Interactively Guiding Semi-Supervised Clustering via Attribute-based Explanations

Shrenik Lad and Devi Parikh

Virginia Tech

Abstract. Unsupervised image clustering is a challenging and often ill-posed problem. Existing image descriptors fail to capture the clustering criterion well, and more importantly, the criterion itself may depend on (unknown) user preferences. Semi-supervised approaches such as distance metric learning and constrained clustering thus leverage user-provided annotations indicating which pairs of images belong to the same cluster (must-link) and which ones do not (cannot-link). These approaches require many such constraints before achieving good clustering performance because each constraint only provides weak cues about the desired clustering. In this paper, we propose to use image attributes as a modality for the user to provide more informative cues. In particular, the clustering algorithm iteratively and actively queries a user with an image pair. Instead of the user simply providing a must-link/cannot-link constraint for the pair, the user also provides an attribute-based reasoning e.g. "these two images are similar because both are natural and have still water" or "these two people are dissimilar because one is way older than the other". Under the guidance of this explanation, and equipped with attribute predictors, many additional constraints are automatically generated. We demonstrate the effectiveness of our approach by incorporating the proposed attribute-based explanations in three standard semi-supervised clustering algorithms: Constrained K-Means, MPCK-Means, and Spectral Clustering, on three domains: scenes, shoes, and faces, using both binary and relative attributes.

1 Introduction

Image clustering is the problem of grouping images such that similar images fall in the same clusters and dissimilar images fall in different clusters. Image similarity as perceived by humans is difficult to capture by existing image descriptors. Moreover, the notion of similarity itself is often ill-defined. For instance, one user may want to cluster a group of faces such that people of the same gender, race and age fall in the same cluster, while a different user may want to cluster faces based on accessories such as glasses, makeup, etc. Clearly, without supervision, clustering is an ill-posed problem.

Semi-Supervised clustering approaches [1–8] leverage user-provided pairwise constraints either to learn an appropriate distance metric in the feature space (distance metric learning [5–8]) or to guide a clustering algorithm towards correct clusters (constrained clustering [1–4]). These pairwise constraints are either

Fig. 1: An Illustration of our approach. The system interactively queries the user with a pair of images, and solicits "must-link" and "cannot-link" response along with an attribute-based explanation. Using attribute predictors, the system can automatically generate additional pairwise constraints.

must-link or cannot-link, indicating that the two images in the pair should belong to the same cluster or different clusters resp. These constraints allow the user to inject their domain knowledge or preferences in the clustering algorithm.

A fundamental problem with these approaches is that they require a large number of pairwise constraints in order to achieve decent performance. This is because the pairs of images to be annotated are typically randomly chosen, and are likely to be redundant. Active clustering approaches [9–11] reduce the number of required constraints by iteratively and actively selecting a pair to be annotated by a user in the loop. But each constraint still remains a weak low-level indication of the desired clustering. For instance, active PCK-Means [10] requires 5000 constraints to achieve 30% accuracy on a dataset of 500 face images as reported in [9].

We propose accessing the user's mental model of the desired clustering via a mid-level semantic representation i.e. visual attributes. Some attributes like *wearing glasses* or *having four legs* are binary in nature while others like *smiling* and *attractive* are relative [12–15]. Both binary and relative attributes have been shown to improve various computer vision tasks like image search [16, 17] and classification [13, 14, 18, 19]. In this paper, we show that attributes can be used for achieving more accurate clusterings when using semi-supervised clustering algorithms.

Specifically, for an actively selected image pair, the user provides a must-link or cannot-link constraint, along with an attribute-based explanation for that constraint. This explanation can be in terms of binary attributes e.g. "these two shoes are similar because both are *red* and *have high-heels*", or in terms of relative attributes e.g. "these two people are dissimilar because one is significantly *older* than the other". This form of an explanation is intuitive for a user. Equipped

with pre-trained attribute predictors, the machine can infer a large number of constraints from just one user response. For instance, in the first example the machine can identify all shoes that are red and have high-heels and add must-link constraints between all possible pairs. In the second example the machine can identify pairs of images where the difference in age is even larger than the query pair, and add cannot-link constraints between these pairs. This results in significant gains in clustering accuracy. The scores of the attribute predictors can be used to assess the confidence of these automatically added constraints.

We demonstrate the effectiveness of our approach by incorporating binary and relative attribute-based explanations in three diverse semi-supervised clustering algorithms: Constrained K-Means [1], MPCK-Means [2] and Spectral Clustering [20] on three different domains: scenes (SUN [21]), faces (PubFig [13]) and consumer products (Shoes [22]) using real human studies. We also evaluate our system on user-specific or personalized clustering and show that our approach can be used to guide the system towards different clustering outputs.

2 Related Work

Semi-Supervised clustering approaches are either constraint-based [1–4] or metric-based [5–8]. Both these approaches use pairwise constraints (must-link and cannot-link) to guide a clustering algorithm towards correct clusters or to learn an appropriate distance metric for the given data. Basu et. al. [2] propose an integrated framework that incorporates constrained-clustering and distance metric learning. These approaches rely on large number of constraints in order to achieve satisfactory clustering performance even on small datasets with low-dimensional features.

Active clustering works like [9,11] reduce the number of constraints by querying a user on actively chosen pairs instead of completely random pairs. But these approaches are specific to the clustering algorithms they use. Biswas et. al. [9] present an active clustering algorithm that goes through all possible pairs and then re-clusters the data in order to identify the most informative pair. Hence, they use a simple Minimal Spanning Tree based clustering algorithm that can be run many times in a reasonable amount of time. Wauthier et al. [11] propose an active learning algorithm specifically for spectral clustering. Moreover, even though these approaches reduce the number of constraints, each constraint remains a weak indication of the desired clustering. Our attribute-based approach on the other hand allows the user to convey rich information which can be propagated to other unlabelled pairs in the dataset. It is also not specific to any clustering algorithm. We show that our approach can improve performance across different clustering algorithms including a mix of constraint- and metric-based approaches.

Attributes are mid-level concepts that have been extensively used for a variety of tasks in computer vision [13–19,22–26]. The vocabulary of attributes can be pre-defined or it can be discovered [27,28]. Recently, Attributes have been used as a mode of communication between humans and machines [16,18,19]. Donahue and Grauman [19] use spatial annotations and binary attributes to al-

low an annotator to provide an explanation for a particular class label. Whittle Search [16] uses relative attributes to allow a user to convey (potentially actively solicited [29]) feedback to a search engine to quickly find the desired target image. [15] uses attributes to avoid semantic drift in a bootstrapping approach by enforcing known relationships between categories (eg: "banquet halls are bigger than "bedrooms). An exhaustive set of such relationships are provided to the system as input (spirit to zero shot learning [14]). In our approach, we use attributes to propagate constraints to pairs of images in an interactive semi-supervised clustering setting. Parkash and Parikh [18] use relative attributes feedback (e.g. "this image is too open to be a forest") to identify additional negative examples from an unlabelled pool of images to train a classifier. Attributes have also been used for fine-grained classification with a human-in-the-loop who conveys domain knowledge by answering attribute-based questions at test time to aid the machine [25]. To the best of our knowledge, ours is the first work on interactive semantic explanation-based clustering of images. These explanations are used to propagate domain knowledge to other pairs of images in terms of both must-link and cannot-link constraints. The "domain knowledge" can easily be user-preferences, making our approach a natural fit to personalized clustering.

Crowd-in-the-loop clustering has been explored for dealing with a large collection of images. Crowdclustering [8,30] shows small subsets of images to MTurk workers and they are asked to annotate the images with keywords. Similarity of image pairs is then determined based on how many keywords they share. Tamuz et al. [31] learn a similarity measure for images in the form of a kernel matrix by posing triplet comparison queries to a crowd. Our approach can be used to make such crowd-in-the-loop efforts more cost effective by reducing human effort via attribute-based explanations.

The rest of this paper is organized as follows: Sec 3 briefly describes three semi-supervised clustering approaches that we augment using attribute-based explanations. Sec 4 describes the details of incorporating these explanations. Sec 5 presents our experimental results. We conclude the paper in Sec 6.

3 Semi-Supervised Clustering Approaches

Semi-Supervised clustering incorporates background knowledge in the form of pairwise constraints: must-link and cannot-link. Let M be the set of must-link constraints and C be the set of cannot-link constraints. The pairwise constraints lead to two types of transitive closure: 1) $(a, b) \in M$ and $(b, c) \in M \implies (a, c) \in M$. 2) $(a, b) \in M$ and $(b, c) \in C \implies (a, c) \in C$.

In the following subsections, we briefly describe the three semi-supervised clustering approaches that we augment using our attribute-based approach.

3.1 Constrained K-Means

Constrained K-Means or COP K-Means [1] is a modification of K-Means to incorporate pairwise constraints. Specifically, during the assignment step of K-Means, instead of assigning every datapoint to its nearest cluster, the datapoint is assigned to the nearest cluster which would not result in violating any constraints.

The centroid estimation step remains the same. We use our own implementation of COP K-Means. During the assignment step, there are situations when a point cannot be assigned to any of the clusters because every assignment would result in violating some constraint. In such cases, we assign the point to the cluster with minimum violations.

3.2 MPCK-Means

Metric Pairwise Constrained K-Means or MPCK-Means [2] combines distance metric learning and constrained clustering in a unified framework. It minimizes the following objective function:

$$\sum_{\boldsymbol{x}_i} \Big(\|\boldsymbol{x}_i - \boldsymbol{\mu}_{l_i}\|^2_{D_{l_i}} - \log\left(\det(D_{l_i})\right) \Big) + \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in M} w_{ij} \mathbb{1}[l_i \neq l_j] + \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in C} w_{ij} \mathbb{1}[l_i = l_j] \tag{1}$$

Here, $\boldsymbol{x}_i$ denotes the i^{th} datapoint, l_i is the cluster-id of $\boldsymbol{x}_i$, $\boldsymbol{\mu}_{l_i}$ is the corresponding centroid, D_{l_i} is the matrix that parameterizes the Mahalanobis distance metric for cluster l_i, w_{ij} is the weight (or importance) of the pairwise constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\mathbb{1}$ denotes the indicator function. The objective function penalizes those assignments which violate many constraints (last two terms of objective) and also assignments where the datapoints are far from their respective centroids w.r.t. the learnt distance metric (first term of objective). The function is minimized by an Expectation Maximisation (EM) approach, where the E-step assigns points to the closest centroids according to the learnt distance metric, and the M-step estimates the centroids and learns the distance metric. We use the code provided by the authors. In our experiments, we enforce a common distance metric for all clusters. This was found to perform better [2].

3.3 Spectral Clustering

We use the spectral clustering algorithm from [20]. It consists of the following steps:

- Compute the $N \times N$ affinity matrix A for the N datapoints. $A_{ij} = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$ for $i \neq j$, $A_{ii} = 0$. The parameter σ controls how fast the affinity falls with increasing distances.
- Compute the Laplacian L of the affinity matrix and find its top K eigenvectors.
- Stack the eigenvectors as columns to form the $N \times K$ matrix E and normalize the rows of E to form a $N \times K$ matrix Y.
- Finally, cluster the N rows of matrix Y into K clusters using K-Means. The assignment of each row provides the cluster-id for each of the N datapoints.

Kamvar et al. [32], present a semi-supervised version of this spectral clustering algorithm. If $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a must-link constraint, then $A(i,j)$ and $A(j,i)$ are set to 1 (zero distance) and if $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a cannot-link constraint then $A(i,j)$ and $A(j,i)$ are set to 0 (∞ distance). In our experiments, we set the value of σ as the average pairwise distance between datapoints.

4 Approach

Let U denote the set of N images $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ that we want to cluster into K groups. Let $\{a_1, a_2, \ldots, a_Q\}$ be the Q attributes in the predefined attributes vocabulary associated with the images. Recall that M and C are the set of must-link and cannot-link constraints respectively and we represent them by symmetric matrices M and C of dimensions $N \times N$, where $M(i,j)$ denotes whether image $\boldsymbol{x}_i$ and image $\boldsymbol{x}_j$ are must-linked and similarly $C(i,j)$ denotes whether image $\boldsymbol{x}_i$ and image $\boldsymbol{x}_j$ are cannot-linked. Initially M and C are empty, and will get filled up as we query the user on pairs of images at each iteration. Ideally, we would like to fill these matrices as accurately and in as few user iterations as possible. Our approach is as follows:

1. Cluster images into K clusters using unsupervised K-Means.
2. Identify the most uncertain pair (Sec 4.1) and present it to the user as query.
3. User provides a must-link or cannot-link label for the pair along with an attribute-based explanation (Sec 4.2). Perform transitive closure on all accumulated ground truth constraints.
4. Convert attribute explanation to additional (possibly noisy) constraints (Sec 4.3).
5. Cluster the images with the updated set of constraints using a semi-supervised clustering algorithm.
6. Repeat from step 2.

4.1 Active selection of pair

A pair of very similar images are likely to be must-links (ML), and images that are very different are likely to be cannot-links (CL). A must-link or cannot-link label on a pair of images that is neither too similar nor too dissimilar is likely be to be most informative for a semi-supervised clustering algorithm.

Let d_{ij} denote the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ in low-level feature space. Let *label* be a random variable that takes two states, ML (must-link) and CL (cannot-link). Let P^{ij}_{ML} and P^{ij}_{CL} be the probabilities of the pair (i,j) being a ML and CL respectively i.e.

$$P^{ij}_{ML} = P(label = ML \mid d_{ij}), \tag{2}$$

$$P^{ij}_{CL} = P(label = CL \mid d_{ij}) \tag{3}$$

Using Bayes rule

$$P(label = ML \mid d_{ij}) \propto P(d_{ij} \mid label = ML) * P(label = ML) \tag{4}$$

$$P(label = CL \mid d_{ij}) \propto P(d_{ij} \mid label = CL) * P(label = CL) \tag{5}$$

The distributions $P(d_{ij} \mid label = ML)$ and $P(d_{ij} \mid label = CL)$ are approximated as Gaussians. Their parameters are estimated from the pairwise distances of all pairs in same clusters and different clusters respectively, according to the current clustering. $P(label = ML)$ and $P(label = CL)$ are the proportion of pairs in same and different clusters. Similar in spirit to active learning for training classifiers [33], we select the pair with the highest entropy under the distribution $P(label \mid d_{ij})$ and present it to the user to solicit *label*.

4.2 Attribute-based Explanation

We assume that the vocabulary of attributes is pre-defined and the attribute predictors have been trained offline. Presence of binary attributes can be predicted by standard classification tools [13, 14, 23]. Relative attribute models on the other hand predict the relative strength of attribute presence in images using ranking functions for each attribute in the vocabulary [12]. We use attributes (both binary and relative) to allow the user to convey to the machine the semantics that guide the similarity measure between images. This notion of similarity may be based on common sense knowledge, specific domain knowledge, or user's preferences.

In case of binary attributes, we allow attribute-based explanations of the following form: "These two faces are *similar* because both are *young* and *white*", "These two scenes are *dissimilar* because one is *natural* and other is not". In this way, the user can convey similarity along which visual properties suffices to make images similar, and discrepancies along which attributes are sufficient to make the images dissimilar. With relative attributes, the explanation for a must-link pair is of the form "These two shoes are *similar* because both are similarly *shiny* and *formal*" and "These two people are *dissimilar* because one is significantly *younger* than the other". In addition to relevance of attributes, this allows the user to indicate the required sensitivity (or the lack of it) to discrepancies along an attribute that the clustering algorithm should have to achieve the desirable clustering. This rich explanation allows the machine to infer much more than a single constraint from the user's response (as described in the next subsection).

One could argue that instead of actively querying the user, perhaps the user can describe the clustering criterion to the system in terms of attributes from the very beginning. There are two concerns with such an approach. First, if the attribute vocabulary is very large, indicating the relevance of each attribute would be cumbersome. This is especially the case if the user were to specify the desired sensitivity to differences in attribute strengths for each attribute (e.g. how similar does the age of two people need to be for them to fall in the same cluster? is a certain difference between two shoes in terms of heel height sufficient to put them in different clusters?). In our approach, the user can simply look at a pair of images and provide a response specific to those images. Second, the criterion for clustering a large number of images may not be crystal clear in the user's mind till they see specific pairs of images and are forced to think explicitly about the desired clustering output with respect to those images.

4.3 Incorporating Attribute-based Explanation

Binary Attributes: Without loss of generality, lets assume that the cannot-link feedback on an image pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is "These two images are *dissimilar* because one is a_q and the other is not". The dissimilar label in the feedback provides a ground truth constraint that $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is cannot-linked and hence $C(i,j) = C(j,i) = 1$. The attribute-based explanation suggests that pairs of images where one has a_q and the other does not are likely to belong to different clusters. The machine can thus infer additional pairs $S_1 \times S_2$ as cannot-link constraints in the

matrix C, where $S1$ is the set of images where a_q is predicted to be present using the pre-trained binary classifier for a_q, and S_2 is the set of images where a_q is predicted to be absent.

If the must-link feedback given by the user on the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is of the form "These two images are *similar* because both are $a_{q_1}, a_{q_2}, \ldots a_{q_T}$", it suggests that having attributes $a_{q_1}, a_{q_2}, \ldots a_{q_T}$ in common are sufficient to qualify two images as being similar. After adding the ground truth must-link constraint for the pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the machine then uses the attribute classifiers of $a_{q_1}, a_{q_2}, \ldots a_{q_T}$ to find images that have all these attributes present in them. If the set of images is S, then all pairs in $S \times S$ are added as must-link constraints in the matrix M.

Relative Attributes: If the cannot-link feedback on image pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is "These two images are *dissimilar* because $\boldsymbol{x}_i$ is way too a_q than $\boldsymbol{x}_j$", the machine first adds a ground truth cannot-link constraint $C(i,j) = C(j,i) = 1$. It then creates the set S_G which consists of all images where degree of presence of a_q is predicted to be greater than $\boldsymbol{x}_i$ and the set S_L which consists of all images where degree of presence of a_q is predicted to be less than $\boldsymbol{x}_j$. The pairs in $S_G \times S_L$ are added as cannot-link constraints in the matrix C, because all such pairs have a difference in relative strengths of a_q even larger than $(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which the user indicated is already too large for them to belong to the same cluster. For a must link feedback "These two images are *similar* because both are equally $a_{q_1}, a_{q_2}, \ldots a_{q_T}$", machine computes S_S, the set of images having the degree of presence of each attribute $a_{q_1}, a_{q_2}, \ldots a_{q_T}$ between the ranges specified by $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The pairs in $S_S \times S_S$ are added as must-link constraints in the matrix M along with the ground truth constraint $M(i,j) = M(j,i) = 1$.

Note that representing images in the attribute-space instead of a low-level feature-space is not likely to diminish the value of such attribute-based explanations. Information about which attributes are relevant to the clustering and which ones are not is still valuable. Moreover, we do not assume that the desired clustering is precisely defined by the available vocabulary of attributes. We only assume that some information about the clustering criterion can be explained in terms of some of the available attributes, and hypothesize that this information is significantly richer than the pairwise constraints alone. Both these hypotheses are empirically validated in our experiments (Sec 5.2).

Purity of Constraints: The constraints generated by the machine from attribute-based explanations are not expected to be 100% accurate because of various factors like error of attribute classifiers/rankers, discrepancy in perception of attributes between the machine and user, erroneous feedback on the part of the user, inconsistencies between the clustering criterion and attributes, etc. On average, we found the accuracy of our attribute-based constraints to be 86% for cannot-links and 60% for must-links. We solicit attribute explanations from the user only up to a certain point (100 iterations in our experiments). This is because the vocabulary of attributes is limited. After a point, attributes do not add too many new constraints (especially with binary attributes). For instance, we find that from 20th to 100th iterations, the number of newly added constraints is only 13% of what are added in the first 20 iterations. From that point onwards, the user provides only must-link and cannot-link labels for pairs and the number of impure constraints goes down with iterations. This gives the

user an opportunity to fine tune the clustering after the broad brush strokes of attributes give them a head start.

Hard and Soft Constraints: The constraints created by the machine can be incorporated in two ways: hard and soft. In the hard setting, all the generated constraints are given confidence = 1 i.e. they are treated as being as important and reliable as ground truth constraints. On the other hand, in the soft setting, the machine-generated constraints are assigned confidences between 0 and 1 based on the confidence of attribute classifiers in predicting the attributes. This may provide robustness to incorrect attribute predictions. With binary attribute-based constraints, the confidence of a generated cannot-link constraint $(\boldsymbol{x}_{i'}, \boldsymbol{x}_{j'})$ is $c_{i'} * c_{j'}$, where $c_{i'}$ and $c_{j'}$ are the confidences of presence and absence of the attribute a_q in images $\boldsymbol{x}_{i'}$ and $\boldsymbol{x}_{j'}$ respectively. In case of a generated must-link constraint $(\boldsymbol{x}_{i'}, \boldsymbol{x}_{j'})$, $c_{i'}$ is the average confidence of presence of the q_T attributes in image $\boldsymbol{x}_{i'}$, and $c_{j'}$ is the average confidence of presence of the q_T attributes in image $\boldsymbol{x}_{j'}$. The final confidence assigned to the pair is $c_{i'} * c_{j'}$. When the cannot link feedback on pair $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ involves a relative attribute a_q, we first sort all the images according to the ranking function for attribute a_q. The confidence of a generated cannot-link constraint $(\boldsymbol{x}_{i'}, \boldsymbol{x}_{j'})$ is proportional to the total number of images that lie between $\boldsymbol{x}_{i'}$ and $\boldsymbol{x}_i$ and between $\boldsymbol{x}_{j'}$ and $\boldsymbol{x}_j$ in the sorted list. In case of a must-link feedback, the confidence of a generated must-link constraint $(\boldsymbol{x}_{i'}, \boldsymbol{x}_{j'})$ is proportional to how close $\boldsymbol{x}_{i'}$ and $\boldsymbol{x}_{j'}$ are in the sorted orderings on average across the q_T attributes. All the confidences are normalized to lie between 0 and 1. When multiple attribute explanations indicate the same pair to be cannot-linked or must-linked, we assign a confidence based on the explanation leading to the highest confidence.

Incorporation of Soft Constraints: MPCK-Means directly allows for soft-confidences in their formulation via the weights w_{ij} associated with each constraint in Equation 1. We set the weight w_{ij} to be the confidence computed for $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ from user feedback. In spectral clustering, the affinity matrix A captures the similarity of datapoints. If the confidence of a cannot-link constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is c, we set $A(i, j) = 1 - c$, and if the confidence of a must-link constraint $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is c, we set $A(i, j) = c$. COP K-Means does not allow handling of soft constraints and so we modify it slightly. During the assignment of a point to a cluster, instead of computing the number of violated constraints, we assign a datapoint to the cluster with the least sum of confidences of violated constraints.

5 Experiments and Results

5.1 Experimental Setup

Datasets: We demonstrate our approach on three domains. **Scenes:** We use the SUN Attributes dataset [34] that consists of indoor and outdoor scenes along with pre-trained attribute classifiers for 102 attributes like natural, open, enclosed, warm etc. Scene categories in the SUN dataset [21] are organized according to a hierarchy, where the first level has super-ordinate categories like indoor and outdoor scenes, the second level has basic-level categories like sports,

Fig. 2: These are the 6 ground truth clusters in SUN600 dataset. They correspond to categories like transportation, industrial regions, sports/recreation etc.

transportation, desert, etc. and the third (last) level has more than 700 fine-grained categories. We choose six out of 16 categories from the second level of the hierarchy and create a dataset of 600 images called SUN600. The six categories cover scenes of various types like transportation, home/hotel indoor scenes, sports/recreation, etc. An illustration of the clusters can be seen in Fig 2. We use GIST [35] features for this dataset. **Faces:** We use the Public Figures Face Database [13] which consists of face images of 60 different public figures in their development set. Attribute predictions for 73 binary attributes like male, white, smiling, wearing glasses etc are available with the dataset. We use 570 images involving 38 people (Sec 5.3 has details regarding how the clusters are defined). The features used are pyramid HOG features [36]. **Shoes**: Berg et al. [22] provide a shoes dataset which is also used in [16]. Kovashka et al. [16] provide ranking predictors for 10 relative attributes like shiny, formal, open, sporty etc. We choose 1000 images belonging to four different categories namely, boots, flats, high-heels, and sneakers, and refer to this dataset as Shoes1000. We use the 960-d GIST features provided with the dataset. For SUN600 and Shoes1000 we assume that the categories are the desired clusters.

Human Studies: We collect attribute-based explanations for image pairs through real human studies on Amazon Mechanical Turk. For each dataset, we run unsupervised K-Means 300 times and choose the max-entropy pair each time according to our formulation (Sec 4.1). We present these pairs to users on MTurk and solicit attribute-based explanations. In our experiments, we pick a query at each iteration from this pool of pairs. This allows us to collect data offline and run exhaustive experiments on the collected data. Our MTurk interface shows the desired clustering on top by displaying 4 randomly chosen images from each cluster. These images are chosen offline for each dataset and are fixed across all HITs. Note that this is just to inform the workers of the desired clustering. In a real application, the users already have a clustering in mind. The clustering is not described to users in any other form. The users observe the clusters and visualize the similarity measure. They are presented with a pair and asked to indicate whether the two images belong to the same or different clusters. They are also asked to select attribute(s) from a list as explanation. In case of a cannot-link (CL) label, users are asked to select the main property (attribute) that makes the two images different. In the case of a must-link (ML) label, users are asked to select as many attributes as necessary to make the two images similar. Multiple

attributes for CL can be easily incorporated, but a single attribute is least time consuming. Single attributes are often insufficient to establish similarity, hence users are free to select more for ML responses. One HIT had 5 image pairs and workers were paid 5 cents per HIT. To ensure good quality of responses, we took a majority vote over 3 different workers for the ML/CL response.

Baselines: We compare our approach with the following baselines

(a) K-Means: Unsupervised K-Means clustering without any pairwise constraints.
(b) Random: User provides a must-link or cannot-link constraint on a randomly chosen pair at each iteration. No attribute-based explanation is taken.
(c) Semi-Random: User provides constraint on a pair chosen randomly from outside the transitive closure of all previously labelled pairs. This is a stronger baseline than (b) which may solicit feedback on redundant pairs. No Attribute-based explanation is taken.
(d) Max-entropy: User provides constraints on the highest entropy pair as described in Sec 4.1. No attribute-based explanation is taken. A comparison to (c) semi-random baseline demonstrates the effect of our entropy formulation even when constraints are chosen from outside the transitive closure.
(e) Many-Random: The above baselines add only 1 constraint per iteration. To evaluate if our approach is benefiting from just more constraints as opposed to *meaningful* constraints, we compare with a baseline that expands the set of constraints every iteration by adding as many must-link and cannot-link constraints as our approach generates, but between random pairs of images. We experiment with both hard and soft constraints.
(f) Attributes-hard: User provides a constraint on the highest entropy pair along with an attribute-based explanation. Machine generates hard constraints.
(g) Attributes-soft: User provides a constraint on the highest entropy pair along with an attribute-based explanation. Machine generates soft constraints.

Evaluation Metric: We use the Jaccard's coefficient (JCC) to quantify the clustering accuracy. It is defined as follows:

$$\text{Jaccard's coefficient (JCC)} = \frac{SS}{SS+SD+DS}$$

where SS is the number of pairs which belong to the same clusters in the ground truth (GT) as well as the clustering output, SD is the number of pairs which belong to same clusters in GT but are incorrectly placed in different clusters by the clustering algorithm and DS is the number of pairs which belong to different clusters in GT but are placed in same clusters by the algorithm.

5.2 Results

The semi-supervised clustering algorithm can be any one of the 3 algorithms described in Sec 3. Results on Shoes1000 and SUN600 with some of the clustering algorithms are shown in Fig 3. The results for more clustering algorithms can be found in our supplementary material. Similar trends are observed when other metrics like F-Measure, Rand-Index, or NMI are used to measure performance.

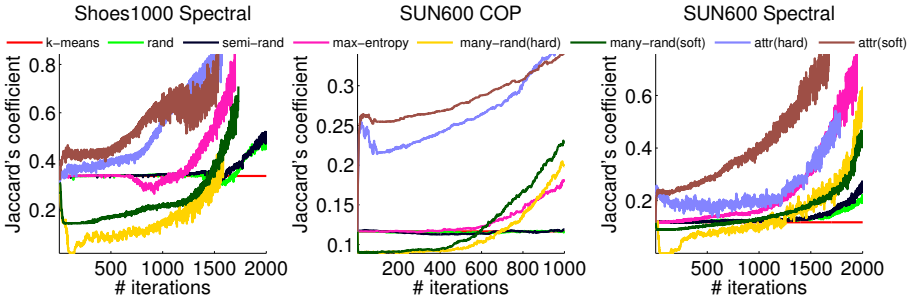

Fig. 3: Results of our approach on SUN600 and Shoes1000

Results on PubFig will be presented when we show personalized clustering experiments (Sec 5.3). We observed that spectral clustering had lot of variance in performance between consecutive iterations, and so we averaged results across 6 random runs. We now discuss various aspects of our results in detail.

Active selection of pairs: In Fig 3 we see that for each of the non-attribute baselines, the initial region is nearly flat. This demonstrates that semi-supervised clustering algorithms do not gain much until a large number of pairs have been labelled [2,9]. The max-entropy baseline starts learning earliest followed by semi-random and pure-random. On average, we found that max-entropy picks pairs that the current clustering had classified incorrectly (placed in different clusters when they belong to the same cluster or vice versa) 40% of the times, while semi-random picks such pairs only 25% of the times. This shows that our active selection of image pairs is quite effective at picking informative pairs. Note that even when no information is gained through the constraint itself (60% of the times), the attribute-based explanation will still be informative.

Attribute explanations: Our proposed approach in Fig 3 starts learning significantly earlier than non-attribute baselines, and outperforms these baselines across the board. Clustering accuracy increases by 15-20% in the first 100 iterations. This would have taken 20 times more constraints using the semi-supervised clustering algorithms without attribute explanations. The many-random baseline on the other hand performs even worse than the unsupervised version for many iterations. This shows that the additional constraints generated using the attribute explanations are important not just because of their quantity, but also because of their relevance to the desired clustering. Clustering in attribute space instead of low-level feature space does not diminish the effect of attribute-based explanations. For SUN600, unsupervised K-Means clustering in attribute space gives 20% accuracy (compared to 12% in low-level feature space). Attribute explanations in this case still lead to 15% performance gain after 100 iterations.

Timing Analysis: In interactive applications like these, it is important to consider the time spent by users in answering queries. In our approach, we observed that the time spent by MTurk workers per question (with attributes explanation) was around 17 secs on average as compared to 5 secs without attributes explanation. Moreover, as mentioned earlier, attribute explanations are taken only upto first 100 iterations. Using attributes, 70% accurate clusters can

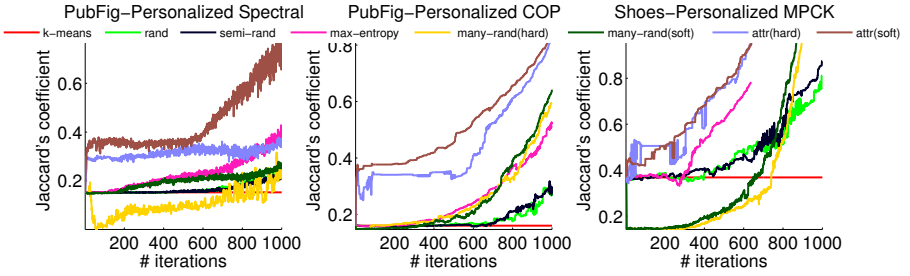

Fig. 4: Results of our approach on PubFig-Personalized and Shoes-Personalized

be obtained for Shoes1000 with just 40% of the user time as required by random approach. The algorithm can propagate the attribute explanations and identify the highest entropy pair in less than a second. The overall time per iteration is dominated by the underlying clustering algorithm (typically 5-20 secs).

Binary and Relative Attributes: We show clustering results using both relative attributes (Shoes1000) as well as binary attributes (SUN600). Both types of attributes are able to provide significant gains to the clustering algorithms as seen in Fig 3. We observed that the workers may use the same set of attributes in a future explanation. When binary attributes are used, the same attribute does not add any new information. This is because the constraints generated depend only on the attribute predictors, which do not change. For relative attributes on the other hand, repeated use of the same attributes can add new information because the constraints generated are relative to the image pairs on which the feedback was provided. Relative attributes allow the machine to be more specific while generating constraints. We found that the number of incorrect constraints are less (30% in Shoes1000) compared to for binary (40% in SUN600). Relative attributes also allow the user to provide more fine-grained feedback and are therefore more suitable for personalized clustering.

Hard vs. Soft Constraints: In Sec 4.3 we described two ways of incorporating the attribute explanations: hard and soft. The motivation behind soft constraints is to provide robustness to noisy attribute predictions. Comparing the soft and hard baselines in Fig 3, it can be observed that soft constraints usually perform better than hard constraints for both our approach and the many-random baseline In case of spectral clustering with SUN600, soft baseline gives a significant gain over hard throughout the clustering process.

5.3 Personalized Clustering

Personalized Clustering: Apart from the usual common-sense or domain-knowledge based clustering, we also evaluated our system for user-specific or personalized clustering. The ideal way to do this is by showing the image dataset to an MTurk worker and asking them to visualize a clustering and guide our system iteratively using attribute-based explanations. The resulting clusters can then be evaluated by the same user. However, MTurk is not the right platform for such extensive long duration tasks. Having sufficient users come to our lab

would limit the number of experiments we can conduct. Hence, we adopt the following strategy to simulate personalized clustering. We first create a ground truth (GT) clustering from a given set of images based on certain criteria (simulated preferences). We then solicit attribute-based explanations consistent with the desired clustering from MTurk workers using the same setup described in Sec 5.1. Since each worker sees the same clustering, responses from different workers can be thought of as coming from the same user.

We create the personalized GT clusters using two approaches: 1) By fixing the definition of each cluster in terms of attributes (PubFig) and 2) By merging several basic level categories to form clusters (Shoes). For PubFig, we create 4 clusters: {C1: white-male, C2: not-white-male, C3: female-lighthair, C4: female-darkhair}. We call this dataset PubFig-personalized. We used category level ground truth annotations to find the people that satisfy the various definitions. A total of 38 categories (people) satisfied atleast one of the cluster definitions. We used 15 images for each person resulting in 570 images total. For Shoes, the GT has the following configuration {C1: Athletic shoes-Sneakers, C2: Boots-Rainboots, C3: Clogs-Flats, C4: High Heels-Pumps-Stiletto}. We call this Shoes-personalized and it consists of 450 images covering 9 basic categories.

Results on Shoes-personalized and PubFig-personalized are shown in Fig 4. Our approach significantly outperforms all baselines in both the datasets. This shows the power of our approach in allowing users to more effectively inject their preferences in semi-supervised clustering algorithms as compared to existing approaches. Other trends like soft baseline performing better than hard, max-entropy better than random, etc. are observed in personalized clustering as well. Especially with PubFig-personalized, the results are exceptionally good. This may be because the clusters were created from attribute-based definitions. Note however that it is quite natural for humans to visualize groups or clusters based on certain attributes. Our approach can be used for organizing collections of personal photos or shopping products for efficient browsing and retrieval.

6 Conclusion

We presented an interactive approach for augmenting semi-supervised clustering approaches with attribute-based explanations. Using such a rich mode of communication, a user can convey his clustering criterion to the machine without having to annotate a large number of pairs of images. The clustering criterion can be domain knowledge which can be provided by a crowd, or it can be user preferences. We showed that by providing attribute-based explanations, the machine can get significant gains in clustering quality. We showed the generality of our approach by incorporating the attribute-based explanations in three diverse semi-supervised clustering algorithms, using both binary and relative attributes, in three different domains. Future work involves discovering a vocabulary of attributes simultaneously while clustering images and learning attribute models on-the-fly instead of using pre-trained attribute predictors.

Acknowledgements: This work was supported in part by a Google Faculty Research Award to DP.

References

1. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: In ICML, Morgan Kaufmann (2001) 577–584 1, 3, 5
2. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the Twenty-first International Conference on Machine Learning. ICML '04, New York, NY, USA, ACM (2004) 11– 1, 3, 5, 12
3. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. Machine Learning **74**(1) (2009) 1–22 1, 3
4. Yi, J., Zhang, L., Jin, R., Qian, Q., Jain, A.: Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion. In Dasgupta, S., Mcallester, D., eds.: Proceedings of the 30th International Conference on Machine Learning (ICML-13). Volume 28., JMLR Workshop and Conference Proceedings (May 2013) 1400–1408 1, 3
5. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 15, MIT Press (2003) 505–512 1, 3
6. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: Proceedings of the 24th International Conference on Machine Learning. ICML '07, New York, NY, USA, ACM (2007) 209–216 1, 3
7. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10** (June 2009) 207–244 1, 3
8. Yi, J., Jin, R., Jain, A., Jain, S., Yang, T.: Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In: Advances in Neural Information Processing Systems (NIPS). (2012) 1781–1789 1, 3, 4
9. Biswas, A., Jacobs, D.W.: Active image clustering: Seeking constraints from humans to complement algorithms. In: CVPR, IEEE (2012) 2152–2159 2, 3, 12
10. Basu, S., Banjeree, A., Mooney, E., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: In Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04. (2004) 333–344 2
11. Wauthier, F.L., Jojic, N., Jordan, M.I.: Active spectral clustering via iterative uncertainty reduction. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '12, New York, NY, USA, ACM (2012) 1339–1347 2, 3
12. Parikh, D., Grauman, K.: Relative attributes. In: ICCV. (2011) 2, 7
13. Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and simile classifiers for face verification. In: ICCV. (2009) 2, 3, 7, 10
14. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR. (2009) 2, 3, 4, 7
15. Shrivastava, A., Singh, S., Gupta, A.: Constrained semi-supervised learning using attributes and comparative attributes. In: ECCV. (2012) 2, 3, 4
16. Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: Image search with attribute feedback. In: CVPR. (2012) 2, 3, 4, 10
17. Kumar, N., Belhumeur, P., Nayar, S.: Facetracer: A search engine for large collections of images with faces. In: ECCV. (2010) 2, 3
18. Parkash, A., Parikh, D.: Attributes for classifier feedback. In: ECCV. (2012) 2, 3, 4
19. Donahue, J., Grauman, K.: Annotator rationales for visual recognition. In: ICCV. (2011) 2, 3, 4

20. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, MIT Press (2001) 849–856 3, 5
21. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: CVPR. (2010) 3, 10
22. Berg, T., Berg, A., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: ECCV. (2010) 3, 10
23. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR. (2009) 3, 7
24. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS. (2007) 3
25. Branson, S., Wah, C., Babenko, B., Schroff, F., Welinder, P., Perona, P., Belongie, S.: Visual recognition with humans in the loop. In: ECCV. (2010) 3, 4
26. Farhadi, A., Endres, I., Hoiem, D.: Attribute-centric recognition for cross-category generalization. In: CVPR. (2010) 3
27. Rastegari, M., Farhadi, A., Forsyth, D.: Attribute discovery via predictable discriminative binary codes. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., eds.: Computer Vision ECCV 2012. Volume 7577 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2012) 876–889 3
28. Parikh, D., Grauman, K.: Interactively building a discriminative vocabulary of nameable attributes. In: CVPR, IEEE (2011) 1681–1688 3
29. Parikh, D., Kovashka, A., Grauman, K.: Whittlesearch: Image search with relative attribute feedback. 2013 IEEE Conference on Computer Vision and Pattern Recognition **0** (2012) 2973–2980 4
30. Gomes, R.G., Welinder, P., Krause, A., Perona, P.: Crowdclustering. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K., eds.: Advances in Neural Information Processing Systems 24. (2011) 558–566 4
31. Tamuz, O., Liu, C., Belongie, S., Shamir, O., Kalai, A.T.: Adaptively learning the crowd kernel. CoRR **abs/1105.1033** (2011) 4
32. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: In IJCAI. (2003) 561–566 6
33. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: CVPR, IEEE (2009) 2372–2379 7
34. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: CVPR. (2012) 10
35. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV (2001) 10
36. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval. CIVR '07, New York, NY, USA, ACM (2007) 401–408 10