

Interactively Guiding Semi-Supervised Clustering via Attribute-based Explanations

Shrenik Lad

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Devi Parikh, Chair
Dhruv Batra
Lynn Abbott

April 27, 2015
Blacksburg, Virginia

Keywords: Computer Vision, Semi-Supervised Clustering, Attributes, Human-Machine
Communication

Copyright 2015, Shrenik Lad

Interactively Guiding Semi-Supervised Clustering via Attribute-based Explanations

Shrenik Lad

(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 work, 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.

This work was supported in part by a Google Faculty Research Award to Devi Parikh.

Dedication

I dedicate this thesis to my M.S. advisor, Dr. Devi Parikh. I cannot imagine a more helpful and resourceful advisor than Dr. Parikh.

Acknowledgments

Firstly, I would like to thank my parents and my sister who supported my decision to pursue Masters, who have been my well wishers all my life. I would like to thank Dr. Devi Parikh for being such a great advisor. Thank you for your guidance and support in almost every aspect of grad school. I want to thank my friends Ravi and Rama who forced me to take GRE and consider the possibility of grad school, which turned out to be one of the best decisions of my life. Both of you have been great advisors in my journey so far and I am sure you will continue to be in the future. Most importantly, I want to thank my friend Purva for supporting me during all ups and downs, for making me feel confident after every hard decision, and for helping me navigate whenever I am lost.

I would like to thank Dr. Dhruv Batra for teaching Machine Learning. I couldn't have understood ML better in any other class. You have been a great teacher and a superb go-to person whenever needed. I would also like to thank the advisory committee for their valuable feedback on the work. I thank my labmates for all intellectual discussions we had in lab meetings and group meetings. Rama, Ram, Senthil, Harsh, Tanmay, Faruk, Arjun, Laddha, Clint, Prakriti: grad school wouldn't have been fun without you.

Contents

1	Introduction	1
2	Related Work	4
3	Attributes	6
3.1	Binary Attributes	6
3.2	Relative Attributes	7
4	Semi-Supervised Clustering	9
4.1	Constrained K-Means	9
4.2	MPCK-Means	10
4.3	Spectral Clustering	10
5	Approach: Attribute-based Explanations	12
5.1	Active selection of pair	13
5.2	Attribute-based Explanation	13
6	Approach: Incorporating Attribute Explanations	15
7	Experiments and Results	18
7.1	Experimental Setup	18
7.2	Results	20
7.3	Personalized Clustering	22

8	Conclusion	26
	Bibliography	27

List of Figures

1.1	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. This leads to significant improvements in the performance of a diverse set of semi-supervised clustering algorithms on a variety of datasets.	2
3.1	An illustration of the Animals with Attributes (AwA) dataset with binary attributes. The attributes are defined at the category level. (Image taken from http://attributes.kyb.tuebingen.mpg.de/ Used under fair use, 2015.) . .	7
3.2	Relative attributes: While it is clear that (a) is <i>smiling</i> and (c) is not, for (b) it is more intuitive to say that he is <i>smiling</i> less than (a) but more than (c). Similarly, scene (e) is less <i>natural</i> than (d) but more so than (f). (Image taken from https://filebox.ece.vt.edu/~parikh/relative.html Used under fair use, 2015.)	8
7.1	These are the 6 ground truth clusters in SUN600 dataset. They correspond to categories like transportation, industrial regions, sports-recreation, etc. (Images taken from SUN Attributes dataset [1]. Used under fair use, 2015) . . .	19
7.2	Results of our approach on SUN600 and Shoes1000	21
7.3	Result on PubFig-Personalized dataset while clustering in the attribute feature space. The clustering algorithm is COP K-Means.	23
7.4	These are the four clusters in our PubFig-Personalized dataset. These images were shown to MTurk workers without any other information. The workers visualized the similarity measure and answered pairwise questions. (Face images taken from PubFig dataset [2]. Used under fair use, 2015)	24
7.5	Results of our approach on PubFig-Personalized and Shoes-Personalized . . .	25

Chapter 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 [3, 4, 5, 6, 7, 8, 9, 10] leverage user-provided pairwise constraints either to learn an appropriate distance metric in the feature space (distance metric learning [7, 8, 9, 10]) or to guide a clustering algorithm towards correct clusters (constrained clustering [3, 4, 5, 6]). These pairwise constraints are either 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 [11, 12, 13] 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 [12] requires 5000 constraints to achieve 30% accuracy on a dataset of 500 face images as reported in [11].

We propose accessing the user’s mental model of the desired clustering through visual attributes, a mid-level semantic representation. Some attributes like *wearing glasses* or *having four legs* are binary in nature while others like *smiling* and *attractive* are relative [14, 2, 15, 16]. Both binary and relative attributes have been shown to improve various



Figure 1.1: 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. This leads to significant improvements in the performance of a diverse set of semi-supervised clustering algorithms on a variety of datasets.

computer vision tasks like image search [17, 18] and classification [19, 20, 15, 2]. In this work, 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 annotation indicating whether the two images should belong to the same cluster or not. The user also provides an attribute-based explanation for that image pair. 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. Figure 1.1 shows an illustration of our proposed approach for image clustering.

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 [3], MPCK-Means [4] and Spectral Clustering [21] on three different domains: scenes (SUN [1]), faces (PubFig [2]) 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.

Contributions: We propose a novel human-machine interactive framework for semi-supervised image clustering. Instead of just providing must-link/cannot-link annotations on image pairs, the user also conveys the clustering criteria to the machine by providing semantic attribute-based explanations. The machine generates additional constraints on its own by propagating the explanations to other unlabeled pairs in the data. We propose attribute explanations involving both binary as well as relative attributes. We incorporate attribute explanations in three diverse semi-supervised clustering algorithms and show improved results due to these explanations.

Chapter 2

Related Work

Semi-Supervised clustering approaches are either constraint-based [3, 4, 5, 6] or metric-based [7, 8, 9, 10]. 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. [4] 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 [11, 13] 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. [11] 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. [13] 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 [15, 23, 2, 24, 22, 25, 18, 17, 19, 20, 26, 16]. 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 [17, 19, 20]. Donahue and Grauman [20] use spatial annotations and binary attributes to allow an annotator to provide an explanation for a particular class label. Whittle Search [17] 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. [16] uses attributes to avoid semantic drift in a bootstrapping approach by enforcing known relationships between categories (e.g., “banquet halls” are bigger than “bedrooms”). An exhaustive set of such relationships are provided to the system as input (similar in spirit to zero shot learning [15]). In our approach, we use attributes to propagate constraints to pairs of images in an interactive semi-supervised clustering setting. Parkash and Parikh [19] 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 [30, 10] 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.

Chapter 3

Attributes

Attributes are mid-level properties of images, for e.g., *shiny*, *striped*, *smiling*, etc. They are visual and hence machine detectable. They have human-designated names which makes them human understandable. Attributes can either be described at category level (e.g., all chairs are *wooden*) or at image level (e.g., this chair is *wooden*). As described in Chapter 2, attributes have been shown to help various tasks in Computer Vision like recognition, image description, segmentation, etc. In the literature, two kinds of attributes have been proposed: binary and relative. These are explained in the following two sections.

3.1 Binary Attributes

A binary attribute can take two states: it is either present or absent in an image. Given a vocabulary of Q predefined attributes, an image can be described by the states taken by these attributes in the image. An example of a commonly used animals dataset [15] with binary attributes is shown in Fig. 3.1. Since ground truth states of attributes are not known without human involvement, it is important to be able to predict attributes on novel images. Learning binary attributes can be posed as standard classification problem, where each attribute is a two-class problem (“present” and “absent”). Existing approaches have looked at various classification models to learn attributes independently [15, 23].

Certain attributes are often correlated, for e.g., *natural* scenes often *have greenery* in them, a *four-legged* animal cannot be *two-legged* at the same time, etc. Hence, learning attributes jointly can exploit correlations like these which could result in more accurate attribute predictions. Existing work including [32, 33, 34], learn attributes jointly by modelling correlations between them.



Figure 3.1: An illustration of the Animals with Attributes (AWA) dataset with binary attributes. The attributes are defined at the category level. (Image taken from <http://attributes.kyb.tuebingen.mpg.de/> Used under fair use, 2015.)

3.2 Relative Attributes

For various attributes like *smiling*, *natural*, etc., it is not possible to objectively tell whether the image has the attribute or not. Figure 3.2 shows some examples of this. Parikh and Grauman [14] proposed relative attributes, where the idea is to indicate the strength of an attribute in an image with respect to other images. For e.g., in Fig. 3.2, person (b) is *smiling* less than person (a) but more than person (c). Relative attributes are more natural, and allow more detailed human supervision.

Learning relative attributes involves learning a ranking function for each attribute, where each ranking function estimates a *real-valued* rank for images indicating the relative strength of the attribute presence in them. The supervision required for learning the ranking function involves ordered pairs of images $O_q = \{(i, j)\}$ such that $(i, j) \in O_q \implies i > j$, i.e., image i has a stronger presence of attribute a_q than j , and unordered pairs of images $S_q = \{(i, j)\}$ such that $(i, j) \in S_q \implies i \sim j$, i.e., i and j have similar relative strengths of a_q . Interested readers can refer to [14] for detailed procedure of learning relative attributes.



Figure 3.2: Relative attributes: While it is clear that (a) is *smiling* and (c) is not, for (b) it is more intuitive to say that he is *smiling* less than (a) but more than (c). Similarly, scene (e) is less *natural* than (d) but more so than (f). (Image taken from <https://filebox.ece.vt.edu/~parikh/relative.html> Used under fair use, 2015.)

Chapter 4

Semi-Supervised Clustering

Semi-Supervised clustering incorporates background knowledge in the form of pairwise constraints: must-link and cannot-link. A must-link constraint between image i and j implies that they must belong to the same cluster and a cannot-link constraint between i and j implies that they must belong to different clusters. These constraints are either used to learn an appropriate distance metric or are directly enforced by the clustering algorithm when assigning points to clusters. 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 $(b, c) \in M \implies (a, c) \in M$, meaning if a and b are must-linked and b and c are must-linked, then a and c should be must-linked.
2. $(a, b) \in M$ and $(b, c) \in C \implies (a, c) \in C$, meaning if a and b are must-linked and b and c are cannot-linked, then a and c should be cannot-linked.

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

4.1 Constrained K-Means

Constrained K-Means or COP K-Means [3] 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.

4.2 MPCK-Means

Metric Pairwise Constrained K-Means or MPCK-Means [4] combines distance metric learning and constrained clustering in a unified framework. It minimizes the following objective function:

$$\sum_{\mathbf{x}_i} \left(\|\mathbf{x}_i - \boldsymbol{\mu}_{l_i}\|_{D_{l_i}}^2 - \log(\det(D_{l_i})) \right) + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in M} w_{ij} \mathbb{1}[l_i \neq l_j] + \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} w_{ij} \mathbb{1}[l_i = l_j] \quad (4.1)$$

Here, \mathbf{x}_i denotes the i^{th} datapoint, l_i is the cluster-id of \mathbf{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 $(\mathbf{x}_i, \mathbf{x}_j)$, 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 Maximization (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 [4]. In our experiments, we enforce a common distance metric for all clusters. This was found to perform better than using a different metric for each cluster[4].

4.3 Spectral Clustering

Spectral clustering algorithms cluster points using eigenvectors of matrices derived from the data. We use the spectral clustering algorithm from Ng, Jordan and Weiss [21]. It consists of the following steps:

- Compute the $N \times N$ affinity matrix A for the N datapoints.
 $A_{ij} = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$ 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 .
- Obtain the matrix Y by normalizing the rows of E to sum to 1.
- 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. [35], present a semi-supervised version of this spectral clustering algorithm. If $(\mathbf{x}_i, \mathbf{x}_j)$ is a must-link constraint, then $A(i, j)$ and $A(j, i)$ are set to 1 (zero distance) and if $(\mathbf{x}_i, \mathbf{x}_j)$ 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.

Chapter 5

Approach: Attribute-based Explanations

Let U denote the set of N images $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ that we want to cluster into K groups. Let $\{a_1, a_2, \dots, a_Q\}$ 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 \mathbf{x}_i and image \mathbf{x}_j are must-linked and similarly $C(i, j)$ denotes whether image \mathbf{x}_i and image \mathbf{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 5.1) 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 5.2). Perform transitive closure on all accumulated ground truth constraints.
4. Convert attribute explanation to additional (possibly noisy) constraints (Sec 6).
5. Cluster the images with the updated set of constraints using a semi-supervised clustering algorithm.
6. Repeat from step 2.

5.1 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 to be most informative for a semi-supervised clustering algorithm.

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

$$P_{ML}^{ij} = P(label = ML | d_{ij}), \quad (5.1)$$

$$P_{CL}^{ij} = P(label = CL | d_{ij}) \quad (5.2)$$

Using Bayes rule

$$P(label = ML | d_{ij}) \propto P(d_{ij} | label = ML) * P(label = ML) \quad (5.3)$$

$$P(label = CL | d_{ij}) \propto P(d_{ij} | label = CL) * P(label = CL) \quad (5.4)$$

The distributions $P(d_{ij} | label = ML)$ and $P(d_{ij} | label = CL)$ 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 approximated as the proportion of pairs in same and different clusters. Similar in spirit to active learning for training classifiers [36], we select the pair with the highest entropy under the distribution $P(label | d_{ij})$ and present it to the user to solicit $label$.

5.2 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 [23, 15, 2]. 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 [14]. 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.

Chapter 6

Approach: Incorporating Attribute Explanations

Binary Attributes: Without loss of generality, we assume that the cannot-link feedback on an image pair $(\mathbf{x}_i, \mathbf{x}_j)$ 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 $(\mathbf{x}_i, \mathbf{x}_j)$ 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 S_1 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 $(\mathbf{x}_i, \mathbf{x}_j)$ is of the form “These two images are *similar* because both are a_{q_1} and a_{q_2} and ... and a_{q_T} ”, it suggests that having attributes $\{a_{q_1}, a_{q_2}, \dots, a_{q_T}\}$ in common are sufficient to qualify two images as being similar. After adding the ground truth must-link constraint for the pair $(\mathbf{x}_i, \mathbf{x}_j)$, the machine then uses the attribute classifiers of $\{a_{q_1}, a_{q_2}, \dots, a_{q_T}\}$ 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 $(\mathbf{x}_i, \mathbf{x}_j)$ is “These two images are *dissimilar* because \mathbf{x}_i is much more a_q than \mathbf{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 \mathbf{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 \mathbf{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 $(\mathbf{x}_i, \mathbf{x}_j)$, 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} and a_{q_2} and ... and a_{q_T} ”, machine com-

putes S_S , the set of images having the degree of presence of each attribute $\{a_{q_1}, a_{q_2}, \dots, a_{q_T}\}$ between the ranges specified by \mathbf{x}_i and \mathbf{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 7.2).

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 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 $(\mathbf{x}_{i'}, \mathbf{x}_{j'})$ 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 $\mathbf{x}_{i'}$ and $\mathbf{x}_{j'}$ respectively. In case of a generated must-link constraint $(\mathbf{x}_{i'}, \mathbf{x}_{j'})$, $c_{i'}$ is the average confidence of presence of the q_T attributes in image $\mathbf{x}_{i'}$, and $c_{j'}$ is the average confidence of presence of the q_T attributes in image $\mathbf{x}_{j'}$. The final confidence assigned to the pair is $c_{i'} * c_{j'}$. When the cannot link feedback on pair $(\mathbf{x}_i, \mathbf{x}_j)$ 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 $(\mathbf{x}_{i'}, \mathbf{x}_{j'})$ is proportional to the total number of images that lie between $\mathbf{x}_{i'}$ and \mathbf{x}_i and between $\mathbf{x}_{j'}$ and \mathbf{x}_j in the sorted list. In case of a must-link feedback, the confidence of a generated must-link constraint

$(\mathbf{x}_{i'}, \mathbf{x}_{j'})$ is proportional to how close $\mathbf{x}_{i'}$ and $\mathbf{x}_{j'}$ 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 4.1. We set the weight w_{ij} to be the confidence computed for $(\mathbf{x}_i, \mathbf{x}_j)$ from user feedback. In spectral clustering, the affinity matrix A captures the similarity of datapoints. If the confidence of a cannot-link constraint $(\mathbf{x}_i, \mathbf{x}_j)$ is c , we set $A(i, j) = 1 - c$, and if the confidence of a must-link constraint $(\mathbf{x}_i, \mathbf{x}_j)$ 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.

Chapter 7

Experiments and Results

7.1 Experimental Setup

Datasets: We demonstrate our approach on three domains.

- (a) **Scenes:** We use the SUN Attributes dataset [37] 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 [1] 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, 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 7.1. We use GIST [38] features for this dataset.
- (b) **Faces:** We use the Public Figures Face Database [2] 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 7.3 has details regarding how the clusters are defined). The features used are pyramid HOG features [39].
- (c) **Shoes:** Berg et al. [22] provide a shoes dataset which is also used in [17]. Kovashka et al.[17] 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.



Figure 7.1: These are the 6 ground truth clusters in SUN600 dataset. They correspond to categories like transportation, industrial regions, sports-recreation, etc. (Images taken from SUN Attributes dataset [1]. Used under fair use, 2015)

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 5.1). 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 5.1. 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 as well as the clustering output, SD is the number of pairs which belong to same clusters in ground truth but are incorrectly placed in different clusters by the clustering algorithm and DS is the number of pairs which belong to different clusters in ground truth but are placed in same clusters by the algorithm.

7.2 Results

The semi-supervised clustering algorithm can be any one of the 3 algorithms described in Sec. 4. Results on Shoes1000 and SUN600 with some of the clustering algorithms are shown in Fig 7.2. 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. Results on PubFig will be presented when we show personalized clustering experiments (Sec 7.3). We observed that spectral clustering had lot

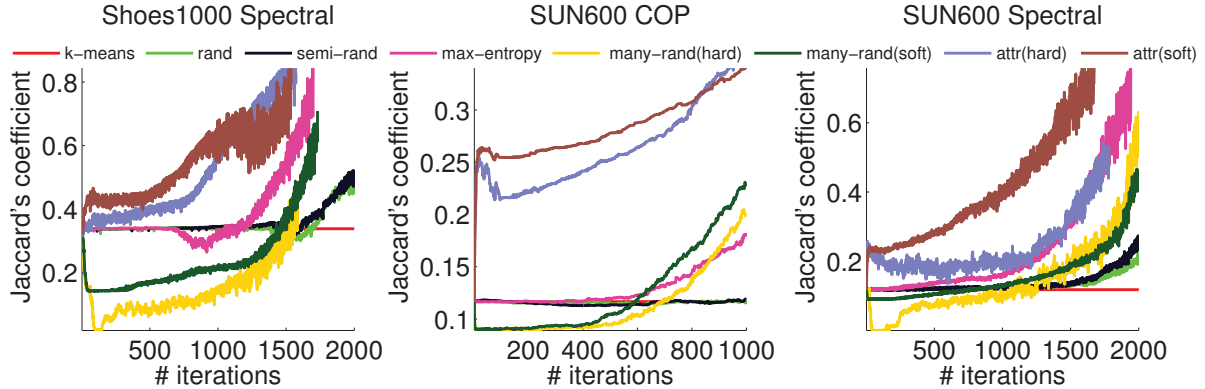


Figure 7.2: Results of our approach on SUN600 and Shoes1000

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 7.2 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 [4, 11]. 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 7.2 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 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 7.2. 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. 6, 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 7.2, 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.

Clustering in Attribute space: We also experimented with clustering in the feature space of attributes. Our attribute-based explanations provide significant gains even while clustering in the attribute space. Figure 7.3 shows the results on PubFig-Personalized dataset (introduced later in Sec. 7.3) when the 73 attribute predictions are used as features. The attr-soft baseline in pHoG feature space is also plotted in the same graph to compare the two feature spaces. Clustering in attributes space performs better than low level features (pHoG) as the unsupervised K-Means baseline itself is at 25% compared to 15% while using pHoG.

7.3 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

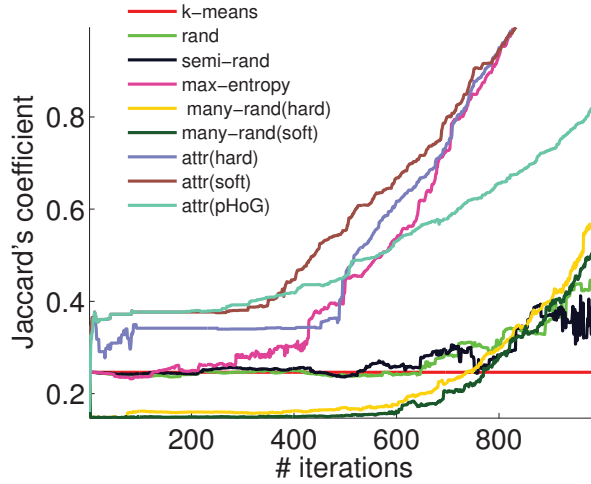


Figure 7.3: Result on PubFig-Personalized dataset while clustering in the attribute feature space. The clustering algorithm is COP K-Means.

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 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 7.1. 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 ground truth 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. An illustration of the clusters in Pubfig-Personalized can be seen in Fig. 7.4. For Shoes, the ground truth 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 7.5. 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

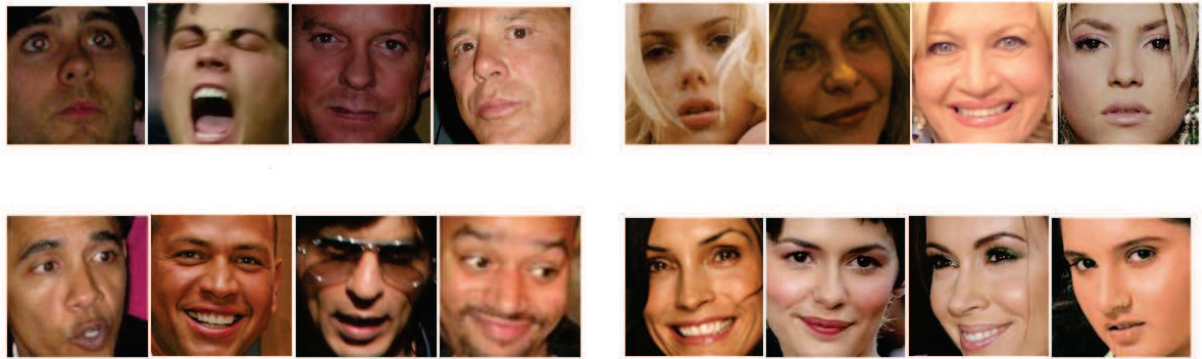


Figure 7.4: These are the four clusters in our PubFig-Personalized dataset. These images were shown to MTurk workers without any other information. The workers visualized the similarity measure and answered pairwise questions. (Face images taken from PubFig dataset [2]. Used under fair use, 2015)

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.

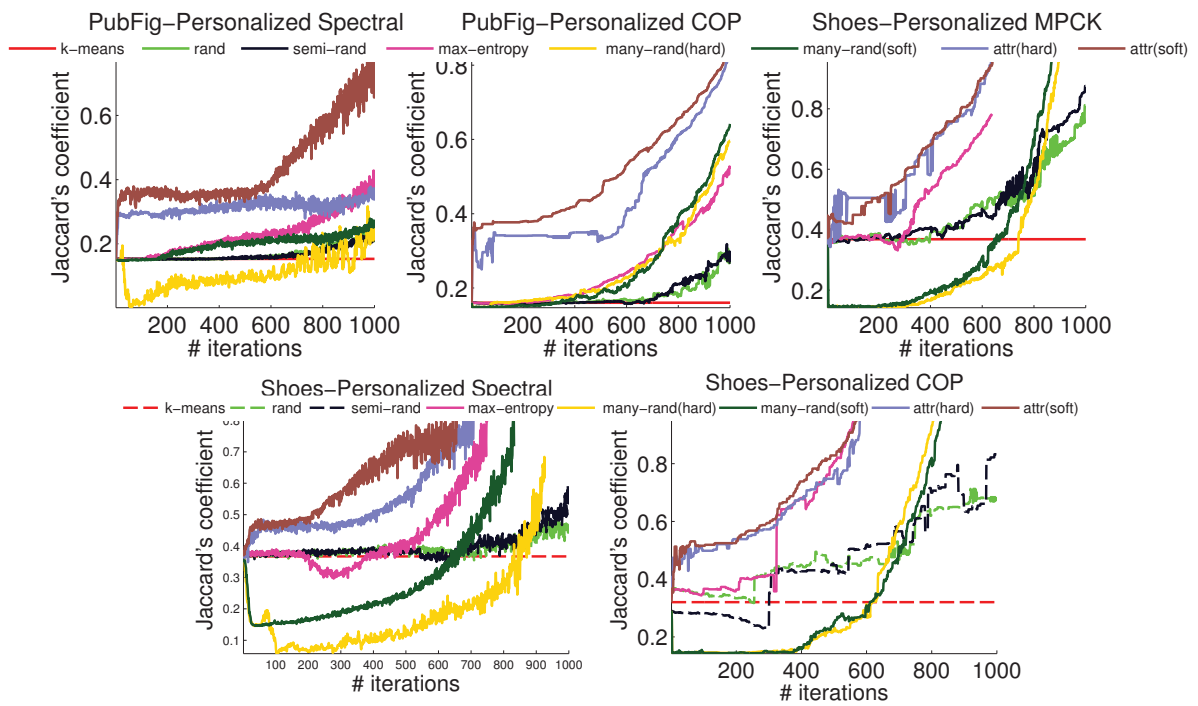


Figure 7.5: Results of our approach on PubFig-Personalized and Shoes-Personalized

Chapter 8

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 personalized 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.

Bibliography

- [1] Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Computer Vision and Pattern Recognition (CVPR). (2010)
- [2] Kumar, N., Berg, A., Belhumeur, P., Nayar, S.: Attribute and simile classifiers for face verification. In: International Conference on Computer Vision (ICCV). (2009)
- [3] Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means clustering with background knowledge. In: International Conference on Machine Learning (ICML). (2001)
- [4] Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: International Conference on Machine Learning (ICML). (2004)
- [5] Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. *Machine Learning* **74**(1) (2009) 1–22
- [6] Yi, J., Zhang, L., Jin, R., Qian, Q., Jain, A.: Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion. In: International Conference on Machine Learning (ICML). (2013)
- [7] 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 (NIPS). (2003)
- [8] Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: International Conference on Machine Learning (ICML). (2007)
- [9] Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research (JMLR)* (2009)
- [10] 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

- [11] Biswas, A., Jacobs, D.W.: Active image clustering: Seeking constraints from humans to complement algorithms. In: Computer Vision and Pattern Recognition (CVPR). (2012)
- [12] Basu, S., Banerjee, 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)
- [13] Wauthier, F.L., Jojic, N., Jordan, M.I.: Active spectral clustering via iterative uncertainty reduction. In: International Conference on Knowledge Discovery and Data Mining (KDD). (2012)
- [14] Parikh, D., Grauman, K.: Relative attributes. In: International Conference on Computer Vision (ICCV). (2011)
- [15] Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Computer Vision and Pattern Recognition (CVPR). (2009)
- [16] Shrivastava, A., Singh, S., Gupta, A.: Constrained semi-supervised learning using attributes and comparative attributes. In: European Conference on Computer Vision (ECCV). (2012)
- [17] Kovashka, A., Parikh, D., Grauman, K.: Whittlesearch: Image search with attribute feedback. In: Computer Vision and Pattern Recognition (CVPR). (2012)
- [18] Kumar, N., Belhumeur, P., Nayar, S.: Facetracer: A search engine for large collections of images with faces. In: European Conference on Computer Vision (ECCV). (2010)
- [19] Parkash, A., Parikh, D.: Attributes for classifier feedback. In: European Conference on Computer Vision (ECCV). (2012)
- [20] Donahue, J., Grauman, K.: Annotator rationales for visual recognition. In: International Conference on Computer Vision (ICCV). (2011)
- [21] Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems (NIPS). (2001)
- [22] Berg, T., Berg, A., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: European Conference on Computer Vision (ECCV). (2010)
- [23] Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: Computer Vision and Pattern Recognition (CVPR). (2009)
- [24] Ferrari, V., Zisserman, A.: Learning visual attributes. In: Advances in Neural Information Processing Systems (NIPS). (2007)

- [25] Branson, S., Wah, C., Babenko, B., Schroff, F., Welinder, P., Perona, P., Belongie, S.: Visual recognition with humans in the loop. In: European Conference on Computer Vision (ECCV). (2010)
- [26] Farhadi, A., Endres, I., Hoiem, D.: Attribute-centric recognition for cross-category generalization. In: Computer Vision and Pattern Recognition (CVPR). (2010)
- [27] Rastegari, M., Farhadi, A., Forsyth, D.: Attribute discovery via predictable discriminative binary codes. In: European Conference on Computer Vision (ECCV). (2012)
- [28] Parikh, D., Grauman, K.: Interactively building a discriminative vocabulary of nameable attributes. In: Computer Vision and Pattern Recognition (CVPR). (2011)
- [29] Kovashka, A., Grauman, K.: Attribute pivots for guiding relevance feedback in image search. In: International Conference on Computer Vision (ICCV). (2013)
- [30] Gomes, R.G., Welinder, P., Krause, A., Perona, P.: Crowdclustering. In: Advances in Neural Information Processing Systems (NIPS). (2011)
- [31] Tamuz, O., Liu, C., Belongie, S., Shamir, O., Kalai, A.T.: Adaptively learning the crowd kernel. In: International Conference on Machine Learning (ICML). (2011)
- [32] Wang, Y., Mori, G.: A discriminative latent model of object classes and attributes. In: European Conference on Computer Vision (ECCV). (2010)
- [33] Wang, X., Ji, Q.: A unified probabilistic approach modeling relationships between attributes and objects. International Conference on Computer Vision (ICCV) (2013)
- [34] Jayaraman, D., Sha, F., Grauman, K.: Decorrelating semantic visual attributes by resisting the urge to share. Computer Vision and Pattern Recognition (CVPR) (2014)
- [35] Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: International Joint Conference on Artificial Intelligence (IJCAI). (2003)
- [36] Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: Computer Vision and Pattern Recognition (CVPR). (2009)
- [37] Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Computer Vision and Pattern Recognition (CVPR). (2012)
- [38] Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal on Computer Vision (IJCV) (2001)
- [39] Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Conference on Image and Video Retrieval (CIVR). (2007)