# Active Clustering of Document Fragments using Information Derived from Both Images and Catalogs

Lior Wolf, Lior Litwak, Nachum Dershowitz
The Blavatnik School of Computer Science
Tel Aviv University

Roni Shweka, Yaacov Choueka
The Friedberg Genizah Project
Jerusalem, Israel

## Abstract

*Many significant historical corpora contain leaves that are mixed up and no longer bound in their original state as multi-page documents. The reconstruction of old manuscripts from a mix of disjoint leaves can therefore be of a paramount importance to historians and literary scholars. Previously, it was shown that visual similarity provides meaningful pair-wise similarities between handwritten leaves. Here, we go a step further and suggest a semi-automatic clustering tool that helps reconstruct the original documents. The proposed solution is based on a graphical model that makes inferences based on catalog information provided for each leaf as well as on the pairwise similarities of handwriting. Several novel active clustering techniques are explored, and the solution is applied to a significant part of the Cairo Genizah, where the problem of joining leaves remains unsolved even after a century of extensive study by hundreds of human scholars.*

## 1. Introduction

Written text is an ideal source for understanding historical life. Handwritten texts, such as community documents, notebooks, personal letters and commercial records can all contribute to a fuller understanding of a given place and time. Many large collections of such material are currently being digitized and made available on the Internet, including: (1) 350,000 fragments of discarded medieval codices, scrolls, letters and documents, discovered in the 1890s in the attic of a synagogue in old Cairo and being digitized by the Friedberg Genizah Project [5, Chap. 11]; (2) 2,000,000 images of 70,000 pre-1900 Taiwanese deeds and court papers in the Taiwan History Digital Library [3]; and (3) 30,000 vellum fragments from the Dead Sea Scrolls found in Qumran, now undergoing multispectral imaging [9].

Scholars have expended a great deal of time and effort on manually rejoining leaves of the same original book or pamphlet, and on piecing together smaller fragments. In the case of the Cairo Genizah, whose fragments are dispersed world-wide, this may involve visiting numerous libraries, usually as part of research on a particular topic or literary work.

Recently, computer vision tools were shown to be effective in automatically identifying potential *joins* between pairs of leaves of the same original book, so that they may be verified by human experts [22].[1] This system employs modern image-recognition tools such as local descriptors, bag-of-features representations and discriminative metric learning techniques, that are modified for the problem at hand by applying suitable preprocessing, and by using task-specific key-point selection techniques.

This pairwise approach was shown to provide significant value to the scholarship of the Genizah, and approximately 1000 new joins of significant importance were verified [22]. This is to be compared with the overall number of joins found in over a century of Genizah research, by hundreds of researchers, which numbers only a few thousand.

In this work, we integrate across the leaves the pairwise visual similarity as well as catalog-derived information to obtain solutions for a more general task: the reconstruction—to the extent possible—of the original documents. Computationally, this problem is a challenging clustering problem: the scale of the problem is tens or hundreds of thousands, the number of clusters is unknown a priori, the clusters' sizes vary greatly, and the underlying similarity measure is based on incomplete and noisy data. On the other hand, the work of scholars over the last century can be used to seed the clustering, as well as to learn suitable metrics and parameters of the underlying clustering problem.

The problem is solved in an *active* manner, where experts can provide some feedback to the reconstruction system, validate its riskiest hypothesis, and help the system overcome critical points of ambiguity. Such expert intervention is necessary, since the system does not have the do-

[1] Joins are groups of multiple leaves originating from one original manuscript. Sometimes 'join' refers to a single pair of leaves, even if it's part of a larger known join. The distinction should be clear from context.

main knowledge required to validate or reject challenging borderline cases. Whereas a human expert can read fragments and match them by content, the underlying OCR task is challenging due to poor conservation of the documents and wide variability in scripts. In our experience, the popular Tesseract OCR software [16] fails to produce reliable results even for square Hebrew letters, despite their limited variability among scribes and despite retraining with suitable samples. Other scripts are even more challenging.

## 2. Related Work

We use a graphical model approach for clustering. Earlier contributions [15] assume a known number of classes. A recent contribution in the field of pedestrian tracking uses a formalization that is similar to ours in many key aspects [13]. Pedestrians are grouped based on the similarity in their trajectories using a graphical model formulation where inference is made for the trajectory of each individual out of a list of hypothesis trajectories and for pairwise binary "same group" variables. Transitivity constraints are enforced by considering all triplets of pedestrians. Our framework is similar with some important modifications: each data-point in our formulation is associated with multiple properties that are unique to our problem and require careful modeling; the scale in which we solve the problem is several orders of magnitude larger, which requires structural adaptations; and due to the need of human feedback, our framework is both semi-supervised and active.

The semi-supervised cues in our system are given by a list of known joins. The system also makes queries to the user during the clustering process, making it active. Previous contributions in the domain of active clustering typically try to define classifier-based models for each detected cluster, either as a preprocessing step before clustering [2] or within iterations [7]. The query strategy is then based on active learning techniques for supervised learning. This is in contrast to our work, where the query selection strategy is closely related to the inference engine, and we propose a new scheme that seem to outperform conventional active learning criteria. Another difference from our framework is that, in the active clustering literature we are aware of, properties are not assigned to each data-point.

In the context of computer vision, the active clustering of face images in albums or social networks [17] is a practical problem for which commercial solutions (e.g., Picasa) involve active user participation. Little is known about the actual algorithms used in such systems; however, it seems that in many cases clustering is performed first in a conservative manner to obtain many relatively small groups that are then identified by users to form larger groups. This process may repeat iteratively. A simulation of face image clustering using a simplified semi-automatic model was done in [6] to compare the quality of face descriptors, but the literature on the subject is scarce. Note that typically the user knows the people involved and is able to make instantaneous decisions, which cannot be expected from the experts using our system.

The literature on computer-assisted writer identification based on handwriting is mostly concerned in the supervised case, where there are handwriting samples for each writer. Contributions in the unsupervised case are employed in small scales. Recently, 24 Greek inscriptions were clustered with high confidence to six writers based on multiple morphological criteria [12]; 14 samples from a 15th century book were clustered to two groups by applying repeated $k$-means clustering to a vector containing several numerical properties of the handwriting [1].

As mentioned above, in a recent work on the Cairo Genizah [22, 21], the related task of finding pairs of fragments written in the same hand was addressed. It was shown how handwriting data, especially when combined with prior knowledge of script styles, physical measurements, and subject classification, can produce a reliable system. Clustering was applied based on pairwise similarities on a very coarse level to obtain 18 groups of script styles [21]. In this work, we build a clustering tool for working at a much finer scale that aggregates joined pairs into multi-page joins, integrating pair similarity across multiple documents, and reinforcing the join similarity of pairs.

## 3. Graphical Model

One of the ultimate goals of Genizah scholarly work is that of reconstructing—to the extent possible—the leaves of documents (books, pamphlets, letters, etc.) as they were originally bound, before being discarded and later dispersed across the globe. To achieve this goal we make use of pairwise image similarities as well as on the available catalog data and physical measurements of the Genizah fragments.

The catalog data we employ contains four properties of the written text including: subject (Bible, liturgy, etc.), material (paper, vellum), script type (square Ashkenazi, cursive Spanish, etc.), and the existence of cantillation signs (full, partial, none). These are represented by variables $h_i = [h_i^r]$, where $i$ is the leaf index, and $r = 1 \dots 4$ for the four properties (see Section 7 for details).

The four classifications were obtained from the publicly available database of the Friedberg Genizah Project (www.genizah.org), where multiple catalogs were digitized and combined. The data cannot be taken at face value and is therefore part of the inference: Many of the leaves are associated with partial classification or no classification at all. It is often the case that different leaves of the same known join have contradicting classifications. Moreover, even the same leaf might have multiple conflicting classifications. The per-leaf models $\xi_i(h_i)$ capture the fidelity of the inferred variables $h_i$ to the provided catalog data.

In addition to inferring integrated catalog information for each leaf, we also infer the grouping of the leaves. This is encoded as group variables $l_{ij}$. If leaves $i$ and $j$ belong to the same group, the variable's value is 1, and is 0 otherwise. The classification compatibility models $\psi_{ij}(h_i, h_j, l_{ij})$ capture the compatibility of the various leaf classifications in case they belong to the same join. If they do not belong to the same join, the classification of one is irrelevant for the classification of the other.

In contrast to the catalog information, the physical measurements are computed automatically by the authors of [22] and are assumed correct. There are seven measurements, including: the number of text lines, the mean height of the text lines and the average spacing between them, the height and width of the leaf, and the height and width of the text area itself. The models $\varphi_{ij}(l_{ij})$ describe the compatibility of two leaves based on their physical measurements. The estimation of the parameters of the models $\xi_i$, $\psi_{ij}$, and $\varphi_{ij}$ is described in Section 7.

Even more crucial to the success of the clustering process than the above mentioned models is the model derived from the pairwise handwriting-based image similarity of $i$ and $j$. It is provided as pseudo-probabilities that are extracted by the process described in Section 7.2, and stored in the pairwise models $\gamma_{ij}(l_{ij})$.

Lastly, transitivity is enforced by adding models $\chi(l_{ij}, l_{ik}, l_{jk})$ for every three leaves $i$, $j$ and $k$ that capture the constraints [13] $l_{ij} \wedge l_{ik} \rightarrow l_{jk}$, $l_{ij} \wedge l_{jk} \rightarrow l_{ik}$, and $l_{jk} \wedge l_{ik} \rightarrow l_{ij}$. That is, such models are created for every lexicographically ordered pair $(i,j) < (i,k) < (j,k)$, and are set to be $\chi(1,1,0) = \chi(1,0,1) = \chi(0,1,1) = \log\frac{1}{1000}$ and $\log\frac{997}{5000}$ otherwise.

The integrated log-probability of the grouping variables and the inferred catalog information variables incorporates the above mentioned models:

$$\begin{aligned}\log P(\{h_i\}, \{l_{ij}\}) \quad &= \quad \sum_i \xi_i(h_i) + \sum_{ij} \psi_{ij}(h_i, h_j, l_{ij}) \\ &\quad + \sum_{ij} \varphi_{ij}(l_{ij}) + \sum_{ij} \gamma_{ij}(l_{ij}) \\ &\quad + \sum_{ijk} \chi(l_{ij}, l_{ik}, l_{jk}) - \log Z \ ,\end{aligned}$$

where $Z$ is the partition function that depends on the various models and which ensures that the probabilities sum to 1.

## 4. Inference

We perform inference by employing the Dual Decomposition MRF optimization method (DD), discussed in [8]. This method is used to approximate a solution of a primal problem (minimum overall model energy, in our case) by relaxing its Lagrangian dual-problem and decomposing that dual into many efficiently solvable subproblems. Utilizing this method, the lower bound of the minimal model energy is iteratively improved, while producing solutions (label assignments for each model variable) with gradually decreasing energy levels. In practice, the process is not monotonic and convergence after a reasonable number of iterations is not guaranteed for large-scale problems. To perform the DD iterations, we use the code made available by the authors of [20], augmenting it with a tailor-made decomposition process and an inference engine relevant for our model.

The decomposition into subproblems is performed as follows. In the primal problem, we have five types of factors, which can be divided into two groups: the first representing the probabilistic models of the random variables $h_i$ and $l_{ij}$, namely the unary factors $\xi_i(h_i)$, $\varphi_{ij}(l_{ij})$, and $\gamma_{ij}(l_{ij})$, and the second representing mutual relations between variables, which includes the factors $\psi_{ij}(h_i, h_j, l_{ij})$ and $\chi(l_{ij}, l_{ik}, l_{jk})$. The implied factor graph is decomposed into two layers: the data layer, containing all variables and all factors except for $\chi(l_{ij}, l_{ik}, l_{jk})$, and the constraint layer, which includes the grouping variables $l_{ij}$, the unary factors $\gamma_{ij}$ and $\varphi_{ij}$, and the transitivity factors $\chi(l_{ij}, l_{ik}, l_{jk})$. We further decompose each layer into a set of tree-subproblems, such that the union of all trees cover all variables and factors. This is done in the data layer by building one tree per leaf $i$ (root leaf), which contains variables $h_i$, $h_j$, $l_{ij}$ for that leaf $i$ and various other leaves $j \neq i$ (see Sec. 5 for how these are selected), as well as the factors which involve only these variables. The constraint layer is factored similarly, by building one tree per each variable $l_{ij}$, $i < j$ as the root, which contain all $l_{ik}$, $l_{jk}$ variables that are linked to $l_{ij}$ through the transitivity factor $\chi_{ijk}$. Max-Product Belief Propagation is applied for performing inference on each tree, using the libDAI software package [11].

The primal solution is obtained at every iteration by employing a heuristic similar, but not identical, to that described in [8], which in contrast to the original heuristic does not employ the internal subproblem Belief Propagation messages, and uses instead the marginal beliefs. This modified heuristic assigns all variables a fixed order in which the catalog classification variables $h_i$ appear before all grouping variables $l_{ij}$.

Let the ordered variables be denoted $x_1 \ldots x_{n'}$. These variables are scanned one by one. A variable $x_t$ for which all subproblems support the same assignment is labeled accordingly. Otherwise, it is assigned a label $y_t^*$, which minimizes, over all assignments $y$ that were suggested by the subproblems, the estimated contribution of this variable to the total primal-model energy that is based on the variables known so far:

$$\begin{aligned}E_t(y) \quad &= \quad - \sum_{f \in F_t} \log(f(x_t = y | x_1 = y_1^*, \ldots, x_{t-1} = y_{t-1}^*)) \\ &\quad - \log \left( \frac{1}{|\{S | x_t \in S\}|} \sum_{\{S | x_t \in S\}} P_S(x_t = y) \right) \ , \qquad (1)\end{aligned}$$

where the first summation is over the group $F_t$ of factors that are fully assigned by the variables $x_1, \dots, x_t$ and which contain $x_t$, and the second summation is over the marginal beliefs $P_S$ of the subproblems $S$ that contain $x_t$.

## 5. Constraint Subsampling

The graphical model presented in Section 3 cannot be applied naively in order to cluster large collections. For a collection of $n$ leaves, there are $O(n^3)$ factors (dominated by transitivity factors) and $O(n^2)$ subproblems in the Dual Decomposition. Subsampling of variables and constraints is therefore performed to keep the problem manageable.

The subsampling of variables is based on the domain-based expectation that the Genizah collection would contain many small groups and therefore many of the $l_{ij}$ variables are expected to be 0. Thus we remove pairwise variables that have a very low probability of being one. Recall that the pairwise variable $l_{ij}$ participates in several models. If any one of these models indicates that the link is very unlikely, we eliminate the variable. That is, if the visual similarity is extremely low, the catalog classifications are almost entirely contradictory, or the physical measurements of leaves $i$ and $j$ vary significantly, then we drop the variable. The computation of these probabilities is explained in Section 7. A probability threshold of $10^{-2}$ is used, which discards up to 99% of the grouping variables.

A helpful side effect of dropping many variables $l_{ij}$ is the potential removal of factors involving these variables. Furthermore, many of the ternary transitivity models $\chi$ are reduced to binary models in case exactly one of the variables is set to 0 (i.e., removed; if more than one variable out of the three is removed, the model is eliminated altogether).

Nevertheless, since the transitivity constraints dominate the number of overall factors, sampling is performed on these variables as well. We consider the probability for a penalty in the transitivity term, which is naively estimated as the sum of the probabilities of the three assignments $(1, 1, 0)$, $(1, 0, 1)$, $(0, 1, 1)$ independently of the rest of the graph:

$$\gamma_{ij}(1)\gamma_{ik}(1)\gamma_{jk}(0)+\gamma_{ij}(1)\gamma_{ik}(0)\gamma_{jk}(1)+\gamma_{ij}(0)\gamma_{ik}(1)\gamma_{jk}(1) \ .$$

Those factors $\chi$ with an estimated probability of less than 0.01 are discarded. The same process is also applied to the factors based on physical measurements $\varphi$. Overall, a further $40\%$ reduction in transitive factors is obtained.

## 6. Active Learning

Active learning in the context of Bayesian Networks (for general inference problems) is usually employed in the literature for learning the parameters of the graphical model itself [18] or its structure [19]. Here, we employ active learning as part of the inference process itself.

The queries are presented to the user during the iterations based on the data available at the end of the last iteration. This way, the user need not wait for the system to finish the next iteration, and the system dose not need to wait for the user. In the DD framework, some of the iterations are non-improving due to an overshooting step size. Information obtained from those iterations is discarded.

The query variable is selected from among the variables for which not all subproblems agreed on an assignment at the end of the last iteration. We employed three alternative strategies for selecting the next query variable: (i) uncertainty based, (ii) energy impact based, (iii) structure complexity based (see below). Before the start of the next DD iteration, the parameters of the primal problem are updated according to the expert feedback and reprojected to the dual problems.

The most common query selection strategy used in active learning is uncertainty sampling [14]. In the variant we employ the variable is selected for query $x_H^*$ based on entropy maximization:

$$x_H^* = \arg\max_{x_t} - \sum_{y \in Y_t} P(x_t = y) \log P(x_t = y) \ ,$$

where $x_t$ is one of the grouping variables $l_{ij}$ or the catalog classification variables $h_i$, $y$ ranges over $Y_t$, the set of possible labels of $x_t$ suggested by the subproblems, and the probabilities are estimated using the energy terms computed in (Eq. 1), $P(x_t = y) \propto e^{-E_t(y)}$.

An alternative query strategy we employ is to select the variable in which changing the assigned label would create the largest impact on the energy of the model. In a sense, this strategy selects the variable with the lowest amount of uncertainty, in a sharp contrast to the entropy-based strategy. A similar selection strategy was recently employed in the context of the approximate inference method called Cutset Conditioning in which a subset of variables is instantiated to make the rest of the graph singly connected [4]. The variable $x_t$ is evaluated based on the energy of the assignment $y_t^*$ (Eq. 1) in comparison to the set of assignments $Y_t$ suggested by the various subproblems:

$$x_E^* = \arg\max_{x_t} \frac{1}{|Y_t|} \sum_{y \in Y_t} E_t(y) - E_t(y^*) \ . \qquad (2)$$

Lastly, we employ a third alternative, based on minimizing the structural inference complexity invested in each variable. Intuitively, the human expert is asked to assist in those places where most of the computational work is required, which can be viewed as a measure of practical uncertainty.

After each iteration, we record, for every variable $h_i$ and $l_{ij}$ in the model, the total complexity of the Belief Propagation algorithm on all the tree-subproblems containing that variable. This complexity is simply given by the number of non-zero elements in the model matrices of each tree, and provides an estimate on the computational effort exercised by the machine for each variable in the model. Since we sum over the subproblems, the estimated structural complexity per variable also depends on the number of trees that contain that variable, which in turn depends on the underlying graph connectivity. The selected query variable $x^*_C$ is the one with the largest structural complexity. Somewhat surprisingly, this simple novel heuristic seems to experimentally outperform the other strategies presented.

## 7. Data and Model Estimation

We have evaluated our methods on a large subset of the Cairo Genizah that was made public for benchmark purposes (the other leaves presented at the Friedberg Genizah Project's website are available only for viewing). This subset, presented in [22] contains 31,315 leaves, all from the New York (ENA), Paris (AIU), and Jerusalem (JNUL) collections. For the purpose of testing the quality of pairwise similarities, the leaves have already been divided into pairs, labeled as same-join or not, and grouped into splits. Since our goal is different, these splits are not relevant to us.

The available leaves are divided, according to the current lists of known joins, into 1,208 known joins of size up to 72 (see Fig. 4(a)). A large majority of leaves (81.6%) are singletons that are not grouped with other leaves.

### 7.1. Model Parameter Estimation

The parameters of the various models [$\xi_i(h_i)$, $\psi_{ij}(h_i, h_j, l_{ij})$, $\varphi_{ij}(l_{ij})$, and $\gamma_{ij}(l_{ij})$] are estimated empirically using the available join information.

The models involving catalog classifications are based on the transition probabilities from one category classification to another. These are captured by matrices $P^r$, $r = 1 \dots 4$ for subject classification (88 labels), material classification (2 labels), script type (30 labels), and cantillation (3 labels). The row indices of these matrices correspond to the inferred label, and the columns to the given labels. For modeling purposes, a NULL label is added to each category to denote the case where no catalog information of a certain category is provided for a leaf. However, NULL is not allowed as an output label in the inference process.

For each catalog classification category and for each classification label we record the number of times mixing of labels has occurred in the database. There are two cases: the case of multiple catalog entries for a single leaf, and the case of multiple catalog classifications of different leaves of the same known input join (since most manuscript joins contain just one script style, are made of one material, etc.).

The elements of the transition matrices $P^r_{uv}$, $u \neq v$, which denote the probability of mixing label $u$ of category $r$ with label $v$, are estimated as the ratio of the number of times these categories were mixed and the times each category occurred in leaves that either contain multiple catalog entries or appear as part of known joins. $P^r_{uu}$ is computed by considering the fraction of instances where no mixing occurred. Lastly, $P^r_{uv}$, where $v$ is the label NULL, is the frequency of label $u$ in the dataset, disregarding all leaves in which the label of category $r$ is not given.

The input catalog data associated with each leaf $i$ is recorded by the vectors $g^r_i$, where $g^r_i(u)$ denotes the frequency of label $u$ of category $r$ in the available entries for fragment $i$. Typically, one label per category is 1, and the rest are 0. Quite frequently the NULL label is 1, since the catalog information of category $r$ for leaf $i$ is missing.

The first model $\xi_i(h_i)$ captures the fidelity of the estimated catalog classification data to the input classifications. It is estimated as a sum over the four classifications. Each operand $r = 1 \dots 4$ is of the form $\log h^{r\top}_i P^r g^r_i$.

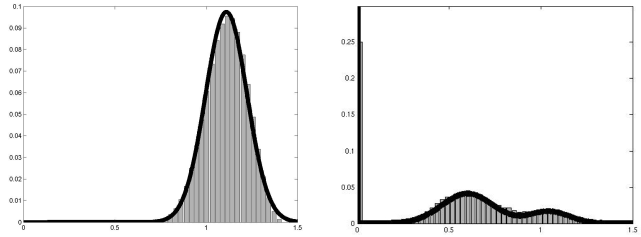
The second model $\psi_{ij}(h_i, h_j, l_{ij})$ ensures that leaves of the same join have similar classifications. Here, too, there is an operand for each of the four classification categories. Each is of the form $\psi^r_{ij}(h^r_i, h^r_j, 1) = \log h^{r\top}_j P'^r h^r_i$, where $P'^r$ is a transition matrix that contains the columns of $P^r$ that correspond to all labels except for the NULL label (after renormalization to form a double stochastic matrix). For leaves of different joins, $\psi^r_{ij}(h^r_i, h^r_j, 0)$ is uniform over the various label values.

The parameters of the physical measurements compatibility model $\varphi_{ij}(l_{ij})$ are estimated based on the seven physical measurements of each leaf (Section 3). There are seven operands $\varphi^s_{ij}(l_{ij})$, $s = 1 \dots 7$, each based on the distance between the measured values of fragment $i$ and fragment $j$. Denote measurement $s$ of fragment $i$ by $m^s_i$. A probabilistic model on the distribution of within join variability of measurements $P^s(|m^s_i - m^s_j|)$ is formed by fitting a Gaussian to the data obtained from the known joins. $\varphi_{ij}(1) = \sum_s \log P^s(|m^s_i - m^s_j|)$, and $\varphi^s_{ij}(0) = 1$.

### 7.2. Image Similarity Modeling

The similarity employed is based on a bag of visual keywords approach as done in [22]. Each leaf is represented as a histogram of keyword prototypes by considering the connected components of the binarized images. To filter out broken letter parts and dark patches arising from stains and border artifacts, the size of the connected component is compared to the height of the lines, which is estimated separately by using horizontal projections.

Each connected component is described by a SIFT descriptor vector [10]. For encoding of the entire leaf, a dictionary is constructed by detecting connected components in a small dataset of 500 documents and clustering the associated SIFT vectors by employing $k$-means. Given a dictionary, a histogram-based methods is used to encode each manuscript leaf as a vector, i.e., we count, for each cluster-



(a) (b)

Figure 1. Fitting parametric distributions to histogram of distances between pairs of leaves that are not known to be joins (a) and pairs known to be joins (b). The first distribution is estimated by a Gaussian, while the second is estimated by a mixture of three Guassians.

center in the dictionary, the number of leaf descriptors closest to it. To account for the variability in fragment sizes, the histogram vector is normalized to sum to 1.

To compare two leaves $i$ and $j$, we compute the $L2$ distance $d_{ij}$ between them. Then, in order to obtain $\gamma_{ij}$ we consider two distributions: the distribution of the similarity values between leaves that are not known to be joins, and the distribution for pairs of leaves that are known joins. We model each distribution parametrically. The first distribution $P(d_{ij}|\text{not join})$ is modeled as a Gaussian distribution, while the second $P(d_{ij}|\text{join})$ is parameterized as a mixture of three Gaussians (see Fig. 1). The prior $P(\text{join})$ is estimated conservatively with accordance to the prevalence of known joins to be $0.001$, and Bayes' rule is used to compute $P(\text{join}|d_{ij})$. The pairwise similarity model is simply given as $\gamma_{ij}(1) = \log P(\text{join}|d_{ij})$ and $\gamma_{ij}(0) = \log(1 - P(\text{join}|d_{ij}))$

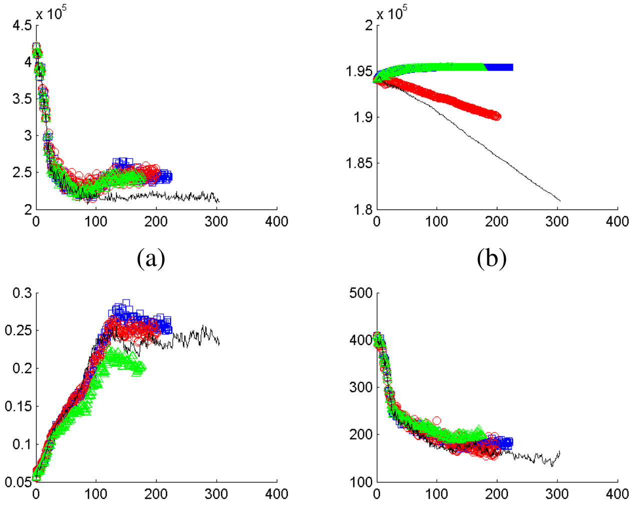
Note that while previous work [22, 21] has put great emphasis on improving the pairwise image similarity, the current version of our system uses, for simplicity purposes, a basic similarity score. More elaborate scores and the application of metric learning tools are expected to further improve our system and are left for future implementations.

# 8. Results

We have conducted two sets of experiments. The first set is used to evaluate the suitability of each query selection method within the active learning framework. To allow for speedy iterations and to reduce the work time of the human expert, these are evaluated on a smaller subset of 5,000 Genizah leaves. The second set of experiments is the deployment of our system to the entire set of 31,315 leaves that were made public.

## 8.1. Comparing Active Learning Query Strategies

To allow a fair comparison among the various alternatives we perform these experiment using a simple interactive protocol in which one query is presented after every DD iteration. A typical iteration for the subset of 5,000 leaves takes 45 seconds, making the presented experiments feasible. However, we did not repeat the experiment multiple times due to the required human effort.



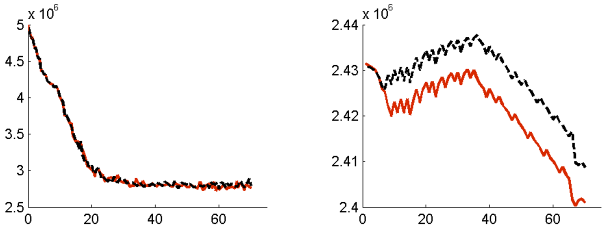
(a) (b)

(c) (d)

Figure 2. Test runs on 5,000 leaves. All plots show results for no-query (blue box), entropy criteria (green triangle), energy difference criteria (red circle), and selection based on structural complexity (black line). (a) Primal energy throughout the iterations. (b) Dual energy in each iteration. (c) Average entropy over all indecisive variables. (d) Mean energy difference (Eq. 2) over all indecisive variable.

We compare four query selection strategies (see Section 6): (a) fully automatic, without any query; (b) query selection of the variable $x^*_H$ with maximum entropy; (c) selection of the variable $x^*_E$ with the largest impact on the energy level; (d) selection of the variable $x^*_C$ with the highest structural complexity. All methods are run till convergence.

The results are presented in Fig. 2. As can be seen (Fig. 2(a)), the overall lowest energy is achieved by the structural complexity method. Since queries are addressed, the structure of the graphical model might change between iterations, therefore the dual energy is not monotonic (Fig. 2(b), note that the scale differs from Fig. 2(a)). The structural complexity criteria has the largest impact on the dual energy, i.e., the dual energy, which is a lower bound on the primal energy, drops considerably during the iterations. Fig. 2(c) reveals that the mean entropy per indecisive variables (those with conflicting subproblems) drops the most, as expected, by selecting $x^*_H$, followed by $x^*_C$. The mean effect of indecisive variables on the energy (Fig. 2(d)), the criterion maximized by $x^*_E$, drops the most for this query as well as for $x^*_C$. Lastly, the runtime per iteration drops the most, as expected, by $x^*_C$ (not shown for lack of space).



(a) (b)

Figure 3. Convergence of two independent runs on 31,315 leaves. (a) Primal energy throughout the iterations. (b) Dual energy (note: different y-scale).

## 8.2. Reconstructing a Large Chunk of the Genizah

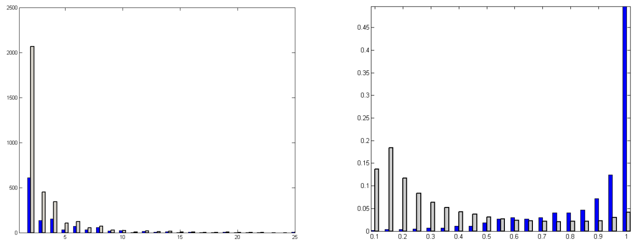
Due to the required human effort, the runs on the entire data set were less comprehensive and only the structural complexity method that was believed to be the best choice was tested. Each complete iteration takes 25 minutes, and the human expert provides feedback for the next iteration while the current iteration runs. We ran two experiments, each for 70 iterations, which differ at the stage at which the human operator took a break. For example, in one of the experiments, the operator took a first break after two hours, and in the second, three hours passed before the first break.

The energy levels of the system throughout the runs are captured in Fig. 3(a) for the primal energy and 3(b) for the dual one. The regions in which the dual energy monotonically increases correspond to the rest periods of the human operator in which the computer was not halted. After 30 hours, each experiment was terminated, and the transitive closure of the lowest energy result obtained by the two runs was further evaluated as the clustering output.

The change to the grouping landscape by the incorporation of new joins is depicted in Fig. 4(a). As can be seen, there is a shift to larger joins due to the clustering process. Fig. 4(b) depicts the distribution of similarity values $e^{\gamma_{ij}}$ for pairwise variables that were either assigned $l_{ij} = 0$ or $l_{ij} = 1$. As can be seen, in both cases the entire range of values is used, although, as expected, larger values of $\gamma_{ij}$ are more likely to lead to a value of 1.

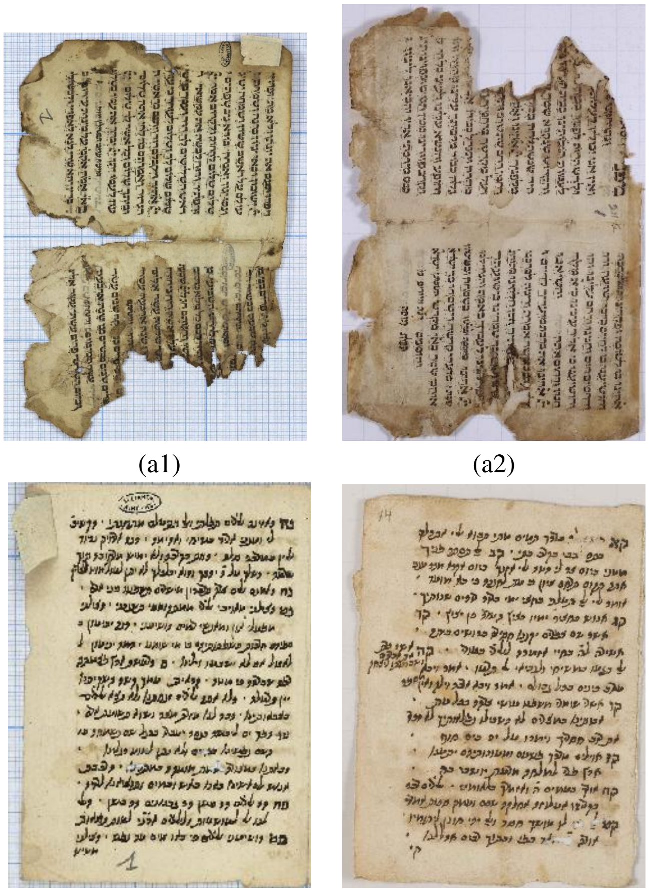
Examples of newly found joins are presented in Fig. 5. In order to quantitatively evaluate the new joins, we have evaluated two groups of joins that were affected by the clustering process. Among the 50 modified joins (graph cliques, not just pairs) with the highest mean $\gamma_{ij}$ value, 100% were found to be correct. Among the 50 modified joins with the lowest mean $\gamma_{ij}$ value 78% were verified.

Fig. 6 shows results in which ambiguous or missing catalog data was inferred. While the success rates for these were not computed yet, it seems that many of the outcome classifications are correct. Finally, the subsampling of the transitivity constraints leads to violations of transitivity, as can be seen in Fig. 7. A fraction of 1.8% of the new edges violate a missing transitivity constraint.



(a) (b)

Figure 4. Results of the run on 31,315 leaves. (a) the histogram of join sizes: known (solid blue bar) and after the run (grey bar). For clarity, singletons and sizes larger than 25 were omitted from the plot. (b) the distribution of the image similarity measure for each of the two classes "newly found joins" (solid blue) and "inferred to be non-joins" (grey).



(a1) (a2)

(b1) (b2)

Figure 5. Examples of newly-found joins. (a1) a page from AIU; (a2) a joined page from ENA; (b1) a page from AIU; (b2) a joined page from ENA, a third joined page from ENA is omitted for lack of space.

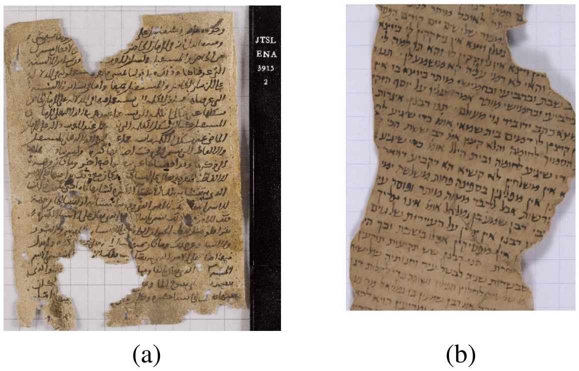


Figure 6. Pages with initially ambiguous catalog data. (a) Initially a singleton with no catalog data other than material (paper), it is found to be a philosophical work with a Naskhi script type. (b) Initially the subject classification had three equally probable options, two cantillation options and a very broad script type categorization. The inference determines it to be a religious exegesis essay with a Tiberian semi-cursive script and partial cantillation.

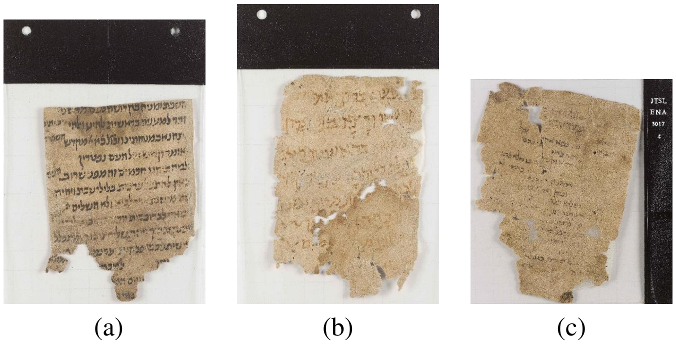


Figure 7. Example showing a transitivity constraint that was violated in the final assignment. While (a) and (b) are known to be joins, and (a) is similar in handwriting and catalog data to (c), (b) and (c) do not look the same, and the transitivity constraint was left out during the constraint subsampling stage. In the clustering assignment (a) and (c) were incorrectly joined.

## 9. Discussion and Future Work

Despite the enormous efforts and expenses invested in digitization of manuscripts, computer tools to help analyze them are severely wanting. The methods presented here are applicable to other corpora as well. In a broader context, since clustering is ill-posed, active clustering is a key enabling technology for large-scale tagging of image collections. We help bridge a gap in the literature and show a system that without much optimization is scalable to tens of thousands of objects on a conventional PC. Lastly, our newly formulated query selection strategy, which is based on the notion of structural complexity, seems to outperform other criteria suggested in the past for active learning or for approximate inference.

## References

[1] M. Aussems and A. Brink. Digital palaeography. In *Palaeography and Codicology in the Digital Age*, 2009. 2

[2] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *International Conference on Data Mining*, 2004. 2

[3] S. P. Chen, J. Hsiang, H. Tu, and M. Wu. On building a full-text digital library of historical documents. In *Int. Conference on Asian Digital Libraries*, 2007. 1

[4] F. Eaton and Z. Ghahramani. Choosing a variable to clamp: Approximate inference using conditioned belief propagation. In *Int. Conf. on AI and Statistics*, 2009. 4

[5] M. Glickman. *Sacred Treasure, the Cairo Genizah*. 2010. 1

[6] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV09*. 2

[7] R. Huang and W. Lam. An active learning framework for semi-supervised document clustering with language modeling. *Data Knowl. Eng.*, 68:49–67, January 2009. 2

[8] N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007. 3

[9] M. Lidman. Google plans to offer Dead Sea Scrolls online. *The Jerusalem Post*, 20 Oct. 2010. 1

[10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 5

[11] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 2010. 3

[12] M. Panagopoulos, C. Papaodysseus, P. Rousopoulos, D. Dafi, and S. Tracy. Automatic writer identification of ancient Greek inscriptions. *PAMI*, 2009. 2

[13] S. Pellegrini, A. Ess, and L. V. Gool. Improving data association by joint modeling of pedestrian trajectories and groupings. In *ECCV*, 2010. 2, 3

[14] B. Settles. Active learning literature survey. Computer Sciences TR 1648, U Wisconsin–Madison, 2009. 4

[15] N. Shental, A. Zomet, T. Hertz, and Y. Weiss. Pairwise clustering and graphical models. In *NIPS*, 2003. 2

[16] R. Smith. An overview of the Tesseract OCR engine. In *Int. Conf. on Document Analysis and Recognition*, 2007. 2

[17] Z. Stone, T. Zickler, and T. Darrell. Autotagging facebook: Social network context improves photo annotation. *CVPR Workshop on Internet Vision*, 2008. 2

[18] S. Tong and D. Koller. Active learning for parameter estimation in Bayesian networks. In *In NIPS*, 2000. 4

[19] S. Tong and D. Koller. Active learning for structure in Bayesian networks. In *Int. Joint Conf. on AI*, 2001. 4

[20] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *ECCV*, pages 596–609, 2008. 3

[21] L. Wolf, N. Dershowitz, L. Potikha, T. German, R. Shweka, and Y. Choueka. Automatic paleographic exploration of Genizah manuscripts. *Codicology and Palaeography in the Digital Age II*, 2011. 2, 6

[22] L. Wolf, R. Littman, N. Mayer, T. German, N. Dershowitz, R. Shweka, and Y. Choueka. Identifying join candidates in the Cairo Genizah. *IJCV*, 2010. 1, 2, 3, 5, 6