CONFIRM - Clustering Of Noisy Form Images using Robust Metrics

Chris Tensmeyer
Department of Computer Science
Brigham Young University
tensmeyer@byu.edu

Tony Martinez
Department of Computer Science
Brigham Young University
martinez@cs.byu.edu

Abstract

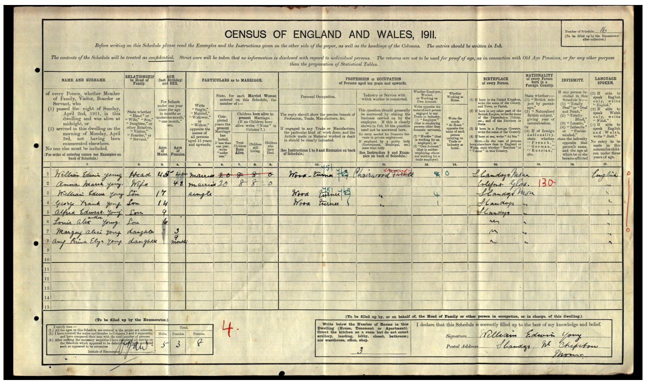
The ability to automatically cluster large collections of noisy form images according to form type would improve the efficiency of organizations that currently do this by hand. Some noisy form collections contain form types that are structurally very similar, but should cluster apart. To address this issue, we propose CONFIRM - Clustering Of Noisy Form Images using Robust Metrics. CONFIRM uses a novel technique to match form text and rule lines to create vector representations of each form. A Random Forest classifier is then used to learn a pairwise similarity metric for use in Spectral Clustering. Validation is provided on the NIST tax forms as well as several historical forms datasets.

1 Introduction

Much genealogical data has been made available by converting paper forms, such as census records or death certificates, into digital images. It is often desirable to group these forms according to form structure, so that structure-specific processing can take place. For example, a template of field locations for a particular structure can be fit to forms with that structure for field segmentation. Currently, grouping forms according to structure is often done manually, which is accurate but costly. We present CONFIRM (Clustering Of Noisy Form Images using Robust Metrics), an approach that attempts to automate this process by clustering form images according to structural similarity.

While there is no agreed upon definition in the literature [2], the definition of a form type which we adopt for this work is that two forms are of the same type if the location and labels of the form fields are identical. For example, adding or removing a form field from an existing type results in a new type. The type of a form is agnostic to information entered in the fields and to cosmetics such as printing color. Figure 1 shows two examples of similar but different form types, and Figure 2 shows two forms of the same type.

The main contribution of this work is to present CONFIRM, which extends the *HVP-RF* model in [5] to use form-specific features in conjunction with a novel form matching algorithm. Additionally, we address the problem of clustering very similar form types. Chen et. al. [3] note the difficulty introduced by similar form types in a classification setting, but to our knowledge this has not been addressed in an unsupervised setting.


CENSUS OF ENGLAND AND WALES, 1911.

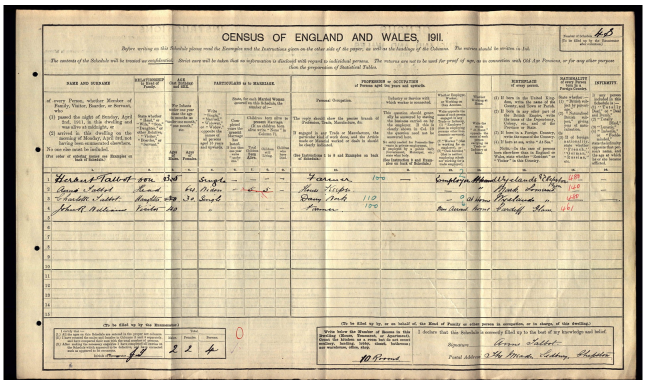

CENSUS OF ENGLAND AND WALES, 1911.

Figure 1: Two similar form types from the 1911 England and Wales Census that should cluster apart. The first form has an additional column (the last) entitled *Language Spoken*.

Clustering algorithms may have difficulty deciding if the differences between similar types are significant or are noise. For example, compare the between form variance illustrated in Figure 1 to the within form variance shown in Figure 2.

We validate CONFIRM using cluster purity and V-measure on the NIST Tax Forms [4], a standard forms dataset, as well as on several historical collections furnished by Ancestry.com. As CONFIRM is an extension of the *HVP-RF*, we have implemented the *HVP-RF* model as a comparative baseline.

2 Related Work

We review two works that address form image clustering by structure. There are related problems of clustering by field value and form classification which are omitted for space.

The first work, by Saund [8], uses features extracted from de-

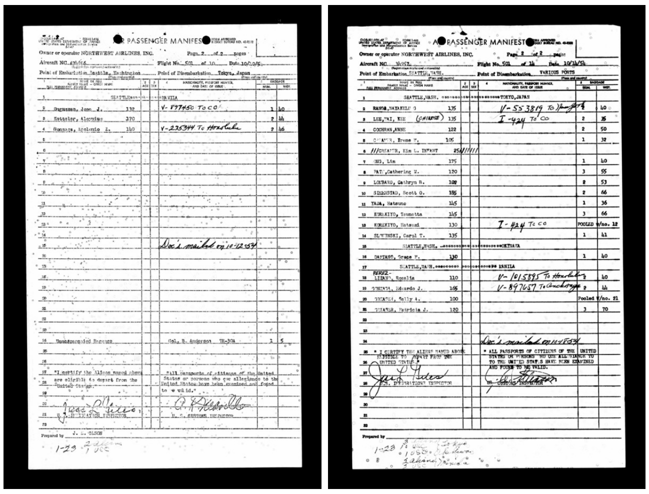

PASSENGER MANIFEST
PASSENGER MANIFEST

Figure 2: Forms of the same form-type from the *WassPass* dataset that should cluster together. The structure can be somewhat obscured by noise and entered information.

tected horizontal and vertical lines to perform clustering. Each form is encoded as a histogram of both simple and complex line junctions patterns. Then a greedy similarity based clustering is performed using the Common-Minus-Difference similarity metric. Doing this, Saund was able to achieve a 100% pure clustering on the NIST tax forms, albeit one form type was split between two clusters. However, this approach would fail if some types of forms have identical line structure, but different field labels. This can occur in practice, as it does in the 1911 England and Wales Census, if two types are language translations of each other.

More recently Kumar and Doermann [5] proposed a clustering method, *HVP-RF*, that encodes form images as histograms of generic computer vision features. Then a Random Forest classifier [1] is used to learn a pairwise similarity matrix, which then becomes input to Spectral Clustering [9]. As CONFIRM is an extension of this method, detail is given in Section 3.3. Even without form-specific features, *HVP-RF* clusters the NIST tax forms with 100% purity and the correct number of clusters.

3 Methodology

CONFIRM consists of three steps: feature detection, feature matching, and learning a similarity metric. We assume that form images are deskewed and are rescaled to uniform width and height.

3.1 Feature Detection

Each text line in the form image can be characterized by a string and a bounding box. Text line detection is done by using Optical Character Recognition (OCR). To remove falsely detected text lines, CONFIRM discards punctuation, digits, and lines with fewer than 4 characters.

Horizontal and vertical rule lines are detected by passing directed edge filters over the image and using connected component analysis on the filtered images. Each rule line is then represented as a starting point, a length, and an orientation (horizontal or vertical).

3.2 Feature Matching

Matching is performed between two forms, e.g. f_1 and f_2, to see if the features of f_1 match corresponding features in f_2. If f_1 has N features, this results in a vector v of length N, where $v_k \in [0,1]$ indicates how well the k^{th} feature of f_1 was matched by some feature in f_2. The matching algorithm for text lines determines the value of the v_k that correspond to text lines of f_1, and the same occurs for rule lines.

Text Lines

Two text lines should match if their strings are similar and if their bounding boxes are not too far apart. More formally, two text lines t_1, t_2 match, when the following conditions hold:

$$dist(t_1,t_2) \leq a \qquad \frac{edit(t_1,t_2)}{max(len(t_1),len(t_2))} \leq \frac{1}{d}$$

when a and d are user-set thresholds, $dist$ is Euclidean distance between bounding boxes, $edit$ is string edit distance [6], and $len(t)$ is the number of characters in the transcription of t. The true translation offset between forms is unknown, so a is a liberal estimation of offset that also prevents text lines from opposite sides of the page from matching. The threshold d represents a tolerance of one character error in the OCR for every d characters in the longer string. Empirically we have empirically set a to be 25% of the larger image dimension and set d to 5.

Matches are found by sequentially examining all pairs of text lines to see if they meet the matching criteria. When matches are found, both text lines are marked as matched and are excluded from matching other text lines. Remaining unmatched text lines are then considered for prefix and suffix matches to account for OCR segmentation errors. If the combined prefix and suffix meet the above criteria for matching another line, then all three lines are marked as fully matched.

This algorithm sets $v_k = 1$ if the k^{th} feature is a text line and that text line is marked as matched, and sets it to 0 otherwise.

Rule Lines

Here we present adaptation of string edit distance for matching sorted sequences of parallel line segments. The edit distance [6] of two strings s_1 and s_2 of lengths n and m respectively is given by:

$$edit(s_1,s_2) = d(n,m)$$

$$d(i,j) = \begin{cases} min \begin{cases} d(i-1,j)+1 \\ d(i,j-1)+1 \\ d(i-1,j-1)+\mathbb{1}(s_{1i} \neq s_{2j}) \end{cases} \\ 0 \qquad \text{if } i \text{ or } j \leq 0 \end{cases}$$

The three edit operations, corresponding to the terms of the min function, are insertion, deletion, and match or substitution,

Table 1: Edit operations for sequences of parallel lines. Let l_1 and l_2 be lines from different sequences.

Operation	Description	Cost
Deletion	l_1 is deleted	$len(l_1)$
Match	l_1 and l_2 match	0
Contain	l_1 contains l_2	$len(l_1) - len(l_2)$
Overlap	l_1 and l_2 overlap	len of non-overlap
Connect	l_1 bridges gap between two lines in other sequence	Two Overlaps or Contains
Transpose	swap sequence position of adjacent lines	0

respectively. While strings are sequences of discrete symbols and the corresponding edit operations and costs are discrete, each line segment consists of multiple continuous values so the line edit operations and costs we use are continuous. The line segment edit distance has the same mathematical form as string edit distance, but the terms in the min function include other operations and costs.

When rule lines are faded in a form image, the extracted lines are often truncated or broken into multiple pieces. The edit operations shown in Table 1 attempt to correct these kinds of errors. For example, the Connect operation matches a whole line to corresponding fragmented parts in the other sequence.

Computing the edit distance gives us the optimal set of edit operations, which optimally groups lines by similarity. Each line is associated with a single edit operation, which has a particular cost. We normalize that cost to the range $[0, 1]$ by dividing by the cost of deleting the line, which is the maximum cost operation. If the k^{th} feature is a rule line with a normalized edit cost of c, then we set $v_k = (1 - c)$.

3.3 Learning Pairwise Similarities

Given a collection F of forms to cluster, CONFIRM randomly selects a small (e.g. 10-50) subset of forms T to be *templates*. Ideally, T should contain a form for every type, but our results indicate this is not an absolute requirement for good performance. CONFIRM then matches each $f_i \in F$ against each $t_j \in T$ to produce vectors v_{ij} which indicate how well each feature in each t_j is matched by f_i. Then let u_i be the concatenation of all vectors v_{i*} be a new feature vector for f_i.

A similarity metric is then learned using a technique proposed by Kumar and Doerman in [5], which is briefly explained here.

Each u_i becomes a row in a matrix M, so each column corresponds to how well a particular text or rule line in some $t \in T$ was matched. Then an auxiliary matrix A is formed with the same dimensions as M. Each element in each column in A is set to a randomly drawn value from the corresponding column in M. The values within each column of A have approximately the same univariate statistics as the columns of M. The difference is that the columns of A are not correlated, or in other words, there is no structural dependence between features in A.

These two matrices M and A are used to train a binary Random Forest (RF) Classifier [1], with rows of M labeled as the positive class and rows of A labeled as the negative class. To distinguish rows of M from rows of A, the RF must discover the relevant correlations between features in M. Each individual tree in the RF uses a random subset of the features to make splitting decisions before terminating in class specific leaf nodes. Two structurally different forms in M share the same label according to the RF, but may arrive at different positive-class leaf nodes in the trees because the relevant structural feature correlations are different between the forms. The similarity of two forms is then computed as the percentage of trees in which both forms arrive at the same leaf node, indicating that they follow the same path in that tree.

As in [5], CONFIRM then uses Spectral Clustering [9] to cluster forms based on the pair-wise similarities learned by the RF.

4 Experiment

A number of datasets (shown in Table 2) are used to compare the *HVP-RF* model [5] and CONFIRM using a varying number of randomly selected templates. Note that the difference in the models is how forms are initially encoded. *HVP-RF* uses histograms of generic computer vision features, and CONFIRM initially uses form-specific features which are then matched against templates to produce vector features.

Dataset Name	Types	Forms	Notes
*WashPass**	2	2,000	Typeset field entries
*PADeaths**	5	4974	Imbalanced Types
*PADeath-Balanced**	5	491	100 forms per type
*Wales**	11	6,354	Imbalanced Types
*Wales-Balanced**	24	4,800	200 forms per type
NIST [4]	20	11,185	Well separated types

*provided by Ancestry.com

Table 2: Information on datasets and subsets used in experiments.

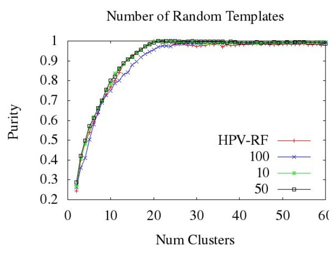
We report two clustering metrics, purity and V-measure [7], which rely on ground truth type labels, on the average of 5 different random seeds. Cluster purity is measured by assigning each cluster to its most frequent ground truth type and counting the percentage of correctly labeled forms. While purity is a good metric, it generally increases as the number of clusters increases because 100% purity can be trivially obtained by placing each form in its own cluster. Therefore we also report V-measure to give a more balanced picture of the trade off between purity and number of clusters. V-measure is an entropy based measure that first computes homogeneity and completeness measures and then computes their harmonic mean, which skews V-measure towards the lower of the two individual measures. Homogeneity is similar to purity, but considers the distribution of types within a cluster, instead of just the majority type. Completeness measures the entropy of the distribution of each type over the clusters so that high completeness scores occur when each type is concentrated in a small number of clusters.

The number of Code Words of *HVP-RF* (a hyper-parameter) was set by cross validation on the *Wales* dataset, and the same RF parameters as in [5] were used for both models.

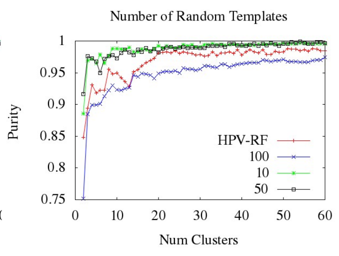
5 Results

The graphs in Figure 3 indicate that CONFIRM performs as well as or better than *HVP-RF* on all datasets with as little as 10 random templates. Using 100 templates tended to do worse, most likely because the higher dimensionality of the features required more trees in the RF to get asymptotic behavior. Template sets of size 20, 30, and 75 were also tried, but in all cases performed similarly to templates sets of size 50.

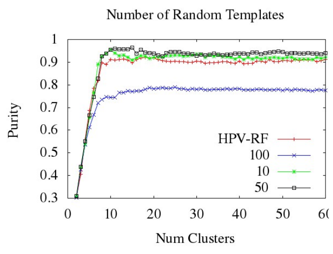
On the clean, well-separated NIST dataset, both algorithms perform similarity, achieving 100% purity at 20 clusters. The performance difference is most noticeable on the Wales-Balanced dataset which has the most similar types of any dataset. Note that even though Wales-Balanced has 24 types, selecting 10 random seeds yields good performance indicating that not all types need to be represented in the randomly chosen templates. The performance gap is small for Wales, but CONFIRM tends to cluster apart the smallest type (comprising 2.2% of the forms) and *HVP-RF* does not until the number of clusters greatly exceeds the number of types. This trend is also seen in PADeaths where two of five types compose 75% of the dataset. *HVP-RF* hits a purity peak at 20 clusters, while CONFIRM is highest around 8-9 clusters. In PADeaths-Balanced CONFIRM and *HVP-RF* levels off with fewer clusters.



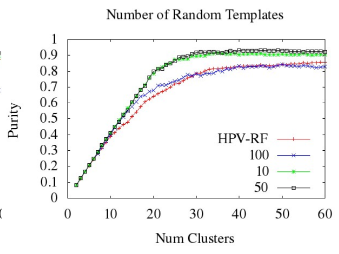
(a) NIST



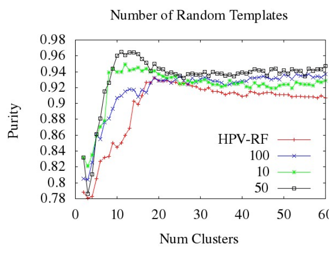
(b) WashPass



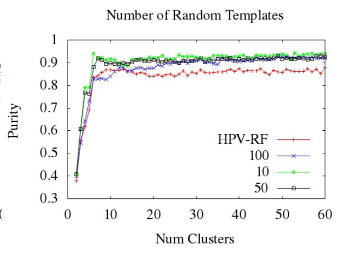
(c) Wales



(d) Wales-Balanced



(e) PADeaths



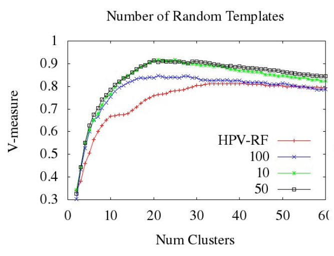
(f) PADeaths-Balanced

Figure 3: Cluster Purity curves for CONFIRM differing the number of randomly chosen templates. Even selecting 10 templates out performs *HVP-RF* by a large margin, though using 100 templates likely generated too many features for the number of trees (2000) used in the RF.

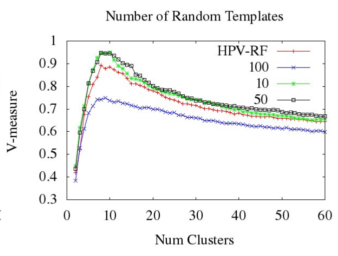
Examining the graphs of V-measure (Figure 4) indicate that CONFIRM has overall better cluster quality (except for NIST where performance is similar). Because the differences in V-measure are similar to the differences in purity, graphs for some of the datasets are omitted.

6 Conclusion

Here we have presented CONFIRM and shown that it outperforms a state-of-the-art model across several datasets in form image clustering tasks. The use of form specific features and encoding forms via feature matching against randomly selected templates enables CONFIRM to differentiate between different form types that appear similar in image space.



(a) Wales-Balanced



(b) Wales-Small

Figure 4: V-measure for selected Datasets (others omitted for brevity). The difference is larger for the balanced dataset.

References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[2] Nawei Chen and Dorothea Blostein. A survey of document image classification: problem statement, classifier architecture and performance evaluation. *International Journal of Document Analysis and Recognition (IJDAR)*, 10(1):1–16, 2007.

[3] Siyuan Chen, Yuan He, Jun Sun, and Satoshi Naoi. Structured document classification by matching local salient features. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 653–656. IEEE, 2012.

[4] D. L. Dimmick, M. D. Garris, and Wilson C. L. Nist structured forms reference set of binary images (sfrs). Online, 1991.

[5] Jayant Kumar and David Doermann. Unsupervised classification of structurally similar document images. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1225–1229. IEEE, 2013.

[6] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*, volume 10, page 707, 1966.

[7] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420. Citeseer, 2007.

[8] Eric Saund. A graph lattice approach to maintaining dense collections of subgraphs as image features. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1069–1074. IEEE, 2011.

[9] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.