Anticipating human actions by correlating past with the future with Jaccard similarity measures : supplementary material

Basura Fernando
IHPC, A*STAR, Singapore.

Samitha Herath
Dept of Data Science & AI, Monash University

1. How verb, noun models are fused with action model?

We train three separate models, one for noun prediction ($y_n = f_n(X)$), one for verb prediction ($y_v = f_v(X)$) and another one for action prediction ($y_a = f_a(X)$). Let us say, the i^{th} action y_a^i is a composition of k^{th} noun (y_n^k) and j^{th} verb (y_v^j). Therefore, at test time, the action score for y_a^i is obtained from composition of noun and verb prediction scores by multiplying them $y_n^k \times y_v^j$ and the score prediction obtained from $f_a(X)$. Therefore, the final score is given by the following

$$y_a^i + y_n^k \times y_v^j \tag{1}$$

where each action a^i is a composition of noun n^k and verb v^j.

2. Architecture for action anticipation using Fisher Vectors

Fisher vector sequence has 64 dimensional features. Let use denote the observed video feature sequence by V_{obs} which contains features from $t = t_{obs}$ seconds and ends at $t = t_{obs} + T$ seconds. We denote the future sequence by V_f which starts at $t = t_{obs} + T + \delta_t$ and ends at $t = t_{obs} + T + \delta_t + T$ where T is typically 2. We use δ_t of one. The feature summarizing architecture for Fisher vectors are shown in figure 1.

3. More ablation on Jaccard losses.

We raise the concern about cosine similarity due to two reasons. First, it is not a smooth loss. Secondly, it overly rely on the angle between two vectors and not so much on the magnitude. However, L2 distance considers both the magnitude and the angle between vectors. However, it is not bounded. Can a combination of L2 and cosine similarity perform better than Jaccard Vector Similarity? We answer this question here. We report results in Table 1.

4. Ablation on action anticipation architecture.

The full model used for action anticipation is presented in the Equation 6 of the main paper. For ease of reference

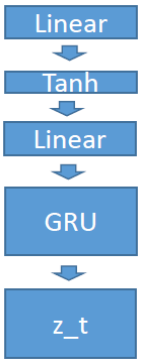

Figure 1. Feature summarizing network for Fisher vectors on Breakfast dataset.

Method	JHMDB	JHMDB	UCF101
Observation (%)	10	20	20
L2 loss	50.4	51.5	67.5
Cosine loss	51.8	54.5	66.9
L2 + Cosine	56.6	63.1	70.1
JVS	**62.6**	**64.7**	**72.3**

Table 1. Results on early action prediction to show the the impact of JVS loss.

we state the model equation here as well.

$$\beta[L_{CE}(y_o, \hat{y_o}) + L_{CE}(y_f, \hat{y_{of}})] + L_{CE}(y_f, \hat{y_f}) + \lambda exp(-\phi(\mathbf{z_h}, \mathbf{z})) \tag{2}$$

Now we add a new term β to in equation 5 for observation branch to evaluate the impact of it. Additional result in shown Table 2. When we set $\beta = 0$ and $\lambda = 1$ we see that the observed branch (β) helps when $\lambda = 1$ and does not not help when $\lambda = 0$.

Our baseline model contains on the future loss and therefore the overall loss of the baseline model is given by

$$L_{CE}(y_f, \hat{y_{of}}) \tag{3}$$

When we set the λ term to zero the model reduces to the following loss

$$L_{CE}(y_o, \hat{y_o}) + L_{CE}(y_f, \hat{y_{of}}) + L_{CE}(y_f, \hat{y_f}) \tag{4}$$

Dataset	Breakfast		Epic-Kitchen	
Modality	FV	(R(2D+1D))	(R(2D+1D))	
Measure	Accuracy		Top 1	Top 5
$\lambda = 0.0$ and $\beta = 0.0$	23.4	24.3	9.82	24.48
$\lambda = 0.0$ and $\beta = 1.0$	23.9	24.6	10.01	24.82
$\lambda = 1.0$ and $\beta = 0.0$	27.6	27.1	14.25	30.46
$\lambda = 1.0$ and $\beta = 1.0$	28.6	28.0	15.20	32.54

Table 2. The impact of components of our model on action anticipation on Breakfast and Epic-Kitchen55 datasets.

Dataset	Breakfast		Epic-Kitchen	
Modality	FV	(R(2D+1D))	(R(2D+1D))	
Measure	Accuracy		Top 1	Top 5
Baseline	23.4	24.3	9.82	24.48
Eq. 6 ($\lambda = 0.0$)	23.9	24.6	10.01	24.82
JVS	28.6	28.0	15.20	32.54
JCC	28.6	28.1	14.12	32.16
JFIP	30.3	30.9	13.89	33.31

Table 3. The impact of $\lambda exp(-\phi(\mathbf{z_h}, \mathbf{z})$ on action anticipation on Breakfast and Epic-Kitchen55 datasets.

Predictor	Noun		Verb	
Measure	Top 1	Top 5	Top 1	Top 5
JVS	26.37	45.75	41.34	78.05
JCC	26.25	45.94	41.72	78.33
JFIP	23.94	45.30	39.99	78.08
ALL	27.27	49.97	43.55	79.10

Table 4. Verb and noun anticipation results on Epic Kitchen55 validation set.

We evaluate the impact of these models in Table 3.

$$\beta[L(y_o, \hat{y_o}) + L(y_f, \hat{y_{of}})] + L(y_f, \hat{y_f}) + \lambda exp(-\phi(\mathbf{z_h}, \mathbf{z})) \quad (5)$$

It is interesting to see that model with $\lambda = 0.0$ performs only slightly better than the baseline model. Once, we incorporate the term $\lambda exp(-\phi(\mathbf{z_h}, \mathbf{z}))$, the results improve significantly for all datasets and features.

5. Noun and verb anticipation accuracy

In this section we evaluate the noun and verb anticipation performance on Epic Kitchen55 dataset. We use rgb, optical flow and object streams to train our model. As before we use Resnet18(2D+1D) network pre-trained on Kinetics dataset for action recognition. Then we fine-tune these models using our anticipation architecture and loss functions. Results are shown in Table 4.