UAC: An Uncertainty-Aware Face Clustering Algorithm

Biplob Debnath, Giuseppe Coviello, Yi Yang[*] and Srimat Chakradhar
NEC Laboratories America, Inc.
Princeton, New Jersey, USA
{biplob,giuseppe.coviello}@nec-labs.com, yangyi@gmail.com, and chak@nec-labs.com

Abstract

We investigate ways to leverage uncertainty in face images to improve the quality of the face clusters. We observe that popular clustering algorithms do not produce better quality clusters when clustering probabilistic face representations that implicitly model uncertainty – these algorithms predict up to 9.6X more clusters than the ground truth for the IJB-A benchmark. We empirically analyze the causes for this unexpected behavior and identify excessive false-positives and false-negatives (when comparing face-pairs) as the main reasons for poor quality clustering. Based on this insight, we propose an uncertainty-aware clustering algorithm, UAC, which explicitly leverages uncertainty information during clustering to decide when a pair of faces are similar or when a predicted cluster should be discarded. UAC considers (a) uncertainty of faces in face-pairs, (b) bins face-pairs into different categories based on an uncertainty threshold, (c) intelligently varies the similarity threshold during clustering to reduce false-negatives and false-positives, and (d) discards predicted clusters that exhibit a high measure of uncertainty. Extensive experimental results on several popular benchmarks and comparisons with state-of-the-art clustering methods show that UAC produces significantly better clusters by leveraging uncertainty in face images – predicted number of clusters is up to 0.18X more of the ground truth for the IJB-A benchmark.

1. Introduction

Analyzing video streams from surveillance cameras is becoming crucial for businesses and organizations to maximize their return on investment on video surveillance systems. For example, shopping malls, equipped with surveillance cameras, analyze the video streams to gain insights into shopper statistics to provide a better and personalized experience to their customers. Table 1 shows a few examples of insights that are useful in a shopping mall scenario. All these insights can be easily generated by first clustering faces detected in the surveillance videos. As shown in Figure 1, faces are extracted from surveillance cameras and stored in a database. These faces are then clustered to determine unique persons and to derive customer insights shown in Table 1.

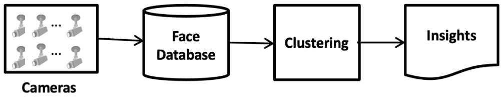


Figure 1: A high level view of the target setup

1. Number of shoppers visiting every day, week, month, or quarter
2. Number of returning shoppers visiting every day, week, month, or quarter
3. Number of shoppers visiting a specific store (e.g., toy store, theatre, etc.)
4. Number of shoppers visiting a specific set of stores of interest

Table 1: Customer insights from mall surveillance videos

Faces captured by surveillance videos are inherently noisy. Consequently, facial features learned by conventional, deterministic face embedding models [21, 4, 18, 13, 24] can be ambiguous, or certain facial features may not even be present in the input face, leading to noisy representations. Consequently, clustering algorithms based on these noisy representations tend to produce incorrect results.

Recently, probabilistic face embeddings (PFE [23] and DUL [3]), which represent each face image as a multivariate Gaussian distribution in the latent space, have been proposed to improve the accuracy of face recognition when face images are noisy. In addition, PFE [23] proposes a new similarity function augmented with uncertainty information to compute the similarity between two probabilistic embeddings. Using such embeddings and similarity functions, one would expect face clustering algorithms to produce better quality clusters when more information like uncertainty is available. However, as we show in Section 2, popular face clustering algorithms do not produce better quality face clusters when using probabilistic embeddings with uncertainty information.

Inspired by the use of uncertainty information to improve face recognition [23], and principal component analysis [8], we investigate ways to leverage uncertainty in face images to improve face clustering tasks. In this paper, we make the following contributions:

1. We show that popular face clustering algorithms do not produce better quality clusters when additional informa-

[*] Now works at Google and this research has been performed while employed at NEC Laboratories America, Inc.

tion like uncertainty is implicit in the face representations or the similarity function.

2. We empirically analyze the causes for this unexpected behavior and identify excessive false-positives and false-negatives (when comparing face-pairs) as the main reasons for poor quality clustering.
3. We propose a novel uncertainty-aware clustering algorithm, UAC, that explicitly leverages uncertainty information during clustering to intelligently decide when a pair of faces are similar or when to discard a predicted cluster due to the high uncertainty estimate for the cluster.
4. We propose a new cluster quality metric, *Purity Adjusted Amplification Score (PAAS)*, that more accurately reflects the quality of clusters when data uncertainty is high.
5. Extensive experimental results on several popular benchmarks, and comparisons with state-of-the-art clustering methods, show that UAC produces an order of magnitude better clusters by leveraging uncertainty in face images.

2. Motivation

With the increased face recognition accuracy by using uncertainty-aware probabilistic face embeddings, one would expect a similar improvement for clustering tasks. However, we find that these embeddings do not help to improve the clustering accuracy when data uncertainty is high. For example, Table 2 shows the number of clusters predicted by popular clustering algorithms using PFE [23] embeddings of face images in public benchmarks like LFW [9] and IJB-A [10], which include faces with low and high uncertainty, respectively. The number of predicted clusters for the LFW benchmark is very close to the ground truth (face images have very low uncertainty), while for the IJB-A, the predicted clusters are overestimated by 5.97X, 7X, and 9.6X by the DBSCAN [5], AHC [15], and GCN-V [30] clustering algorithms, respectively. This is problematic because clustering results are used to compute various analytics queries like the ones shown in Table 1.

We also find that popular face cluster quality metrics (e.g., Pairwise F-Score, BCubed F-score, NMI, etc.) do not adequately reflect the true quality of clusters when data uncertainty is high (we provide detailed results in Section 4.2). For example, for the IJB-A case the DBSCAN [5] predicts 5.97X more clusters, however popular quality metrics still report high scores: *Purity = 0.97, BCubed F-Score = 0.84, Pairwise F-Score = 0.78 and NMI = 0.94*. Thus, we rethink how to evaluate cluster quality to rank different clustering algorithms when data uncertainty is high.

3. Related Work

Existing face clustering methods can be broadly categorized into two groups: unsupervised and supervised.

Algorithm	**LFW** (*Data Uncertainty = Low*)		**IJB-A** (*Data Uncertainty = High*)	
	Expected Clusters	Predicted Clusters	Expected Clusters	Predicted Clusters
DBSCAN	5749	5777	500	3483
AHC	5749	5841	500	5322
GCN-V	4600	4667	400	3214

Table 2: Predicted number of clusters for the popular clustering algorithms using probabilistic embedding and mutual likelihood score (MLS) similarity method [23].

Unsupervised methods, such as K-Means [14], DBSCAN [5], Agglomerative Hierarchical Clustering (AHC) [15], etc. use similarity scores to find clusters. In addition to the similarity function, K-Means uses the number of clusters (i.e., K), while other algorithms use similarity threshold as well as few other parameters (e.g., $minPts$, linkage method, etc.). We cannot use K-means [14] algorithm because the value of K is what we are trying to estimate for our collections of faces. In order to improve similarity scores, recently, some works have focused on learning new similarity functions using deep learning. For example, Lin et al. [11] propose a new function based on the density affinity of the local neighborhoods; Otto et al. [16] propose an approximate rank order metric based on the shared nearest-neighbors information; and PAHC [12] proposes a similarity function that measures similarity between CNN features by evaluating linear SVM margins, and SVM is trained based on the nearest neighbor information. Among all unsupervised methods, DBSCAN is one of the most popular density-based algorithms. It has been successfully used in many real-world applications and has received the SIGKDD test-of-time award in 2014 [22]. Agarwal et al. [1] provides a survey of the many variants of the DBSCAN algorithm for handling noise, which is very different from the data uncertainty issue considered in this paper. We estimate noise from the face image, while DBSCAN finds noisy data points based on the number of nearest-neighbors and reachability information.

Recently some supervised clustering methods have been proposed in order to learn cluster patterns. For example, graph convolutional networks (GCNs) learn cluster representations from the nearest-neighbor graphs [30, 6, 31, 28]; and Tapaswi [25] et al. propose a method to carve the feature space into equal-sized balls. Although these supervised algorithms have achieved good results for some datasets, these algorithms require hyperparameters tuning as well as repeated training for each dataset. In addition, our evaluation shows these algorithms do not work well when data uncertainty is high.

To the best of our knowledge, no face clustering algorithm explicitly considers uncertainty information. Most of the above works use nearest-neighbors information in the feature space. However, when data uncertainty is high, the similarity estimate becomes incorrect, which generates incorrect nearest-neighbors estimate – thus, these algorithms may not be effective when data uncertainty is high.

4. Impact of Uncertainty on Clustering Tasks

In this section, we first describe how uncertainty is captured with a probabilistic embedding. Next, we evaluate the impact of probabilistic embedding and uncertainty-aware similarity function on the clustering tasks. Our evaluation shows that these enhanced embeddings do not improve the quality of the clusters when data uncertainty is high. Finally, we further understand the reasons for poor clustering accuracy and present our key insights to consider uncertainty during clustering effectively.

4.1. Uncertainty estimation

Probabilistic face embeddings provide a distributional estimation [23, 3] instead of a deterministic point estimation [4, 21, 13, 18, 24, 27, 20] in the latent space for each input face image. It represents each face as a multivariate Gaussian distribution, $\mathcal{N}(\mu, \sigma^2)$. The mean of the distribution estimates the most likely latent feature values while the span of the distribution, or variance, represents the noise or *uncertainty* of these estimates.

In this paper, we use PFE [23] for generating probabilistic embeddings. Given a pre-trained face recognition (FR) model, the mean vector μ is the deterministic embedding generated by the FR model. Then, PFE adds an extra branch to the FR model to learn the variance vector σ^2. The extra branch is trained using the mutual likelihood score (MLS). While PFE learns σ^2 separately, DUL [3] learns both μ and σ^2 simultaneously. Now, given a probabilistic embedding of a face as $\mu_1, \mu_2, ..., \mu_D, \sigma_1^2, \sigma_2^2, ..., \sigma_D^2$, where D is the feature dimension, then the estimated uncertainty is the harmonic mean of the variances across all dimensions:

$$uncertainty = \frac{D}{\sum_{i=1}^{D} \frac{1}{\sigma_i^2}}$$

As an example, consider two popular face benchmarks, LFW [9] and IJB-A [10]. Figure 2 shows the distribution of uncertainty in face images for both datasets. These datasets exhibit different degrees of data uncertainty. The uncertainty for the face images in the LFW dataset is less than 0.0015. However, faces in IJB-A exhibit much higher uncertainty when compared with faces in LFW. There are faces in IJB-A with uncertainty greater than 0.0030! Therefore, face images in LFW have less noise than face images in IJB-A.

Probabilistic face embedding model. For our evaluation, we use a pre-trained PFE model from github [32]. This model uses a 64-layer residual network trained with AM-Softmax [26] on the MS-Celeb-1M dataset [7]. The dimension of the deterministic embedding (μ) is 512. Thus, the feature dimension of a PFE embedding (μ and σ^2) is 1024.

4.2. Clustering with PFE

In this paper, we use DBSCAN [5], AHC [15], GCN-V [30] algorithms to cluster latent feature vectors from probabilistic face embedding. We consider two cases. The first case "Deterministic + Cosine" works as a baseline because it represents the class of face embeddings [4, 21, 13, 18, 24, 27, 20] which does not consider the uncertainty information. It uses only the mean vector μ of a PFE embedding. In contrast, the second case "Probabilistic + MLS" represents the class of face embeddings [23, 3] which uses uncertainty augmented face representation and similarity function. It uses both μ and σ^2 vectors of a PFE embedding.

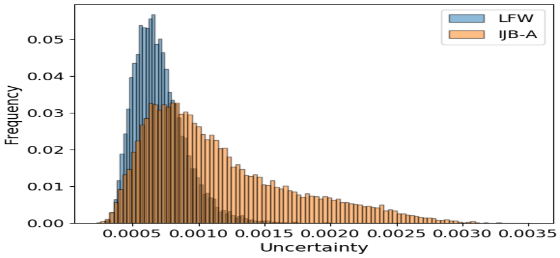


Figure 2: Data uncertainty distribution in LFW [9] and IJB-A [10] benchmarks. LFW has low data uncertainly, while IJB-A has high data uncertainty.

4.2.1 Similarity functions

We use cosine similarity for the deterministic embedding and MLS for the probabilistic embedding.

Cosine Similarity. To calculate cosine similarity, we do not consider the variance vector $\sigma_1^2, \sigma_2^2, ..., \sigma_D^2$ of a probabilistic representation. [1]. The cosine similarity score for a pair of latent vectors (x_i, x_j) is calculated as follows:

$$cosine(x_i, x_j) = \frac{\sum_{l=1}^{D} \mu_i^{(l)} * \mu_j^{(l)}}{\sqrt{\sum_{l=1}^{D} \mu_i^{2(l)}} \sqrt{\sum_{l=1}^{D} \mu_j^{2(l)}}}$$

where D is feature dimension and $\mu_i^{(l)}$ refers to the l^{th} dimension of the μ_i of x_i.

Mutual Likelihood Score (MLS). PFE [23] proposes a similarity function to take into account the data uncertainty in a face image. MLS for a pair of latent vectors (x_i, x_j) is calculated as follows:

$$MLS(x_i, x_j) = \sum_{l=1}^{D} \left(\frac{\left(\mu_i^{(l)} - \mu_j^{(l)}\right)^2}{\sigma_i^{2(l)} + \sigma_j^{2(l)}} + log\left(\sigma_i^{2(l)} + \sigma_j^{2(l)}\right) \right)$$

where $\mu_i^{(l)}$ refers to the l^{th} dimension of the μ_i of x_i and similarly for $\sigma_i^{2(l)}$.

4.2.2 Evaluation of clustering with PFE

We use two popular benchmarks for the evaluation:

LFW. The Labeled Faces in the Wild [9] contains 13,233 face images of 5,749 subjects. Of the 5,749 subjects, 4,069 individuals have only one face image each. These face

[1] We also considered different ways to augment the cosine similarity function by incorporating uncertainty information (i.e., the cosine similarity score depends on both μ and σ_2). However, we observed that the augmented cosine similarity score does not improve the quality of clustering. Therefore, we are omitting further discussion of results using the augmented cosine similarity function.

Embedding + Similarity	**Algorithm**	**LFW** (uncertainty: $mean = 0.00070, std = 0.00020$)						**IJB-A** (uncertainty: $mean = 0.00114, std = 0.00055$)					
		Predicted Clusters	Purity	BCubed F-Score	Pairwise F-Score	NMI	PAAS	Predicted Clusters	Purity	BCubed F-Score	Pairwise F-Score	NMI	PAAS
Deterministic + Cosine	DBSCAN	5799	0.9966	0.9932	0.9985	0.9986	0.9891	2267	0.9206	0.8393	0.4082	0.9251	0.3158
Deterministic + Cosine	AHC	5121	0.5827	0.7065	0.0262	0.8007	0.5791	4293	0.9716	0.6747	0.6391	0.8965	0.2095
Deterministic + Cosine	GCN-V	4697	0.9977	0.9873	0.9596	0.9967	0.9867	2259	0.9685	0.7790	0.7531	0.9223	0.2814
Probabilistic + MLS	DBSCAN	5777	0.9952	0.9933	0.9968	0.9986	0.9893	3483	0.9688	0.8445	0.7832	0.9353	0.3032
Probabilistic + MLS	AHC	5841	0.9966	0.9899	0.9966	0.9978	0.9864	5322	0.9921	0.7109	0.6932	0.9049	0.2203
Probabilistic + MLS	GCN-V	4667	0.9976	0.9763	0.8014	0.9943	0.9893	3214	0.9864	0.7828	0.7692	0.9215	0.2703

Table 3: Experimental results for the DBSCAN [5], Agglomerative Hierarchical Clustering (AHC) [15], and GCN-V [30] algorithms. In these experiments, we set $minPts$ to 1, cosine similarity threshold to 0.5, and MLS similarity threshold to 2650. For AHC, we set the linkage method to $average$. For GCN, 20% data is used for training and in the k-Nearest-Neighbors graph k is set to 80.

images were acquired by retrieving images of celebrities and public figures and retaining only those images where a face is detected using an off-the-shelf face detector, Viola-Jones [19]. As a consequence, the variation in the facial pose is limited.

IJB-A. The IARPA Janus Benchmark A (IJB-A) [10], a publicly available media in the wild dataset containing 25,813 face images of 500 subjects. IJB-A is designed for unconstrained scenarios, and it has the key features: (a) wider geographic variations in subjects, (b) full pose variation, (c) a mix of faces from images and videos.

Table 3 shows our evaluation results. In addition to the predicted number of clusters,we report values for several popular metrics that are frequently used to estimate the overall quality of clustering [17, 25, 28, 30, 31]: Purity[2] [2], BCubed F-Score [2], Pairwise F-Score [2], and Normalized Mutual Information (NMI) [2]. The gray-colored columns report the PAAS metric, which we explain in Section 5.2. For all these metrics, a value close to 1.0 indicates better cluster quality.

DBSCAN. The number of predicted clusters for LFW is very close to the ground truth 5749 for both the "Deterministic + Cosine" and "Probabilistic + MLS". Also, the values of all the cluster quality metrics are almost 1.0 for both cases. This result is expected because, as shown earlier, images in the LFW dataset are generally of good quality, and they have low uncertainty. So, by considering uncertainty in "Probabilistic + MLS" during clustering, we do not see a big advantage.

In contrast, the number of predicted clusters for the IJB-A dataset is much larger compared to the ground truth subjects of 500: clustering with "Deterministic + Cosine" predicts 3.53X times more clusters compared to the ground truth, and clustering with "Probabilistic + MLS" over-predicts the number of unique persons by a factor of 5.97X. Also, contrary to what one would expect, clustering with "Probabilistic + MLS", which uses uncertainty, over-predicts clusters by 2.44X when compared with clustering using "Deterministic + Cosine". It is also surprising that clustering using "Probabilistic + MLS" has a higher score for the popular cluster quality metrics like Purity, BCubed F-Score, Pairwise F-Score, and NMI metrics – the Pairwise F-Score is almost twice as high (0.7832 vs. 0.4082) when compared with clusters obtained by "Deterministic + Cosine".

Agglomerative Hierarchical Clustering (AHC). Our results show that AHC [15] shows similar trends as the DBSCAN. This is not surprising, as AHC is also an unsupervised algorithm. In fact, when the $linkage.method = single$, it produces the same results as DBSCAN. Here, we set $linkage.method = average$. For IJB-A, "Probabilistic + MLS" predicts 9.6X more clusters, while "Deterministic + Cosine" predicts 7.6X more clusters. Overall, we do not see any significant benefit from using "Probabilistic + MLS" when data uncertainty is high.

GCN-V. GCN-V [30] also shows similar trends as the DBSCAN[3]. It is a supervised clustering algorithm. We use 20% of the data for training in order to learn cluster patterns. Training a GCN network is a cumbersome task as it requires tuning hyperparameters and labeling the dataset. Surprising, even with these extra efforts, GCV-V does not show better performance for the IJB-A – "Probabilistic + MLS" predicts 7X more clusters, while "Deterministic + Cosine" predicts 4.64X more clusters. Overall, we find that even "Probabilistic + MLS" cannot handle high data uncertainty.

Cluster quality metric.. Our results in Table 3 show that widely used metrics [2] like Purity, BCubed F-score, Pairwise F-score and NMI alone do not adequately capture the accuracy of the clustering algorithm in the presence of uncertainty. These metrics fail to penalize algorithms that over-cluster (generates too many clusters).

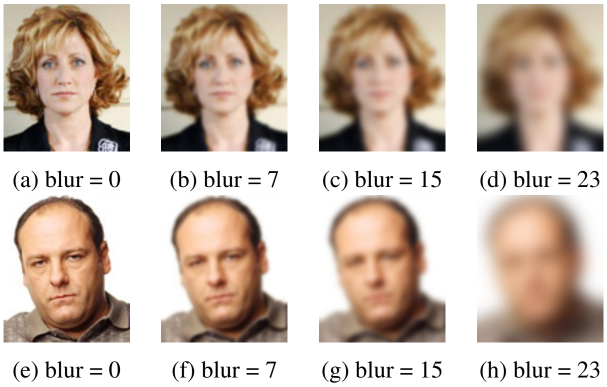
4.3. Deeper analysis of clustering with PFE

We perform a controlled experiment to better understand the various sources of clustering inaccuracy in the presence of uncertainty. We select two images (of different people) and systematically increase the uncertainty in the images by introducing more Gaussian blur[4]. Figure 3 shows the two images of two different persons and their corresponding blurred images as we gradually increase the degree of uncertainty (i.e. degree of Gaussian blur).

[2] We report the weighted clustering purity (WCP) score used in [25]

[3] Other GCN variants (i.e., GCN-(V+E) [30], linkage based GCN [28], affinity graph-based GCN [31]) also exhibit similar trends.

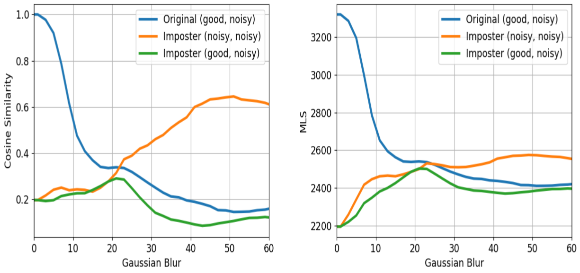
[4] Gaussian blur is used here for the illustration purposes. We considered many other sources of noise (i.e., occlusion, random Gaussian noise, etc.), and they all exhibit similar behaviors as the Gaussian blur.



(a) blur = 0 (b) blur = 7 (c) blur = 15 (d) blur = 23

(e) blur = 0 (f) blur = 7 (g) blur = 15 (h) blur = 23

Figure 3: Face images used in the controlled experiment



(a) Deterministic (b) Probabilistic

Figure 4: Effect of increasing uncertainty on "Deterministic + Cosine" (left) and "Probabilistic + MLS" (right).

Figure 4 shows the effect of uncertainty on cosine similarity score and MLS. We consider three different cases:

1. **Genuine (good, noisy)**: We start with an original, good quality face image, and compare it with increasingly blurred versions of the original image. Since the two face images in the pair belong to the same person, we refer to the pair as a *genuine* pair. The effect of increasing blur on the "Deterministic + Cosine" is shown by the blue line in Figure 4a. Similarly, the effect of increasing blur on the "Probabilistic + MLS" is shown by the blue line in Figure 4b.
2. **Imposter (noisy, noisy)**: We start with two images of two different persons. Since the two images belong to different persons, we call the pair as an imposter pair. Then, we gradually apply increasing blur to both the images. The degree of blur is the same for both the images. The effect of increasing blur on the "Deterministic + Cosine" is shown by the orange line in Figure 4a. Similarly, the effect of increasing blur on the "Probabilistic + MLS" is shown by the orange line in Figure 4b.
3. **Imposter (good, noisy)**: Again, we start with two images of two different persons. Both original images are of good quality (they exhibit very low uncertainty). Since the two images are of different persons, we have an imposter pair. Then, we gradually apply increasing blur to only one of the images (while leaving the other image as is). The effect of increasing blur on the "Deterministic + Cosine" is shown by the green line in Figure 4a. Similarly, the effect of increasing blur on the "Probabilistic + MLS" is shown by the green line in Figure 4b.

4.3.1 Effect on deterministic embedding

Consider the results shown in Figure 4a. We typically set a threshold on the cosine similarity score to determine if two face images belong to the same person. A threshold of 0.4 appears to be a good choice so that genuine pairs are correctly matched to be the same person. As the Gaussian blur is varied from 0 to 17, we are able to correctly classify an original image and its blurred version to be the same person. As we increase the blur beyond 17, the uncertainty in the blurred image increases and we are no longer able to conclude that the original image and its blurred version are a match (i.e., we have a *false negative*). This will cause the clustering algorithm to place the original image and its blurred version into different clusters, and we may end up with a lot of unnecessary clusters. We refer to this situation as the *false-negative* problem.

Figure 4a illustrates another important problem. Assuming a threshold of 0.4 for the cosine similarity score, many pairs in the Imposter(noisy, noisy) category (orange curve) will be incorrectly declared as a match as the Gaussian blur increases beyond 20. This will result in false-positives. In this case, face images of two different people will be placed in the same cluster, and erroneously, far fewer clusters will be generated (compared to the ground truth or actual number of clusters). We refer to this situation as the *false-positive* problem.

4.3.2 Effect on probabilistic embedding

Consider the results shown in Figure 4b. Again, we typically set a threshold on MLS to determine if two face images belong to the same person. A threshold of 2650 appears to be a good choice so that genuine pairs (blue line) are correctly matched to be the same person. As the Gaussian blur is varied from 0 to 17, we are able to correctly classify an original image and its blurred version to be the same person. As we increase the blur beyond 17, the uncertainty in the blurred image increases and we are no longer able to conclude that the original image and its blurred version are a match (i.e., we have a false negative). This will cause the clustering algorithm to place the original image and its blurred version into different clusters, and we may end up with a lot of unnecessary clusters. So, we have a false-negative problem.

However, unlike the case in Figure 4a, Figure 4b does not have the false-positive problem. Again, assuming a threshold of 2650 for MLS, many pairs in the Imposter(noisy, noisy) category (orange curve) will be correctly declared as a non-match as the Gaussian blur increases beyond 20. So, by using uncertainty information, MLS avoids the false-positive problem. Accordingly, results of clustering are also not polluted by false-positives.

4.4. Key insights

If the uncertainty in face images is low, both “Deterministic + Cosine” and “Probabilistic + MLS” perform well. However, as uncertainty increases, we see a divergence in the behavior of the two cases.

Unlike a similarity function based on the cosine similarity score, the MLS-based similarity function does not suffer from false-positives caused by noisy imposter pairs. Therefore, by leveraging uncertainty information, MLS can satisfactorily address the false-positive problem. However, both cosine similarity score, and MLS, are unable to address the false-negative problem adequately. Here, genuine pairs (where both images in the pair belong to the same person, but one of the images has high uncertainty) are missed.

Based on the above key insight, the next section introduces a new clustering algorithm that leverages the uncertainty information.

5. Clustering with Uncertainty Estimates

We first present an uncertainty-aware clustering algorithm. Next, we present a new cluster quality assessment metric.

5.1. Uncertainty-aware clustering algorithm

Clustering algorithms rely on similarity scores for finding similar faces. These algorithms implicitly assume that similarity scores are reliable, which is only true when the input set consists of mostly good quality faces (for example, LFW [9] dataset). However, similarity scores become unreliable (as shown in Figure 4) whenever the input set contains a mix of good and noisy faces, and consequently, clustering algorithms generate incorrect results.

What if, in addition to the similarity score, the uncertainty information about a face is explicitly made available to the clustering algorithm instead of being embedded in a representation or similarity function? Can a clustering algorithm leverage this additional information and improve clustering? We show that a clustering algorithm can use explicit uncertainty information to assess the trustworthiness of a similarity score, take appropriate actions to avoid the *false-positive* and *false-negative* problems, and improve the quality of clusters – this is the basis of our new clustering algorithm, UAC.

To address the *false-positive* and *false-negative* problems, we leverage uncertainty information. We classify a face pair into one of four classes, as shown in Figure 5. Here, x-axis and y-axis correspond to uncertainty estimates of $face_x$ and $face_y$, respectively; and the dotted lines correspond to the uncertainty threshold u_t. We use this threshold to group uncertainty values into LOW and $HIGH$ based on whether they are below or above the u_t, respectively. We focus on the following four cases:

1. $\{face_x(LOW), face_y(LOW)\}$. In this case, both deterministic and probabilistic embeddings provide accurate estimates (because data uncertainty is low), and clustering algorithms correctly cluster the two faces.

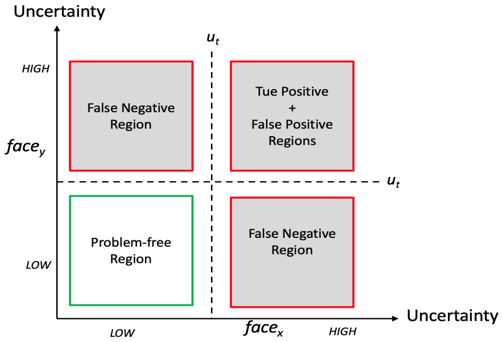


Figure 5: The accuracy of the similarity estimate of a face-pair $(face_x, face_y)$ varies with the uncertainty level.

2. $\{face_x(HIGH), face_y(LOW)\}$. In this case, both deterministic and probabilistic embeddings provide inaccurate estimates (because data uncertainty of the $face_x$ is high). As observed in Figure 5, this case leads to *false-negatives* (faces from the same person are incorrectly deemed to be dissimilar due to a low similarity score).
3. $\{face_x(LOW), face_y(HIGH)\}$. This is similar to the case above.
4. $\{face_x(HIGH), face_y(HIGH)\}$. In this case, both deterministic and probabilistic embeddings provide inaccurate estimates (due to the high data uncertainty of both faces). We observe very high similarity scores irrespective of whether images are of the same person or two different persons. The former case corresponds to the *true positive*, while the second case corresponds to the *false positive*.

To handle the above-mentioned four cases, we propose a new, uncertainty-aware clustering algorithm, UAC. It has two key phases: a) cluster formation by explicitly leveraging uncertainty and b) cluster uncertainty estimation and pruning.

Case-specific similarity thresholds: UAC varies the similarity threshold based on the uncertainty of $face_x$ and $face_y$. We assume that the base similarity threshold is ϵ (with a default value of 0.50). UAC takes different actions based on the uncertainty of the two faces in a face pair:

1. $\{face_x(LOW), face_y(LOW)\}$. The similarity threshold remain unchanged.
2. $\{face_x(HIGH), face_y(LOW)\}$. We lower the similarity threshold to $\epsilon - \Delta_{HL}$. For example, if $\Delta_{HL} = 0.05$, then the new similarity threshold is $(0.50 - 0.05)$ or 0.45. By lowering the similarity threshold, UAC can avoid *false-negatives*.
3. $\{face_x(LOW), face_y(HIGH)\}$. This case is similar to the case above.
4. $\{face_x(HIGH), face_y(HIGH)\}$. Since it is hard to distinguish between the *true positive* and *false positive* cases due to lack of adequate information content, to be

safe, we ignore similarity estimates when the uncertainty level of both images are high by raising the similarity threshold to ∞. In the three cases above, when there is at least one face with a LOW uncertainty, we can trust the similarity score to a certain extent. However, when both faces have $HIGH$ uncertainty, there is no rational basis to trust the similarity score.

Uncertainty of clusters. After clusters are formed using case-specific thresholds, UAC performs one more step. It assigns an uncertainty estimate to each cluster, as follows:

$$uncertainty(C_i) = \frac{\sum_{m=1}^{|C_i|} uncertainty(face_m)}{|C_i|}$$

where cluster C_i consists of $|C_i|$ (similar) faces, and $uncertainty(face_m)$ is the uncertainty of $face_m$ (computed using the formula in Section 4.1). If the $uncertainty(C_i)$ is above the u_t threshold, then UAC considers C_i as a noisy cluster, which is excluded from the final clustering results.

Algorithm 1 describes the key steps in UAC. To cluster n faces, it creates an undirected graph G with n nodes. Initially, G has no edges. Next, UAC performs an all-pairs comparison and adds an edge (i, j) whenever it finds that the similarity score of face pair (f_i, f_j) is the same or above the appropriate similarity threshold (which is varied based on the uncertainty case of face-pair (f_i, f_j). Then, UAC finds the connected components of G – each component corresponds to a cluster. Finally, UAC estimates the uncertainty of each cluster and returns the clusters that have LOW uncertainty. Overall, UAC performs $O(\frac{n^2}{2})$ similarity comparisons to add E edges in G and needs $O(n + E)$ operations to find the connected components.

5.2. Cluster quality metric

Many famous metrics like Purity, BCubed F-score, Pairwise F-score, and NMI have been proposed to evaluate cluster quality. We use these metrics to evaluate the quality of our clusters. However, to better evaluate cluster quality in the presence of data uncertainty, we introduce a new metric, Purity Adjusted Amplification Score (PAAS), which is defined as follows:

$$PAAS = \frac{purity}{amplification}$$

Amplification. It measures the degree of over-clustering with respect to the ground truth. For each person, we count the number of different clusters that the faces similar to this person are assigned to, and then we estimate the amplification as the harmonic mean of all counts:

$$amplification = \frac{I}{\sum_{i=1}^{I} \frac{1}{count_i}}$$

where $count_i$ denotes the count of different clusters for the faces corresponding to the i-th person and I is the total number of persons. A good clustering algorithm should get an amplification score close to 1, and bad ones should score much larger than 1. However, the best amplification is easy to achieve when a clustering algorithm assigns all faces to a single cluster[5].

Algorithm 1 UAC Pseudocode

```
Require: Faces $(f_1, \dots, f_n)$, similarity threshold $(\epsilon)$, uncertainty threshold $(u_t)$
  $G \leftarrow Graph(n)$                       ▷ Initialize undirected graph G with $n$ nodes
  for $i \leftarrow 1, 2, \dots, n$ do
    for $j \leftarrow i, i+1, \dots, n$ do
      $u_i \leftarrow uncertainty(f_i) > u_t$ ? HIGH : LOW
      $u_j \leftarrow uncertainty(f_j) > u_t$ ? HIGH : LOW
      $sim\_threshold \leftarrow \infty$
      if $u_i = LOW$ and $u_j = LOW$ then
        $sim\_threshold \leftarrow \epsilon$
      else if $u_i = LOW$ and $u_j = HIGH$ then
        $sim\_threshold \leftarrow \epsilon - \Delta_{HL}$
      else if $u\_i = HIGH$ and $u\_j = LOW$ then
        $sim\_threshold \leftarrow \epsilon - \Delta_{HL}$
      else if $u\_i = HIGH$ and $u\_j = HIGH$ then
        $sim\_threshold \leftarrow \infty$
      end if
      if $similarity(f_i, f_j) \geq sim\_threshold$ then
        Add $edge(i, j)$ in $G$                 ▷ Mark that $f_i$ and $f_j$ are similar
      end if
    end for
  end for
  $S \leftarrow G.connectedComponents()$        ▷ Find connected components in G
  $C \leftarrow \emptyset$                                          ▷ Good Clusters
  for all $s \in S$ do
    if $uncertainty(s) \leq u_t$ then
      Add $s$ in $C$
    end if
  end for
return $C$
```

Purity. It is computed as follows [2]: first, each cluster is assigned to the most frequent ground truth identity; next, cluster assignment accuracy is estimated as the ratio of the total number of correctly assigned faces and the total number of faces. Purity values lie in between 0 and 1, and a good clustering algorithm should get a score close to 1. However, one can achieve a perfect purity score by forming one cluster per face.

PAAS is a composite metric that is the ratio of purity and amplification, and it measures contradictory qualities of a clustering algorithm. It is easy for a random clustering algorithm to get a perfect score in either amplification or purity, but it is rare for a random algorithm to get perfect scores for both. PAAS score is a value between 0 and 1, and it can be used to compare different clustering algorithms.

6. UAC Evaluation on Noisy Datasets

IJB-A. Table 4 shows the evaluation results for our uncertainty-aware clustering (UAC) algorithm for the IJB-A benchmark. As we increase the uncertainty threshold (u_t), the number of predicted clusters increases slowly, and compared to the ground truth, the quality of the clusters decreases. For example, when $\Delta_{HL} = 0.0$ and we vary u_t from 0.0012 to 0.0014 for the "Deterministic + Cosine" case, the number of predicted clusters increases from 547 to 651. This is in contrast to the results in Table 3 where

[5] Amplification is different from the inverse purity [2] metric, which also achieves the best score when all faces are placed in a single cluster and its value lies in between 0 and 1.

Embedding + Similarity	**Uncertainty Threshold** (u_t)	Δ_{HL}	**IJB-A** (Expected Clusters = 500)					
			Predicted Clusters	Purity	BCubed F-Score	Pairwise F-Score	NMI	PAAS
Deterministic + Cosine	0.0012	0.00	547	0.9931	0.9904	0.9906	0.9969	0.9128
Deterministic + Cosine	0.0013	0.00	587	0.9934	0.9890	0.9901	0.9965	0.8954
Deterministic + Cosine	0.0014	0.00	651	0.9936	0.9863	0.9887	0.9955	0.8570
Deterministic + Cosine	0.0012	0.05	521	0.9825	0.9849	0.9801	0.9956	0.9040
Deterministic + Cosine	0.0013	0.05	565	0.9824	0.9831	0.9791	0.9950	0.8860
Deterministic + Cosine	0.0014	0.05	621	0.9775	0.9776	0.9678	0.9934	0.8499
Deterministic + Cosine	0.0012	0.10	449	0.8549	0.9016	0.7347	0.9723	0.7924
Deterministic + Cosine	0.0013	0.10	501	0.8768	0.9152	0.7551	0.9758	0.7967
Deterministic + Cosine	0.0014	0.10	556	0.8908	0.9238	0.7979	0.9786	0.7875
Deterministic + Cosine	0.0012	0.15	218	0.2876	0.4212	0.0137	0.5173	0.2494
Deterministic + Cosine	0.0013	0.15	261	0.3283	0.4653	0.0157	0.5714	0.2729
Deterministic + Cosine	0.0014	0.15	317	0.3561	0.4951	0.0178	0.6129	0.2801
Probabilistic + MLS	0.0012	0.00	525	0.9863	0.9873	0.9825	0.9963	0.9187
Probabilistic + MLS	0.0013	0.00	553	0.9866	0.9868	0.9823	0.9961	0.9097
Probabilistic + MLS	0.0014	0.00	602	0.9870	0.9853	0.9818	0.9955	0.8796
Probabilistic + MLS	0.0012	10.0	522	0.9806	0.9841	0.9769	0.9956	0.9145
Probabilistic + MLS	0.0013	10.0	551	0.9811	0.9838	0.9771	0.9954	0.9051
Probabilistic + MLS	0.0014	10.0	597	0.9817	0.9826	0.9767	0.9949	0.8763
Probabilistic + MLS	0.0012	20.0	514	0.9778	0.9827	0.9731	0.9951	0.9108
Probabilistic + MLS	0.0013	20.0	548	0.9786	0.9822	0.9734	0.9949	0.9013
Probabilistic + MLS	0.0014	20.0	596	0.9792	0.9810	0.9731	0.9944	0.8718
Probabilistic + MLS	0.0014	30.0	584	0.9737	0.9774	0.9654	0.9934	0.8683
Probabilistic + MLS	0.0012	30.0	502	0.9625	0.9731	0.9450	0.9927	0.8957
Probabilistic + MLS	0.0013	30.0	541	0.9728	0.9782	0.9650	0.9938	0.8950

Table 4: UAC evaluation on IJB-A benchmark when uncertainty threshold (u_t) and Δ_{HL} are varied. We set the cosine similarity threshold to 0.50 and MLS threshold to 2650 (i.e., same as Table 3). The blue-colored rows indicate the recommended range of u_t and Δ_{HL}.

implicit consideration of uncertainty in face representations and similarity function ("Probabilistic + MLS" case) lead to a prediction of over 2700+ clusters.

In UAC, for a fixed u_t, as we increase the Δ_{HL}, the number of predicted clusters are progressively closer to the ground truth. However, there is a trade-off. Increasing the Δ_{HL} value helps to address the *false-negative* problem, where multiple clusters are created for faces of the same person. However, beyond a certain point, increasing Δ_{HL} can also create the *false-positive* problem where faces of different persons are included in a single cluster. Consequently, this lowers the clustering quality. Thus, there is a sweet spot for Δ_{HL} where the clustering accuracy is the highest.

Similarly, we observe a sweet spot for u_t. As we increase u_t, both *false-positive* and *false-negative* problems become more prevalent. A higher Δ_{HL} achieves best trade-off when u_t is higher. However, if u_t is too high then similarity scores become unreliable and Δ_{HL} is not effective anymore.

When we set $u_t \leq 0.0013$, and $\Delta_{HL} \leq 0.05$ for the "Deterministic + Cosine" and $\Delta_{HL} \leq 20$ for "Probabilistic + MLS" (marked in blue color rows), we achieve the best results. In this range, UAC predicts 0.02X - 0.18X more clusters than the ground truth while achieving very high scores in other cluster quality metrics. In contrast, uncertainty-unaware algorithms like DBSCAN [5], AHC [15], and GCN-V [30] predict 4.5X - 9.6X more clusters (results shown in Table 3).

The gray-colored columns in Table 3 and Table 4 show the score of our PAAS metric. Compared to popular metrics like Purity, BCubed F-score, Pairwise F-score and NMI, PAAS does not report a high score when a clustering algorithm produces over-clusters (i.e., it predicts too many clusters compared to the ground truth). PAAS penalizes over-clusterings more compared to other metrics. Specifically, when data uncertainty is high (i.e., a dataset like IJB-A), PAAS metric can help to select a better clustering algorithm. For example in Table 4, based on the PASS score we can rank clusters produced by the "Deterministic + Cosine" with $u_t = 0.0012$ and $\Delta_{HL} = 0.05$ higher than the "Deterministic + Cosine" with $u_t = 0.0013$ and $\Delta_{HL} = 0.05$, although other metrics report very close score for both settings.

Algorithm	**YTF** (Expected clusters = 1594)					
	Predicted Clusters	Purity	BCubed F-Score	Pairwise F-Score	NMI	PAAS
DBSCAN ($minPts = 01$)	7937	0.9555	0.8011	0.6912	0.9494	0.4359
DBSCAN ($minPts = 05$)	3499	0.9556	0.8102	0.6993	0.9536	0.5935
DBSCAN ($minPts = 10$)	3204	0.9565	0.8153	0.7041	0.9548	0.6162
UAC ($\Delta_{HL} = 0.00$)	2521	0.99999	0.8574	0.8703	0.9466	0.7297
UAC ($\Delta_{HL} = 0.05$)	2041	0.9711	0.8632	0.8692	0.9476	0.7600

Table 5: UAC evaluation results on YTF [29] benchmark.

YouTube Faces Database (YTF). We also report experiments results on YTF [29]. It contains 3,425 videos with 611,246 faces from 1594 people. YTF has higher data uncertainty ($mean = 0.00136$) than IJB-A ($mean = 0.00114$). For YTF, we compare UAC with DBSCAN [5] using "Deterministic + Cosine". In particular, we evaluate the effect of the $minPts$ parameter (which controls the desired minimum size of clusters) of DBSCAN. We set the cosine similarity threshold to 0.80. For UAC, we set $u_t = 0.0012$ and set $\Delta_{HL} \leq 0.05$. For DBSCAN, we set $1 \leq minPts \leq 10$. Table 5 shows our evaluation results. For DBSCAN, as we increase $minPts$ the number of predicted clusters comes closer to the ground truth. Even with $minPts = 10$, UAC outperforms DBSCAN algorithm – it predicts 0.28X more clusters (2041) with high scores in other metrics, while DBSCAN predicts almost twice as many clusters (3204) as the ground truth (1594). Again, it shows that leveraging uncertainty explicitly can improve the quality of clustering.

In general, $minPts$ is very hard to set for unknown datasets as it depends on the data distribution. In particular, if a dataset contains many small clusters, then an incorrect $minPts$ setting can discard all clusters. For example, LFW has 4069 ground-truth clusters with one face image each, and a $minPts \geq 2$ simply discards all of them. In contrast, UAC does not require information about the minimum cluster size – rather, UAC discards a cluster when the uncertainty estimate of that cluster exceeds u_t, which can be set without any knowledge of the data distribution.

7. Conclusion

We investigated new ways to improve the accuracy of a clustering task by leveraging uncertainty information. Popular clustering algorithms with uncertainty-augmented probabilistic embedding and similarity functions do not automatically improve clustering accuracy when data uncertainty is high. However, by considering uncertainty information explicitly during clustering and choosing different similarity thresholds, we show that it is possible to improve clustering accuracy significantly for the probabilistic and deterministic embeddings.

References

[1] Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. Chapman & Hall/CRC, 2013. 2

[2] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4), 2009. 4, 7

[3] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[4] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 1996. 2, 3, 4, 8

[6] Senhui Guo, Jing Xu, Dapeng Chen, Chao Zhang, Xiaogang Wang, and Rui Zhao. Density-aware feature embedding for face clustering. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[7] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *arXiv:1607.08221*, 2016. 3

[8] Jochen Görtler, Thilo Spinner, Dirk Streeb, Daniel Weiskopf, and Oliver Deussen. Uncertainty-aware principal component analysis. *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 2020. 1

[9] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 2, 3, 6

[10] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, M. Burge, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3, 4

[11] Wei-An Lin, Jun-Cheng Chen, Carlos D. Castillo, , and Rama Chellappa. Deep density clustering of unconstrained faces. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[12] Wei-An Lin, Jun-Cheng Chen, and Rama Chellappa. A proximity-aware hierarchical clustering of faces. In *IEEE Conference on Computer Face and Gesture Recognition (FG)*, 03 2017. 2

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 3

[14] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2), 1982. 2

[15] Daniel Müllner. fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python. *Journal of Statistical Software, Articles*, 53(9), 2013. 2, 3, 4, 8

[16] Charles Otto, Dayong Wang, and Anil K. Jain. Clustering millions of faces by identity. *The IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(2), Feb. 2018. 2

[17] Charles Otto, Dayong Wang, and Anil K. Jain. Face Clustering: Representation and Pairwise Constraints. *The IEEE Transactions on Information Forensics and Security*, 13(7), July 2018. 4

[18] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015. 1, 3

[19] Michael Jones Paul Viola. Robust real-time object detection. *International Journal of Computer Vision*, 2001. 4

[20] Aruni RoyChowdhury, Xiang Yu, Kihyuk Sohn, Erik Learned-Miller, and Manmohan Chandraker. Improving face recognition by clustering unlabeled faces in the wild. In *The European Conference on Computer Vision (ECCV)*, 2020. 3

[21] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 3

[22] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Transactions on Database Systems*, 42(3), July 2017. 2

[23] Yichun Shi and Anil K. Jain. Probabilistic face embeddings. In *The International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3

[24] Yaniv Taigman, Ming Yang, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *In: IEEE CVPR*, 2014. 1, 3

[25] Makarand Tapaswi, Marc T. Law, and Sanja Fidler. Video face clustering with unknown number of clusters. In *The International Conference on Computer Vision (ICCV)*, 2019. 2, 4

[26] F. Wang, J. Cheng, W. Liu, and H. Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7), 2018. 3

[27] Mei Wang and Weihong Deng. Deep face recognition: A survey. In *arXiv:1804.06655*, 2020. 3

[28] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[29] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 8

[30] Lei Yang, Dapeng Chen, Xiaohang Zhan, Rui Zhao, Chen Change Loy, and Dahua Lin. Learning to cluster faces via confidence and connectivity estimation. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3, 4, 8

[31] Lei Yang, Xiaohang Zhan, Dapeng Chen, Junjie Yan, Chen Change Loy, and Dahua Lin. Learning to cluster faces on an affinity graph. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[32] Yichun Shi. Probabilistic Face Embeddings - Github. `https://github.com/seasonSH/Probabilistic-Face-Embeddings`, 2019. 3