Collaborative Similarity Metric Learning for Face Recognition in the Wild

Batuhan Gundogdu
Department of Electrical and Electronics Engineering
National Defense University, Naval Academy
Istanbul, Turkey
* E-mail: mbgundogdu@dho.edu.tr

Abstract: This paper proposes a joint Siamese neural network training methodology for face recognition that collaboratively makes use of different face image features. This joint training procedure takes two (or possible more) features of two images, and outputs a similarity score. In this paper, we investigate two novel collaborative similarity metric learning techniques for two face features: (a) scale-invariant feature transform (SIFT) and (b) local binary pattern (LBP). We train two Siamese neural networks jointly as a means of early fusion. We provide theoretical as well as empirical comparisons of the proposed models against the baseline systems and yields comparable results to the state of the art techniques.

1 Introduction

The mind-blowing blast in the amount of digital content, brought by the ever-increasing usage of the social media, initiated the recent challenge of working on face images in the wild, where the images possess an increased amount of variations due to lighting, facial expression, pose, age, scale, accessories, occlusions, background and misalignment, rather than just frontal passport pictures. Such a task is generally referred to as unconstrained face recognition or verification in the literature. In this paper, we propose a collaborative or joint distance metric learning methodology for the task of unconstrained face verification. In face verification, the systems are required to determine if two images belong to the same person or not. Considering that the brand new smart phones and tablets are desired to be equipped with a reliable face verification technology, the geometrical contribution of each feature in the discrimination of two images [14]. To approach this problem, we offer a methodology for learning the distance metric jointly, using multiple feature representations of the images.

1.1 Contributions of the Paper

In this paper, we propose a Siamese neural network-based SML methodology to facilitate joint learning of a similarity metric that exploits multiple feature representations of the input images. The key contributions of the paper is as follows:

- The sigma distance that was recently proposed for speech features in a keyword search task [15], is utilized for face images and compared with the other DML/SML techniques in the literature.
- Two collaborative similarity metric learning (CoSiL) methodologies are proposed and discussed both mathematically and empirically. For experiments, two image representations: SIFT and LBP are combined and the SML networks are trained jointly to investigate a more efficient means of early fusion. The proposed CoSiL networks are given in Figure 2; named mass CoSiL and attention CoSiL, for the reasons to be provided along with the mathematical interpretations in Section 2.2.
- The proposed CoSiL methodologies are tested on one of the most challenging test sets in the literature, the Labeled Faces in the wild (LFW) dataset [16], under the image restricted, label-free outside data paradigm, and compared with the similar work in the literature.

The primary approach to this problem has been the investigation of representation and distance metric learning (DML) or similarity metric learning (SML) methodologies [1]. Many studies such as [2–6] investigated obtaining better representations and descriptors to facilitate face verification. On the other hand, many other studies aimed at learning a better discriminative distance metric [7–10], while some investigated learning both distance metric and image representations together [11–13]. Most of the distance metric learning methods focus mainly on learning a linear/nonlinear function of images, so that a desired kinship cost is minimized when two (or more) images are inputted to this function. General application depends on using a single or a concatenation of multiple hand-crafted feature representations of the images as the input. Some studies, however, employ learning these functions from raw images using deep convolutional neural networks [13]. Distance metrics learned on a single feature representation potentially fail to utilize the complementary information brought by different features. An approach to overcome this conundrum is fusing several DML/SML outputs trained with single feature [10]. Another common approach to exploit information content of several features is concatenating them on the input side and training a larger DML model. Although this approach provides the desired effect, it generally suffers from its inability to interpret the geometrical contribution of each feature in the discrimination of two images [14]. To approach this problem, we offer a methodology for learning the distance metric jointly, using multiple feature representations of the images.
The paper is organized as follows: In Section 1 we provide an introduction and the literature review of the subject. In Section 2, the methodology is presented as well as the theoretical background. One the notation and the objectives are set in this section, we discuss in detail the differences and similarities of this work to the literature in Section 3. In Section 4, the experiments are presented after which the conclusions are drawn in Section 5.

1.2 Literature Review

The most substantial work on the unconstrained face verification was initiated with the LFW challenge [17]. Early work on the task focused on descriptor-based methods [18, 19] to conduct the same vs different discrimination using known distance metrics or classifiers such as Linear Discriminant Analysis (LDA) or Support Vector Machines (SVM).

Distance/similarity metric learning methodologies had recently been studied in the literature [20] and were brought to the unconstrained face verification task by [7]. The primary objective in the DML methodology proposed in [7] is learning a Mahalanobis distance on the given representation space. Similarly, Nguyen and Bai proposed the Cosine similarity metric learning methodology in which the cosine similarity cost is optimized per kinship, in the learned projection space [21]. Cao proposed optimizing for a generalized similarity function, obtained by a subtracting Euclidean distance measure from an inner product measure calculated in two different subspaces to be learned [22]. Zheng et al. proposed the logistic similarity metric learning methodology which optimizes over the logistic loss function, calculated with respect to the cosine similarity and the kinship label of the images [23]. Schroff et al. proposed training deep convolutional neural networks that works directly raw images and maps them to Euclidean spaces in which the the desired kinship discrimination is achieved [24].

Detailed explanations of the related work, along with their mathematical representations are deferred to Section 3, in which we provide a thorough comparison of the work proposed in this paper and the related work in the literature, as well as their (dis)similarities.

2 Methodology

In this section, we provide mathematical and theoretical background of the model proposed.

2.1 Assumptions and Preliminaries

This paper presents a methodology for the task referred to as Similarity Metric Learning-SML.

- The “Similarity” in SML is function \( f(x, y) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R} \) of the representations of two images \( x \) and \( y \in \mathbb{R}^d \), such that this function outputs a high value for images that belong to the same person, and a low value for images that belong to different people. For notational simplicity, we denote images by their vector representations \((x, y)\) and we call \( x \) and \( y \) friends if they are extracted from the images of the same person, and foes otherwise.

- The “Metric” in SML is in fact a misnomer. The metric spaces are defined with a distance measure \( d(x, y) \), that satisfies the axioms: (i) non-negativity, (ii) symmetry, (iii) triangle inequality and (iv) identity of discernibles [25]. On the other hand, since there is no formal definition of a “similarity metric”, axioms for \( f(x, y) \) are defined per application. Hence, we do not claim that we conform to the axioms of metric spaces when we call the name “similarity metric”, as in other work in the literature.

- The “Learning” in SML is the essence of this work. SML is actually an optimization task. Given a set of pairs \( X = \{x_t, y_t\} \) and their labels \( r_t \) denoting if they are friends or foes, for \( t = 1 \cdots T \); the goal is to find a function \( f(x, y) \) such that it is higher for friends than foes for all pairs that are not seen in \( X \). Therefore, a proper generalization and learning methodology is desired on top of the cost minimization procedures.

As stated in the introduction section, we propose a methodology for jointly training a combination of several Siamese neural networks, which have shared layers along each feature path. For the sake of simplicity, we will be building our methodology using two of the most common image representations, i.e. (i) SIFT [26, 27] and (ii) LBP [28, 29]. However, the methodology and simply be generalized to using pairs of other features and using multiple features.

2.2 Collaborative Similarity Metric Learning-CoSiL

The CoSiL model, follows the Siamese neural network structure called the sigma distance, that was recently proposed for speech features in to be used in dynamic time warping [30]. Given \( x, y \in \mathbb{R}^d \), the similarity measure is obtained by projection onto a new space and calculating the inner product \( \langle x, y \rangle \). This Mahalanobis inner product is then applied to a sigmoid function so that the similarity function represents a measure of probability of the input vectors being friends:

\[
\sigma(x, y) = \sigma(x^TWWy + b) = \frac{1}{1 + e^{-z}}
\]

where

\[
\sigma(z) = \frac{1}{1 + e^{-z}}
\]

\(\sigma(z)\) is the sigmoid function that reaches 1 and 0 asymptotically as \( z \) reaches \(+\infty\) and \(-\infty\), respectively. This basis SML methodology, which works for a single feature can be seen graphically in Figure 1. In training, we use the training set of triplets \((x_1, y_1, r_1)\) where \( r_1 \) is the label indicating the friendship of the inputs \( x_1 \) and \( y_1 \). As the labels \( r_1 \), we use 1 for friends and 0 for foes and aim to minimize the cross-entropy (CE) objective function. If we call the total set of parameters in the network \( \Theta \), the CE cost function is defined as:

\[
J_{CE}(\Theta; x_1, y_1, r_1) = -r_1 \log(f) - (1 - r_1) \log(1 - f)
\]

where \( f(x_1, y_1) \) is expressed as \( f \) for the sake of simplicity. The JSML model, on the other hand, takes two (or more) feature representations of the images as input and outputs a similarity function. Let \( x_1, y_1 \in \mathbb{R}^{d_1} \) and \( x_2, y_2 \in \mathbb{R}^{d_2} \) be the first and second pair of features for two images, we aim to learn a function \( f(x_1, y_1, x_2, y_2) \) that exploits the complementary information content in both feature representations. For this, we propose two different JSML network topologies: (i) mass JSML and (ii) attention JSML.

2.2.1 Mass JSML: The first topology we refer to as the mass JSML merges the two input after the inner product layer. There are two Siamese neural networks that are shared across the same feature representation. After both input features are projected onto their
corresponding hyper-spaces, the inner products are added before the sigmoid function application:

\[ f(x_1, y_1, x_2, y_2) = \sigma(x_1^T W_1 y_1 + x_2^T W_2 y_2 + b) \]  

(4)

Using the CE cost function, the gradients for each parameter can be calculated as follows:

\[ \Delta b = \frac{dJ}{db} = \frac{dJ}{df} \frac{df}{dz} \frac{dz}{db} = (1 - f) \sigma'(z) \]  

(5)

\[ \Delta W_{1,2} = \frac{dJ}{dW_{1,2}} = \frac{dJ}{df} \frac{df}{dz} \frac{dz}{dW_{1,2}} = (1 - f) \sigma'(z) (x_1 y_{1,2} + x_2 y_{2,2}) \]  

(6)

where \( z = x_1^T W_1 y_1 + x_2^T W_2 y_2 + b \).

**Mathematical Interpretation of mass-JSML:** It is easy to see that each shared projection matrix is updated to the direction decided by the corresponding input feature (6). In other words, \( W_1 \) is updated to the direction \( W_1 (x_1 y_{1,2}^T + y_1 x_{1,2}^T) \) and vice versa. The main help we obtain in adding before the sigmoid is seen on the step size of this step. The update rule of the network is very similar to the single feature SML network, except with one difference. The step size of the update for each shared parameter is decided by the term \( (r - f) \), which carries the collective decision information using both features. Also, the bias term \( b \) is updated by the joint decision of the two input pairs, hence it is less susceptible to outliers and noisy decisions.

### 2.2.2 Attention JSML:

The second topology conducts the merging of the different features, after the individual decisions of the two Siamese neural networks are made. We refer to this system as the attention JSML since the decisions themselves decide how much “attention” each network will receive in training. The two Siamese neural networks are still shared across the same feature representation. The sigma similarities for each feature are calculated and their weighted sum is taken to obtain a joint similarity measure:

\[ f(x_1, y_1, x_2, y_2) = a_1 f_1(x_1, y_1) + a_2 f_2(x_2, y_2) \]  

(7)

where \( a_1 + a_2 = 1 \). \( a_i \) are the weight parameters, which can be taken as uniform or learned from the input. In our implementation, we took \( a_1 = a_2 = 0.5 \)

\[ f_i(x_1, y_1) = \sigma(x_i^T W_i y_1 + b_i) \]  

(8)

With the definition \( z_i = x_i^T W_i y_1 + b_i \), the gradients with respect to the CE cost function is calculated as follows:

\[ \Delta b_i = \frac{dJ}{db_i} = \frac{dJ}{df_i} \frac{df_i}{dz_i} \frac{dz_i}{db_i} \]  

\[ \Delta W_i = \frac{dJ}{dW_i} = \frac{dJ}{df_i} \frac{df_i}{dz_i} \frac{dz_i}{dW_i} \]  

(9)

since it is easy to see

\[ \frac{df_i}{dz_j} = 0, \quad \frac{dz_i}{W_j} = 0 \quad \text{and} \quad \frac{dz_i}{b_j} = 0 \quad \text{for} \quad j \neq i \]

The partial derivatives in this topology bring a normalization coefficient, which is useful in terms of joint learning:

\[ \Delta b_i = \frac{(r - f)}{f(1 - f)} a_i f_i(1 - f_i) \]  

\[ \Delta W_i = \frac{(r - f)}{f(1 - f)} a_i f_i(1 - f_i) W_i (x_i y_i^T + y_i x_i^T) \]  

(10)

**Mathematical Interpretation of attention-JSML:** In (10) the term \( (r - f) \) also exists in bare SML [30] and mass-JSML (5). It decides the direction and the size of the update step. In other words, if the two input vectors are friends but the system output is close to zero, the bias is increased (as much as the difference), and decreased otherwise. For the shared matrices, the projected outer products (6) are added or subtracted by this scale. In attention-JSML, however, this value is scaled with the following term:

\[ a_i f_i(1 - f_i) = a_i (a_1 f_1 + a_2 f_2)(1 - a_1 f_1 - a_2 f_2) \]  

(11)

It is not straightforward to see how (11) contributes to the update. Apart from first term, \( a_i \), which is a pre-decided or learned attention, the fractional term, in fact manages the update for each channel in the Siamese neural network based on their decisions. Given that \( 0 \leq f_i \leq 1 \) due to the sigmoid layer, we can observe the magnitude of this term for various values of \( f_1 \) and \( f_2 \) for a better interpretation. Taking \( a_1 = a_2 = 0.5 \), the surface plot of the scaling term, in \( f_1 \) and \( f_2 \)’s range can be seen in Figure 3. When \( f_1 = f_2 \), in other words the two Siamese networks are in perfect accordance, the attention coefficient is 1 for all values. For off-diagonal values where
$f_1 \neq f_2$, the coefficient reinforces or annihilates the step according to the balance between the two decisions. Particularly, when $f_1 < f_2$ and both of the values are close to zero, it means that the current feature ($f_1$) is potentially detecting correctly and it has a large coefficient. Likewise, when $f_1 > f_2$ and both of the values are close to one, the coefficient is again larger than 1 (yellow edges on the surface curve). On the other hand, when there is a discrepancy between $f_1$ and $f_2$ scores, the coefficient reduces as much as the discrepancy. The blue edges that touch the zero level depict such regions.

This feature of the automatic scaling of the Siamese networks brought by attention-JSML is the reason for its name. Due to its particularly automated choice of update steps, attention-JSML possesses a slow and distinct learning curve. The comparison of the learning curves along with the training details will be provided on the next section.

2.3 Training Topologies and Implementation Details

The flowchart of the proposed networks are illustrated in Figure 2. The top figure presents the contemporary approach of concatenating the features and applying a single SML network. In our implementation, the two inputs (SIFT and LBP) are whitened by Principal Component Analysis (PCA) and dimension reduction is applied to 300 dimensions for both features. This preprocessed data was obtained from [10]. In each of the three networks, we added a batch normalization before the shared layers for faster convergence. Furthermore, a dropout of rate 0.7 was added between the batch normalization and shared layer, which was experimented to be crucial to avoid over-fitting to the training set. In implementation, we used Keras toolkit with Tensorflow back-end. In the optimization procedure we used the *adam* optimizer [31] with the following parameters:

- $\alpha = 0.001$ (learning rate)
- $\beta_1 = 0.9 (1^{st} \text{ moment estimate exponential decay})$
- $\beta_2 = 0.999 (2^{nd} \text{ moment estimate exponential decay})$
- $\epsilon = 10^{-8}$ (denominator de-nullifier)
- $\lambda = 0.001$ (learning rate decay)

The scalar weights were initialized with random Gaussian distributed numbers. For the shared matrices, the experimental analyses showed that better convergence points were obtained when they are initialized with identity matrices for the corresponding size. We also applied an early stopping with a *patience* parameter of 1000 epochs to avoid over training. The neural network topology summaries for the mass-JSML and attention-JSML training are given in 1 and Table 2, respectively.

The training epoch behavior of the JSML models can be seen in Figure 4. The figure demonstrates the change of validation accuracy with respect to training epochs for the mass-JSML, attention-JSML as well as the contemporary concatenation-based SML training. Since the early stopping trick is implemented, each model lasts for different time spans. It can be seen that the mass-JSML, not only converges earlier than the baseline, but it also converges to a better performing point. Attention-JSML, on the other hand, possesses an expectedly interesting training behavior. The convergence is late, so that it has a very poor performance at the time the mass-JSML has already converged to its optimum. However, after enough training time, it surpasses the baseline and the mass-JSML. This is due to the addition of smart attention allocation, which is also learned by the each path in the Siamese neural network.

![Surface of the attention coefficient](image)

**Fig. 3:** Surface plot of the attention coefficient

![Validation accuracy vs epochs](image)

**Fig. 4:** Validation accuracy vs epochs for the proposed models and the baseline concatenation-based model
3 \ Comparison with the Similar Work in the Literature

Having introduced the proposed model in the previous section, this section compares the work with the literature mathematically and points out the similarities and dissimilarities. We believe that such an explanation is crucial to evaluate the originality of the work. For this reason, some of the most similar works to this paper will be explained here. The notations in the original papers are altered to match the explanation here to facilitate direct comparison. The comparison of performances with the following works will be provided in the next chapter along with other related work.

Guillaumin et al. proposed learning a Mahalanobis metric over the representation space [7]. Defining the learned Mahalanobis distance:
\[ d_{W}(x, y) = (x - y)^T W (x - y) \]
(12)
define a probability that the two images belong to the same person to be \( p = \sigma(b - d_{W}(x, y)) \), which is very similar to (4) in our work. This value is then used the cross entropy cost (3), as in our work. Some of the key differences are that we learn a similarity metric based on projected inner products and use it in the sigmoid function. That way, what corresponds to the \( W \) matrix in our work is forced to be symmetric and positive definite.

Nair and Hinton, proposed using Rectified Boltzmann Machines (RBM) with rectified linear units to extract features that are used in distance metric learning [8]. They propose calculating cosine distance of the extracted features from the two images, which is defined in (13). They also describe the probability of being the same person to be \( p = \sigma(-\alpha w d + b) \) with \( w \) and \( b \) being trainable parameters, which is very similar to [7] and this work.
\[ d_{cos}(x, y) = 1 - \frac{x^T y}{||x|| ||y||} \]
(13)
Nguyen and Bai proposed finding a projection space on which the cosine similarity was optimized [21]. The cosine similarity on the new space with the projection matrix \( W \), was defined in [21] to be
\[ CS(x, y, W) = 1 - d_{cos}(Wx, Wy) \]
(14)
The shared projection matrix is very similar to the methodology in this work. While in [21] the target-output control is done by the L2-normalization of the cosine distance, this work uses the sigmoid nonlinearity. Furthermore, Nguyen and Bai use a hinge-like loss to obtain an optimized projection matrix, whereas the cost is cross entropy in this work.

Using this generalized similarity measure, the learning problem is done via the following constraint optimization procedure
\[
\min_{W_1, W_2} \sum_{\forall x_t, y_t \in \mathcal{X}} \xi_t + \frac{1}{2} ||W_1 - I||^2_2 + ||W_2 - I||^2_2, \\
\text{s.t. } \tau_t f(W_1, W_2)(x_t, y_t) \geq 1 - \xi_t, \\
\xi_t \geq 0
\]
(18)
with \( \xi_t \)‘s being slack variables. One other difference from our work is that they take \( r_t \in \{-1, 1\} \), while we take \( r_t \in \{0, 1\} \) as the target values in optimization. This dual formulation corresponds to the following cost function, which can be directly compared with our cross entropy and (15):
\[
J(W_1, W_2) = \sum_{\forall x_t, y_t \in \mathcal{X}} (1 - \tau_t f(W_1, W_2)(x_t, y_t))
\]
(19)
Hu et al. proposed the discriminative deep metric learning (DDML) approach, which is also based on Siamese neural networks [32]. This work proposes taking image representations to another feature subspace, by applying shared neural network operations and then calculating the Euclidean distance on the this new space:
\[
d_{f}^2(x, y) = ||f(x) - f(y)||^2_2
\]
(20)
where,
\[
h^{(1)} = \sigma(W^{(1)} x + b^{(1)}) \\
h^{(2)} = \sigma(W^{(2)} h^{(1)} + b^{(2)}) \\
\ldots
\]
(21)
\[
f(x) = h^{(M)} = \sigma(W^{(M)} h^{(M-1)} + b^{(M)})
\]
The distance value calculated in the new space is used to enforce a margin between friends and foes via a threshold value. The pairs that do not conform to this margin are designed to contribute to the cost. As the cost function, the generalized logistic loss function is used, which is a smoothed approximation of the rectified linear unit function [33].
\[
J = \sum_{t} \log \left( 1 + \exp \left( \beta (1 - \tau_t (\tau - d_{f}^2(x_t, y_t))) \right) \right)
\]
\[
+ \lambda \sum_{m} \left( ||W^{(m)}||^2_2 + ||b^{(m)}||^2_2 \right)
\]
(22)
In (22), the first term is the generalized logistic function, steepness of which is decided by \( \beta \). The part inside the exponential is contributed to the cost is it is non-zero, which means the distance is inside the margin(\( \tau \)). One point to make here is that although DDML, like our study, use the term "metric", \( d_{f}(x, y) \) is not guaranteed to conform to the axioms of metric spaces, on the space that spans \( x \) and \( y \). It is, however, an Euclidean metric space on the space that spans \( h^{(M)} \). Such "metric" learning approaches are, in fact, representation or space learning approaches such that the target distance measure performs better on the new feature space.

Zheng et al. proposed the logistic similarity learning approach [23], which also aims to learn a shared projection matrix to optimize over the learned cosine similarity (4), similar to [21]. The interesting part is that they use generalized logistic loss function (22) as in [32] which penalizes for wrong signed cosine similarities of pairs.
\[
J = \sum_{t} \log \left( 1 + \exp \left( - \tau_t (CS(x_t, y_t, W_t) - b) \right) \right)
\]
\[
+ \lambda ||W - W_0||^2_2
\]
(23)
where the positive constants \( b \) and \( T \) are used to shift the decision boundary and control the steepness, respectively. Although the original works adopt very different techniques, their objective functions
The total distance is then defined as:

\[ d^2_J(x, y) = \sum_{k=1}^{K} a_k d^2_{f_k}(x^k, y^k) \]

(24)

The cost function of DDML involves the weighted sum of each DDML cost function, given with (22), plus a penalty term calculated with the squared differences of \( d_{f_k} \)'s for the same image. The accordance of different feature representations for the same image is enforced with this term.

Another very relevant work addresses the problem of joint similarity learning as a multi-view metric learning [14]. Similar to DDML, they define a joint distance measure obtained by the summation of individual and shared neural network activations. Given a pair of different representations of the same image, \( x^k \) and \( x^\ell \); two projection matrices are defined:

- \( W^k_k \): specific projection matrix for feature \( k \), symmetric positive definite
- \( W^k_{k, \ell} \): common projection matrix for features \( k \) and \( \ell \), symmetric positive definite

The total distance is then defined as:

\[ d^2_h(x, y) = \sum_{k=1}^{K} (x_k - y_k)^T W^k_k (x_k - y_k) \]

(25)

\[ + \sum_{k=1}^{K} \sum_{\ell=1}^{L} (x_k - y_k)^T W^k_{k, \ell} (x_k - y_\ell) \]

The methodologies of [13] and [14] and be compared to our work in that all three work suggest a joint learning of similarity metric, for multiple features. While these two studies work on finding neural network-based representations for Euclidean space, our proposed approach is incorporates this idea to the inner product similarity-based realm.

4 Experiments

We conducted our experiments on the LFW dataset, which is one of the most challenging datasets in the literature. The information about the dataset along with the training procedures, and the numerical results are provided in the following subsections.

4.1 Experiment Set-up

The experiments were conducted on the view-2 configuration of the LFW dataset, which can be viewed and downloaded at http://vis-www.cs.umass.edu/lfw/. LFW includes 13,233 images which belong to 5,749 different people. One other aspect that makes this dataset difficult in machine learning tasks is that more than 70% of the people in the dataset have only 1 image, which constitute 30% of the images in the dataset. Sample image pairs from the dataset can be seen in Figure 5. It can be observed that two pictures of the same person can possess a considerable degree of variation whereas pictures of different people may even look more similar than those of same people.

The face verification set-up, which is referred to as view-2 dataset of LFW, has 6,000 pairs of images, with equal number of friends and foes. To measure the robustness of the proposed algorithms, the dataset is divided into ten subsets, to allow a 10-fold cross validation experiment. The reported numerical numbers in this paper and the works in literature are the average accuracy and the standard deviation measured along 10 different experiments. In each of the 10 experiments, the models are trained with 9 of the subsets and tested with the 10th subset. No experience is allowed to be transferred along different experiments, hence the models are initialized randomly from the beginning from each of the experiments. One other very critical point about the experiment is that the people in the training-test split are mutually exclusive. In other words, none of the people that are seen in the training set exist in the test set. This feature is referred to as the unseen pair match problem.

The SIFT and LBP features are obtained from [10], which use the aligned version of the LFW dataset (LFW-a) [34]. Therefore, we report our results for the image restricted, label-free outside data paradigm. In this setting, the identity information of the face images are unknown and only the friend/foe information is known. Although no outside data usage is explained in the methodology, we do not fit for the image restricted, no outside data protocol because we use the LFW-a dataset, which relies on an alignment algorithm that use label-free outside data.

For each of the 10 subsets, we have conducted individual SML experiments with the features and JSML experiments that use both features. We denote the experiments with the following labels:

- SML-s : SIFT features as input, and the shared SML structure in Figure 1.
- SML-l : LBP features as input, and the shared SML structure in Figure 1.
- conSML : Concatenation of LBP and SIFT features as input and the shared SML structure, shown in Figure 2a.
4.2 Numerical Results

As stated in the experiment set-up, there are 10 test sets each of which have 300 friends and 300 foes. As an initial interrogation of the proposed approach we compared our similarity learning approach with the two most widely used distance metrics, i.e. Euclidean distance and cosine distance. The distance equivalent of the sigmoid similarity in (4), is simply obtained by calculating the complement, \( d_{\sigma} = 1 - f(x, y) \), since it has the nice feature that allows it to be interpreted as a probability measure [30].

### Table 3

<table>
<thead>
<tr>
<th>DISTANCE/KINSHIP</th>
<th>foes</th>
<th>JSML-m</th>
<th>JSML-a</th>
<th>SML-s</th>
<th>SML-l</th>
<th>conSML</th>
<th>SDML-s</th>
<th>SDML-l</th>
<th>cosSML</th>
<th>JSML-m</th>
<th>JSML-a</th>
</tr>
</thead>
<tbody>
<tr>
<td>euclidean+SIFT</td>
<td>0.5930</td>
<td>0.0685</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosine+SIFT</td>
<td>0.5130</td>
<td>0.1122</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosine+LBP</td>
<td>0.5930</td>
<td>0.0685</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>euclidean+Concat.</td>
<td>0.5930</td>
<td>0.0685</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosine+Concat.</td>
<td>0.5130</td>
<td>0.1122</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDML-s</td>
<td>0.3313</td>
<td>0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDML-l</td>
<td>0.3313</td>
<td>0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cosSML</td>
<td>0.3313</td>
<td>0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSML-m</td>
<td>0.3313</td>
<td>0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JSML-a</td>
<td>0.3313</td>
<td>0.0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 4

**Table 4 Experimental Results on LFW View-2 Dataset**

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SML-s</td>
<td>300</td>
</tr>
<tr>
<td>SML-l</td>
<td>300</td>
</tr>
<tr>
<td>cosSML</td>
<td>300</td>
</tr>
<tr>
<td>conSML</td>
<td>300</td>
</tr>
<tr>
<td>JSML-m</td>
<td>300</td>
</tr>
<tr>
<td>JSML-a</td>
<td>300x2</td>
</tr>
</tbody>
</table>

5 Discussion and Conclusion

6 References


[16] Huang, G.B., Learned - Miller, E. ‘Labelled faces in the wild: Updates and new reporting procedures’. (University of Massachusetts, Amherst, 2014). UM-CS-14-003


