

LEARNING DISCRIMINATIVE FEATURES AND METRICS FOR MEASURING ACTION SIMILARITY

Yang Yang

Mubarak Shah

University of Central Florida

ABSTRACT

Measuring the similarity of human actions in videos is a challenging task. Two critical factors that affect the performance include low-level feature representations and similarity metrics. However, finding the right feature representations and metrics is hard. In this paper, we describe a novel approach that *jointly* learns both of them from the data, while current approaches either only learn one or not learn at all. We propose a generative plus discriminative learning method based on gated auto encoders to simultaneously learn the features and their associated metrics. Our method differs from existing feature or metric learning methods in two ways: 1) while other methods treat feature learning and metric learning as independent tasks, we argue that they should be learned jointly since features and metrics are tightly inter-dependent; 2) our method learns more discriminative features than its purely generative counterparts.

1. INTRODUCTION

Measuring the similarity of two human actions is an important task with many applications. It is challenging since matching actions intimately tied to the invariance modeling: two actions belong to the same category if they are invariant under some classes of allowable transformations. Modeling action invariances has received a fair amount of attention in the past. Invariances are common modeled from two perspectives: (1) the feature invariance. Researchers in action recognition have proposed various video descriptors such as HoG[1], sparse spatial-temporal features[2], MBH[3], ISA[4], STIP[5] to (partially) achieve invariance to view-points, illumination changes, camera motions, etc. Recently, feature learning becomes popular as the learned features fit better to a particular task/dataset than the handcraft ones, thus yield better performance. (2) Model invariance by learning metrics.

Predefined distance metrics, such as Euclidean, χ^2 and histogram intersection empirically do not work well on measuring the similarity of high-dimension features. Metric learning [6, 7, 8, 9, 10, 11] can achieve better performance by finding the “good” distance measurement in high dimensional feature spaces that is more suitable for a specific task. However, as the features and metrics are tightly inter-dependent

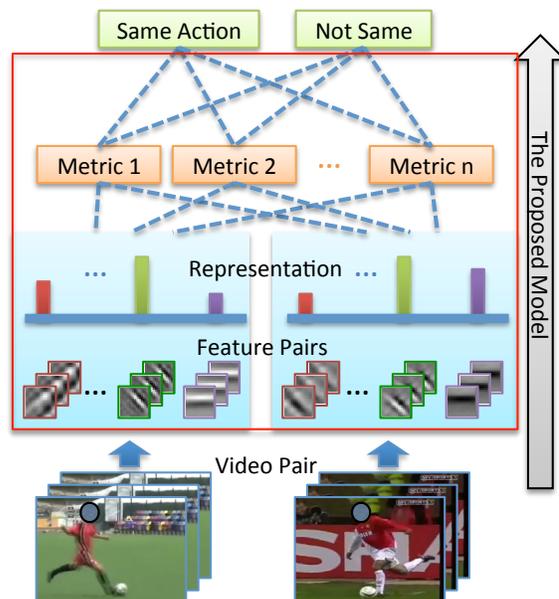


Fig. 1. An illustration of the proposed model. The model (as shown in the red rectangle) learns the features and metrics simultaneously. The features are discovered as the spatial-temporal feature pairs which find the similarity between two videos. We show three pairs of feature sets marked with red, green and purple. The model learns multiple metrics to model the complex transformations between two videos. From the multi-metrics outputs, we can further build a classifier to get the final labels which tell whether the two input videos contain the same action or not.

to each other, designing them separately will likely degrade the overall performance. In this paper, we propose to learn the similarity metrics and the feature representations *jointly*. Figure 1 illustrates the proposed model. More specifically, we learn the spatial-temporal feature pairs and multiple metrics which can model the complex action transformations. In this way, the features and the metrics will co-operate to achieve an optimal solution.

Oftentimes two distinct actions share the same scene background (this happens a lot in sport videos). Existing generative feature learning approaches [4] tend to be distracted

by the common scene instead of learning discriminative features to tell apart two actions.

In order to improve the discriminative ability of the learned features, we propose a new learning method using both generative and discriminative objectives based on gated auto encoders [12].

Our model differs from the existing methods in two ways: first it learns the features and metrics jointly, while others either fix one of them or learn them separately; second, our approach optimizes an objective function with both discriminative and generative terms, which gives our model better discriminative ability. Experiments with qualitative and quantitative results on action verification demonstrate the efficacy of our approach.

2. THE MODEL

The proposed model is illustrated in Fig. 2. Given a video pair x and y together with the “same or not” binary label $c \in \{[0, 1], [1, 0]\}$, the model learns the features U, V , the metrics Z and classifier T simultaneously by minimizing a hybrid objective function consisting of both generative and discriminative terms. The generative term guarantees good reconstructions of the input videos, while the discriminative term produces good predictions on whether two videos are the same or not. We will describe the details in the following subsection. We start with introducing the auto-encoders [13] for unsupervised feature learning, then move to gated auto-encoders for pair matching specifically for video pairs. Finally, we describe our hybrid joint learning of features and metrics.

2.1. Preliminaries: Auto-encoders

Auto-encoders (AE)[13] has been widely used as a basic learning module. It is an unsupervised learning architecture used to pre-train deep networks. The underlying idea of this module is to minimize the reconstruction error of the inputs. A typical structure of an auto-encoder is plotted in the red box in Fig. 2. More specifically, suppose we have a set of spatial-temporal cuboids $\{x^{(n)}\}$ randomly sampled from input video X , where $x^{(n)} \in \mathbb{R}^{D_x}$. The conventional auto-encoders minimize the following reconstruction loss:

$$L_{AE} = \sum_n \|x^{(n)} - U_2 s(U_1 x^{(n)} + b_1) + b_2\|^2 \quad (1)$$

where $U_1 \in \mathbb{R}^{N_f \times D_x}$ is a weight matrix which maps the visible nodes to hidden nodes, $U_2 \in \mathbb{R}^{D_x \times N_f}$ is a weight matrix which reconstructs the visible node from the hidden node. $b_1 \in \mathbb{R}^{N_f}$ is a hidden bias vector, $b_2 \in \mathbb{R}^{D_x}$ is an input bias vector. $s(x) = \frac{1}{1+\exp(-x)}$ is a non-linear sigmoid function. To simplify the formulation, we ignore the regularization term and use linear activation, zero biases and tied

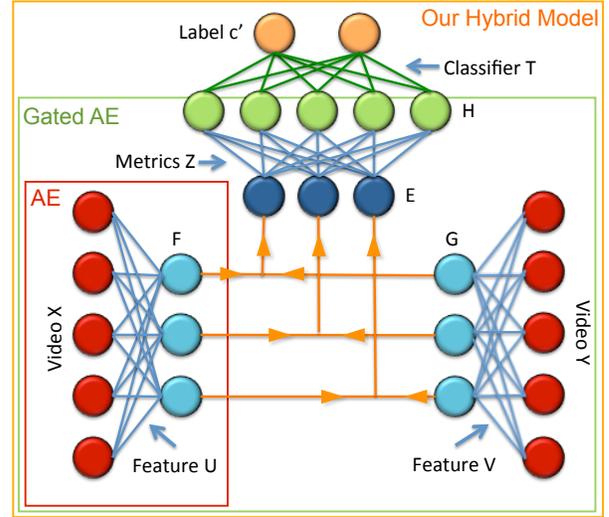


Fig. 2. An illustration of the proposed neural networks. Video X and Y are the input video pair. c' is the predicted label telling whether X, Y are the same action. U, V are the learned feature pairs. F, G are the feature representation of video X, Y . Z is the multi-metrics learned together with U, V . H is the hidden unit. T is the learned classifier. E is computed by the element-wise multiplication of F, G . Each video X or Y is encoded with auto-encoders. Their relations (transformations) are modeled by a gated auto encoder, which learns the features and metrics jointly. Our hybrid model incorporates supervised information on top of the generative encoders to achieve a good balanced between preserving information and good discriminative ability.

weights ($U = U_1 = U_2^T$). Hence, the loss function of auto-encoders can be simplified as:

$$L_{AE} = \sum_n \|x^{(n)} - U^T f^{(n)}\|^2 \quad (2)$$

where, we let hidden units $f^{(n)} = Ux^{(n)}$ as the left light blue nodes in the neural network shown in figure 2. From $f^{(n)}$, we can reconstruct $x^{(n)}$ using $x^{(n)} = U^T f^{(n)}$.

From Eqn. 2 we see that the auto-encoder models the relationship between the input units X and the hidden units F by minimizing the reconstruction error. The learned weights U serve as the filters (features) which will be used in the feature extraction process once the learning is done.

2.2. Gated Auto Encoder

The conventional auto-encoders models individual image x which only emphasizes on the content presented in x itself. When dealing with video pairs $(x^{(n)} \in \mathbb{R}^{D_x}, y^{(n)} \in \mathbb{R}^{D_y})$, we use gated auto encoder (GAE) to model the relationship between x and y (the green box in Fig. 2).

Gated auto encoder models the complex transformations that make two videos the “same”(same action) though the appearance can be very different(different view point, background, etc.). One can think that each hidden unit in H can contribute a “basis transformation” to model the overall dependency between x and y . The activation of hidden units not only depend on x , but also on y . The k^{th} hidden unit activation is:

$$h_k(x; y) = \sum_{j=1}^{D_y} \sum_{i=1}^{D_x} w_{ijk} x_i y_j, \quad (3)$$

where $W \in \mathbb{R}^{D_x \times D_y \times D_h}$ is the learned model. D_x, D_y and D_h are the dimensions of x, y and h respectively. This shows that in the conditional model, hidden variable activities are now given by a simple basis expansion of x and y .

In other words, one can think of the outputs x' as a function of the input image y , in that different inputs y give rise to different transformation over x . The output x' are given by a basis expansion of y and h :

$$x'_i(h; y) = \sum_{k=1}^{D_h} \sum_{j=1}^{D_y} w_{ijk} y_j h_k. \quad (4)$$

Further, we can simplify this model by factorizing W [14] and we can rewrite Eqn. 3 as the following:

$$h^{(n)} = Z e^{(n)}, \text{ where } e^{(n)} = f^{(n)} \cdot g^{(n)} \quad (5)$$

Here \cdot indicates the element-wise multiplication, and $f^{(n)} = U x^{(n)}, g^{(n)} = V y^{(n)}$. As shown in Fig.2, U, V are the filters (features) applied on x and y respectively. Z is the multi-metrics related to each hidden unit h . To encode the relationship between x and y , the hidden units h correlate x and y by using element-wise products(shown in Fig. 2 orange line) between the filter response f, g of x and y as inputs to the hidden variables h . The reason for using multiple hidden units is to model complex transformations. Different y causes different h .

Similarly, given hidden units h and y , we can reconstruct x' by rewriting Eqn. 4 as:

$$x'^{(n)} = U^T (e^{(n)} \cdot g^{(n)}), \text{ where } e^{(n)} = Z^T h^{(n)}. \quad (6)$$

To learn the features U, V , and metrics Z , we can deploy the standard learning criteria, minimize the reconstruction error using gradient-based optimization on the loss function:

$$L_{GAE}(U, V, Z) = \sum_n \|x'^{(n)} - x^{(n)}\|^2. \quad (7)$$

In our verification task, we are interested in the joint distribution over x and y as oppose to the conditional one. As x and y can be interchangeable in the pair matching problem, in practice, we train the model symmetrically by reconstructing

both y from x and x from y . The overall objective function as the sum of the two asymmetric objectives is defined as:

$$L_{GAE-sym} = \sum_n \|y'^{(n)} - y^{(n)}\|^2 + \sum_n \|x'^{(n)} - x^{(n)}\|^2. \quad (8)$$

2.3. Discriminative Learning

So far, the features U, V and metrics Z of the model are learned by a generative loss function which measures an average reconstruction error between the input x, y and the reconstruction x', y' to preserve most of the information from original signal. However, good reconstruction does not necessarily implies good discriminative ability. In order to improve the discriminative ability we propose a hybrid objective by adding a discriminative term to Eqn. 8:

$$L_{hyb} = L_{GAE-sym} + \alpha L_{dis}, \quad (9)$$

where α is a coefficient balancing $L_{GAE-sym}$ and L_{dis} , and

$$L_{dis} = \sum_n \|c'^{(n)} - c^{(n)}\|_2, \quad (10)$$

here $c'^{(n)} = \text{softmax}(T h^{(n)}) \in \mathbb{R}^2$ (since in our case there is only two output nodes: “same” or “not same”). The label c is a binary 2-dimension vector with one element being 1. T is the classifier to be learned.

The first term of Eqn. 9 models the structure and the dependencies among the input components of x, y . The second term represents the supervised goal ensuring that learned model is good for discriminating the similarity between actions. In the rest of the paper, we will call the model learned with Eqn. 9 **a hybrid model**, the one learned with Eqn. 8 **a generative model** and the one learned with Eqn. 10 **a discriminative model**.

3. EXPERIMENTAL RESULTS

The Action Similarity Labeling (ASLAN) dataset [15] is a recent action verification benchmark which includes thousands of video clips collected from YouTube, and over 400 complex action classes. A “same/not-same” challenge is provided, which transforms the action recognition problem from a multi-class labeling task to a binary decision one. The goal is to answer the question of whether a pair of video clips presents the same action or not. We use View1 to select the best parameters and test on View2 using the same evaluation criteria of [15]. The performance is reported based on average performance of ten separate experiments in a leave-one-out cross validation fashion. Each of the ten splits contains 300 pairs of same action videos and 300 not-same pairs. All the videos are first resized to 240×360 . The cuboid size is $16 \times 16 \times 10$ pixels. The number of features is 300 for both video inputs and the number of metrics (H) is set as 40.

	Accuracy+std err	AUC
HOG [16]	58.55 ± 0.8 %	61.59
HOF [16]	56.82 ± 0.6 %	58.56
HNF [16]	58.67 ± 0.9 %	62.16
MIP [16]	62.23 ± 0.8 %	67.5
MBH [3]	59.85 ± 0.8 %	61.5
ISA [4]	59.11 ± 0.7 %	60.3
Generative feature	61.49 ± 0.7 %	65.5
Discriminative feature	59.13 ± 0.6 %	62.2
Hybrid feature	62.05 ± 0.9 %	67.1

Table 1. The average accuracy of different single features with only pre-defined metric $\sqrt{\sum(a \cdot b)}$ on ASLAN dataset. HOG, HOF, HNF, MIP are the best performance on ASLAN reported by [15, 16]. MBH and ISA which has been demonstrated as the state-of-the-art features on several action benchmarks. The last third row is the generative learned feature (using Eqn.8), the last second row is the discriminative learned feature (using Eqn.10), and the last row is the hybrid features learned by Eqn.9. The performance is reported as accuracy with standard error and Area Under the Curve (AUC). One can see that the learned features (last three rows) perform almost equal with other features.

We first quantitatively compare our proposed method with multiple algorithms, including HOG, HOF, HNF, MIP which gives best performance on ASLAN reported by [15, 16], MBH and ISA which has been demonstrated as the state-of-the-art features on several action benchmarks. Besides, we also test the performance of the generative model (learned using Eqn.8) and the discriminative model (learned using Eqn.10). The performance is tested in terms of “feature only” and “feature+metrics”. The accuracy with standard error and AUC is shown in table 1 for “feature only” comparison. The aim of this experiment is to first test the features without the effect or aid of the learned metrics. In this experiments, we use the hand designed features (HOG, HOF, MIP, HNF, MBH) and the learned features (ISA, Generative, Discriminative and Hybrid) followed by the pre-defined metric $\sqrt{\sum(a \cdot b)}$ as suggested by [15]. From table 1 we can see that the learned features either learned by a generative or hybrid objective perform almost equally well with other state-of-the-art features.

We further compare the proposed model in terms of “feature+metrics” as shown in table 2. For the methods which only focus on designing the features, we use CSML [10] to learn the metrics on top of those feature representations. The CSML has been demonstrated to have the best results on MIP as reported in [16]. We also compare our proposed hybrid model with the generative model (learned by Eqn.8). Since the generative model only gives the hidden units response, we train a linear SVM on top of it. Compare table 2 with table

	Accuracy+std err	AUC
HOG+CSML [16]	60.15 ± 0.6 %	64.2
HOF+CSML [16]	58.62 ± 1.0 %	61.8
HNF+CSML [16]	57.2 ± 0.8 %	60.5
MIP +CSML [16]	64.62 ± 0.8 %	70.4
MBH+CSML [3, 10]	61.67 ± 0.9 %	63.26
ISA+CSML [4, 10]	60.97 ± 0.9 %	62.64
Generative model	65.71 ± 0.7 %	70.16
Discriminative model	62.46 ± 0.6 %	68.16
Hybrid model	68.55 ± 0.8 %	72.43

Table 2. The average accuracy of different models on ASLAN dataset. Each model is composed of features and metrics. All the models with CSML design the features and metrics separately. In contrast, the generative model and hybrid model learn the features and metrics simultaneously. Compare with table 1, one can see that the performance can be improved using metric learning. Learning the metrics and features jointly is better than learning them separately as our proposed model is better than MIP+CSML by 4% on average accuracy. Moreover, learning them discriminatively and generatively, is better than pure generative method as the hybrid model also incorporates the label information for classification tasks.

1, we see that using metric learning method such as CSML improves the overall performance of low-level features on ASLAN dataset in general. Further, by combining the co-trained metrics and the hybrid features, our hybrid model outperforms the MIP+CSML method by 4% on average accuracy and 2% on AUC. This demonstrate that to learn the features and metrics jointly can boost the overall performance given that the hybrid features perform equally with MIP in terms of ‘feature only’ as shown in table 1. Moreover, the hybrid model is better than the pure generative model by 2 – 3% as shown in the last two rows in table 2. This means that learning the features and metrics in a discriminative and generative manner is more suitable for classification tasks.

4. CONCLUSION AND FUTURE WORK

We propose to jointly learn the features and metrics directly from raw pixels of videos for measuring action similarity. We consider the discriminative property of features and metrics, and simultaneously learn them by using both a supervised and an unsupervised objective. Extensive experiments results demonstrate the efficacy of our approach. The future work will be incrementally learn the discriminative set of features instead of a fixed size of feature set for large scale action recognition.

5. ACKNOWLEDGEMENTS

The research presented in this paper is supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20071. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

6. REFERENCES

- [1] Navneet Dalal and Bill Triggs, “Histograms of oriented gradients for human detection,” in *International Conference on Computer Vision & Pattern Recognition*, 2005.
- [2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *VS-PETS*, 2005.
- [3] Heng Wang, Alexander Kl, Cordelia Schmid, and Cheng-lin Liu, “Action recognition by dense trajectories,” in *CVPR*, 2011.
- [4] Quoc V. Le, Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng, “Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis,” in *CVPR*, 2011.
- [5] Ivan et al. Laptev, “Learning realistic human actions from movies,” in *CVPR 2008*, 2008.
- [6] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart J. Russell, “Distance metric learning with application to clustering with side-information,” in *NIPS*, 2002.
- [7] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller, “Kernel principal component analysis,” in *ICANN*, 1997.
- [8] Boris Babenko, Piotr Dollár, and Serge J. Belongie, “Task specific local region matching,” in *ICCV*, 2007.
- [9] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, 2000.
- [10] Hieu V. Nguyen and Li Bai, “Cosine similarity metric learning for face verification,” in *ACCV*, 2010.
- [11] Sumit Chopra, Raia Hadsell, and Yann LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *CVPR*, 2005.
- [12] Roland Memisevic, “Gradient-based learning of higher-order image features,” in *ICCV*, 2011.
- [13] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *ICML*, 2008, pp. 1096–1103.
- [14] Roland Memisevic and Geoffrey E. Hinton, “Learning to represent spatial transformations with factored higher-order boltzmann machines,” *Neural Computation*, vol. 22, no. 6, 2010.
- [15] Orit Kliper-Gross, Tal Hassner, and Lior Wolf, “The action similarity labeling challenge,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.
- [16] Orit Kliper-Gross, Yaron Gurovich, Tal Hassner, and Lior Wolf, “Motion interchange patterns for action recognition in unconstrained videos,” in *ECCV*, 2012.