Clustering Benchmark for Characters in Historical Documents

Martin Jenckel
DFKI
67655 Kaiserslautern
Email: Martin.Jenckel@dfki.de

Dr. Saqib Bukhari
DFKI
67655 Kaiserslautern
Email: Saqib.Bukhari@dfki.de

Prof. Dr. Prof. h.c. Andreas Dengel
DFKI
67655 Kaiserslautern
Email: Andreas.Dengel@dfki.de

Abstract—In document analysis clustering found many applications, especially in the field of Optical Character Recognition (OCR). Depending on the task these algorithms however perform very differently. Especially when clustering single characters of historical documents state of the art algorithm still lack in performance and semi-manual methods are used [1]. Therefore we take a look at some standard feature and clustering algorithm combinations and compare their performance on a character corpus extracted from the medieval novel "Narrenschiff".

Keywords—clustering, historical documents

I. INTRODUCTION

Clustering in general and working on historical documents specifically leads to a number of challenges and problems that algorithms have to deal with.

- Deformed characters: e.g. due to damaged documents or scanning artifacts
- Inhomogeneous class distribution: Some characters are overrepresented while others appear very rarely.
- Inhomogeneous class differences: Certain characters are more "similar" (depending on the features) to each other while others are very different.

The first and the last are mainly dependent on the selection of features, but the clustering algorithm can also account for them to some degree.

II. BENCHMARKING

A. Data

We use a randomly selected three page subset of the medieval novel "Narrenschiff" as our data. Segmenting the pages into their single characters automatically would introduce additional errors, therefore we manually segmented the pages into lines and characters as well as assigned the corresponding ground truth, but ignored side- and handwritten notes. This results in \( \approx 2400 \) characters representing 69 different character classes.

B. Feature Selection

In order to extract comparable features from the segmented characters, each character is first down sampled to 32 pixel in the larger dimension while keeping the aspect ratio. Then it is centered and zero padded to a 32x32 image.

1) Raw Features: For comparison we will also use the raw features. The image is unraveled into a single feature vector of length 1024.

2) Principle Component Analysis: The Principle Component Analysis (PCA) is another standard method for feature extraction and dimensionality reduction [2]. We chose to use each images projection onto the first 100 principle components as the feature vectors.

3) Shape Context: Shape context (SC) is a set of features introduced by [3] and aims to uniquely describe shape contours based on their relative radial and angular positions. The main properties of the generated features are angular invariance, size invariance and their ease to compute. As the feature vector we use the average histogram over all contour points. The grid was given by 12 angular spaces and 5 radial ones resulting in 60 cells. The maximum radius for each character was set to its maximum distance between any two contour points.

C. Clustering

1) K-means: K-means is probably the most used and best known clustering algorithm [2]. The initial seeds are chosen randomly and the euclidean distance to determine closeness.

2) DBSCAN: DBSCAN is a density based clustering algorithm that assigns points to clusters based on the local density [4]. If there is a number of points \( i \) within the distance \( r \) of any
point, than this point is considered a core-point of this cluster. We set \( i = 5 \) and optimized \( r \) until we had the maximum number of clusters while keeping the number of noise points small.

### III. Results

We evaluated the clustering by assigning the class to each cluster which is represented the most. All characters of other classes are therefore considered errors. For DBSCAN all noise points are also considered errors. For k-Means the accuracies are averaged over five initial seeds.

We would like to point out, that the raw features performed very similar to the PCA and SC was considerably worse. In terms of clustering, k-Means yielded better results with all three features, even when underestimating the actual number of classes. When increasing \( k \) and allowing for some over fitting the accuracy is increased as well. This is not problematic since this type of clustering does not have to be generalized.

For shape context the similarity between “u” and “n” as well as “p” and “d” is due to the angular invariance inherent to the feature and is well represented in the confusion, when combined with k-Means. The confusion for all combinations with DBSCAN in comparison show the algorithms inability to properly differentiate the classes and mixing them in a single cluster instead.

### IV. Discussion

The results are in line with the challenges we mentioned before. The inhomogeneous distribution of the classes could not be handled by either clustering algorithm. Even an accuracy of 0.91 corresponds to detecting less than half of the classes. Arguably this problem would be reduced by having more data, however this would at the same time reduce the over fitting effect and most likely decrease over all performance.

DBSCAN does not depend as much on the class distribution but more on the inter class differences. However none of the features provided a suitable basis. PCA by construction enhances the inter class variances and therefore performed best, though PCA itself suffers from the overrepresentation of some classes such that the principle components are biased towards these classes.

None of the methods was particularly good at dealing with deformed and/or damaged characters. Shape context though is very reliant on the character shapes and damages, missing parts or artifacts are represented more distinctively in the features. Our method of averaging and defining closeness do not seem to retain all of the methods positive properties.

Through the work on this benchmark we recognize the importance of features and algorithms that tackle the challenges mentioned in the introduction. Especially inhomogeneous class distributions are critical, because they are poorly represented in the overall accuracy. In our case about half of the classes were not represented in the clustering. Our conclusion is to focus on features that highlight the differences between the classes rather than identifying the similarities within each class. At the same time clustering algorithm that can robustly recognize clusters of various sizes and densities are necessary.

### ACKNOWLEDGMENT

The authors would like to thank the Kallimachos project for providing the data and the ELKI Team for providing the necessary tools [5].

### REFERENCES


