A Generative Approach to Zero-Shot and Few-Shot Action Recognition

Ashish Mishra[*,1] , Vinay Kumar Verma[†,1], M Shiva Krishna Reddy[*], Arulkumar S[*]
Piyush Rai[†] and Anurag Mittal[*]
[*]Indian Institute of Technology Madras [†]Indian Institute of Technology Kanpur
{vkverma,piyush}@cse.iitk.ac.in, {mishra,shiva,aruls,amittal}@cse.iitm.ac.in

Abstract

We present a generative framework for zero-shot action recognition where some of the possible action classes do not occur in the training data. Our approach is based on modeling each action class using a probability distribution *whose parameters are functions of the attribute vector representing that action class. In particular, we assume that the distribution parameters for any action class in the visual space can be expressed as a linear combination of a set of basis vectors where the combination weights are given by the attributes of the action class. These basis vectors can be learned solely using labeled data from the known (i.e., previously seen) action classes, and can then be used to predict the parameters of the probability distributions of unseen action classes. We consider two settings: (1) Inductive setting, where we use only the labeled examples of the seen action classes to predict the unseen action class parameters; and (2) Transductive setting which further leverages unlabeled data from the unseen action classes. Our framework also naturally extends to few-shot action recognition where a few labelled examples from unseen classes are available. Our experiments on benchmark datasets (UCF101, HMDB51 and Olympic) show significant performance improvements as compared to various baselines, in both standard zero-shot (disjoint seen and unseen classes) and generalized zero-shot learning settings.*

1. Introduction

Action Recognition is an important problem in Computer Vision in which knowledge about a sequence of actions is learned from a large collection of video clips. It is a challenging task due to the inherent variability in actions, non-deterministic occlusion patterns, abrupt changes in illumination, cluttered dynamic background, and noisy videos. Knowledge about an action is inferred usually by learning from the labelled data in a supervised manner. Even as more complex models are being built, it is a common observation that the number of categories of actions is progressively increasing (for example, one of the earliest benchmark datasets KTH has 6 categories while Olympic, HMDB and UCF datasets have 16, 51, and 101 categories, respectively). Consequently, annotating videos of this growing number of categories can be a very cumbersome task and consequently restricts the scalability of a fully supervised action recognition for a large number of categories.

To circumvent this problem, Zero-Shot Learning (ZSL) of actions has been actively pursued [35, 34, 21]. In the conventional Action Recognition framework, only the classes present in the training data can be recognized by the model during the test phase. In Zero-Shot Learning, however, the model is expected to recognize and categorize action classes that did not appear in the training phase at all. The information about the unseen classes is provided via other modalities such as language in the form of textual descriptions, *word2vec* [19] or human annotated attributes. Essentially, the model has to learn to recognize the unseen action classes based on the knowledge acquired from the data instances of the seen action classes. Zero-shot learning is typically defined in two settings: (1) the conventional setting, in which set of classes for the training and test instances are disjoint ($Y_{tr} \cap T_{te} = \varnothing$); and (2) the generalized zero-shot (GZSL) setting, in which the set of classes for the training and test instances may have an overlap [16, 20]. The generalized zero-shot setting is considered much harder than standard setting (disjoint setting) since the learned models tend to be biased towards predicting seen classes at training time (as they are learned solely from the unseen class training data). While much of the prior work in ZSL has focused on the conventional setting, the focus has recently shifted to the more realistic GZSL setting.

In this work, we present a simple generative approach for zero shot action recognition, which works in both standard as well as generalized ZSL setting. Our approach models each action class as a probability distribution in the visual space where the parameters of this distribution are assumed to be a linear combination of a set of “basis” parameters,

[1] Both authors contributed equally.

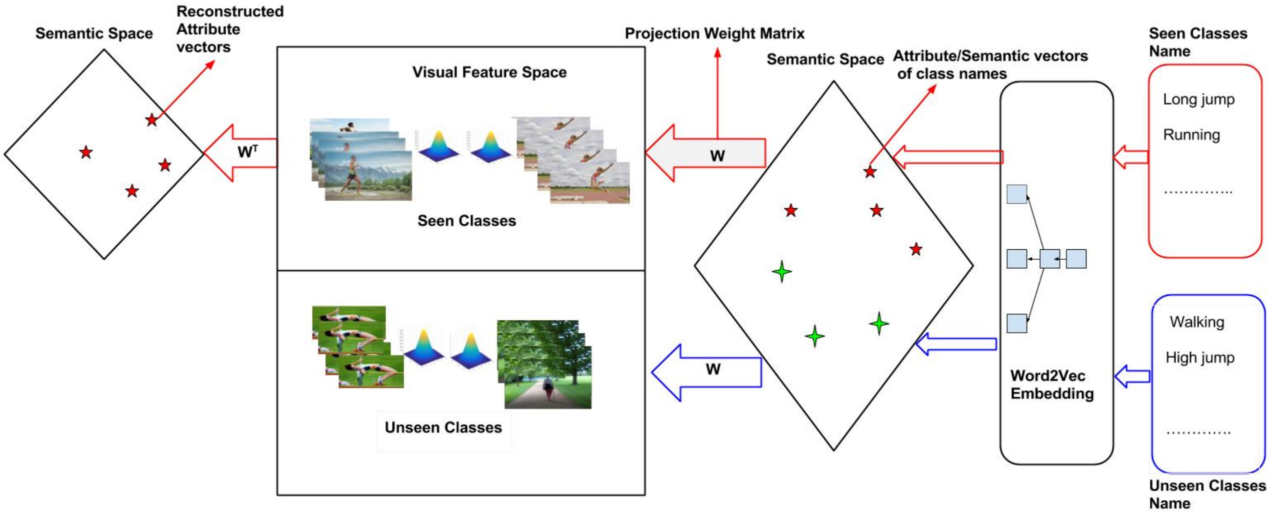

Figure 1. **Proposed Model**: Each class attribute is projected to the visual space, In the visual space each class is represented by a Gaussian distribution. To avoid information loss, a reconstruction regularizer is added.

where the combination weights are given by the (known) attribute vector of that class. This is akin to assuming that each action class can be represented as a combination of a set of "prototype" action classes. The complete architecture of our model is shown in Fig. 1.

Once the basis vectors are learned using the training instances from seen action classes, the parameters of an unseen action class distribution can be easily computed via a weighted combination of the learned basis vectors, with weights being the attributes of the respective unseen class. The loss function (More details in Methodology section) is formulated in such a way that, for each seen action class, the weighted combination of the basis vectors is close to the maximum likelihood estimate (MLE) of the class distribution's parameters. The MLE estimate can also be replaced by maximum-a-posteriori (MAP) estimate. Our approach is akin to the one proposed recently in [29] for zero-shot learning, though our focus an application is specifically the zero-shot *action recognition* problem. In addition, we add an additional "reverse direction" regularizer to encourage reconstruction ability of the class attribute vectors from the parameters of the seen class distributions so as to minimize the information loss. Note that this is akin to an autoencoder (cf, Fig. 1). One of the appealing aspects of our model is that it admits a simple closed-form solution.

Our main contributions can be summarized as follows

- We provide a probabilistic generative approach for zero-shot learning (ZSL) where each action class is represented by a Gaussian distribution (although the Gaussian can be replaced by other distributions without changing the rest of our model).
- We show that our approach, although simple, generalizes well to the unseen classes in the inductive setting and improves over the state-of-the-art.
- We show that our approach can be easily generalized to the transductive setting where unlabeled data from unseen classes are available at training time.
- Our model also naturally extends to be "few-shot learning" setting where a few examples of each unseen class are available as well. In particular, the parameters of the class distribution can be updated easily given a few additional labeled examples from that class. Through extensive experimentation on three benchmark datasets, we show that our simple approach gives significant performance gains in all three settings over the state-of-the-art methods.
- Finally, since our approach is generative, we can also synthesize novel examples for any unseen class by sampling from the respective class distribution. Since we can now have labeled data from seen as well as unseen classes, it is possible to train a classifier in the generalized zero-shot setting which is much harder than the standard (disjoint) setting.

2. Methodology

For the zero-shot action recognition setting, we denote the total number of seen action classes by S and the total number of unseen action classes by U. We take a generative classification based approach to the action recognition problem where we assume that the data instances of each action class (seen/unseen) c are generated by a distribution

$p(\mathbf{x}|\boldsymbol{\theta_c})$. Without loss of generality, and for simplicity of exposition, we will assume these distributions to be Gaussians (note that our approach can be used with other distributions as well). In the Gaussian case, the parameters $\boldsymbol{\theta_c}$ consist of the mean vector $\boldsymbol{\mu_c} \in \mathbb{R}^D$ and a diagonal covariance matrix $\boldsymbol{\Sigma}_c = \text{diag}(\boldsymbol{\sigma_c^2})$, where $\boldsymbol{\sigma_c^2} \in \mathbb{R}_+^D$. We assume a diagonal covariance matrix to reduce the total number of parameter estimated and prevent overfitting especially when the number of examples from each class is small. However, other forms for the covariance matrix can also be used.

Given labeled data from the seen classes, it is straightforward to estimate the parameters $\boldsymbol{\mu_c}, \boldsymbol{\sigma_c}$ using Maximum Likelihood Estimation (MLE) or Maximum-a-Posteriori (MAP) estimation. For example, using MLE, the mean is estimated as $\boldsymbol{\mu_c} = \frac{1}{N_c}\sum_{i=1}^{N_c} \mathbf{x_i}$ and $\boldsymbol{\sigma_c^2} = \text{diag}(\frac{1}{N_c}\sum_{i=1}^{N_c}(\mathbf{x_i} - \boldsymbol{\mu_c})(\mathbf{x_i} - \boldsymbol{\mu_c})^\top)$ where N_c denotes the number of labeled examples from class c.

However, this approach cannot be used to estimate the parameters $\boldsymbol{\theta_c}$ ($c = S+1, ..., S+U$) of unseen classes due to unavailability of labeled data corresponding to unseen classes. To resolve this problem, we model the parameters $\boldsymbol{\theta_c} = (\boldsymbol{\mu_c}, \boldsymbol{\sigma_c^2})$ of each seen/unseen clas as a function of the respective class attribute vector $\mathbf{a_c}$, i.e., $\boldsymbol{\theta_c} = f(\mathbf{a_c})$. In the zero-shot learning setting, the class attribute vector $\mathbf{a_c} \in \mathbb{R}^K$ is either provided by a human expert or as the WORD2VEC embedding of the name of the action.

The function f can be linear or nonlinear and can be learned using the labeled data instances of seen classes in visual feature space. Once learned, the function f can be used to predict $\boldsymbol{\theta_c}$ for all the unseen class actions $c = S+1, \ldots, S+U$ using their respective class attributes.

A simple choice of f is a linear model that maps the class attributes $\mathbf{a_c}$ to the class parameters $\boldsymbol{\theta_c}$. In the Gaussian class distribution case, for the mean $\boldsymbol{\mu}_c$, such a linear function f can be defined as

$$\boldsymbol{\mu_c} = f_\mu(\mathbf{a_c}) = \mathbf{W}_\mu \mathbf{a_c} \tag{1}$$

Note that the above linear model represents the mean $\boldsymbol{\mu}_c \in \mathbb{R}^D$ as a weighted linear combination of K basis vectors $\mathbf{W}_\mu = [\mathbf{w}_{\mu_1}, \mathbf{w}_{\mu_2}, .., \mathbf{w}_{\mu_\mathbf{K}}] \in \mathbb{R}^{D\times K}$ is a set of *learned* basis vectors in the visual space.

The basis vectors $\mathbf{W}_\mu$ can be learned using the seen class training data. In particular, given the empirical estimates $\hat{\boldsymbol{\mu}}_{\boldsymbol{c}}, c = 1, \ldots, S$ of means of the seen class distributions, we can use $(\mathbf{a_c}, \hat{\boldsymbol{\mu}}_{\boldsymbol{c}})$ as "training data" to learn the regression model $\mathbf{W}_\mu$ that maps $\mathbf{a_c}$ to $\hat{\boldsymbol{\mu}}_{\boldsymbol{c}}$.

While the above model can be seen as mapping the class attribute vector $\mathbf{a_c}$ to the class mean $\boldsymbol{\mu}_c$, we further impose the condition that the class means can also be used to *reconstruct* the class attribute vector via a "reverse map" akin to an autoencoder, i.e., $\mathbf{a_c} = \mathbf{W}_\mu^\mathbf{T}\boldsymbol{\mu_c}$, which leads to

$$\boldsymbol{\mu}_c = \mathbf{W}_\mu \mathbf{a_c} = \mathbf{W}_\mu \mathbf{W}_\mu^\top \boldsymbol{\mu_c} \tag{2}$$

A similar procedure can be employed for learning the mapping from the class attributes $\mathbf{a_c}$ to the variance parameters $\boldsymbol{\sigma}_c^2$ of the distribution of class c via another set of basis vectors $\mathbf{W}_{\sigma^2}$. Sections 2.1 and 2.2 provide more details.

Once the basis vectors $\mathbf{W}_\mu, \mathbf{W}_{\sigma^2}$ (which define the functions f_μ and f_σ) are learned, we can use them to estimate the parameters (e.g., $\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2$) of the distribution of each unseen class. For example, given the class attribute vector $\mathbf{a_c}$ of an unseen class $c = S+1, \ldots, S+U$, we can estimate $\boldsymbol{\mu_c}$ simply as $\boldsymbol{\mu_c} = \mathbf{W}_\mu \mathbf{a_c}$.

The mapping f (which is essentially a regression model) from the class attribute vector to the parameters of the class distribution can be linear or nonlinear. We describe both these cases in the next two sections.

2.1. Linear Regression

Given the labeled data from seen classes $c = 1, ..., S$, we can estimate their class distribution parameters using MLE. We can then learn the functions f_μ and f_{σ^2} using training data of the form $(\mathbf{a_c}, \boldsymbol{\mu_c})_{c=1}^S$ and $(\mathbf{a_c}, \boldsymbol{\sigma_c^2})_{c=1}^S$. In the linear regression approach $\boldsymbol{\mu}_c = f_\mu(\mathbf{a_c})$ and $\boldsymbol{\sigma}_c^2 = f_\sigma(\mathbf{a_c})$, we assume the functions f_μ and f_σ to be linear projections with weight matrices, $\mathbf{W}_\mu$ and $\mathbf{W}_\sigma$, making this problem equivalent to the following regression problem:

$$\boldsymbol{\mu}_c = \mathbf{W}_\mu \mathbf{a_c} \quad \text{s.t.} \quad \mathbf{a_c} = \mathbf{W}_\mu^\mathbf{T} \boldsymbol{\mu_c}$$

$$\rho_c = \log \boldsymbol{\sigma}_c^2 = \mathbf{W}_{\sigma^2} \mathbf{a_c} \quad \text{s.t.} \quad \mathbf{a_c} = \mathbf{W}_{\sigma^2}^\mathbf{T} \boldsymbol{\sigma_c^2}$$

The projection matrices $\mathbf{W}_\mu$ and $\mathbf{W}_{\sigma^2}$ can be easily learned using a multi-output ridge regression problem with training data $(\mathbf{a_c}, \boldsymbol{\mu_c})_{c=1}^S$ and $(\mathbf{a_c}, \boldsymbol{\sigma_c^2})_{c=1}^S$. These problems have simple closed form solution and we omit the equations here for brevity. We give details equations for the nonlinear case, as shown below.

2.2. Nonlinear Regression

For the non-linear regression, we first map the attributes $\{\mathbf{a_c}\}_{c=1}^S$ to the kernel space using the kernel function k which is defined as a nonlinear mapping ϕ. Using the Representer theorem [24], we can re-formulate the regression problem in kernel space as given in Eq. 3. Note that instead of computing the $\phi(\mathbf{a_c})$ explicitly, we have to compute only the dot product $\phi(\mathbf{a_c})^T\phi(\mathbf{a_{c'}}) = k(\mathbf{a_c}, \mathbf{a_{c'}})$ for the non-linear mapping of the two class c and c'. Let $\mathbf{K}$ be the kernel matrix of size $S \times S$ containing pairwise similarities of the attributes of the seen classes, $\mathbf{M}$ be the $D \times S$ matrix containing the means of the distributions of all the seen classes, then the attribute to mean nonlinear mapping can be learned by solving the following problem

$$\begin{aligned} \min_{\mathbf{W}_\mu} ||\mathbf{M} - \mathbf{W}_\mu \mathbf{K}||_F^2 + \lambda_\mu ||\mathbf{W}_\mu||_2^2 \\ \text{s.t.} \quad \mathbf{K} = \mathbf{W}_\mu^* \mathbf{M} \end{aligned} \tag{3}$$

Eq 3 shows our main objective function. Here the first term can be interpreted as learning an optimal weight matrix that projects the attribute space to the visual space using the kernel regression. The second term ensures that we can reconstruct the attribute vector from the visual space and acts as a regularization term. Akin to an autoencoder [9], we assume the two mappings to be reverse of each other

$$\mathbf{W}_{\mu}^{*} = \mathbf{W}_{\mu}^{T}$$

Therefore the complete objective can be written as:

$$\begin{aligned}\mathbf{W}_{\mu}^{*} = \underset{\mathbf{W}_{\mu}}{\operatorname{argmin}} ||\mathbf{M} - \mathbf{W}_{\mu}\mathbf{K}||_F^2 + \lambda_{\mu}||\mathbf{W}\mu||_2^2 \\ + \lambda_1||\mathbf{K} - \mathbf{W}_{\mu}^{\mathbf{T}}\mathbf{M}||_F^2\end{aligned} \tag{4}$$

The next section provides details of the optimization procedure used for solving Eq. 4

2.2.1 Optimization

Noting $Tr(\mathbf{K}) = Tr(\mathbf{K}^T)$ and $Tr(\mathbf{W}_{\mu}^T\mathbf{M}) = Tr(\mathbf{M}^T\mathbf{W}_{\mu})$, Eq. 4 can be written as:

$$\begin{aligned}\mathbf{W}_{\mu}^{*} = \underset{\mathbf{W}_{\mu}}{\operatorname{argmin}} ||\mathbf{M} - \mathbf{W}_{\mu}\mathbf{K}||_F^2 + \lambda_{\mu}||\mathbf{W}\mu||_2^2 \\ + \lambda_1||\mathbf{K}^T - \mathbf{M}^T\mathbf{W}_{\mu}||_F^2\end{aligned} \tag{5}$$

Taking the derivative of Eq. 5 and equating to zero we have.

$$\mathbf{M}\mathbf{M}^T\mathbf{W}_{\mu} + \mathbf{W}_{\mu}\lambda_1\mathbf{K}\mathbf{K}^T + \lambda_{\mu}\mathbf{W}_{\mu} = (1+\lambda_1)\mathbf{M}\mathbf{K}^T \tag{6}$$

$$\mathbf{M}\mathbf{M}^T\mathbf{W}_{\mu} + \mathbf{W}_{\mu}(\lambda_1\mathbf{K}\mathbf{K}^T + \lambda_{\mu}) = (1+\lambda_1)\mathbf{M}\mathbf{K}^T \tag{7}$$

The above equation has the form

$$\mathbf{AW} + \mathbf{WB} = \mathbf{C} \tag{8}$$

This is a well-known Sylvester equation which can be solved using the Bartels-Stewart algorithm [5] efficiently, and several off-the-shelf solvers exist (we used a MATLAB implementation for the same). The various quantities in the above equation are defined as

$$\mathbf{A} = \mathbf{M}\mathbf{M}^T \tag{9}$$

$$\mathbf{B} = \lambda_1\mathbf{K}\mathbf{K}^T + \lambda_{\mu} \tag{10}$$

$$\mathbf{C} = (1+\lambda_1)\mathbf{M}\mathbf{K}^T \tag{11}$$

Likewise, the nonlinear model $f_{\sigma_i^2}$ can be learned by solving:

$$\begin{aligned}\mathbf{W}_{\sigma^2}^{*} = \underset{\mathbf{W}_{\sigma^2}}{\operatorname{argmin}} ||\mathbf{R} - \mathbf{W}_{\sigma^2}\mathbf{K}||_F^2 + \lambda_{\sigma^2}||\mathbf{W}_{\sigma^2}||_2^2 \\ + \lambda_2||\mathbf{K} - \mathbf{W}_{\sigma^2}^T\mathbf{R}||_F^2\end{aligned} \tag{12}$$

Again, taking derivatives and setting to zero gives

$$\mathbf{R}\mathbf{R}^T\mathbf{W}_{\sigma^2} + \mathbf{W}_{\sigma^2}(\lambda_2\mathbf{K}\mathbf{K}^T + \lambda_{\sigma^2}) = (1+\lambda_2)\mathbf{R}\mathbf{K}^T \tag{13}$$

The above equation is also in the form of $\mathbf{AW} + \mathbf{WB} = \mathbf{C}$

$$\mathbf{A} = \mathbf{R}\mathbf{R}^T \tag{14}$$

$$\mathbf{B} = \lambda_2\mathbf{K}\mathbf{K}^T + \lambda_{\sigma^2} \tag{15}$$

$$\mathbf{C} = (1+\lambda_2)\mathbf{R}\mathbf{K}^T \tag{16}$$

Given the learned parameters $\mathbf{W}_{\mu_c}$ and $\mathbf{W}_{\sigma_c^2}$, the parameters of data distribution for unseen classes $c = S+1, \ldots, S+U$ are estimated as:

$$\boldsymbol{\mu}_c = \mathbf{W}_{\mu}\mathbf{k_c}, \quad \& \quad \boldsymbol{\sigma}_c^2 = \exp(\rho_c) = \exp(\mathbf{W}_{\boldsymbol{\sigma}^2}\mathbf{k_c}) \tag{17}$$

Where $\mathbf{k_c} = [\mathbf{k}(\mathbf{a_c}, \mathbf{a_1}), ..., \mathbf{k}(\mathbf{a_c}, \mathbf{a_S})]$ denotes an $S \times 1$ vector of kernel-based similarities of the class attribute vectors of the unseen class c with the class attribute vectors of all the seen classes.

In the aforementioned procedure for estimation of the unseen class distribution parameters uses only seen class labelled data. In this setting, the unseen classes unlabeled data have not been used. This setting is called as an *inductive setting*. If we have access to the unseen classes test instances at the training time, we can use these to improve the estimation of distribution parameters of unseen classes. This is the *transductive setting* which we describe next.

2.3. Transductive setting

One of the unique advantages of the proposed generative approach is that unlabeled data from unseen classes can be leveraged to improve the parameter estimates ($\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c$). In zero-shot learning, training and test data could possibly come from different domains. Therefore, it is very likely that parameters learned in the training, will not work well for the test data. This phenomenon is called domain shift. An illustrative view of the domain shift can be seen in Fig. 2. One way to overcome this issue is to use unlabeled data to further fine-tune the parameters learned by the inductive approach which only uses the labeled data from the seen action classes. In the transductive setting [34], we assume that the test data is also available at the training time. This data can help mitigate the bias towards the seen classes. In this work, we handle the domain shift problem by initializing the parameters $\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c$ using the learned basis vectors from the inductive learning phase, which are then fine-tuned using the unlabeled test data from the unseen classes using the an Expectation-Maximization (EM) algorithm.

Since each class distribution is assumed to be a Gaussian, this EM based procedure is equivalent to a Gaussian mixture model (GMM) on the unlabeled test data $(\mathbf{x_n})_{n=1}^{N_u}$ from unseen classes. This GMM has U mixture components, with each corresponding to an unseen class and is initialized by

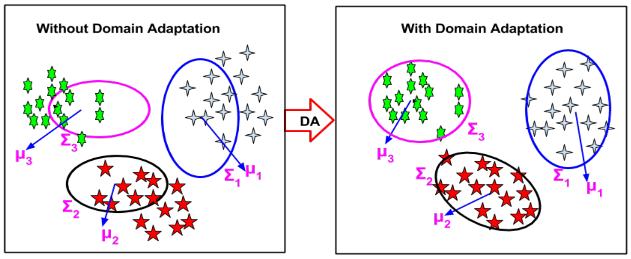

Figure 2. **Domain Adaptation illustrative example**: Each class attribute is projected to the visual space, In the visual space each class are represented by a distribution. Because the seen and unseen class are disjoint, there is a problem of domain shift.

the estimated parameters of unseen classes $(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)_{c=S+1}^{S+U}$ in the inductive setting. The procedure for transductive setting can be briefly summarized as follows

1. **Initialize:** Let the initial estimate of the unseen class parameters be $\boldsymbol{\Theta} = (\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)_{c=S+1}^{S+U}$ where $\boldsymbol{\mu}_c = \mathbf{W}_\mu \mathbf{a_c}$, $\boldsymbol{\sigma}_c^2 = \exp(\mathbf{W}_{\sigma^2}\mathbf{a_c})$. Here $\mathbf{W}_\mu$ and $\mathbf{W}_{\sigma^2}$ are estimated from seen class data using equations 4, 12 (assuming we have used the nonlinear regression model in the inductive phase).

2. **Expectation Step:** Infer the probabilities for each example $\mathbf{x_n}$ belonging to each of the unseen classes $c = S+1, ..., S+U$ as

$$p(y_n = c|\mathbf{x_n}, \boldsymbol{\theta}) \propto \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$$

   where the class priors $p(c)$ are assumed to be uniform.

3. **Maximization Step:** Use the inferred class labels to re-estimate $\Theta = (\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)_{c=S+1}^{S+U}$. These updates have closed form solution as in the standard GMM.

4. Go to step 2 if not converged.

2.4. Few-shot Action Recognition

In few-shot action recognition, we have a small number of labeled examples for each of the unseen classes. Since our method assumes a Gaussian distribution for each class, we can easily extend our zero-shot action recognition method to few-shot action recognition. To this end, we treat the initial estimate obtained using the previous approach as the prior. Due to the conjugate nature of the Gaussian, we can update the estimates $(\mu_c, \sigma_c^2)_{c=S+1}^{S+U}$ obtained from zero-shot action recognition method in a straightforward manner when such labeled data for unseen classes is provided. In particular, given a small number of labeled data $(\mathbf{x_n})_{n=1}^{N_c}$ for unseen class c the parameters of this class can be directly updated as:

$$\boldsymbol{\mu}_c^{FS} = \frac{\boldsymbol{\mu} + \sum_{n=1}^{N_c} \mathbf{x_n}}{1+N_c} \tag{18}$$

$$\boldsymbol{\sigma}_c^{2(FS)} = \left(\frac{1}{\boldsymbol{\sigma}_c^2} + \frac{N_c}{\boldsymbol{\sigma}^{2*}}\right)^{-1} \tag{19}$$

where $\boldsymbol{\sigma}^{2*} = \frac{1}{N_c}\sum_{n=1}^{N_c}(\mathbf{x_n} - \boldsymbol{\mu}_c)^2$ denotes empirical variance of N_c observations from the unseen class c.

2.5. Extension to Other Distributions

Finally, we would like to emphasize that although we have consider Gaussians to model each class, our approach applies to any parametric distribution $p(\mathbf{x}|\theta_c)$ as it essentially boils down to learning a mapping from the class attribute vectors $\mathbf{a}_c$ to the distribution parameters θ_c. The choice of a Gaussian or an exponential family distribution [29], due to conjugacy, makes our estimation procedure particularly simple in the transductive and few-shot settings, but our framework is not restricted to these. Other density estimation methods such as deep generative models can also be used [32].

3. Related Work

ZSL can be viewed as an interplay of three subproblems: a visual representation of data instances (feature representation), semantic representation of all classes such as *word2vec* representation [19], and learning a function which establishes the relationship between visual representations and semantic representations of each class[13, 14].

For visual (or feature) representation of class instance, popular hand-crafted features such as HOG [3], HOF [2], ITF [30] were designed. However, the proven utility of deep features for many tasks such as Object Recognition [10, 25, 27], Object Detection [4], etc., has made features from well performing CNNs such as [15], Two-Stream CNN[18], 3DCNN [6] ubiquitous for Action Recognition tasks including the zero shot setting. By using 3DCNN features in ZSL, a significant boost in accuracy has been observed [31]. Semantic representation of a class provides additional, complementary information to the visual features of the classes. Typically, two types of semantic representations have been widely used in the ZSL literature: attribute representations [12] and word vector representations [19]. Attribute representations are manually annotated vectors for each class based on the gesture and motion appearance of the objects in the video. Word Vector representations are automatically learned from a large amount of textual data (Wikipedia Corpus). Word2vec models have been used successfully for extracting semantic word vectors from class names [31, 7, 35]. The core step in ZSL is to find a function or projection matrix which can establish a relationship between visual space and semantic space in such a way that visual features of classes map close to their semantic features and vice versa. For example, we would like to have visual features of 'running' map close to semantic features of 'running' and far away from an unrelated action such as 'eating'.

Note that our framework is similar in spirit to such methods with a key difference: Instead of learning a mapping between the semantic feature and visual features, we learn a mapping from the semantic features and the parameters of the *distributions* representing the classes.

Most methods for zero-shot learning are evaluated on image classification whereas only a few methods have been proposed for zero-shot action recognition in the literature [31, 35, 34, 21]. Such methods typically assume the inductive or the transductive setting. The most popular approach to ZSL is learning a linear compatibility between the visual and semantic space [1]. [23, 9] provide novel regularizations while learning a linear compatibility function. ESZSL [23] models the relationship between features and attributes as a linear compatibility function while explicitly regularizing the objective. UDA [8] uses a domain adaptation technique by using unlabeled data of unseen classes for better estimation of the parameters.

Our model is inspired by the recently proposed model [29], which is a simple generative approach for zero-shot learning. However, their model does not have the reconstruction regularizer (autoencoder-style reverse mapping) from visual to attribute space and their focus is on image classification whereas here we have focused on action recognition. In another recent work, [9] proposed a semantic auto-encoder for zero-shot learning which introduced the reconstructability regularizer. This paper works only in the inductive setting and their approach is not generative. Our generative approach can be seen as a combination of the generative approach of [29] with auto-encoder style regularizer proposed by [9].

Among prior works on zero-shot action recognition in transductive setting, [35] proposed a transductive framework for zero-shot action recognition, which uses unlabeled unseen class data for training the model. In their work, they introduced a manifold-regularized regression and a data augmentation strategy to enhance the performance. They have also introduced a multi-task visual-semantic mapping for zero-shot action recognition. In addition, they used prioritized auxiliary data augmentation for domain adaptation and improved the mapping between visual and semantic spaces.

Because of the generative nature of our proposed approach, we can *synthesize* the data from unseen class based on attribute and train the classifier. This approach helps to reduce the baisness in the case of Generalize Zero-Shot Learning. The efficacy of the proposed approach for the GZSL as well as ZSL can be seen from the experiment on three standard datasets.

4. Experiments

Datasets and Settings: We evaluate our proposed method in three of the most challenging video action recognition datasets, UCF101 [26], HMDB51 [11] and Olympic [17], widely used as benchmark datasets. We report mean accuracy along with standard deviation on 30 independent test runs with random train/test class splits.

- **UCF101:** [26] is human action recognition data set with 101 different classes of actions and total of 13320 video clips. In our experiments, we split the classes into 51 seen and 50 unseen class respectively. '

- **HMDB51:** [11] is the one of the most challenging human action recognition dataset with 51 different classes of human actions and total number of 6766 video clips. Each class has more than 100 video clips. For the evaluation of our model, we perform a 26/25 split for seen and unseen classes respectively.

- **Olympic:** [17] This dataset has 783 videos from 16 different classes with seen/unseen class split being 8/8.

Dataset	#videos	#classes	seen/unseen	Attribute dim
UCF101	13320	101	51/50	115
HMDB51	6676	51	26/25	N/A
Olympic	783	16	8/8	40

Table 1. Dataset details and their train test split on all the three dataset used in our experiment.

Visual features: The quality of visual features directly affect the efficacy of the model. We use deep features as they have been shown to be successful in many computer vision tasks. In our experiments, we use the latest convolutional 3D(C3D) visual features provided by [28]. This model was pre-trained on the sports-1M dataset. We extract the outputs of fc6 layer for all segments similar to [28] and then averaged over the segments to form a 4096-dimensional video representation which is used as the input visual features.

Class attributes: Two types of class attribute vectors (semantic representation of the classes) are widely used in ZSL: human labeled attributes [12] and automatically learned distributed semantic representations such as word vectors [19]. Word vector representation is learned automatically by a skip-gram model trained on the google news text corpus provided by Google. Each word is represented by a 300 dimensional vector. We experiment on both attribute and word2vec representations. For HMDB51 dataset, to the best of our knowledge, there is no publicly available attribute representations of the classes. Hence only word2vec is used for HMDB51. However, for UCF101 and Olympic datasets, 115 and 40 dimensional attribute vectors are available respectively [26, 17].

Method	Embed	Olympic	UCF101	HMDB51
HAA [16]	A	46.1 ± 12.4	14.9 ± .8	N/A
DAP [13]	A	45.4 ± 12.8	14.3 ± 1.3	N/A
IAP [14]	A	42.3±12.5	12.8 ± 2	N/A
ST [33]	W	N/A	13.0±2.7	10.9±1.5
SJE [1]	W	28.6±4.9	9.9±1.4	13.3±2.4
SJE [1]	A	47.0±14.8	12.0±1.2	N/A
ESZSL [23]	W	39.6±9.6	15.0±1.3	18.5±2
UDA [8]	A	N/A	13.2±1.9	N/A
Bi-dir [31]	A	N/A	20.5±.5	N/A
Bi-dir [31]	W	N/A	18.9±.4	18.6±.7
Ours	A	**50.41±11.2**	**22.74±1.2**	N/A
Ours	W	34.12±10.1	17.33+1.1	**19.28±2.1**

Table 2. Results on inductive setting for standard zero shot learning setting(disjoint setting) for the action recognition. Here A represents the human annotated attribute vectors and W represents the *word2vec* embedding.

Hyper-parameters: Our model consists of four hyper-parameters: λ_μ, λ_1 (Eq. 4) and λ_{σ^2}, λ_2 (Eq. 12) for estimating the projection matrix for mean and variance. The optimal values of hyper-parameters are chosen via cross validation on the seen classes. For cross validation, we randomly fix 1/4th of the seen classes as validation classes and conduct five trials on 30 random splits (same as [31]). For generalized ZSL setting, the number of synthesize examples for unseen classes is also hyper-parameter which we find using cross-validation and observe best model performance for 200 synthesized examples.

4.1. Inductive and Transductive ZSL

In our first set of experiments, we evaluate our model for zero-shot action recognition with inductive and transductive setting and compare with a number of state-of-the-art methods.

Evaluation Metric: We evaluate our model using 30 different splits into seen and unseen classes provided by [31] for UCF101 (51/50), HMDB51 (26/25) datasets. For Olympic dataset, we generate 30 random splits for seen and unseen classes (8/8). We use the average accuracy for all 30 splits as the evaluation metric. For fair comparison, we run five such trials for 30 random splits and present the final accuracy with average and standard deviation.
For generalized zero-shot setting we have evaluated for 30 different splits as above and calculated the average accuracy for seen and unseen classes. The final evaluation metric of our model is on the harmonic mean of the average accuracy of seen and unseen classes, as used in [20, 16, 1].

Inductive setting: In this setting, it is assumed that only the labeled data from the seen classes is available during training. Table 2 shows the experimental results in the inductive setting of the zero-shot action recognition problem. We assume that the train and test classes are disjoint. Note that this assumption is made for all the evaluation settings in this work. In this setting, we obtain an improvement of **3%** over the state-of-the-art on the Olympic dataset. On UCF-101, which is the most used dataset for zero shot action recognition, the proposed model outperforms state-of-the-art on attribute-based semantic representations. For HMDB dataset, the attribute vectors are not available. Hence, we present results only on word2vec embeddings. Our model outperforms the state-of-the-art for this dataset as well. We believe the improvements can be attributed to its inherent nature of sharing information across classes (by modeling each as a basis combination of prototype classes) and its simple estimation procedure.

Method	Embed	Olympic	UCF101	HMDB51
PST [22]	A	48.6±11	15.3 ±2.2	N/A
ST [33]	W	N/A	15.8±2.3	15.0±3
TZWE [34]	A	53.5±11.9	20.2±2.2	N/A
TZWE [34]	W	38.6±10.6	18.0±2.7	19.1 ±3.8
Bi-dir [31]	A	N/A	**28.3±1.0**	N/A
Bi-dir [31]	W	N/A	21.4±.8	18.9±1.1
UDA [8]	A	N/A	13.2±.6	N/A
Ours	A	**57.88±14.1**	24.48±2.9	N/A
Ours	W	41.27±11.4	20.25±1.9	**20.67±3.1**

Table 3. Results on transductive setting for the standard zero shot action recognition. Here A represents the human annotated attribute vectors and W represents the *word2vec* embedding.

Transductive setting: In the transductive setting, it is assumed that the unlabeled data of the unseen classes is also available at train time. Table 3 shows the performance of our model in the transductive setting. The unlabeled data from unseen classes helps us mitigate the bias towards the seen classes. In this setting, our model outperforms the state-of-the-art in the Olympic and HMDB datasets. The performance on the UCF-101 dataset is slightly worse, where [31] has the best performance. However, note that we outperform [31] in the inductive setting.

Method	Embed	Olympic	UCF101	HMDB51
HAA [16]	A	49.4 ± 10.8	18.7 ± 2.4	N/A
SJE [1]	W	32.5±6.7	8.9±2.2	10.5±2.4
ConSE [20]	W	37.6 ± 9.9	12.7 ± 2.2	15.4± 2.8
Ours	A	**52.41±12.2**	**23.74±1.2**	N/A
Ours	W	**42.23±10.2**	**17.45±2.2**	**20.10±2.1**

Table 4. Results on the transductive setting for generalized zero-shot learning setting for the action recognition. Here A represents the human annotated attribute vectors and W represents the *word2vec* embedding.

4.2. Generalized ZSL

In this setting, the test data may come from both seen and unseen classes. In this setting, from the seen classes, we separate 20% of the data for testing and remaining 80% data is used as training data for calculating $\mathbf{W}_{\mu}$ and $\mathbf{W}_{\sigma^2}$ which is used to predict the mean ($\boldsymbol{\mu}_c$) and variance ($\boldsymbol{\sigma}_c^2$) for the unseen classes. One way to handle this setting is to assign each test data-point to the class whose estimated distribution gives the highest score. However, we notice that such an approach is biased towards seen classes since the model has not seen any unseen class examples. In our approach, we propose the following solution to this issue: we synthesize class instances of unseen classes using the $\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c^2$ which are obtained from the transductive setting approach; these class instances are called pseudo class instances for unseen classes. Here we generate 200 instances for each unseen classes. Since we now have labelled data for seen classes and pseudo labelled data for unseen classes, we train SVM classifier for labelled seen classes data and pseudo labelled data for unseen classes. We then pass the test data (unseen class data plus 20% seen class data) to the trained SVM classifier for classification. Table 4 presents the performance of our model in the generalized setting for zero-shot action classification which clearly shows that it significantly outperforms state-of-art on all the datasets.

4.3. Few-shot action recognition

Finally, we experiment with the few shot action recognition setting and present the results. Here only a small number of examples for each of the unseen classes are available during training. Our generative model provides a simple way to update the parameters of the class distribution using equation 18, 19 . It is clear from the Table 5 that availability of the few data points of the unseen classes significantly improves the performance which is now comparable to that of supervised learning. Note that we do not assume any unlabeled data from the unseen classes in this setting. We test our model with varying number of examples of each unseen classes. The plot of accuracy with respect to the number of samples per class is shown in Figure 3.

Dataset	2 samples	3 samples	4 samples	5 samples
UCF101	68.78±3.3	73.49±2.2	76.51±2.1	78.68±1.8
HMDB51	42.10±3.6	47.54±3.3	50.34±3.4	52.58±3.1
Olympic	73.20±7.4	75.35±7.3	80.21 ±7.24	83.81±7.11

Table 5. Inductive setting with few-shot action recognition

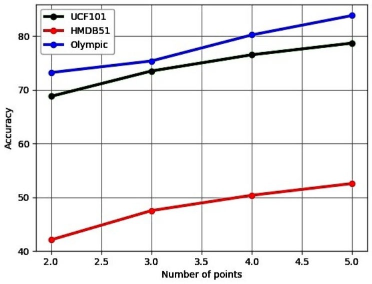

Figure 3. Accuracy vs number of data points for few-shot learning

5. Conclusion

We have presented a simple, probabilistic, generative model based framework for zero-shot action recognition. The proposed approach performs well in both the inductive and transductive setting for the standard (disjoint) and generalized zero-shot learning. The generative aspect of our model unables synthesizing unseen class examples and can effectively work in the generalized ZSL setting. In addition, the ability of leverage unlabeled data (transductive setting) helps address the domain shift problem between seen and unseen classes. A particularly appealing aspect of our model is that it yields a closed form solution for the parameters to make it fast and easy to implement. Experimental results are shown to achieve state-of-the-art performance. The proposed method also generalizes to few-shot action recognition setting, achieving comparable results to *fully supervised* learning using only a few synthesized examples from each unseen class.

Acknowledgments: Thanks to Vismay who always gives his support in discussion and experiment. Special thanks to Prof. Hema A Murthy for her valuable feedback. Vinay Verma acknowledges support from Visvesvaraya Ph.D. fellowship. Piyush Rai acknowledges support from Visvesvaraya Faculty Fellowship and a grant from Tower Research.

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015.

[2] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *computer vision and pattern recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1932–1939. IEEE, 2009.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recogni-*

tion, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE, 2005.

[4] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[5] A. Jameson. Solution of the equation ax+xb=c by inversion of an m*m or n*n matrix. *SIAM Journal on Applied Mathematics*, 16(5):1020–1023, 1968.

[6] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[7] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2452–2460, 2015.

[8] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2452–2460, 2015.

[9] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011.

[12] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[14] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3337–3344. IEEE, 2011.

[17] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3337–3344. IEEE, 2011.

[18] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib. Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *arXiv preprint arXiv:1703.10667*, 2017.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[20] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[21] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang. Zero-shot action recognition with error-correcting output codes. In *Proc. CVPR*, 2017.

[22] M. Rohrbach, S. Ebert, and B. Schiele. Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54, 2013.

[23] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161, 2015.

[24] B. Scholkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. 2001.

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[29] V. K. Verma and P. Rai. A simple exponential family framework for zero-shot learning. *arXiv preprint arXiv:1707.08040*, 2017.

[30] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[31] Q. Wang and K. Chen. Zero-shot visual recognition via bidirectional latent embedding. *arXiv preprint arXiv:1607.02104*, 2016.

[32] W. Wang, Y. Pu, V. K. Verma, K. Fan, Y. Zhang, C. Chen, P. Rai, and L. Carin. Zero-shot learning via class-conditioned deep generative models. In *AAAI Conference on Artificial Intelligence (AAAI-18), Louisiana, USA.*, 2018.

[33] X. Xu, T. Hospedales, and S. Gong. Semantic embedding space for zero-shot action recognition. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 63–67. IEEE, 2015.

[34] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, pages 1–25, 2017.

[35] X. Xu, T. M. Hospedales, and S. Gong. Multi-task zero-shot action recognition with prioritised data augmentation. In *European Conference on Computer Vision*, pages 343–359. Springer, 2016.