Clustering Workflow Requirements Using Compression Dissimilarity Measure

Li Wei  
Department of Computer Science  
University of California, Riverside  
wli@cs.ucr.edu

John Handley  
Nathaniel Martin  
Tong Sun  
Xerox Corporation  
{JHandley, NMartin, TSun}@xeroxlabs.com

Eamonn Keogh  
Department of Computer Science  
University of California, Riverside  
eamonn@cs.ucr.edu

Abstract

Xerox offers a bewildering array of printers and software configurations to satisfy the needs of production print shops. A configuration tool in the hands of sales analysts elicits requirements from customers and recommends a list of product configurations. This tool generates special question and answer case logs that provide useful historical data. Given the unusual semi-structured question and answer format, this data is not amenable to any standard document clustering method. We discovered that a hierarchical agglomerative approach using a compression-based dissimilarity measure (CDM) provided readily interpretable clusters. We compare this method empirically to two reasonable alternatives, latent semantic analysis and probabilistic latent semantic analysis, and conclude that CDM offers an accurate and easily implemented approach to validate and augment our configuration tool.

1. Introduction

Digital printing has dramatically changed the publishing and printing world. Digital presses allow shorter press runs by removing the bottleneck of readying the press for printing. Shorter press runs focus attention on the entire publishing workflow from creating content to delivering finished results because the cost in time and effort must be amortized over fewer final products [1]. Choosing equipment and software to support such workflows becomes more difficult because time and effort must be squeezed from the entire process.

The activities the print shop performs to satisfy their customers’ requirements comprise a workflow [2][4]. Each workflow introduces a set of workflow constraints, i.e., features such as the printing application types, existing or desired pre-press software, operating environments, finishing equipment, print volume, speed, media, etc. Though production print shops specialize on particular types of products such as books or statements, each still has a unique set of workflow constraints introduced by the workflows of that particular production print shop.

To develop sales proposals for a production print shop, sales analysts recommend products to the print shops according to their workflow constraints. These products include hardware such as printers of various capabilities and finishing equipments that turn the printed pages into products such as books and fliers. They also include software such as document repositories, workflow management systems, and digital front ends to convert files produced by print drivers into bitmaps for printing. A consistent set of products that meets a print shop’s needs by supporting all of their workflow constraints is called a workflow configuration.

To help select an appropriate workflow configuration, Xerox sales analysts use the Workflow Configuration Tool. The Workflow Configuration Tool elicits the customer’s printing requirements, such as finishing, printing, and color specifications using a dynamic questionnaire. It then produces a set of equivalent workflows that satisfy the customer’s requirements, and then combines the workflows into workflow configurations using heuristic rules. Here, we try to characterize the workflows based on the questions and answers given to the Workflow Configuration Tool to explore the use of document clustering as a check and augmentation of the heuristic rules.

The rest of the paper is organized as follows. In Section 2, we review the Workflow Configuration Tool. In Section 3, we review document clustering techniques, the Compression-based Dissimilarity Measure (CDM), and hierarchical clustering methods. Section 4 provides an empirical evaluation of our approach. Finally in Section 5, we offer conclusions and thoughts on future work.
2. Workflow Configuration Tool

The Workflow Configuration Tool collects responses to questions in forms of yes/no, free text, multiple (check-box) and single (radio-button) selections. The questions in the Workflow Configuration Tool are designed by domain experts who are specialized in several key production printing workflow spaces: Book Printing (Book), Print-On-Demand (POD), Personalized Communication (PC), Transactional and Promotional Printing (TP). Although there is no unified logical structure across these four workflow spaces, questionnaire structure and choices in Book Printing and POD spaces are closely related while questionnaires in PC and TP share many commonalities.

After the questions are answered, a rule-based workflow auto-generation module generates a set of workflow configurations based upon the product and workflow knowledge patterns in a knowledge base. The user then chooses a workflow configuration that is most suitable to his or her requirements. User’s inputs and the generated workflows are recorded as case logs.

The case logs contain valuable information about the types of configurations customers require, making them a potentially valuable resource. We are interested in extracting information from these logs. Two questions are of particular interest: Can the high-level workflow be detected from the questions and responses to validate the rule-based workflow configuration system? Can the tool be augmented by using historical data to classify new cases?

Although the two problems are quite distinct, document clustering addresses both. Answering the first question involves clustering the case logs and checking them against the workflow configurations generated by the configuration tool. If the clusters of the case logs match with the workflow configurations assigned by the configuration tool, our confidence that the questionnaires have sufficient information to distinguish different customer needs increases; if not, there is some important information the questionnaire misses. Answering the second question involves using the clusters to discover unexpected workflow configurations. If there are clusters of documents that fall outside the known workflow configurations, they indicate the possibility of a new, unknown workflow configuration.

However, the case logs of the Xerox Workflow Configuration Tool pose unusual problems for document clustering. Figure 1 shows a sample segment of a case log. The unusual problems that the case logs pose are:

1) The case logs are in XML format but the XML tags carried formatting information instead of structure information. Two case logs may have the same XML tags but different attributes and values.
2) The case logs are generated from questionnaires and the most typical content is questions followed by simple responses such as yes, no, true, or false.
3) Since the questionnaires are generated dynamically, the sizes of the case logs vary widely. We found that the sizes of the case logs ranged from 10 kilobytes to 24 kilobytes.

3. Document Clustering Techniques

To cluster the logs, we applied the Compression-based Dissimilarity Measure (CDM), an elegant general-purpose measure for strings recently explored by researchers in machine learning and data mining [6]. CDM is effective, efficient, and easy to implement. More importantly, since the CDM approach measures the similarity of the documents based entirely on shared information, it sidesteps the problem of selecting explicit or implicit features of the text, which introduces possible bias, error, or assumptions about the data.
3.1  Kolmogorov Complexity

The key problem of document clustering is to find a good measure to capture the true similarity of documents. Recently, a number of compression based similarity measures have been proposed to capture the amount of information shared between two strings. They may trace their roots to Kolmogorov complexity.

Following the notations in [8], we use \( \text{string} \) to mean a finite binary string. The Kolmogorov complexity \( K(x) \) of a string \( x \) is defined as the length of the shortest program to compute \( x \) on a universal computer, such as a Turing machine. Intuitively, \( K(x) \) is the minimal amount of information required to generate \( x \) by an algorithm. The conditional Kolmogorov complexity \( K(x|y) \) of \( x \) to \( y \) is defined as the length of a shortest program to compute \( x \) if \( y \) is provided as an auxiliary input to the program.

Li, et al. [7] define the distance between two strings \( x \) and \( y \) as

\[
d_{i}(x, y) = \frac{K(x|y) + K(y|x)}{K(xy)}
\]

where \( K(xy) \) is the length of the shortest program that produces \( x \) concatenated with \( y \). They have proved that this distance satisfies the triangle inequality, up to a small error term.

3.2 Compression-based Dissimilarity Measure

Since \( K(x) \) is the best compression that one could possibly achieve for the text string \( x \), universal compression algorithms give an upper bound to the Kolmogorov complexity. Therefore, we could use compression to approximate the distance between two strings. Given a data compression algorithm, we define \( C(x) \) as the compressed size of \( x \), \( C(xy) \) as the compressed size of \( x \) concatenated with \( y \), and \( C(x|y) \) as the compression achieved by first training the compression on \( y \), and then compressing \( x \). So we can approximate (1) by the following distance measure

\[
d_{i}(x, y) = \frac{C(x|y) + C(y|x)}{C(xy)}
\]

We adopted the compression-based distance as our similarity measurement. However, the computation of the distance given in equation (2) requires modifying the chosen compression algorithm to obtain \( C(x|y) \) and \( C(y|x) \). Therefore, Keogh et al. [6] proposed a further simplified distance and showed that the simpler measure is as effective as its more complicated counterpart. Given two strings \( x \) and \( y \), they define the Compression-based Dissimilarity Measure (CDM) as

\[
CDM (x, y) = \frac{C(xy)}{C(x) + C(y)}
\]
insight by comparing the purity of classes within clusters.

By inspection of the dendrogram of the CDM in Figure 2 we see eight fairly well-formed clusters. Their class distributions are shown in Table 1. There are two Book Printing (Book) clusters, one Transactional/ Promotional Printing (TP) cluster, three Print-On-Demand (POD) clusters, one No Workflow (No WF) cluster, and a Personalized Communication (PC) cluster.

We then evaluate the purity of the clusters generated by LSA and PLSA. The purities of the clusters generated by LSA are listed in Table 2. All but a trivial cluster (Book, POD) are impure and there is no distinct No Workflow cluster. The class distributions of PLSA clusters are shown in Table 3. The PLSA clusters are considerably better in purity, but not as good as CDM (more No Workflow instances are mixed with other clusters in PLSA approach).

The most surprising result is that CDM-based hierarchical clustering methods were able to detect a
previously unknown class where the user exits the tool prematurely. These are (now) called “No Workflow.” These cases cannot be detected purely by length or by inspection, because sometimes they are the same as normal logs except for submission. There were 23 cases of “no workflow,” 17 of which PLSA misclassified. Also, CDM with hierarchical agglomerative clustering was able to detect the close relationship between the Book Printing and POD workflows – customers buy similar product configurations for these.

We use the dissimilarity matrices from CDM and LSA to construct 5-nearest neighbor classifiers and present their confusion matrices in Tables 4 and 5 using resubstituted data. The CDM-based classifier has an estimated error rate of 4.5% compared to 29.5% for LSA. A trained PLSA classifier has a resubstitution error rate of 14%. Its confusion matrix is shown in Table 6. In fact, the CDM classifier has errors only in the No Workflow class. Values of \( k = 3, 7, \) and 9 give similar results.

<table>
<thead>
<tr>
<th>Table 4: Confusion matrix for 5-nn classifier using CDM dissimilarity matrix with data resubstitution. Error rate is 4.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
</tr>
<tr>
<td>Book</td>
</tr>
<tr>
<td>TP</td>
</tr>
<tr>
<td>POD</td>
</tr>
<tr>
<td>No WF</td>
</tr>
<tr>
<td>PC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5: Confusion matrix for 5-nn classifier using LSA dissimilarity matrix with data resubstitution. Error rate is 29.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
</tr>
<tr>
<td>Book</td>
</tr>
<tr>
<td>TP</td>
</tr>
<tr>
<td>POD</td>
</tr>
<tr>
<td>No WF</td>
</tr>
<tr>
<td>PC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6: Confusion matrix for PLSA classifier. Resubstitution error rate is 12.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
</tr>
<tr>
<td>Book</td>
</tr>
<tr>
<td>TP</td>
</tr>
<tr>
<td>POD</td>
</tr>
<tr>
<td>No WF</td>
</tr>
<tr>
<td>PC</td>
</tr>
</tbody>
</table>

5. Conclusions and Future Work

Gathering requirements and generating recommendations for production printing systems can be automated using a question and answering tool. The output of this tool contains a wealth of data that can be mined. These logs contain unusual text that exhibit little of the expected semantic structure of documents. We have established empirically that standard document clustering methods like LSA and PLSA do not perform as well as CDM on these semi-structured data. Moreover, CDM was able to identify an unanticipated class of case logs, ones that were the result of an aborted questionnaire. We have shown that a CDM approach is surprisingly effective, interpretable, and simple to implement using general-purpose, off-the-shelf software. It also allows for a very accurate classifier to be constructed. This success adds to the growing scope of applications for CDM.

6. References