clusters results in non-intuitive centers. For diverse clusters, the mean tends to shrink towards the centroid of the observations, and the order that the observations are introduced to the weighted mean calculation can have small yet meaningful impacts on the resulting measure of center. This issue is mitigated in this paper by using exemplars rather than the raw weighted mean as the measure of center for the K-means algorithm, but finding a way to calculate a stable mean for any cluster type could improve and simplify our algorithm.
As with every distance measure, the one presented in this paper favors certain properties over others in assessing similarity. It would be interesting, and easy, to implement distance measures that prioritize other features of handwriting. For example, distance measures that do not take edge shape into account would prioritize edge location more heavily. On the other hand, discounting the location of an edge within a graph would lead to a measure that cares only about how similar the edge shapes are and not at all about where the edges lie in space. The distance measure that we implement is used because it leans evenly on many types of distance without over emphasizing any specific aspects of the graphs.

2.7 References


CHAPTER 3. A STATISTICAL ANALYSIS OF FORENSIC HANDWRITING WITH INSIGHTS REGARDING WRITER STYLES

A paper prepared for the *Proceedings of the National Academy of Sciences*

Amy Crawford, Alicia Carriquiry, and Danica Ommen

3.1 Abstract

We develop a statistical model that allows all styles of writing (cursive, print, connected print) and gives a probabilistic evaluation of authorship of a questioned document given a closed-set of candidate writers. Such probabilistic statements can support examiner conclusions and be used by the community to provide free and open-source methods for searching a collection of documents. Writing is treated as a sequence of disjoint graphical structures, which are extracted using an automated and open source process. The graphs are grouped based on the similarity of their shapes through a $K-$means clustering template Forgy (1965); Lloyd (1982). Writers can be characterized by the rate at which they emit graphs to each cluster. The cluster memberships serve as data for a Bayesian hierarchical model with a mixture component. The rate of mixing between two parameters in the hierarchy gives an indication of writing style.

3.1.1 Significance Statement

Forensic analysis of handwritten samples has been in use for almost a century. A celebrated implementation of handwriting comparison occurred in 1935, and helped convict Bruno Hauptmann of the kidnapping of famous aviator Charles Lindbergh’s baby. To this day, document examiners visually compare a questioned document with reference samples and provide a subjective assessment of similarity. We propose a modeling approach to estimate the posterior probability of authorship of a questioned document given a set of potential writers. Writing is treated as disjoint graphs, which
are clustered dynamically into 40 groups. Cluster membership serves as data for a hierarchical mixture model, where the mixture parameter is associated with writing style.

### 3.2 Introduction

There has been a movement brewing in the forensic science community toward a more objective and probability-based evaluation of evidence. A variety of governing bodies have made this effort explicit and public, emphasizing the need for research to support scientific and statistical foundations of evidence evaluation (President’s Council of Advisors on Science and Technology (US), 2016; National Research Council, 2009; National Institutes of Standards and Technology, 2019).

We take a parametric modeling approach to the evaluation of a questioned writing sample against a finite set of genuine writing samples from known sources. The methods and conclusions presented in this article are part of a body of work that aims to close the gap with respect to these needs for the evaluation of handwritten documents, in particular.

Handwriting can arise as forensic evidence in the form of forged signatures, bomb threats, bank notes, ransom notes, and many others, and traditional analysis of such materials is done by trained Forensic Document Examiners (FDEs). Details regarding the role of the FDEs and traditional analysis procedures can be found in standards documents (such as Scientific Working Group for Forensic Document Examination (2015a), Scientific Working Group for Forensic Document Examination (2015c), and Scientific Working Group for Forensic Document Examination (2015b)) produced by the Scientific Working Group for Forensic Document Examination (SWGDOC). Usually, a comparison of the questioned document (the evidence, the writing of unknown source) is made against a set of reference material of known origin. The examiner gives their opinion, whether by testimony in court or within reporting documents, about the similarity between documents and indicates whether they believe the questioned writing originated from the same source (writer) as

---

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement #70NANB15H176 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, University of California Irvine, and University of Virginia.
the reference material or not. They usually follow a reporting scale that has been accepted by the community of FDEs. The process has clear inputs, and relatively clear results, but the analysis that falls in-between is considered to be done by a “black box.” The examiner can point out features of the writing that they consider especially informative given their conclusions, but can never fully express how the human brain aggregates the information it takes in. On top of that, each examiner is a unique black box, with unique perspectives and biases that have been developed over years of formative experiences. See Kassin et al. (2013) and Dror and Pierce (2019) for discussion on bias in forensic science and standards developed to address them.

A more objective approach is one we take here. A statistical model is used to aggregate features extracted from handwritten documents in a systematic and transparent way. Using the model, we estimate probabilistic outcomes regarding the writership of a document of questioned source, with respect to a finite set of genuine handwriting samples from known sources. This approach is not meant to replace the FDE, but rather, to provide a statistical framework to support examiner source conclusions. Given a questioned document, our approach searches a collection (or closed-set) of known writers and gives an estimated probability of source within the closed-set. The closed set can be very large, but the author of the questioned document is assumed to be included in the set. This approach has gained interest in the community for applications in relatively contained situations, such as school bomb threat letters, where the closed-set of reference writers would be the students at the school.

To accomplish the analysis we consider writing as a sequence of disjoint graphical structures with nodes and edges (see Figure 3.1). The graphs we extract through a decomposition of connected ink can and often correspond to characters, but can also capture parts of, or groups of connected characters. Others have taken a similar approach, treating writing elements as graphs. For example, with the goal of isolating the writing order, Parziale et al. (2014) treat characters as graphs that can be traversed from first element written to last. Miller et al. (2017) extract graphemes, which are not necessarily disjoint, from writing and treat them as graphs to be compared by topology (connection pattern of a graph) and geometric features.
To capture general tendencies for any given writer, we use a template to sort each graph into a group, or cluster, that best represents it. The template is generated by a $K-$Means clustering algorithm and a variety of writing samples (See Chapter 2). The rate at which a writer emits graphs to each of the forty clusters reflects the structure of their writing characters across a document. We fit a Bayesian hierarchical model to the frequency data. The model allows for distinct writing patterns, and provides insight into two writing styles, cursive and print, and the combination style of writing that spans the space in between.

### 3.3 Handwriting as Data

Turning a handwritten sample into data that is usable by any statistical algorithm requires a series of image processing and feature extraction steps. In this section we briefly describe those steps.

Document processing is handled by the R package **handwriter**, developed at the Center for Statistics and Applications in Forensic Evidence (CSAFE). Handwritten documents, stored as image files with a 72dpi resolution, are first binarized to turn each pixel black or white and isolate the writing from background noise. Then the isolated writing is reduced to a one pixel wide skeleton, and the skeleton is broken into smaller pieces (often corresponding to letters) based on a set of rules.
Each of these pieces is treated as a graphical structure with nodes and edges. Figure 3.1 shows the letter 'f' represented as one such graph. Greater detail regarding the document processing workflow can be found in Chapter 2.

3.3.1 Feature Extraction with K-Means Clustering

The clustering algorithm from Chapter 2, with minor changes, was used to create a clustering template for feature extraction. One hundred documents from three open source repositories served as data for the template creation.

1. Center for Statistics and Applications in Forensic Evidence (CSAFE) Handwriting Database Crawford et al. (2019a,b), 25 documents. Each participant in this dataset provided writing samples on three occasions, or sessions. We selected 25 "London Letters" provided during the third session, each from a distinct writer.

2. Computer Vision Lab (CVL) Database Kleber et al. (2013), 25 documents. A subset of participants in this dataset provided writing samples from seven prompts. One is in German, and omitted here. We selected 25 documents from the subset, each from a unique writer, comprised of six unique writing prompts.

3. IAM Handwriting Database Marti and Bunke (2002), 50 documents. The writing prompts in this dataset are very diverse. We selected 50 documents, each with a distinct prompt and writer.

In each iteration of the K-Means algorithm defined in Chapter 2, graphs are sorted into one of 40 clusters based on their similarity to each cluster exemplar. Then, within each cluster, an new exemplar is calculated to serve as the measure of center for current cluster members. All of the graphs are then re-sorted into the clusters using the updated exemplars, and the process is repeated until there are no cluster assignment changes.

We make three small adjustments to the calculations of this algorithm. First, the measure of graph-to-graph distance is adjusted to rely more heavily on shape and less on size. This is done to
accommodate the variety of letter sizes that come with incorporating a large number or writers from a variety of data sources. There are three contributions to the distance measure: shape, straight line displacement (capturing size, in a way), and location (anchoring the elements in space). Originally given equal weight, we take half weight on the straight line displacement and double weight on the shape component. Next, to facilitate stable convergence of the algorithm, we calculate the iterative average graph by including letters in order of which was most similar to the exemplar during the sorting process. Last, it often happens in the edge matching process of the distance comparison that the edge of one graph matches with nothing (a ghost edge) in another graph because the graphs have a different number of edges. When this occurs, we change the contribution of such an edge comparison to be the length of the real edge squared, rather than just the length of the real edge. This gives a larger penalty to long edges of one graph that do not find a mate in the other graph, yielding a larger distance between graphs in such circumstances.

We implemented the modified algorithm with the above changes 20 times. Each implementation was given a different set of 40 starting values to initiate cluster exemplars. The starting values were chosen by stratified sampling of the template generating dataset. Exemplars from the implementation with the smallest within-cluster sum of squares in the final assignment was selected to serve as the template moving forward.

Ten of the 40 final cluster assignments in the template are pictured in Figure 3.2. The clusters are numbered for labelling purposes in the algorithm, and are meaningless outside distinguishing between the 40 groups. Cluster 32 has members with a vertical stroke in the center of the graph and appendage strokes stemming from the center. Members of this cluster tend to be ‘t’s, ‘f’s, ‘k’s or other similar forms. Cluster 34, tends to include compound style letters with multiple loops. These members are an ‘sh’, ‘gh’, or ‘ft’ style combination with a short looped first component and tall thin second component that did not satisfy the rules to be broken apart during processing. Cluster 15 holds similar graphs, but with fewer compound members. Cursive ‘f’s, ‘g’s, ‘d’s, or any other form with a structure dominated by two stacked loops. Cluster 39 captures many ‘u’s without
leading or trailing ink, indicating that these letters were not connected to their neighbors. Many letters like the lowercase ‘a’ and ‘o’ that do not complete their loops are also included here.

Now, we use the clustering template as a feature extraction tool for the documents that will serve as data for a writership analysis. These documents come from the first data collection session for all 90 writers in the CSafe Database Crawford et al. (2019a,b). For each writer, we use three London Letter Osborn (1929) prompts for model fitting and one Wizard of Oz L. F. Baum, illustrated by W.W. Denslow (1900) prompt is held out as a ”questioned document” for testing.

All documents are processed by handwriter to extract graphs from the writing. Each graph is then given a cluster assignment based on similarity to the cluster template exemplars, ten of which are shown in red in Figure 3.2. Similarity is assessed by the same distance measure employed in the clustering algorithm for template creation. We utilize these cluster assignments as compositional data vectors for statistical modeling. In similar spirit, Saunders, et al. train classifiers to categorical data extracted from handwriting in Saunders et al. (2011).

We move forward with the goal of building a statistical model to distinguish between writers using the shape of the extracted data vectors. For each document, we tally the number of graphs assigned to each cluster, creating a vector of length 40. Scaled vectors for three documents, by
Figure 3.3  Scaled cluster assignment vectors for three documents in the training dataset. The top is a document by Writer 12, the middle by Writer 66, and the bottom by Writer 124. For comparability, the vectors have been scaled so that the cluster assignments sum to one, rather than the number of graphs in each document. The result is a proportional breakdown of each document.

three different writers, are shown in Figure 3.3. Notice the difference in overall shape of the vectors in the figure. In the Bayesian hierarchical model to follow, we use three training documents to estimate the general cluster fill rate tendencies for each of the 90 writers. Being Bayesians, we let those writer-wise fill rates learn from a grand fill rate that captures the general large scale behavior of all writers. Although the writers tend to exhibit similar cluster fill patterns overall, we observe two sub-patterns emerge in vector shapes across writers (see Figure 3.3). As such, in the modeling framework we will consider letting writer-wise fill rates learn from a mixture of two grand fill rates, where the mixing parameter is allowed vary from writer to writer.

3.4 Methods

After the documents have been processed by *handwriter* and the cluster assignment feature vector has been extracted from each, we have all the components necessary for developing a hierarchical model. In the model, we capture the cluster fill tendencies of each writer as a learned parameter. Each writer in the closed set will have a parameter representation, and when a new
3.4.1 Bayesian Hierarchical Mixture Model

Let $Y_{w(d)}$ be a vector of length $K = 40$ that denotes the number of graphs assigned to each cluster for document, $d = 1, 2, 3$, within known writer, $w = 1, \ldots, 90$. Let

$$Y_{w(d)} \sim \text{Multinomial}(\pi_w), \quad (3.1)$$

where $\pi_w$ is the 40-simplex (a 40-dimensional vector in which the elements sum to 1) that captures the cluster fill probabilities for writer $w$. Let the prior distribution for the $\pi_w$ be a Dirichlet distribution with writer-wise mixture over the parameter space

$$\pi_w \sim \text{Dirichlet}(\rho_w \alpha_1 + (1 - \rho_w) \alpha_2), \quad (3.2)$$

where $\alpha_1$ and $\alpha_2$ are two distinct parameter vectors of length $K$ that can be conceptualized as pseudo cluster counts for two writer types. The mixing parameter for each writer in the dataset, $\rho_w$, determines the combination of the two types for writer $w$. We use final level priors for these parameters of

$$\rho_w \sim \text{Beta}(1, 1), \quad (3.3)$$

$$\alpha_{1,i}, \alpha_{2,i} \sim \text{Gamma}(2, 0.25), \forall i = 1, 3, \ldots, K. \quad (3.4)$$

To handle potential label switching, we let the second element of each vector be separated and restricted a priori by

$$\alpha_{1,2} \sim \text{Gamma}(5, 0.25), \quad \text{and} \quad (3.5)$$

$$\alpha_{2,2} \sim \text{Gamma}(1, 0.25) \mathbb{I}[\alpha_{2,2} < \alpha_{1,2}]. \quad (3.6)$$

This ensures that the alpha vector with the larger second element is always considered as $\alpha_1$. It will become clear that the second element of the $\alpha$ vectors is suitable as a constraining entry to handle vector-wise label switching in the next portion of the paper.
3.4.2 Posterior Estimation

We estimated posterior distributions of model parameters using Markov Chain Monte Carlo (MCMC). The chain was thinned by 5, with a burn-in period of 1000 iterations before 4000 iterations were stored.

A $\pi_w$ simplex for a particular writer $w$ captures the relative frequency with which the writer produces writing that falls into each of the 40 clusters. Figure 3.4 shows posterior samples of elements from the $\pi_w$ simplices, colored by the posterior mean $\hat{\rho}_w$ for each writer $w$. For brevity, vector elements corresponding to only the 10 clusters occurring most frequently in the training data are shown. Notice that writers with high values of $\hat{\rho}_w$ (red densities) tend to produce writing from clusters 34, 35, and 15 at a higher rate. The corresponding template elements are shown in Figure 3.2(b), 3.2(c), and 3.2(i) respectively. These are clusters that tend to capture more loopy and connected writing structures. The same writers tend fill clusters 39, 9, and 31, shown in Figure 3.2(g), 3.2(h), and 3.2(j) at a lower rate. These clusters include elements that tend not to be connected to the letters around them (little leading or trailing ink) and do not include many loops. Trends like this are found throughout the 40 elements of the $\pi_w$ vectors, some elements showing more dominant patterns in $\rho$ values than others. The patterns we see here point to a link between mixing parameter $\hat{\rho}_w$ and a writer’s style.

3.4.3 Writing Style Insights

We investigate the behavior of writers who fall across the spectrum of $\hat{\rho}_w$. Writers with an estimated mixing parameter near 1 are considered to have a style that more closely follows type 1, with prior information given by $\alpha_1$. We find that type 1 writers tend to make simpler, more broken letter forms. On the other hand, a $\hat{\rho}_w$ near 0 indicates a writer of type 2, who tend to use more connected and formal cursive forms.

Figure 3.5 shows the average relative frequency over the training documents of the ten most popular clusters for 20 writers. Panes are colored and ordered by average posterior $\hat{\rho}_w$ value. Notice
Figure 3.4  Posterior density estimates for the $\pi_{w,i}$ elements shown for $w = 1, \ldots, 90$ and $i = 1, \ldots, 10$ (the ten most frequently occurring clusters in the training data). Panes are labelled with the cluster number they correspond to, and include one density for each writer. Each writer’s density is colored by their average posterior $\hat{\rho}_w$ value. Posterior estimates for Writer 66 are highlighted and outlined.
Figure 3.5  Average relative frequencies of the ten most frequently occurring clusters in the training data taken over the 3 training documents. Data are shown for 20 writers that were selected to show the spectrum of $\hat{\rho}_w$ values. Panes are labeled by writer number and plots in each pane are colored to reflect the average posterior $\hat{\rho}_w$ value estimated for the writer.

Changes in the cluster fill rates as the spectrum progresses from cool to warm, or type 1 to type 2 writers. In particular, Cluster 34 is filled at an extremely low rate for the type 1 writers and grows.

Cluster 34 is the second most populated cluster in the training data, and corresponds to the constrained index in the $\alpha_1$ and $\alpha_2$ vectors of Equations 3.5 and 3.6. The trends shown across Cluster 34 in Figure 3.5 demonstrate the ability that a constraint on this index has to prevent vector-wise label switching between the two group parameters.

Figure 3.6 shows writing samples for 10 of the writers depicted in Figure 3.5. The writers near the top of the figure tend to have non-connected writing elements. Notice that in the blue and green writing there is not much retrace, very few loops, and the presence of letters that do not close, such as the ‘o’s, and ‘a’s in the upper-most sample. Writers in the middle of the figure with yellow tones exhibit more connection between some letters, use more loops. The writers near the bottom in reds tend to use more formal cursive forms, particularly in their letter connections and letters themselves, such as the ‘r’s and ‘s’s.
Figure 3.6  Writing samples taken from the training documents for 10 of the writers from Figure 3.5, colored by average posterior $\hat{\rho}_w$ value.
3.4.4 Posterior Predictive Distributions and Writer Identification Analysis

We use the $\pi_w$ draws in a posterior predictive evaluation of a questioned document with data $Y_{w^*}$, by unknown writer, $w^*$. The data vector for such a questioned document is evaluated via the multinomial likelihood under each of the known writers, $w = 1, \ldots, 90$. Let $w'$ be a particular known writer that is under consideration, and $\pi_{w'}^{(m)}$ be the $m$th MCMC sample of their multinomial parameter vector. Then, the posterior probability of writership for writer $w'$ on the new document $Y_{w^*}$ at MCMC iteration $m$ is

$$p_{w'}^{(m)} = \frac{\text{Multi}(Y_{w^*}; \pi_{w'}^{(m)})}{\sum_{w_i=1}^{90} \text{Multi}(Y_{w^*}; \pi_{w_i}^{(m)})}.$$  \hspace{1cm} (3.7)

Performing this calculation for every known training writer yields the probability vector

$$p^{(m)} = \{p_{w1}^{(m)}, \ldots, p_{w90}^{(m)}\},$$  \hspace{1cm} (3.8)

and we summarize the probability of writership for writers over all MCMC samples as

$$\bar{p} = \{\bar{p}_{w1}, \ldots, \bar{p}_{w90}\},$$  \hspace{1cm} (3.9)

where $\bar{p}_{wi} = \frac{1}{M} \sum_{m=1}^{M} p_{wi}^{(m)}$.

The posterior predictive distributions are used to evaluate the likelihood that a questioned document with data $Y_{w^*}$, by unknown writer, $w^*$, was written by a known writer $w$. We conduct the evaluation for all 90 known writers in the closed set and normalize to spread the total probability of writership for the document $Y_{w^*}$ between them.

Recall that one Wizard of Oz sample was held out for each of the 90 writers. We consider them as "questioned documents" with an unknown source and use the above process to evaluate writership of each within the closed set of known writers. Results are presented in Figure 3.8. Each row of the figure contains results for a questioned document, is in dependant of the others, and sums to one.

3.4.5 Examining Error

Almost 89% (88.73%) of the $\bar{p}_{w}$ assignments made for the set of 90 holdout documents was on the true writer. That is, most of the probability lands on-diagonal in Figure 3.8. The off-diagonal
Figure 3.7 Breakdown of incorrect (off-diagonal) probability assignment for the 90 hold-out documents. In each row, the tiles show the breakdown of all off-diagonal probability assignment. Rows of the plot are independent of one another, and each sums to one.

allocation of probability (11.27%) is placed on incorrect writers. We are interested in the behavior of the off-diagonal assignments.

Consider the $\hat{\rho}_w$ values in the discrete space of $(0,0.2]$, $(0.2,0.4]$, $(0.4,0.6]$, $(0.6,0.8]$, and $(0.8,1]$. Recall that the $\hat{\rho}_w$ parameter is learned in model fitting, and dictates the degree to which writer $w$ aligns with each of the two writing styles. In this application, we know the true writer of each questioned document and the discrete bin in which the $\hat{\rho}_w$ value for that writer falls. We can examine the off-diagonal assignment breakdown by the bin in which the $\hat{\rho}_w$ for the true writer of the questioned document is contained, and the bin that holds $\hat{\rho}_w$ for the writer that we assigned off-diagonal probability to.

Figure 3.7 shows this breakdown, and it is clear that when we assign off-diagonal probability, the recipients tend to have similar $\hat{\rho}_w$ values as the questioned writer. When the model misses, it is on writers that tend to lean the same amount on each of the style parameters, and have similarly shaped training data vectors.
Figure 3.8  Results of the posterior predictive analysis defined in Equations 3.7 - 3.9 for each of the 90 holdout documents. There is one row to summarize writership results of each questioned document. Ground truth writer is labeled on the left-hand side. Columns are labeled by known writers. Cells are colored by elements of their $\bar{p}$ vector defined in Equation 3.9. Rows are independent and sum to one, as all of the posterior writership probability is spread across the writers. Diagonal cells indicate the correct identification for the questioned documents.
3.5 Discussion

The analysis presented for the closed-set writer identification task is grounded in the treatment of handwriting as a sequence of disjoint graphical structures. A cluster based feature extraction process enables meaningful grouping of the graphs and produces data for statistical modelling via the template.

The Bayesian hierarchical model developed for analysis of the data has a multinomial parameter of interest, $\pi_w$, that represents the overall rate at which each writer, $w$, contributes graphs to each of the 40 clusters in their training documents. This parameter is fundamental to evaluating a questioned document against the closed-set of training documents with known author.

The nature of the data, and the multinomial model, is compositional in the sense that each graph in a document must be assigned to one of 40 clusters, and correspondingly, each of the parameter vectors $\pi_w$ must sum to 1. Across writers, the rate of cluster inclusions can take quite different shapes, and in the $\pi_w$ vectors, when one element gets larger, others must become smaller to satisfy the sum-to-one restriction. To accommodate a large variety of $\pi_w$ vector shapes, we include a mixing parameter, $\rho_w$, that allows for each writer to mix between two distinct vectors. It happens that one such vector, $\alpha_1$, corresponds to simplistic, disconnected print writers and the other, $\alpha_2$, to more formal, connected cursive writing. The more weight a writer puts on one or the other is a coarse indication of their writing style.

For model fitting, we used three training documents for most (81) writers included in the analysis, and only two training documents for nine of the writers. It is worth noting that the median on-diagonal probability assignment for those with three training documents is 0.990, and for those with two training documents is 0.925. This is anticipated, as the variability associated with parameter estimation for the two document group will be larger. This partially explains why we see some low on-diagonal probability assignment.

Count data models tend to imply strict relationships between mean and variance. When using such models it is important to consider whether the relationship between the mean and the variance (dispersion) that the model allows will plausibly accommodate the dispersion observed in the
data. In the univariate case, one can investigate the ratio of variance to mean, called the dispersion index, or coefficient of dispersion, under both the model and the data. In the multinomial distribution case, we turn to multivariate extensions of this simple index. The authors assessed the generalized dispersion index of Kokonendji and Puig (2018), similar to the multivariate extension of the coefficient of variation of Albert and Zhang (2010). In addition, a majorization approach to dispersion of multinomial data vectors (Shifeng and Guoying, 2010) was considered. Results of these checks yielded no cause for concern. Some writers’ data showed under-dispersion while others showed over-dispersion for the model fit. Dispersion evaluations by writer proved not to be related to accuracy of the questioned document evaluations.

Through an investigation of probability assigned to incorrect writers, there emerged a clear correlation between the mixing parameter value of the ground truth writer of the questioned documents, and the known writers to which the probability was incorrectly assigned. In order to gain increased accuracy in the posterior predictive analysis, a next step may be to include in the model measurements obtained on each of the graphs, in addition to the rate with which each cluster emerges in writing.

3.6 References


President’s Council of Advisors on Science and Technology (US) (2016). Report to the President, Forensic Science in Criminal Courts: Ensuring Scientific Validity of Feature-comparison Methods. Executive Office of the President of the United States, President’s Council of Advisors on Science and Technology.


CHAPTER 4. AN ANGULAR FEATURE OF HANDWRITING COMPONENTS AND EXTENDED BAYESIAN HIERARCHICAL MODELING FOR FORENSIC WRITER IDENTIFICATION

4.1 Abstract

This chapter presents an extended modeling framework for closed-set forensic writer identification. Documents are processed and parsed into a sequence of disjoint graphical structures. Each graph is given a cluster assignment based on a template developed in Chapters 2 and 3. The frequency at which graphs of each cluster type appear in writing, along with measurements taken on the small graphs, serve as data for analysis by Bayesian hierarchical model. We consider measurements that capture the slant of each graph and compare the measurements within cluster assignments to investigate their ability to increase discrimination between writers. Ultimately, we select an angular feature to describe the primary direction of a graph and incorporate a circular density function in the hierarchical modeling framework. An analysis is conducted to determine whether rotation angles for graphs within every cluster assignment are useful for predicting the writer of a questioned document. A final model is proposed and a variety of applications using three data sources are presented to demonstrate predictive performance of the model.

---

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement #70NANB15H176 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, University of California Irvine, and University of Virginia.
4.2 Introduction

In the last decade, statisticians and computer scientists have worked to apply pattern recognition techniques to compare handwritten documents for forensic evaluations. These algorithmic methods can assist trained Forensic Document Examiners (FDEs) as they perform their job (Leedham and Srihari, 2003). Software packages that have resulted from such research, such as the FLASH ID® (Sciometrics LLC, Chantilly, VA, USA) (Miller et al., 2017), WANDA (Franke et al., 2004), or CEDAR-FOX (Srihari et al., 2007) systems are undeniably useful and accurate tools. The FLASH ID® system (Sciometrics LLC, Chantilly, VA, USA) has an extensive set of detailed measurements that are extracted from each component of writing, leading to near perfect discrimination between writers in a closed set. With the exception of standard settings and document selection, the system runs automatically without the need for frequent examiner intervention for feature extraction or otherwise. However, the system can be expensive and the underlying algorithms are proprietary. WANDA and CEDAR-FOX provide options for interactive intervention by the FDE, such as annotations. These functions provide computer based assistance to the daily workflow of the examiner, but also the opportunity for FDE cognitive biases to influence the resulting decisions. In the spirit of reproducible and open-source research, we have built our own framework for closed-set writer identification tasks using statistically grounded methodology and publicly available datasets. In Chapters 2 and 3 we have laid the foundation for a document analysis pipeline that is transparent, model-based, and provides probabilistic outcomes for questioned writership within a closed-set of known writers. In this chapter, we build upon the framework by extracting features from handwriting that the FDE would consider during examination, with the intention of improving predictive accuracy and robustness.

The analysis pipeline we have developed begins with scanned handwritten documents, stored as png images at 72dpi. An R package called handwriter, currently hosted on GitHub, was developed at the Center for Statistics and Applications in Forensic Evidence (CSAFE) for document processing (Berry, 2019). Details of the package utility can be found in Chapter 2. The primary function in the package takes a png image as input and binarizes the pixels so that each is either black or white.
to isolate handwriting from background noise. Then, the black pixels are reduced to a pixel-wide skeleton of the writing which is decomposed into a series of smaller skeleton structures. It is these smaller structures that we consider as the smallest unit of handwriting. Each is considered as a graphical structure with nodes and edges.

The primary feature extracted from each graph in a document is a cluster assignment based on a template that was created and explained in Chapter 3. The template was generated through a $K$–means (Forgy, 1965; Lloyd, 1982) type algorithm, using novel definitions of center and distance to create 40 clusters. Each graph in a document is given a numerical cluster assignment between 1 and 40 corresponding to the cluster exemplar it is most similar to, according to the novel distance measure. Group assignment via the template facilitates group membership for graphs with similar major structure.

The modeling conducted in Chapters 2 and 3 relies only on the cluster assignment of the graphs in a document. In this chapter, we introduce features that capture the slant of a graph. The fundamental goals of the analysis remains the same as in our previous work. We aim to evaluate a writing sample with unknown origin against a closed-set of writers via a Bayesian hierarchical model that will now include additional measurements. Since graphs within the same cluster assignment are already known to have similar structures, we will compare the new features within cluster.

The rest of the paper is organized in two parts. The first, Section 4.3, is a narrative of the new features we extract from the graphs and their introduction to the hierarchical model. Ultimately, we choose one angular feature to capture the slant of a graph which is supported on the polar coordinate system, and a circular distribution will be incorporated into the modeling framework. For the developmental steps described in Section 4.3 we use the same set of training and testing data from the CSAFE database (Crawford et al., 2019a). A final model is defined in Section 4.3.5, and a prior sensitivity analysis is conducted.

In the second part of the paper, Section 4.4, we use the final model specification to demonstrate model performance on new datasets. Three applications are considered. The first extends the work done in Section 4.3 to other documents in the CSAFE database. We evaluate test documents that
were transcribed months after the training documents. In the second application we combine two data sources, the CSAFE and Computer Vision Lab (CVL) databases (Kleber et al., 2013), and estimate model parameters for a closed-set of 160 writers. Under this model fit, we evaluate a set of test documents for which the writers are contained in the closed-set. We also evaluate a small set of test documents for which the writer is not contained in the closed-set. The third application uses writing from the IAM Handwriting database in a sentence by sentence case (Marti and Bunke, 2002). For each writer in the closed-set, we fit the model using 16 sentences as training data. We evaluate on a decreasing number of test sentences to explore performance behavior on decreasing amounts of data.

4.3 Part I. Feature Extraction and Extended Modeling

4.3.1 Data

The data used throughout Section 4.3 were contributed by the 90 writers in the Center for Statistics and Applications in Forensic Evidence (CSAFE) database (Crawford et al., 2019a). Every participant in this data collection completed three sessions of writing, with a waiting period of at lease three weeks between sessions. At each session, participants were asked to complete a short survey, and transcribe three prompts, three times each (nine samples per writer, per session). The same three prompts were used in each session. The longest, the London Letter (Osborn, 1929), is a common handwriting exemplar. The next longest is a short passage from The Wonderful Wizard of Oz by L. F. Baum, illustrated by W.W. Denslow (1900), chosen for its more natural language than that of the London Letter. The third and shortest prompt is the phrase: “The early bird may get the worm, but the second mouse gets the cheese.”

The feature extraction and model development conducted in Section 4.3 will use three London Letters and one Wizard of Oz sample from the first data collection session of each participant in the CSAFE database. For each writer, we take the London Letters to be included in the training set, and a single Wizard of Oz sample to be part of the testing set. For a small number of the writers (9), we use only two London Letter samples in the training set, the rest (81) contribute all three
samples. Thus, we establish 261 training samples and 90 testing samples to be used throughout the section.

Documents in both the training and testing sets were processed by the handwriter package (Berry, 2019), and the resulting graphs are given cluster assignments using the clustering template results that were described in Chapter 3. On average, a training document is comprised of 376 graphs. The smallest number of graphs extracted from a training document was 298, and the highest was 459. In the testing set, the average number of graphs extracted from a document was 302 (minimum of 228 graphs, maximum of 376 graphs). Documents in the test set were, of course, left out of template creation and will not be used for model fitting. Other terminology used for the testing documents may include “questioned documents”, or “holdout documents”.

4.3.2 Reference Analysis

In Chapter 2, we determined the probability of a questioned document being written by each of the writers in a closed-set with a simple Bayesian hierarchical model. That model is restated here and parameters are estimated using the training and testing documents described in Section 4.3.1. This model specification will serve as the foundation on which we will build more complex models, and provides a baseline for comparison of results as we expand the model. This model relies on the frequency with which writers emit graphs assigned to each of the 40 clusters in their training documents. Saunders et al. (2011) also train classifiers based on categorical data extracted from handwriting.

Let \( k \) denote a cluster assignment \( k = 1, \ldots, K \); for this body of work we have \( K = 40 \). Let \( Y_{w(d)} \) be a \( K \)-dimensional vector that denotes the number of graphs assigned to each cluster for document \( d \), within known writer \( w = 1, \ldots, W \). Throughout Section 4.3 we will have \( W = 90 \) writers in the closed-set. Then, the model is defined as

\[
Y_{w(d)} | \pi_w \overset{ind}{\sim} Multinomial(\pi_w),
\]  

(4.1)
where $\pi_w$ is the 40-simplex that captures the cluster-fill probabilities for writer $w$. Let the prior for $\pi_w$ be the Dirichlet distribution

$$\pi_w | \gamma \sim \text{Dirichlet}(\gamma), \quad (4.2)$$

where $\gamma$ is a vector of length 40 that can be thought of as pseudo cluster counts, across all writers. We specify independent priors for elements of the $\gamma$ vector as

$$\gamma_k \sim \text{Gamma}(2, 0.25), \quad \forall k = 1, \ldots, K \quad (4.3)$$

and we use the $\text{Gamma}(\text{shape, rate})$ parameterization so that the expectation of $\gamma_k$ is defined as $E(\gamma_k) = \frac{\text{shape}}{\text{rate}}$. Although a formal prior sensitivity analysis is not conducted for this model, informal changes in final level prior settings do not impact predictive outcomes.

As was done in Chapters 2 and 3, we conduct a posterior evaluation of a single test document with data $Y_{w^*}$, by unknown writer, $w^*$. Consider a particular writer, $w'$, from the closed-set of writers. The goal is to estimate the probability that this particular writer was the author of the test document. Then, $\pi_{w'}^{(m)}$ is the $m$th MCMC sample of the $w'$ multinomial parameter vector, where $m = 1, \ldots, M$. Evaluate the multinomial likelihood under writer $w'$ at MCMC iteration $m$ as

$$q_{w'}^{(m)} = \text{Mult}(Y_{w^*}; \pi_{w'}^{(m)}). \quad (4.4)$$

The evaluation is conducted for all known writers in the closed-set and stated together as

$$q^{(m)} = \left[ q_{w_1}^{(m)}, q_{w_2}^{(m)}, q_{w_3}^{(m)}, \ldots, q_{w_W}^{(m)} \right], \quad (4.5)$$

and at MCMC iteration $m$, we recognize the writer with the largest likelihood evaluation with a “vote”,

$$q_{w'}^{(m)} = \begin{cases} 1 & \text{argmax}_{w_1, \ldots, w_W} q^{(m)} = w' \\ 0 & \text{o.w.} \end{cases} \quad (4.6)$$
Then, aggregating across all MCMC iterations, evaluate the proportion of MCMC samples in which each writer is most likely to be the true writer of the questioned document by

$$\bar{p} = \frac{1}{M} \left[ \sum_{m=1}^{M} v_{w_1}^{(m)}, \sum_{m=1}^{M} v_{w_2}^{(m)}, \sum_{m=1}^{M} v_{w_3}^{(m)}, \ldots, \sum_{m=1}^{M} v_{w_W}^{(m)} \right],$$

where $$\sum_{i=1}^{W} \bar{p}_{w_i} = 1$$. Given restriction to the closed-set, the $$\bar{p}$$ vector indicates the posterior probability of authorship for each author after accounting for variability in the parameters $$\pi_w$$ through the MCMC samples. The general process shown in Equations 4.4-4.7 will be used to evaluate the holdout documents under each of the models that we will propose. With each new model, the definition of $$q_{w'}^{(m)}$$ will change to reflect the predictive distribution, the rest remains the same.

The set of 90 testing documents are each evaluated in this way, and we obtain an estimated $$\bar{p}$$ vector for each. The estimated probability assignments are depicted in Figure 4.1, where each row of the grid represents the $$\bar{p}$$ vector for a test document. Under this reference analysis, 88.46% (83.33, 93.33) of all probability assignment goes to the true author. The remaining 11.54% is assigned to an incorrect author, and corresponds to off-diagonal elements in Figure 4.1.

The models that we introduce in the following sections include graph-based measurements, in addition to cluster assignments. The measurements are assigned their own data models and build upon the reference model specified in Equations 4.1-4.3.

### 4.3.3 Centroid-based Slope

Huber and Headrick (1999) consider “slant, slope, incline, or lateral expansion” to be a “discriminating element of writing”. We first take a centroid-based approach to caturing this feature of a graph. Centroid locations are calculated as the average position of all pixels in the graph, and are prone to the strong influence of outliers much like traditional arithmetic averages. We present it here as a stepping stone to the second approach, which yields more robust measurements and accurate results of the posterior questioned document analyses.

The centroid-based slope is found by locating the centroid of a graph, and splitting it into two sub-graphs consisting of the pixels to the right and left of the centroid, respectively. We then find
Figure 4.1  Posterior probability of writership as defined in Equations 4.4 - 4.7 for each of the 90 holdout documents, one row for each. True writers are labeled on the left-hand side. Columns are labeled by known writers. Cells are colored by elements of the $\bar{p}$ vectors for each holdout document. Thus, each row sums to one and stands alone.
Figure 4.2 The process of finding centroid-based slopes shown for two graphs. Moving from left to right, the images show a processed graph, the location of graph centroid, and finally the left and right hand centroids with a line connecting them. The slope of that line is taken as the centroid-based slope feature. Slopes for the examples shown here are 0.295 (top) and 1.02 (bottom).

the centroid of each sub-graph, and calculate the slope of the line connecting them. Figure 4.2 demonstrates this process.

One could imagine a scenario in which both the left and right hand centroids are changed dramatically by an ascending or descending stroke that happens to have its tail extend to one side or another of the centroid, altering the sub-graphs. For illustration, consider the graph in the bottom of Figure 4.2, but with a descender that curves, mirroring the descending stroke of a ‘g’. The addition of a few pixels to the right of the centroid and near the bottom of the graph would decrease the slope of the line, pulling it nearer to horizontal like that in the top graph.

Each graph’s cluster assignment is based on its structure and it is reasonable to assume that graphs within cluster assignment will have centroid-based slopes that are more similar. We incorporate slopes into the modeling framework through an average over graphs within each cluster.
For document $d$ by writer $w$, let $S_{w(d),k(j)}$ denote the centroid based slope of the $j^{th}$ graph that is assigned to cluster $k = 1, \ldots, 40$. Define the average slope for any particular cluster $k$, as

$$\bar{S}_{w(d),k} = \frac{1}{Y_{w(d),k}} \sum_{j} S_{w(d),k(j)}, \quad (4.8)$$

where $Y_{w(d),k}$ is the element of the multinomial data vector for the document that corresponds to cluster $k$, and gives the number of graphs that contribute to the average.

To incorporate the $\bar{S}_{w(d),k}$ data into the hierarchical model, consider each average slope as independent given writer and cluster. The full model specification is as follows, where the first three lines are repeated from the reference model in Equations 4.1-4.3.

$$Y_{w(d)}|\pi_{w} \sim \text{Multinomial}(\pi_{w})$$
$$\pi_{w} | \gamma \sim \text{Dirichlet}(\gamma)$$
$$\gamma_{k} \sim \text{Gamma}(2, 0.25)$$

Then, the centroid-based slope component is

$$\bar{S}_{w(d),k}|Y_{w(d)}, \mu_{w,k}, \sigma_{k}^{2} \sim N \left( \mu_{w,k}, \frac{\sigma_{k}^{2}}{Y_{w(d),k}} \right) \quad (4.9)$$

$$\mu_{w,k} | \mu_{k} \sim N(\mu_{k}, 3) \quad (4.10)$$

$$\mu_{k} \sim N(0, 3) \quad (4.11)$$

$$1/\sigma_{k}^{2} \sim \text{Gamma}(0.5, 0.1). \quad (4.12)$$

Notice that the structure in line 4.10 allows the location parameter for the distribution of each writer and cluster combination, $\mu_{w,k}$, to share information across writers.

The posterior analysis of questioned documents for this model is similar to that of the reference model. For a document with unknown writer $w^{*}$, we extract the multinomial data vector $Y_{w^{*}}$. In addition, we measure the centroid based slope of each graph, and take the average within cluster, $\bar{S}_{w^{*},k}$, according to Equation 4.8. Then, to evaluate the posterior probability of writership for a particular writer within the closed-set, $w'$, we consider the probability of writership at each MCMC iteration $m$ as
\[ q_{w'}^{(m)} = \text{Mult} \left( Y_{w'}^*; \pi_{w'}^{(m)} \right) \left[ \prod_{k=1}^{K} N \left( \bar{S}_{w',k}; \mu_{w',k}^{(m)}; \frac{\sigma_k^{2(m)}}{Y_{w',k}} \right) \right]. \quad (4.13) \]

We calculate the vector \( \bar{p} \) by using this definition of \( q_{w'}^{(m)} \) in Equations 4.4-4.7. Each of the 90 holdout documents is evaluated under this model. The result is that 91.48% (87.78, 94.44) of the posterior probability in the \( \bar{p} \) vectors is on the true writers of the corresponding test documents. Figure 4.3 provides more detail about the values in every entry of the \( \bar{p} \) vectors for each holdout document. This an improvement over the 88.46% that we observed in the reference analysis of Section 4.3.2, however we continue to see accumulation of off-diagonal probability, corresponding to probability assigned to writers who did not author the questioned documents.

### 4.3.4 Rotation Angles through Eigen-Decomposition

Next, we present an alternative to the centroid-based slope feature, which we call rotation angle. Like the slope, this feature is extracted from every graph, and again is used to quantify the inclination, lean, or slant of a graph. Through a principal component decomposition, we find the direction of a graph with the greatest variability, and capture the angle of rotation corresponding to that direction, relative to the horizontal.

For a graph with \( n_p \) pixels, consider each pixel as an \((x, y)\) point in Euclidean space. Then the collection of all pixels in the graph can be expressed as \((X, Y) = (x, y)_1, \ldots, (x, y)_{n_p} \). The principal components are calculated through an eigen-decomposition of the covariance matrix, \( \Sigma = \begin{bmatrix} \text{var}(X) & \text{cov}(X,Y) \\ \text{cov}(Y,X) & \text{var}(Y) \end{bmatrix} \). Since the first principal component can be identically defined in two opposite directions we simplify by enforcing that its direction lies in the first or second quadrant of the Euclidean space. The rotation angle of the graph is the angle between the unit vector \((0,1)\) and the directional vector of the first principal component. Call the angle \( \theta \in (0, \pi) \).

Figure 4.4 shows a collection of graphs from both Writer #1 and Writer #95’s training data along with the unit principal component vector on the upper half plane for each. Writer #1 tends to use a connected print style of writing with characters that do not have a prominent slant one
Figure 4.3  Results of the posterior analysis from the model with centroid-based slopes for each of the 90 holdout documents. One row for each document, ground truth writers are labeled on the left-hand side. Columns are labeled by known writers. Cells are colored by elements of the $\bar{p}$ vectors for each holdout document. Thus, each row sums to one and stands alone.
way or another. Writer #95 uses a more formal cursive writing style, and has a consistent slant to the right.

As with the centroid-based slope feature, there is evidence that rotation angle comparisons are more meaningful when considered within cluster. Cluster assignment captures the overall shape of a graph, so two graphs with the same assignment will naturally have rotation angles that are similar. For example, clusters that tend to contain graphs that are taller than they are wide will have members with rotation angles near $\pi/2$, simply because of their structure. What is meaningful, are rotation angle comparisons within that cluster.

For each writer, it is the distribution of rotation angles within each cluster that we aim to capture in the hierarchical model. We consider all clusters when fitting the model, but focus our examination of data and results on Clusters #16, #29, and #31. These clusters are shown in Figure 4.5. Clusters #16 and #29 both include members of the training data that tend to be wider than they are tall. The opposite is true for Cluster #31.

The distribution of rotation angles for three different writer/cluster combinations are shown in Figure 4.6. Data for the Figure are taken from all three training documents for each writer. On the left we show the distribution of rotation angles with traditional histograms. On the right, the angles are represented by a Nightingale rose diagram, or a radial histogram, in the polar coordinate system. See (Brasseur, 2005) for a dialogue regarding the history of the rose diagram.

### 4.3.4.1 A Naive Model for Rotation Angles

Naively, we can treat the rotation angle values as if they are linear on the interval $(0, \pi)$. This approach corresponds to the traditional histograms in Figure 4.6, ignoring the fundamental coordinate system in which the angular measurements should be considered. A more defensible model for the rotation angles is proposed in the next section.

Let $RA_{w,k,j} \in (0, \pi)$ be the rotation angle of the $j^{th}$ graph that is assigned to cluster $k$ across all training documents from writer $w$. Thus, $j$ is nested within the combination of $w$ and $k$, and $j = 0, 1, \ldots, J_{w,k}$. A beta sampling model will be used for the rotation angles for its flexible shape,
Figure 4.4  Writing samples from the training documents of two writers, six graphs from each are included. Graphs are accompanied by a unit vector pointing in the direction of the rotation angle, $\theta$, for the corresponding graph, given by the principal component analysis. Graphs are plotted to fill the space in the figure, so pixels may be shown in different sizes.
and support on an interval. To satisfy the support of a standard beta random variable, we will scale the rotation angles linearly by $\pi$. Define $SRA_{w,k,j} = \frac{RA_{w,k,j}}{\pi}$ so that $SRA_{w,k,j} \in (0,1)$, $\forall w = 1, \ldots, W$, $k = 1, \ldots, 40$, and $j = 1, \ldots, J_{w,k}$. The model is specified as

$$Y_{w(d)}|\pi_w \sim Multinomial(\pi_w)$$
$$\pi_w|\gamma \sim Dirichlet(\gamma)$$
$$\gamma_k \sim Gamma(2, 0.25)$$

where, again, these lines are the same as the reference model specification of Section 4.3.2. The scaled rotation angle data model is

$$SRA_{w,k,j}|\alpha_{w,k}, \beta_{w,k} \sim Beta(\alpha_{w,k}, \beta_{w,k})$$

(4.14)

and we reparameterize the beta distribution through the mean and variance as

$$\mu_{w,k} = \frac{\alpha_{w,k}}{\alpha_{w,k} + \beta_{w,k}}$$

(4.15)

$$\nu_{w,k} = \frac{\mu_{w,k}(1 - \mu_{w,k})}{\alpha_{w,k} + \beta_{w,k} + 1} = \frac{\mu_{w,k}(1 - \mu_{w,k})}{\kappa_{w,k} + 1}$$

(4.16)

$$\Longrightarrow \kappa_{w,k} = \alpha_{w,k} + \beta_{w,k}$$

(4.17)
Figure 4.6  Left, for all Writer/Cluster combinations: A traditional histogram of the rotation angles with 20 bins, a point for each observation is plotted on the x-axis. Right, for all Writer/Cluster combinations: A Nightingale Rose diagram displaying the rotation angles in 20 ‘petals’. A point for each observed rotation angle is plotted on the outermost ring of the diagram.
and assign the following prior distributions

\[ \mu_{w,k} \sim \text{Beta}(2, 2), \]  
\[ \kappa_{w,k} \sim \text{Gamma}(2, 1). \]

For a given writer and cluster, \( \mu_{w,k} \in (0, 1) \) is the mean of the sampling distribution for the rotation angles, expressed as a function of \( \alpha_{w,k} \) and \( \beta_{w,k} \). We place a relatively weak prior, \( \text{Beta}(2, 2) \), on the \( \mu_{w,k} \). Similarly, the variance \( \nu_{w,k} < \mu_{w,k}(1 - \mu_{w,k}) \in (0, 0.5^2) \) is defined. The value \( \kappa_{w,k} \) in the denominator is expressed as a function of \( \alpha_{w,k} \) and \( \beta_{w,k} \). With the relationship \( \kappa_{w,k} + 1 \propto \frac{1}{\nu_{w,k}} \), we observe that \( \kappa_{w,k} \) is a sort of concentration, or precision, parameter.

This approach appropriately captures the shape of rotation angle distributions for a number of writer/cluster combinations. For example, Figure 4.7 illustrates the fitted distributions via MCMC routine to capture the “slant” tendencies of Writer #95 in Cluster #29.

On the other hand, there are difficulties when it comes to the distributions for Writer #1, Cluster #16. The beta distribution is certainly able to take on the ‘U’ shape that is learned from the training data (see Figure 4.6), but by concentrating the mass of the density near 0 and 1. Figure 4.8 shows distributions that were fit to capture the behavior of Writer #1 rotation angles in Cluster #16 for a set of posterior samples. There is indeed a ‘U’ shape in each of these fit distributions, but constraints of the beta distribution make fitting a curve that behaves adequately for both the lower mode and upper mode impossible.

In the posterior analysis implied by this model, Cluster #16 graphs from a questioned document would obtain large likelihood values under known Writer #1 with rotation angles near the boundary of the support, the upper boundary in particular. The same graphs would obtain very small likelihood values under Writer #1 for scaled rotation angles falling in the middle of the support. This behavior supports the conclusion that the model does not adequately fit the data. If one were to sample from this model, the resulting datasets would not look like the observed data.

Figure 4.9 shows a histogram of a holdout document, for which we know that the true writer is Writer #85. There does not seem to be any clear unimodal signal in these rotation angles from
Figure 4.7  A plot depicting elements of the data model $SRA_{w,k,j} \sim \text{Beta}(\alpha_{w,k}, \beta_{w,k})$ with $w = 95$ and $k = 29$. Top: values of $\alpha_{95,29}^{(m)}$ and $\beta_{95,29}^{(m)}$ for 12 iterations, $m$, taken from the middle of the chain and used to plot the 12 corresponding beta densities. Bottom: the scaled rotation angles, $SRA_{w,k,j}, \forall j$, serving as data for parameter estimation.
Figure 4.8 A plot depicting elements of the data model $SRA_{w,k,j} \sim Beta(\alpha_{w,k}, \beta_{w,k})$ with $w = 1$ and $k = 16$. Top: values of $\alpha_{1,16}^{(m)}$ and $\beta_{1,16}^{(m)}$ for 12 iterations, $m$, taken from the middle of the chain and used to plot the 12 corresponding beta densities. Bottom: the scaled rotation angles, $SRA_{w,k,j}$, $\forall j$, serving as data for parameter estimation.
Cluster #16, and as a set, we might want these angles to evaluate the highest under a writer who learned a relatively flat beta curve during model fitting. That is not the case.

The rotation angles in Figure 4.9 are relatively spread out over the x-axis, and not necessarily reflective of the posterior beta distribution fit to the Writer #1 rotation angles depicted in Figure 4.8. However, when the questioned document is evaluated under the model, 0.66 probability is placed on Writer #1 via the $\bar{p}$ vector. The observations near the boundaries of the support lead to large likelihood evaluations under the beta distributions specified by parameters for Writer #1. In particular, the largest Writer #85 observation, which is 3.024, and scales to 0.996, has a great deal of leverage in the posterior evaluation because it falls into upper region of the ‘U’ shaped beta density. Figure 4.10 gives the full breakdown of average probability assignment for the questioned document by Writer #85, omitting labels for values less than 0.02.

When we conduct the posterior predictive analysis on the full holdout set, 92.08% (88.89, 95.56) of the total probability is assigned to the true writers. Much of the incorrect probability assignment can be attributed to behavior similar to that demonstrated with the holdout document for Writer #85 above. We will consider a more appropriate treatment of the rotation angles next.
4.3.4.2 A Circular Model for Rotation Angles

The issue we encountered with a linear treatment of rotation angles was failing to acknowledge the circular nature of the measurements. A more appropriate support for angles is the polar coordinate system. The polar coordinate system establishes that values near 0 and near \( \pi \), previously considered the least similar values, are close together. Ignoring the circular support for the angles did not affect graphs assigned to clusters that are taller than they are wide, such as Cluster #31, because the unimodal peak of the beta density was located in the middle of the support. The issue arises when the cluster describes graphs that are short and wide, like Cluster #29 or Cluster #16.

Figure 4.11(a.1) demonstrates why this occurs for clusters like #16 and #29. A few pixels added to the “corner” of a graph of this nature, can flip the rotation angle from near 0, to near \( \pi \). Although the invariance of the principal component direction keeps the angles securely between 0 and \( \pi \) during original feature extraction, the measurements are meaningful in a full circle. Therefore, we take 2 times each measurement to achieve support on the full polar coordinate space. Figures 4.11(a.2) and 4.11(b) demonstrate the transition to the fully wrapped coordinate system, and a probability density function that is appropriate for circular data.

Moving forward, let \( WRA_{w,k,j} \in [0, 2\pi) \) denote the wrapped rotation angle of the \( j^{th} \) graph that is assigned to cluster \( k \) across all training documents from writer \( w \). As before, \( j \) is nested within
(a.1) A collection of graphs from Writer #1’s training documents assigned to Cluster #16. To the right of each character graph is a depiction of the rotation angle measurement.

(b) Distributions for Writer #95, Cluster #29.

Figure 4.11 A demonstration of rotation angle behavior (a.1) and fully wrapped angles on the entire polar coordinate system (a.2) and (b). Left (a.2 and b): rose diagram with 20 petals for rotation angles on the upper half plane, as originally measured. Right (a.2 and b): a 20 petal rose diagram for fully wrapped rotation angles. A wrapped Cauchy density is shown in red with parameter values set to the MLEs for the corresponding data.
the combination of $w$ and $k$, and $j = 0, 1, \ldots, J_{w,k}$. The relationship to the original measurement is

\[ WRA_{w,k,j} = 2 \times RA_{w,k,j}. \] (4.20)

Common distributions defined on the polar coordinate space include the wrapped normal, the von Mises, and the wrapped Cauchy. The red circular densities shown in Figure 4.11 are wrapped Cauchy distributions with parameters set to the maximum likelihood estimators from the corresponding datasets. The wrapped Cauchy distribution is unimodal and symmetric, defined on the unit circle, and like many probability density functions, can be specified with a variety of parameterizations. We use the specification of Kent and Tyler (1988), who derived maximum likelihood estimates for the model parameters. The probability density function of the wrapped Cauchy distribution is

\[ f(\theta; \mu, \tau) = \frac{1}{2\pi} \frac{1 - \tau^2}{1 + \tau^2 - 2\tau \cos(\theta - \mu)} \quad \theta \in [0, 2\pi), \] (4.21)

where $\mu \in [0, 2\pi)$ is a location parameter, and $\tau \in [0, 1)$ controls the concentration of the model. Setting $\tau = 0$ yields the circular uniform density.

We anticipate an improvement in results when the circular distribution is used in place of the beta for rotation angles in the hierarchical model. The extreme tails of the ‘U’ shaped beta densities, that exhibited asymptotic behaviors in the previous model, are replaced with a solution that recognizes the location of a single mode. Incorporating circular data into a hierarchical model does not require any particularly special consideration. The first three steps are the same as Equations 4.1-4.3 of the reference model specification,

\[
Y_{w(d)}|\pi_w \overset{iid}{\sim} Multinomial(\pi_w) \\
\pi_w|\gamma \overset{iid}{\sim} Dirichlet(\gamma) \\
\gamma_k \overset{iid}{\sim} Gamma(2, 0.25).
\]

The data model for the fully wrapped rotation angles is

\[ WRA_{w,k,j}|\mu_{w,k}, \tau_{w,k} \overset{iid}{\sim} WrappedCauchy(\mu_{w,k}, \tau_{w,k}) \] (4.22)
and, for now, we put non-informative priors on the location and concentration parameters

\[ \mu_{w,k} \sim \text{Uniform}(0, 2\pi) \quad (4.23) \]
\[ \tau_{w,k} \sim \text{Uniform}(0, 1). \quad (4.24) \]

The wrapped Cauchy density is estimated independently for each writer and cluster combination. We will address a more complex model structure for the rotation angle priors in Section 4.3.5.

To evaluate a holdout document by unknown writer \( w^* \), under known writer \( w' \), with this model, we define \( q_{w'}^{(m)} \)

\[ q_{w'}^{(m)} = \text{Mult} \left( \mathbf{Y}_{w'}; \pi_{w'}^{(m)} \left[ \prod_{k=1}^{K} \prod_{j=1}^{J_{w^*,k}} \text{Wr. Cauchy} \left( \text{WRA}_{w^*,k,j}; \mu_{w',k}, \tau_{w',k} \right) \right] \right), \quad (4.25) \]

and the posterior probability of writership is summarized by \( \bar{p} \), which is found by using \( q_{w'}^{(m)} \) in Equations 4.5 and 4.7. The analysis is conducted on the full holdout set and 97.03% (94.44, 98.89) of the total probability is assigned to the ground truth writers of the test documents. This improves upon the 92.08% (88.89, 95.56) correct probability assignment from the beta distributed rotation angle model of Section 4.3.4.1, the 91.48% (87.78, 94.44) from the centroid-based slope model of Section 4.3.3, and 88.46% (83.33, 93.33) from the reference model of Section 4.3.2.

### 4.3.4.3 Cluster Selection for the Circular Model

A natural question is whether wrapped rotation angle comparisons from a reduced number of clusters can still meaningfully separate the questioned documents with respect to the known writers. In this section, an investigation of cluster information is integrated into the modeling framework through a mixture component.

Three ad-hoc ranking systems were considered to determine the importance of the rotation angles in each cluster to the model for predicting writership, one of which can be found in Appendix (Section 4.7). The methods all rely on an adaptation of the traditional within and between sums of squares used to assess clustering results. The first ranking system uses differences in rotation angles, in radians. The second method, included in the appendix, uses the mean resultant vector...
and Euclidean distances between unit vector terminal points. The third system is simply the sum of the resultant vector based ranking and sample size. All three ad-hoc rankings failed to properly capture the relationship between writer and rotation angles within a cluster, and little insight was to be gained from these approaches. They were abandoned for the model based approach described next.

Recall that a $\tau$ value of 0 in the wrapped Cauchy density function yields a circular uniform distribution. If the cluster $k$ rotation angles from writer $w$ are widely distributed around the support $(0, 2\pi)$, we expect the estimated concentration parameter of the wrapped Cauchy distribution $\tau_{w,k}$ to be near 0 to reflect the lack of peaked values. If all writers exhibit this behavior in cluster $k$, it would indicate that the rotation angles of graphs assigned to the cluster likely hold little or no information to aid in identifying the writer of a questioned document.

We can adapt our model to decide how informative each rotation angle distribution is by forcing each $\tau_{w,k}$ parameter to choose between a point mass at 0, or a larger value in the interval $(0, 1)$. When $\tau_{w,k}$ is chosen to be 0 its distribution was not informative enough to justify estimation and evaluation. To implement this model, we introduce a mixture component to the model specified in Equations 4.22-4.24. The multinomial piece from the reference analysis stays the same,

$$Y_{w(d)}|\pi_{w} \overset{ind}{\sim} Multinomial(\pi_{w})$$

$$\pi_{w}|\gamma \overset{iid}{\sim} Dirichlet(\gamma)$$

$$\gamma_{k} \overset{iid}{\sim} Gamma(2, 0.25).$$

The data model for wrapped rotation angles remains wrapped Cauchy,

$$WRA_{w,k,j}|\mu_{w,k}, \tau_{w,k} \overset{ind}{\sim} WrappedCauchy(\mu_{w,k}, \tau_{w,k})$$

and now we force the $\tau_{w,k}$ parameters to choose between a point mass at 0 (effectively diffusing the data distribution to a uniform), or a $Beta(4, 1)$ distribution that allows it to estimate a non-zero value in $(0, 1)$.

$$\tau_{w,k} \sim \phi_{w,k}Beta(4, 1) + (1 - \phi_{w,k})Dirac(0).$$
The final level priors are

\[
\mu_{w,k} \sim \text{Uniform}(0, 2\pi) \quad (4.28)
\]

\[
\phi_{w,k} \sim \text{Bernoulli}(\theta_k) \quad (4.29)
\]

\[
\theta_k \sim \text{Beta}(1, 1) \quad (4.30)
\]

The Beta(4, 1) has mass concentrated on the upper end of the support, as shown in Figure 4.12. By constraining the shape of the Beta component we avoid identifiability issues in the mixture. Results of this model are mildly sensitive to the choice of parameter values in this distribution.

The model was estimated using a single chain, run for 3000 iterations after burn-in. At each MCMC iteration, the Bernoulli distributed mixing parameter \( \phi_{w,k} \in \{0, 1\} \) acts as a density selector for an entire set of rotation angles in a cluster for a writer. Figure 4.13 summarizes the model outcomes for six writer and cluster combinations, along with the average of the estimated posterior density for each parameter \( \phi_{w,k} \), say \( \hat{\phi}_{w,k} = \frac{1}{3000} \sum_{m=1}^{3000} \phi_{w,k}^{(m)} \), which corresponds to the proportion of times that \( \tau_{w,k} \) was sampled from the Beta(4, 1) density rather than the point mass at 0 in the MCMC routine.

In the top left of Figure 4.13, the data from Writer #2 and Cluster #26 is abundant, but not concentrated in any particular direction. These data result in the lowest estimated \( \hat{\phi}_{w,k} \) observed in model fitting. During the MCMC routine, the parameter \( \phi_{2,36} \) took on the value of 1 during only one iteration. On the other extreme, the mixing parameter for Writer #92, Cluster #38 data was sampled as 1 at every iteration. This was not a unique scenario, and a variety of writer,
Figure 4.13 Rose diagrams depicting wrapped rotation angles from six different writer/cluster combinations. Each includes a corresponding wrapped Cauchy distribution shown in red. These are not estimated posterior distributions, rather, each fit using maximum likelihood estimation of the location and concentration parameters. In the title of each plot is the average value of $\bar{\phi}_{w,k}$ ($\bar{\phi}_{w,k} = \frac{1}{3000} \sum_{m=1}^{3000} \phi_{w,k}^{(m)}$), along with the writer and cluster to which the data were assigned.
Figure 4.14 For each cluster $k$ shown on the $x$-axis, a boxplot is constructed using 90 values of $\hat{\phi}_{w,k}$, one for each writer. The red $x$ shows the Monte Carlo estimate for the median of each $\theta_k$, from Equation 4.29.

Cluster combinations exhibit the same behavior. Figure 4.14 gives a summary of the distribution of estimated $\hat{\phi}_{w,k}$ in each cluster $k$. It also includes a summary for the value of $\theta_k$ for each cluster, which lends insight into the overall rate with which rotation angle sets in the cluster tend to select the wrapped Cauchy distribution over the uniform.

There are clusters that hold rotation angles with clear unimodal signal for a great majority of writers. These clusters are on the right side of Figure 4.14, where the $\hat{\phi}_{w,k}$ values are high for most writers, indicating that a wrapped Cauchy distribution is desirable to capture the shape of the wrapped rotation angle distributions.

The main goal of this exercise was to investigate whether any cluster would show a boxplot with all elements near zero. Rotation angles in such a cluster would be best modeled with a circular uniform density across all writers, effectively contributing a constant value to the likelihood evaluation in the predictive analyses. We would consider omitting such a cluster from the rotation
angle modeling, if one was observed. However, the boxplots in Figure 4.14 do not provide any evidence of such a cluster.

This approach is effective in capturing whether there is a strong location signal in the cluster for each writer, but does not indicate whether all of those signals are the same or not. It may be worthwhile to investigate clusters for which the writer distributions are all peaked at the same location. These would also provide very little information to the final analyses.

4.3.5 Final Model Specification

We proceed to specify a final model including all 40 clusters, as justified by the previous section. In this final iteration of model development, we introduce a sharing structure for the location parameter of the wrapped Cauchy data model. The following specification uses Roman letters as final level prior parameters. These final level parameters are the focus of a prior sensitivity analysis to follow.

The first three lines remain the same as the reference model in Equations 4.1-4.3, with the exception of the final level prior parameters left unspecified,

\[ Y_{w(d)} | \pi_w \sim \text{Multinomial}(\pi_w) \]
\[ \pi_w | \gamma \sim \text{Dirichlet}(\gamma) \]
\[ \gamma_k \sim \text{Gamma}(a, b). \]

The wrapped rotation angles are modeled with a wrapped Cauchy distribution, now with a sharing structure for the location parameters,

\[ WRA_{w,k,j} | \mu_{w,k}, \tau_{w,k} \sim \text{Wrapped Cauchy}(\mu_{w,k}, \tau_{w,k}) \] (4.31)
\[ \tau_{w,k} \sim \text{Beta}(c, d) \] (4.32)
\[ \mu_{w,k} \sim \text{Wrapped Cauchy}(\eta_k, \epsilon) \] (4.33)
\[ \eta_k \sim \text{Uniform}(0, 2\pi). \] (4.34)

To demonstrate robustness of the model to final level prior parameter selection, a sensitivity analysis was conducted with two goals in mind. The first goal is to investigate the effect on the final
Table 4.1  Results of the prior sensitivity analysis conducted for the final model specification. The first three lines correspond to varying levels of prior information. The last two lines include parameter settings that affect the concentration of the wrapped Cauchy densities in Equations 4.31 and 4.33. An analysis of the holdout documents was conducted for each set of parameter settings, the percentage of correct assignment is included in the final column with a corresponding 95% credible interval.

<table>
<thead>
<tr>
<th>Situation/Goal</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>True Writer $\bar{p}$ Assignment (%)</th>
<th>(95% credible interval)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non informative</td>
<td>1.1</td>
<td>0.1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>97.05 (94.44, 98.89)</td>
<td></td>
</tr>
<tr>
<td>Moderately informative</td>
<td>2</td>
<td>0.25</td>
<td>2</td>
<td>2</td>
<td>0.3</td>
<td>97.00 (94.44, 98.89)</td>
<td></td>
</tr>
<tr>
<td>Informative</td>
<td>2</td>
<td>0.8</td>
<td>5</td>
<td>5</td>
<td>0.6</td>
<td>96.74 (94.44, 98.89)</td>
<td></td>
</tr>
<tr>
<td>Diffuse Wrapped Cauchy Dstns</td>
<td>2</td>
<td>0.25</td>
<td>2</td>
<td>9</td>
<td>0.2</td>
<td>95.78 (93.33, 97.78)</td>
<td></td>
</tr>
<tr>
<td>Tight Wrapped Cauchy Dstns</td>
<td>2</td>
<td>0.25</td>
<td>9</td>
<td>2</td>
<td>0.8</td>
<td>96.71 (94.44, 98.89)</td>
<td></td>
</tr>
</tbody>
</table>

posterior predictive analysis from varying the amount of information proffered by $a - e$. Parameters $a$ and $b$ express no information through a large variance. Parameters $c$ and $d$ control the beta distribution in line 4.32. To investigate the level of information introduced by these parameters it is reasonable to take $c = d$ in this first prior sensitivity stage to ensure that the center of the prior density remains at 0.5, and increase the values to communicate higher levels of information. The second objective of the prior sensitivity analysis is to determine if the location specified by the beta parameters $c$ and $d$ impacts the posterior predictive accuracy. We first let $c = 2$ and $d = 9$ to concentrate the mass of the beta density near the lower end of the support. This encourages small concentration parameters, and thus a diffuse wrapped Cauchy distribution in line 4.31. Parameter $e$ operates in similar fashion for the wrapped Cauchy of line 4.33, where small values are associated with non-informative specification and a diffuse wrapped Cauchy distribution.

Results of the analysis are included in Table 4.1, and provide evidence that results are relatively robust to prior parameter specification. This is particularly true for the information-based scenarios at the top of the table, which include reasonable values that could be chosen by another researcher who is familiar with the work. The two scenarios in the bottom of the table were included as more of an exercise, as these are not necessarily logical values to choose for parameters $c, d$ or $e$ apriori.
We will use the moderately informative parameter values as the final version of the hierarchical model. Posterior predictive evaluations are done as specified in Equation 4.25, and results of evaluation on the closed-set are 96.97% accurate assignment, as is stated in the table.

The proposed method was developed to produce an indication of which closed-set writers are most similar to the writer of the questioned document. If we wish to make a categorical decision regarding writership for each of the questioned documents, as is common in practice, a threshold would need to be chosen. The threshold would be used to dichotomize the space of $\tilde{p}$, where an identification would be declared for a writer if the corresponding entry of the vector exceeded the threshold. The false positive and true positive rates of such an identification task are calculated across all possible cutoff values. Identification results for this test set, along with others from the CSAFE database are shown via ROC curves in Section 4.4.1.

4.3.6 Part I, Other Considerations

Other features have been considered for inclusion in the model. A noteworthy measurement is one taken from each loop in a graph. Loops are defined as a sequence of pixels with beginning and terminal nodes near each other, with no other structure interfering. We locate the centroid of each loop, and calculate the line that travels through the centroid having the longest distance. Then we find the perpendicular line that also passes through the centroid. Examples of these can be found in Figure 4.15.

The log ratio of length to width of each loop was incorporated into the hierarchical model using a gamma distribution for the data. These data provided only marginal improvement over both the reference model, and the final model, and have been excluded from further consideration.

4.4 Part II. Model Testing and Applications

We now implement the model in Equations 4.31-4.34 on three applications. The nature of the modeling framework requires parameters to be re-estimated for each new training dataset, primarily because there is a parameter vector estimated for each writer in the closed-set.
The first application, however, does not require re-estimation of the model, because we will use the parameter estimates from Section 4.3.5 to evaluate seven additional test sets from the CSAFE writers that have not been considered yet. In Section 4.3 we assessed writership of Wizard of Oz prompts from the first data collection session. In Section 4.4.1 we explore the use of prompts from the second and third data collection sessions as test sets. Through these additional test sets we investigate the effect of time between data collection sessions as well as the effect that different writing prompt content may have on predictive results.

In the second application, we estimate model parameters for a large closed-set comprised of 160 writers, 80 from the CSAFE database and 80 from the Computer Vision Lab (CVL) database (Kleber et al., 2013). As before, we evaluate a test set of documents that we know were authored by writers included in the closed training set. In addition, we examine 14 test documents authored by writers that are not included in the closed-set. Throughout this application we investigate robustness of model behavior when writers from two different data sources are used in the closed set.
The third application is designed to estimate the model parameters using a large amount of training data, but then test its performance on decreasing amounts of holdout data. The question is whether the modeling approach we propose can accurately identify a writer when the questioned document includes only a few graphs, as might be the case with a one-sentence note. The data for this application is a subset of sentence samples from the IAM handwriting database (Marti and Bunke, 2002). We wish to explore whether results degrade as the test writing becomes shorter.

4.4.1 Application 1 (CSAFE data): Extended testing sets

Participants in the CSAFE database provided handwritten samples over three data collection sessions. The sessions took place at least three weeks apart, and the same three prompts were transcribed at each session (Crawford et al., 2019a). In the model development stages of Section 4.3, a variety of hierarchical models were estimated using the London Letter samples from the first data collection session of all 90 writers and one Wizard of Oz sample from each writer served as the testing set.

In this section, we use the final model specification defined in Section 4.3.5, and the parameter estimates used in the final analysis. These are the estimated $\pi_w$ vectors for each writer, the estimated $\mu_{w,k}$ and $\tau_{w,k}$ for each writer and cluster combination, and the $\eta_k$ for each cluster (see Equations 4.31 - 4.34).

For testing we expand our use of the CSAFE database, building testing sets from the previously unexplored prompts and data collection sessions. As before, a holdout set is comprised of one document from each of the 90 writers. Documents yet to be considered for evaluation are the London Letter (LND) and Wizard of Oz (WOZ) samples from the second and third sessions of data collection, and the short phrase (PHR) samples from all three sessions. Test sets were constructed for each of these seven remaining prompt/session combinations. Each of the test sets was evaluated under the model using the posterior analysis defined by Equation 4.25. The percentage of $\bar{p}$ elements assigned to the true writer in each case is provided in Table 4.2.
Table 4.2 Using the final model from Section 4.3.5, writing samples from the seven prompt/session combinations that have not yet been explored are evaluated. * The eighth result shown here is the test set used in Section 4.3.5, same information is provided in Table 4.1. The median, minimum, and maximum number of graphs extracted from the documents in each test set are provided. The percentage of true writer assignment according to $\bar{p}$ is included in the rightmost column. ** Training data used to estimate model parameters, see Section 4.3.5.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Median # of Graphs (Min, Max)</th>
<th>% True Writer $\bar{p}$ Assignment (95% credible interval)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prompt</td>
<td>Session</td>
<td>Median # of Graphs (Min, Max)</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>LND</td>
<td>1</td>
<td>375 (298, 459)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>377 (282, 473)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>375 (277, 471)</td>
</tr>
<tr>
<td>WOZ</td>
<td>1</td>
<td>298 (228, 376)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>298 (226, 392)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>298 (207, 386)</td>
</tr>
<tr>
<td>PHR</td>
<td>1</td>
<td>58 (46, 74)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>58 (43, 76)</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>57 (43, 73)</td>
</tr>
</tbody>
</table>

We observe two clear trends in the results. First, the test sets with longer prompts outperform test sets with shorter prompts with respect to the percent of $\bar{p}$ assigned to the true writers of the questioned documents. In addition to a decrease in the number of graphs, and therefore measurements, associated with shorter prompts, the degradation of results for the PHR samples may be exasperated by the fact that 40 clusters are used for grouping graphs. In particular, the issue comes with the cluster count data vectors, $Y_{w(d)}$, of length 40 that are evaluated using the multinomial distribution. We do not expect a document containing around 50 graphs to robustly represent the rate at which graphs of each cluster type are present for any given writer. It is also important to recognize that the model parameters were estimated based on repetitions of the same London Letter prompt, and may be influenced by the restricted content in the training dataset. In addition to the shorter length of the WOZ and PHR prompts, it is likely that the poorer results are caused by an imposed lack of robustness in the model for evaluation of different content.
The second trend we observe is evidence of degrading results for test sets that were transcribed in later sessions. This is especially apparent for the WOZ and PHR prompts. The London Letter prompts do no exhibit this behavior, and we suspect that this can be attributed to the fact that the content in these test sets are identical to that in the training set, and one expects the model to perform well under these conditions.

ROC curves for a simplified identification decision making task are provided in Figure 4.16 for each of the holdout sets in the table. The curves provide insight into error rates at a variety of thresholds. The identification task we face is not a two-class classification problem in nature, but is presented as such in the figure. This is accomplished by transitioning results to a binary outcome for each holdout document by considering a true positive identification of the true writer if the amount of $\tilde{p}$ in the corresponding index is greater than the cut-off value. Simplification of the 90-class problem comes with our definition of a false positive identification, which is flagged if the amount of $\tilde{p}$ assignment to the true writer is less than the cut-off value. This effectively puts the true writer in one outcome category, and lumps the results for all 89 other writers in the closed-set together in another. The dichotomization is done using a grid of cut-off values in the interval (0,1), and for each holdout set the rate of true positives to false positives at each cut-off are presented by a line in Figure 4.16.

The figure illustrates the behavior that was observed in Table 4.2. Prompts with longer content outperform prompts with shorter content, and samples taken nearer in time to the first session outperform samples taken later in time. In traditional two-class problems, the $x = y$ line represents random guessing. For this application, because the second outcome is considered as the aggregation of 89 outcomes, the identity line represents 0.5 probability assigned to the ground truth writers of the holdout documents, and 0.5 probability assigned to anyone else in the closed-set. This is not necessarily an undesirable outcome in this application, especially if the latter probability is spread widely across incorrect writers, and if there is little test data to be evaluated. Keep in mind that the goal of such an analysis is to provide a probabilistic evaluation of potential authors in a closed-set, to be further examined by a trained Forensic Document Examiner.
Figure 4.16 ROC curves for all eight test sets evaluated under the final model estimated for the CSAFE session #1 LND prompts. For a grid of cut-off values in (0,1), a true positive is declared if the value in the index of $\bar{p}$ corresponding to the ground truth writer is greater than the cut-off. A false positive is declared if the value in that index is less than the cut-off.
4.4.2 Application 2 (CSAFE + CVL data): Combining two data sources and an out-of-set evaluation

The CVL database is comprised of samples from 310 unique writers, 27 of whom transcribed 7 prompts (Kleber et al., 2013). These writers were featured in the clustering algorithm development of Chapter 2, and used in the final template creation outlined in Chapter 3. They are omitted from these analyses. The remaining 283 writers each transcribed the same 5 writing prompts, one in German and the rest in English (Kleber et al., 2013). For consistency, we will omit the German text. From this group of writers, 80 were selected for inclusion in the closed-set for this application. For each writer, samples from three of the four prompts written in English were used for model fitting, the remaining prompt was held out for testing.

The CVL data are used in conjunction with documents from 80 CSAFE writers, leaving some of the writers out of the closed-set during model fitting so that we can use their documents later in an out-of-set analysis. For the 80 CSAFE writers, London Letter samples from the second data collection session are used for model fitting, and a single Wizard of Oz sample is held out for testing. Thus, the full training data set is comprised of documents from 160 writers and \( W = 160 \) so that \( w = 1, \ldots, 160 \) for use in the equations established in Section 4.3. The hierarchical model, as described in Section 4.3.5, is estimated using the new data via MCMC with a single chain of length 4000.

The first test set to be evaluated under the model is comprised of one document from each of the 160 writers. A \( \bar{p} \) vector of length 160 is calculated for each document according to the posterior analysis defined by Equation 4.25, summarising the posterior distribution of probability for each writer in the closed-set. Each of the \( \bar{p} \) vectors are depicted as rows in Figure 4.17. There are six questioned document evaluations for which less than 0.9 probability was assigned to the true writer, and four for which less than 0.5 was assigned to the true writer. The rows corresponding to those four evaluations are labeled in the Figure 4.17. Overall, 97.34% (95.62, 98.75) of the assignment via the \( \bar{p} \) vectors is on the true writers.
Figure 4.17 Results of the posterior analysis under the final model, estimated for 80 CVL and 80 CSafe writers, one holdout document for each writer is evaluated. True writers of each holdout document are labeled on the rows. Columns are labeled by known writers, and cells are colored by corresponding elements of the $\hat{p}$ vectors. Vector elements $< 0.01$ are removed from the plot, showing the gridlines behind. Labels are ordered first by data sources, with CVL writers on the left hand side of the $x$–axis, and CSafe writers on the right, then by writer number. Four holdout documents with the smallest ($< 0.5$) correct probability assignment are labelled.
The second test set we will evaluate under the model includes documents that did not originate from any of the writers in the closed-set. Seven documents were taken from the CVL database having the same prompt as the holdout documents from the CVL writers in the previous analysis. Seven WOZ samples were taken from out-of-set writers in the CSAFE database, making a total of fourteen out-of-set documents to be evaluated. Results of the posterior predictive analysis are given in Figure 4.18. We observe $\bar{p}$ probability assignments that are relatively diffuse in comparison to evaluations for which the test set document writers were in the closed-set.

The hierarchical modeling and evaluation techniques presented in this paper are designed to assign total probability inside the closed-set of known writers for which the model was estimated. While the distinction has not necessarily been made explicit in previous sections, the probability we assign is conditioned on the true writer of the questioned document being contained within the closed-set. There is no opportunity to conclude that the questioned author does not exist in the closed-set. It is up to the Questioned Document Examiner to determine whether one of the writers deemed most probable by the model is the creator of the test document. Thus, the proper interpretation of the rows in Figure 4.1 is: 

"under the assumption that the writer of the questioned document is contained in the closed-set, writer $j$ is most likely to be the writer of the test document with estimated posterior probability $\bar{p}_j$, " where $\bar{p}_j$ is the corresponding element of the vector $\bar{p}$, and $j = 1, \ldots, 160$.

We observe only one row in the figure (1 of 14) for which the amount of probability assigned is greater than 0.9. Recall that all but six evaluations (154 of 160) shown in Figure 4.17 have an element of the $\bar{p}$ vector that is greater than 0.9. This gives indication that for most of the out-of-set test documents there was not one single writer that the model deemed “similar” to the questioned writer. With the one exception, results appear to indicate that none of the writers in the closed-set can be confidently determined to be the writer of any of the out-of-set questioned documents. This is the result we were hoping to obtain.

After evaluation of both test sets (160 questioned documents with true writers in the closed set and 14 questioned documents with true writers out-of-set), there does not appear to be a pattern in
Figure 4.18  Results of the posterior analysis under the final model, estimator 80 CVL and 80 CSAFE writers in the closed-set. Fourteen documents generated by writers outside the closed-set are evaluated. True writers of each holdout document are labeled on the rows. Columns are labeled by known writers, and cells are colored by corresponding elements of the $\mathbf{\bar{p}}$ vectors. Vector elements $< 0.01$ are removed from the plot, showing the gridlines behind. All $\mathbf{\bar{p}}$ elements $\geq 0.1$ are labeled.
the results to indicate that the model considers writers from the same data source as more similar. If this were the case, the model would be picking up on elements outside of the biometric process we wish to capture, such as writing utensil, paper type, or scanning protocol.

4.4.3 Application 3 (IAM data): Test on decreasing number of sentences

The IAM Handwriting Database, developed by the Research Group on Computer Vision and Artificial Intelligence at the University of Bern (Marti and Bunke, 2002), contains handwriting samples from 657 individuals. The forms that participants transcribed were built using content from the Lancaster-Oslo/Bergen (LOB) Corpus (Johansson et al., 1978). After scanning, the writing was segmented into lines, sentences, and words (Zimmermann and Bunke, 2002). For the analysis conducted in this section, we use the sentence-parsed data to investigate the performance of a model that is trained on 16 sentences for each writer in the closed-set, and tested on a decreasing number of sentences.

To build the analysis dataset, the full sentence repository is filtered based on two criteria. First, conscious of the fact that sentences are parsed based on form content, we exclude extremely short sentences (such as “No.” or “They were not.”) from consideration. Eligible sentences are those comprised of 30 or more graphs after decomposition by the handwriter R package. Second, we include writers who provided a sufficient number of eligible sentences. Requiring at least 20 eligible sentences per writer results in 45 viable writers for analysis. If a writer has an abundance of eligible sentences, we select 20 at random. The dataset used for analysis is therefore comprised of 900 sentences from 45 different writers. For this application, \( w = 1, \ldots, W (= 45) \).

We divide the sentences into training and testing sets for model development and evaluation. For each writer, the sentences were ordered by length, with respect to the number of graphs extracted from each. Then, the 16 longest sentences for each writer were assigned to the training set, and the four shortest were held out for testing. For this new closed-set of writers, the model defined in Section 4.3.5 was fit using the known genuine samples in the training set.
Table 4.3 Four amounts of testing data (in number of sentences) are evaluated. The median, minimum, and maximum number of graphs extracted from each test set are provided. The percentage of true writer assignment according to $\bar{p}$ is included in the rightmost column with a 95% credible set.

<table>
<thead>
<tr>
<th># Test Sentences</th>
<th>Median # of Graphs (Min, Max)</th>
<th>% True Writer $\bar{p}$ Assignment (95% credible interval)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>167 (131, 297)</td>
<td>$&gt; 99.99^1$</td>
</tr>
<tr>
<td>3</td>
<td>118 (95, 216)</td>
<td>99.79 (97.78, 100.00)</td>
</tr>
<tr>
<td>2</td>
<td>73 (61, 139)</td>
<td>98.82 (95.56, 100.00)</td>
</tr>
<tr>
<td>1</td>
<td>34 (30, 67)</td>
<td>91.50 (84.44, 95.55)</td>
</tr>
</tbody>
</table>

As noted in Section 4.4.1, the shape of a 40-dimensional vector that serves as an observation for the multinomial data model may not be properly represented by a document containing a small number of graphs. In order to properly learn the multinomial parameters during model fitting, the 16 training sentences for each writer are taken in four groups, to make four “document” length observations on par with the writing samples from both the CSAFE and CVL databases. This effectively constructs four cluster-frequency data vectors, $Y_{w(d)}$, for each writer, and the posterior distributions for the model parameters are estimated via MCMC using a single chain of length 4000.

We begin the test set evaluation by taking all four of the holdout sentences together as a single test sample for each writer. Each test sample is evaluated against the writers in the closed-set according to Equation 4.25, and elements of the resulting $\bar{p}$ vectors are provided in the top left plot of Figure 4.19. Then, we remove the longest of the four holdout sentences and repeat the evaluation process using the shortest three together as a test sample for each writer. The same is done for the two shortest sentences. Finally, the single shortest sentence from the analysis dataset is evaluated for each writer.

All four evaluations are depicted in Figure 4.19. The percentage of $\bar{p}$ probability assignment to the true writers for each test set is provided in Table 4.3 along with variability estimates. As the amount of testing data declines so does the accuracy of model predictions. An especially noticeable

---

$^1$ One mis-vote occurred in one iteration of the MCMC routine.
Figure 4.19  Results of the posterior analysis under the final model, estimated using 16 sentences for each of the 45 writers in the closed-set taken from the IAM database. The four plots are shown on decreasing amounts of test data. True writers of each holdout document are labeled on the rows. Columns are labeled by known writers, and cells are colored by corresponding elements of the $\bar{p}$ vectors. Vector elements $< 0.01$ are removed from the plot, showing the gridlines behind. Elements of the $\bar{p}$ vector corresponding to the ground truth writer are labeled only if they are less than 0.9, and all other elements $\geq 0.1$ are labeled.
Figure 4.20  Distributions of the number of graphs in the holdout sentences across all 45 writers in the IAM closed-set. To be eligible for the analysis dataset, sentences were required to contain a minimum of 30 graphs after decomposition by the handwriter R package.

degradation occurs when we drop from two sentences down to the single shortest test sentence. Aside from the obvious decrease in available testing information, with only one test sentence we move into the space where representation of the 40–dimensional multinomial data vector is likely poor.

Figure 4.20 provides insight into the size of the test documents with respect to the number of graphs in each. We notice that, for most of the writers, the shortest test sentence is comprised of fewer than 40 graphs. Using these as test sentences provides little hope of a robust evaluation based on the cluster count data. Figure 4.21 provides the shortest test sentences for three writers. This gives the reader some insight into how much writing is required for a robust evaluation under the models we estimate.
4.5 Discussion

The analyses in this chapter build upon foundations established in Chapters 2 and 3. In this chapter, we extended the hierarchical model for estimation of writership probabilities by incorporating features extracted from the graphs into the model. These graph-based features were compared within cluster assignment, following the expectation that graphs with the same cluster assignment will be structurally similar. Two such features were presented in detail in Section 4.3. First, the centroid-based slopes for each graph were incorporated, providing general improvement over the use of cluster count data alone, and indicating that a feature aimed at capturing the slant or lean of a graph would be a valuable addition to cluster assignment alone. Unfortunately, the ad-hoc feature demonstrated inconsistent measurement tendencies.

We moved to principal component decomposition to capture the direction with which the pixels of a graph have the largest amount of variability. The first principal component angles for each graph were naively treated as linear measurements before being considered on the polar coordinate system. The wrapped rotation angles were incorporated into the hierarchical model using the circular von Mises distribution. We conducted a model based investigation of whether it would be appropriate to exclude rotation angles for graphs of any particular cluster assignment. The analysis
provided evidence that the feature was important in distinguishing writers within all 40 clusters, and in the final model we included the angles computed for graphs in every cluster.

In Section 4.4 we implemented the hierarchical model with wrapped rotation angles to three different datasets. Each application served a different purpose to stress and study the behaviour of the model. First, using previously unused CSAFE data we observed a degradation of model accuracy on test documents that were transcribed later in time than the training set, as well as on short test writing samples. The second example considered a closed-set of writers that was almost twice as large as the set described in 3. We evaluated a single holdout document for each of the 160 writers in the closed-set, and results were favorable. Using the parameter estimates obtained for these 160 writers, we also evaluated 14 documents by writers that were not included in the closed-set. The final application incorporated data into analysis on a sentence by sentence basis. For 45 writers, model parameters were estimated using 16 training sentences. Testing was conducted using a decreasing number of sentences. Results were noticeably worse when we moved from two test sentences down to a single test sentence.

In general, we observed especially poor results when evaluating test documents with fewer than 70 graphs. This is likely due to the fact that for several of the 40 pre-determined clusters, we observe a frequency of zero graphs in these cases. Consequently, we cannot obtain reliable estimates of the elements of the vector $\pi$ of probabilities. It will be left to future work to explore model fitting and performance for data with cluster assignments from a smaller template, with 20 or fewer clusters.

Results presented here indicate that the analytical framework we propose may prove useful in actual case work. By evaluating a test document using our modeling framework, Forensic Document Examiners can obtain a preliminary, objective and quantitative assessment of the probability of writership of a questioned document when the true writer is included in the pool of potential writers. These initial assessments may be accurate in those cases where the questioned writing has sufficient length.
4.6 References


### 4.7 Appendix to Chapter 4: Ad-Hoc Importance Ordering of Clusters

Three ad-hoc methods for measuring the importance of comparing rotation angles within all 40 clusters were mentioned in Section 4.3.4.3, and one is outlined here. The approach is an adaptation of the common within and between sums of squares that is traditionally used for evaluation of clustering results (Forgy, 1965; Lloyd, 1982). For a cluster, \( k \), we aim to capture the variability of wrapped rotation angles within each writer. This is measured with a *within-writer sum of squares* for each cluster \( k \). Tightly packed rotation angles within a writer provide a clear signal for the writer, and are desirable, but alone are not enough. We also seek clusters in which the signal for each writer is different. We also aim to measure the degree to which each writers’ signal differs from the others. It is desirable to observe concentrations of wrapped rotation angles that are different from writer to writer, so the writers are easily identified. A *between-writer sum of squares* value aims to capture these behaviors.

The traditional total sums of squares decomposition for cluster evaluations relies on three major components: observations, group means, and the grand mean. We define analogous components for the wrapped rotation angles as follows, with fixed \( k \), respecting the angular nature of the data.
1. **Observations, representing a single** \( WRA_{w,k,j} \). Define the point \((x,y)_{w,k,j}\) as the terminal point of the unit vector with angle, \( WRA_{w,k,j} \), that is
\[
(x,y)_{w,k,j} = \left( \sin(WRA_{w,k,j}), \cos(WRA_{w,k,j}) \right). 
\]
(4.35)

The rose diagrams on the right side of Figures 4.11(a.2) and 4.11(b) show these points plotted as black dots on the unit circle.

2. **Group/writer mean, representing a single writer.** For each writer \( w \), the mean resultant vector \( \bar{\tau}_{w,k} \) represents the overall tendency of the writer. Define \((x,y)_{w,k,\bullet}\) as the terminal point of the mean resultant for writer \( w \).

3. **Grand mean, representing all writers.** Similarly, we use a mean resultant vector to represent all observations in the cluster, ignorant of their writer. Define \((x,y)_{\bullet,k,\bullet}\) as the terminal point of the mean resultant vector from all \( WRA_{w,k,j} \) observations in cluster \( k \).

Now, for a fixed cluster, \( k \), we capture the variability within writer by summing over the squared Euclidean distance from the corresponding observations to the writer mean. This is done for every writer in the cluster and aggregated to be the within-writer sum of squares for cluster \( k \), \( WSS_k \).

\[
WSS_k = \sum_{w=1}^{W} \sum_{j=1}^{J_{w,k}} \left| \left| (x,y)_{w,k,j}, (x,y)_{w,k,\bullet} \right| \right|^2 
\]
(4.36)

where \( \left| \left| \cdot, \cdot \right| \right| \) represents usual Euclidean distance (\( L_2 \) norm).

We capture the variability between writers (between-writer sum of squares, \( BSS_k \)) by summing over the squared Euclidean distance from each writer mean to the grand mean. Again, this is done for every writer in the cluster and each contribution is scaled by the number of individual measurements that contributed to each writer-wise calculation, \( J_{w,k} \). The calculation is

\[
BSS_k = \sum_{w=1}^{W} J_{w,k} \left| \left| (x,y)_{w,k,\bullet}, (x,y)_{\bullet,k,\bullet} \right| \right|^2. 
\]
(4.37)

For a cluster \( k \), it is desirable to have small within-writer variability, \( WSS_k \), and high between-writer variability, \( BSS_k \), and therefore, we want the ratio,

\[
RSS_k = \frac{BSS_k}{WSS_k} 
\]
(4.38)
to be large. A large $RSS_k$ indicates that cluster $k$ may have potential to separate well between writers through wrapped rotation angle measurements.

One consideration must be taken for writers that only have one or two graphs assigned to cluster $k$. In this situation, the observation level component $(x,y)_{w,k,j}$, and writer level component $(x,y)_{w,k,*}$, are equal, and the associated contribution in Equation 4.36 will be zero. This behavior expresses an artificial precision and consistency, when in fact, the case that the writer even produces a graph of that cluster is rare. The same scenario provides a very large contribution to the calculation of $BSS_k$. Then, the calculation of $RSS_k$ will tend to favor clusters for which a number of writers contribute very few rotation angles. To adjust for this behavior, in each cluster $k$, exclude writers from the calculations who contribute fewer than 10 graphs. Then the adjusted calculations become

\[
WSS^\text{adj}_k = \sum_{w \in J_{w,k} > 9} \sum_{j=1}^{J_{w,k}} \left\| (x,y)_{w,k,j}, (x,y)_{w,k,*} \right\|^2 \tag{4.39}
\]

\[
BSS^\text{adj}_k = \sum_{w \in J_{w,k} > 9} J_{w,k} \left\| (x,y)_{w,k,*}, (x,y)_{*,k,*} \right\|^2 \tag{4.40}
\]

\[
RSS^\text{adj}_k = \frac{BSS^\text{adj}_k}{WSS^\text{adj}_k}. \tag{4.41}
\]

We calculate $RSS^\text{adj}_k$ for each cluster. Twenty clusters with the highest values are shown in Table 4.4.

<table>
<thead>
<tr>
<th>Cluster # ($k$)</th>
<th>1</th>
<th>14</th>
<th>23</th>
<th>10</th>
<th>4</th>
<th>21</th>
<th>37</th>
<th>25</th>
<th>24</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop. rank</td>
<td>37</td>
<td>39</td>
<td>38</td>
<td>32</td>
<td>29</td>
<td>34</td>
<td>18</td>
<td>30</td>
<td>33</td>
<td>35</td>
</tr>
<tr>
<td>$RSS^\text{adj}$</td>
<td>0.89</td>
<td>0.59</td>
<td>0.54</td>
<td>0.42</td>
<td>0.40</td>
<td>0.40</td>
<td>0.39</td>
<td>0.38</td>
<td>0.37</td>
<td>0.36</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cluster # ($k$)</th>
<th>19</th>
<th>22</th>
<th>6</th>
<th>17</th>
<th>39</th>
<th>3</th>
<th>27</th>
<th>32</th>
<th>29</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pop. rank</td>
<td>20</td>
<td>19</td>
<td>28</td>
<td>40</td>
<td>7</td>
<td>17</td>
<td>5</td>
<td>1</td>
<td>12</td>
<td>16</td>
</tr>
<tr>
<td>$RSS^\text{adj}$</td>
<td>0.35</td>
<td>0.33</td>
<td>0.33</td>
<td>0.32</td>
<td>0.32</td>
<td>0.31</td>
<td>0.31</td>
<td>0.30</td>
<td>0.28</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Table 4.4 Twenty clusters with the highest $RSS^\text{adj}$ values. Cluster #s are arbitrary labels from the clustering algorithm. Pop. rank shows the clusters popularity ranking among the training data, Cluster #32 is assigned to the most graphs and Cluster #17 the fewest.
Model performance is evaluated on subsets of clusters based on their value of $RSS_{k}^{adj}$. Results are presented in Table 4.5. Subsets of size 10 are considered first, where the subset numbers correspond to the rankings of the clusters $RSS_{k}^{adj}$ values. For evaluation, only those clusters in the subset are included in Equations 4.22-4.24 during model fitting, and in the corresponding components of subsequent posterior predictive evaluations. Subsets of size 20 are also considered, including a subset of the 20 clusters that occur most frequently for graphs in the training data set.

Table 4.5 Predictive results for which only rotation angles of graphs assigned to subsets of clusters were considered. The subset correspond to the rankings of the clusters $RSS_{k}^{adj}$ defined above. Subsets of size 10 and 20 were considered. “pop20” is the subset of clusters that are the most commonly occurring among graphs in the training data set (the most populated clusters).

<table>
<thead>
<tr>
<th># in Subset</th>
<th>Subset</th>
<th>Ground Truth Assignment (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1-10</td>
<td>95.00</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>91.39</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>93.76</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>94.54</td>
</tr>
<tr>
<td>20</td>
<td>1-20</td>
<td>95.90</td>
</tr>
<tr>
<td></td>
<td>11-30</td>
<td>94.25</td>
</tr>
<tr>
<td></td>
<td>21-40</td>
<td>95.55</td>
</tr>
<tr>
<td></td>
<td>pop20</td>
<td>96.10</td>
</tr>
</tbody>
</table>

Based on the results presented in Table 4.5, $RSS_{k}^{adj}$ does not appear to be a good indicator of cluster importance with respect to the predictive outcomes. It seems that more clusters included in modeling tend to perform better, overall. It is likely the case that $RSS_{k}^{adj}$ is not adequate for capturing the relationships we intended, or there is not a set of clusters to remove due to lack of ability to separate writers’ rotation angles. We abandoned this approach to further investigate the relationships between writer and cluster in a model-based cluster selection approach described in Section 4.3.4.3.
CHAPTER 5. DISCUSSION AND FUTURE WORK

5.1 Discussion

In this dissertation we propose methodology for evaluating the writership probability of a questioned document, with respect to genuine handwriting samples from known sources. The writers of the genuine samples are assumed to be a finite, closed-set of known writers. Our document analysis pipeline is open-source, reproducible, and accessible. We envision many extensions for this work, which are provided in Section 5.2.

The method begins with a scanned image of handwritten text that is processed by the *handwriter* package and expressed as a sequence of disjoint graphical structures. The clustering algorithm of Chapter 2 is used to produce a template for giving each graph a cluster assignment. The assignment is based on the structural similarity between a graph and the 40 cluster exemplars in the template. The cluster assignment frequencies for the graphs that comprise a document are taken together as a 40−dimensional feature vector. The feature vector captures the tendencies a writer has for emitting graphs of each cluster type in writing. The feature vector is extracted from a number of documents for each writer and serve as multinomial data observations in a Bayesian hierarchical model. In the model, a parameter is estimated for each writer in the closed-set. When a document of questioned source is considered, we extract the 40−dimensional feature vector from it and evaluate it under the posterior parameter estimates for each writer in the closed-set. We provide probability estimates of writership for each writer in the closed-set.

In Chapter 3 we introduce a dataset collected at the Center for Statistics and Applications in Forensic Evidence that contains 90 writers. We note distinct patterns in the shape of the 40−dimension cluster frequency feature vectors for the training documents and explore a model that incorporates a mixture component allowing each writer to leverage information from one of two groups. We find that the groups are indicative of writing style and conduct an analysis of probability
assignment within the closed-set as it relates to group preference (via posterior estimates of model parameters) and writing style.

In Chapter 4 we explore additional measurements of the graphical structures within cluster assignment. In particular, we aim to capture the slant or inclination of a graph. This aids in distinguishing writers and adding a second feature type grants an element of robustness to the model. An angular feature is selected for inclusion in the model, which is subsequently used in three applications. Each application was chosen to stress the model in a particular way so we could observe results under different datasets and scenarios.

Strengths of this approach lie in the interpretability of the hierarchical model. Surely more sophisticated statistical learning methodology could be used in such an application, but such black-box approaches would lose the element of interpretability that is achieved by the methods we use here. In addition, our system is automated in the sense that it does not require examiner input or intervention to perform the writer identification task. This removes an opportunity for cognitive bias to occur. We acknowledge that cognitive bias is not removed completely from the problem since the code writers are human.

The largest limitation of this approach is that evaluations are conducted conditional on a closed-set of writers. Conclusions drawn from the modeling outcomes are only meaningful within the closed-set. In addition, the model must be re-estimated each time the closed-set is to be expanded. This can be time consuming and hinder workflow. Lastly, we note that the handwriter is limited in the sophistication of its background removal techniques for images.

5.2 Future Work

The methods that are applied to documents prior to model specification have room for improvement. We see future work in the removal of background noise and lines from scanned handwriting samples within the handwriter package. In addition, speed of processing and parsing documents can be improved.
The features extracted from writing also provide opportunity for future research directions. First, more graph-based features can be explored such as loops, traversal features, or compactness measurements. These could be incorporated into the current modeling framework in addition to rotation angles. Second, as we have seen through the applications of the model in Chapter 4, it may be the case that a template with 40 clusters may not be appropriate for extracting features when the questioned writing is short. If there is not a sufficient number of graphs to properly represent a 40–dimensional vector of cluster frequencies, the use of a smaller template for feature extraction may be more appropriate.

Of particular interest to the authors is the applicability of the methods to a wide variety of data sources. Aside from the typical data sets we have used here, we wish to evaluate performance on historical documents and handwriting samples in languages other than English.

As far as extensions for the modeling framework are concerned, a clear path forward is to evaluate a pair of documents in an open set of writers using a score based approach to summarizing the similarity of the two documents. This approach is common in the forensic statistics community (see Hepler et al. (2012), Swofford et al. (2018), or Hare et al. (2017)) and comes with its own set of challenges. Taking such an approach would open the analyses up from the closed-set of known writers that we have been conditioning on throughout the three papers presented here.

The final opportunity for future improvements we see for this body of work is the development of a user interface for the software. This would provide a user friendly experience for interested parties, in particular the FDE. This interface would provide details of modeling results along with variability estimates and an interactive environment with which the FDE could explore model components and understand modeling outcomes.

### 5.3 References
