LETTER

Triple Loss Based Framework for Generalized Zero-Shot Learning

Yaying SHEN[†], Qun LI[†a)], *Nonmembers*, Ding XU[††], *Member*, Ziyi ZHANG[†], *and* Rui YANG[†], *Nonmembers*

SUMMARY A triple loss based framework for generalized zero-shot learning is presented in this letter. The approach learns a shared latent space for image features and attributes by using aligned variational autoencoders and variants of triplet loss. Then we train a classifier in the latent space. The experimental results demonstrate that the proposed framework achieves great improvement.

key words: *generalized zero-shot learning, triple loss, image classification, variational autoencoder*

1. Introduction

Zero-Shot Learning (ZSL) is one way proposed to address the challenge of learning from limited labeled data. ZSL aims to recognize unseen classes which are not available during training stage by learning knowledge from seen classes which are available during training stage and some auxiliary information, such as attributes.

In conventional ZSL, train and test classes are disjoint, since train classes only include seen classes and test classes only include unseen classes. We can only measure the performance of the conventional ZSL by classification accuracy on unseen classes. Obviously, this setting basically does not exist in reality. Generalized Zero-Shot Learning (GZSL) is a more practical and challenging variant of ZSL, since train classes under the same setting as the conventional ZSL, but test classes include both seen classes and unseen classes. We measure the performance of the GZSL by the harmonic mean of the classification accuracy on seen and unseen classes.

Early ZSL works focus on embedding image features and attributes to a latent space. We can recognize objects by comparing distances between the representations of image features and attributes in the latent space. In GZSL, embedding-based methods suffer from serious bias owing to the lack of image features of unseen classes during training stage. To alleviate the bias problem of embedding-based mathods in GZSL, feature generation based methods, such as Variational Autoencoder (VAE) [1] based methods and Generative Adversarial Network (GAN) [2] based methods, are proposed. Since unseen classes are not available during training stage, feature generation based methods can generate training features for unseen classes. In this work, we propose a GZSL framework which combines the advantages of the embedding-based and feature generation based methods. Due to the instability of GAN-based loss functions during training stage, we use VAEs in this work.

Image features and attributes are data from different modalities. Recently, cross-modal embeddings are used in GZSL. Cross-modal embeddings are mostly based on autoencoders, such as ReViSE [3]. ReViSE learns a joint representation for data from different modalities by matching their latent distributions. Otherwise, some latent space models use cross-reconstruction to preserve discriminative information in the joint latent representation. Inspired by this, we use VAEs to learn a cross-modal embedding as a latent space. In this work, we train two VAEs to encode and decode features from image features and attributes, respectively. In order to better learn a joint latent representation, we align VAEs by matching their latent distributions and conducting cross-reconstruction. Compared with CVAE [4] which uses the conventional VAE, we use VAE for generating low-dimensional latent features instead of directly generating image features.

Specifically, we use some variants of triple loss to optimize the latent space. Triple loss function [5] is a widely used loss function. The triplet loss can make that data of same classes are closer to each other than those of different classes. Person Re-Identification (ReID) is a branch of image classfication. Several approaches for person ReID have already achieved great performance by using some variants of the triplet loss to train their models. In addition, we explore the impact of different variants of triple loss on the performance of our framework.

Our main contributions are as follow: (1) We propose triple loss based framework for GZSL. (2) We explore the impact of different variants of triple loss on the performance. (3) The proposed framework achieves great improvement on four popular benchmark datasets.

2. Proposed Method

2.1 Problem Definition

The problem definition of GZSL is as follows. Train set $\mathcal{D}_s = \{(x_s, y_s, a_s) | \; x_s \in X_s, \; y_s \in Y_s, \; a_s \in A\}$, where X_s denotes

Manuscript received September 3, 2021.
Manuscript revised December 8, 2021.
Manuscript publicized December 27, 2021.

[†] The authors are with the School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing, China.

[††] The author is with the School of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China.

a) E-mail: liqun@njupt.edu.cn (Corresponding author)

DOI: 10.1587/transinf.2021EDL8079

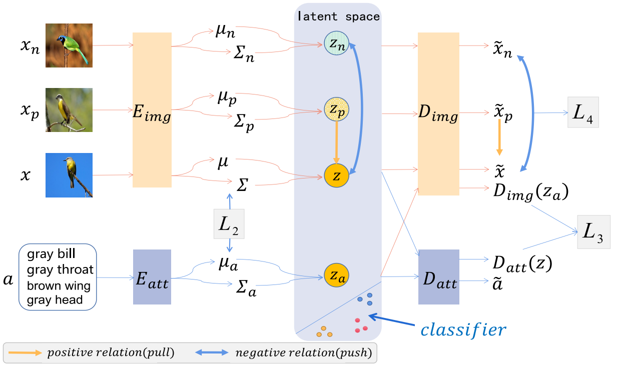


Fig. 1 Overview of our framework.

the image feature set of seen classes, Y_s is the corresponding label set of X_s and A denotes the attribute set. In addition, we use an auxiliary training set $\mathcal{D}_u = \{(y_u, a_u)| \, y_u \in Y_u, a_u \in A\}$, where Y_u denotes the label set of unseen classes. Y_u and Y_s are disjoint. The complete train set is $\mathcal{D}_{tr} = \mathcal{D}_s \cup \mathcal{D}_u$. The test set $\mathcal{D}_{te} = \{(x_{te}, y_{te})| \, x_{te} \in X_s \cup X_u, y_{te} \in Y_s \cup Y_u\}$. The GZSL aims to learn a classifier $f_{GZSL} : X \rightarrow Y_s \cup Y_u$.

2.2 Triple Loss Based Framework

The overview of our framework is shown in Fig. 1. The basic blocks of our framework are two VAEs, one is for image features and the other is for attributes. For learning a joint representation for image features and attributes, we match their latent distributions and conduct cross-reconstruction. Specifically, we use some variants of triple loss to optimize the latent space, as train a softmax classifer in the latent space.

In VAE, the encoder inputs a image feature x and outputs parameters of the Gaussian distribution $\mathcal{N}(\mu, \Sigma)$. Then, a latent vector z is generated from $\mathcal{N}(\mu, \Sigma)$ via the reparametrization trick [1]. We pass the z to the decoder and expect it to reconstruct x. The $D(E(x))$ denotes the reconstructed x. The training loss of a VAE can be formulated as:

$$L_{VAE} = L_{re}((x,D(E(x))) - KL[\mathcal{N}(\mu,\Sigma),\mathcal{N}(0,I)], \tag{1}$$

where L_{re} is the reconstruction loss, the KL is the KL-divergence, and $\mathcal{N}(0,I)$ is standard Gaussian distribution. In this work, we choose L1 norm as the reconstruction loss.

The basic VAE loss of our framework is the sum of two VAEs losses:

$$\begin{aligned} L_1 = & L_{re}(x, D_{img}(E_{img}(x))) - \beta KL(\mathcal{N}(\mu, \Sigma), \mathcal{N}(0, I)) \\ & + L_{re}(a, D_{att}(E_{att}(a))) - \beta KL(\mathcal{N}(\mu_a, \Sigma_a), \mathcal{N}(0, I)), \end{aligned} \tag{2}$$

where x denotes the image feature, a denotes the attribute. The image feature x and attribute a belong to the same class. E_{img} and D_{img} are the encoder and the decoder of image features, respectively. E_{att} and D_{att} are the encoder and the decoder of attributes, respectively. $\mathcal{N}(\mu,\Sigma)$ and $\mathcal{N}(\mu_a,\Sigma_a)$ are latent Gaussian distributions of the image feature x and attribute a, respectively. β is the weight of the KL-Divergence.

Our latent Gaussian distributions of image features and attributes are matched by minimizing their 2-Wasserstein distances. The 2-Wasserstein distance W_{xa} between $\mathcal{N}(\mu,\Sigma)$ and $\mathcal{N}(\mu_a,\Sigma_a)$ simplifies to:

$$W_{xa} = (\| \mu - \mu_a \|_2^2 + \| \Sigma^{1/2} - \Sigma_a^{1/2} \|_{Fro}^2)^{1/2}, \tag{3}$$

where $\| \, \|_{Fro}^2$ denotes Frobenius norm, and the loss L_2 for matching the latent distributions of image features and attributes is:

$$L_2 = W_{xa} + W_{ax}. \tag{4}$$

Cross-reconstruction is achieved by decoding the latent representations derived from another VAE's encoder. The loss L_3 of cross-reconstruction is:

$$L_3 = L_{re}(x,D_{img}(E_{att}(a))) + L_{re}(a,D_{att}(E_{img}(x))). \tag{5}$$

Specifically, we use some variants of triple loss to optimize the latent space. In addition, we explore four variants of the triple loss to improve the performance. Triplet loss can be represented as:

$$L_{tri} = \sum_{x_t,x_p,x_n} [m + d(x_t,x_p) - d(x_t,x_n)]_+, \tag{6}$$

This loss makes sure that, given an anchor point x_t, a positive point x_p belonging to the same class as x_t is closer to the anchor than that of a negative point x_n belonging to another class, by at least a margin m [5]. And we call x_t and x_p as positive pair, x_t and x_n as negtive pair. For an anchor, the distances of its postive pairs merely need to be smaller to any distances of its negative pairs. As proposed in [5], we feed the framework with mini-batches. We form training batches by randomly sampling P classes, and then randomly sampling K images per class, thus resulting in a batch of $P \times K$ images. The anchor point can be any one in the training batch. Then we select positive and negative in the training batch for each anchor point. In this work, we set $m = 0$.

We propose four methods based on four variants of triple loss for reconstructed image features, which are different in triplet selection: (1) All possible triplets. (2) Hardest negative for each positive pair. (3) Random-hard negative for each positive pair. (4) Semi-hard negative for each positive pair.

All possible triplets. That is to simply use all possible triplets with positive triplet loss values. The loss L_{BA} is:

$$L_{BA} = \sum_{i=1}^{P}\sum_{t=1}^{K}\sum_{\substack{p=1 \\ p \neq t}}^{K}\sum_{\substack{j=1 \\ j \neq i}}^{P}\sum_{n=1}^{K}[\, d_{j,t,n}^{i,t,n} \,]_+, \tag{7}$$

$$d_{j,t,n}^{i,t,n} = d(\tilde{x}_t^i,\tilde{x}_p^i) - d(\tilde{x}_t^i,\tilde{x}_n^j), \tag{8}$$

where $\tilde{x}_j^i = D_{img}(E_{img}(x_j^i))$ denotes the reconstructed image features of the j-th image feature of the i-th class in the batch. The $d(x_1, x_2) = ||x_1 - x_2||_2^2$ denotes Euclidean distance between x_1 and x_2.

Hardest negative for each positive pair. This is to select the negative with the maximal positive triplet loss value for each positive pair. The loss L_{BH} is:

$$L_{BH} = \sum_{i=1}^{P} \sum_{t=1}^{K} \sum_{\substack{p=1 \\ p \neq t}}^{K} [\ d_{t,p}\ ]_+, \tag{9}$$

$$d_{t,p} = max([d_{j,t,n}^{i,t,n}|\ j = 1, ..., P; n = 1, ..., K; j \neq i]). \tag{10}$$

Random-hard negative for each positive pair. This is to randomly select the negative with positive triplet loss value for each positive pair. The loss L_{RH} is similar to L_{BH}, different in $\mathrm{d}_{t,p}$:

$$d_{t,p} = rand([d_{j,t,n}^{i,t,n}|j = 1, ..., P; n = 1, ..., K; j \neq i]_+). \tag{11}$$

Semi-hard negative for each positive pair. This is to randomly select the negative with positive triplet loss value greater than 0 but less than a threshold T, for each positive pair. The loss L_{SH} is similar to L_{BH}, different in $d_{t,p}$:

$$\begin{aligned} d_{t,p} = rand[d_{j,t,n}^{i,t,n}|0 < d_{j,t,n}^{i,t,n} < T; j = 1, .., P; \\ n = 1, .., K; j \neq i], \end{aligned} \tag{12}$$

$$T = (\sum_{i=1}^{P} \sum_{t=1}^{K} \sum_{\substack{p=1 \\ p \neq t}}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{P} \sum_{n=1}^{K} [\ \text{-}\ d_{j,t,n}^{i,t,n}\ ]_+)/(P \times K). \tag{13}$$

We express the different variants of triple loss as L_4. Thus, the total loss of our final framework is formulated as:

$$L_{all} = L_1 + \alpha(L_2 + L_4) + \lambda(L_3 + L_4) + L_4, \tag{14}$$

where α and λ are weight coefficients.

3. Experimental Settings and Datasets

In this work, two VAEs are trained for image features and attributes, respectively. All encoders and decoders are multilayer perceptrons (MLPs) containing a hidden layer with ReLU activation. The hidden units of our image feature encoder, image feature decoder, attribute encoder and attribute decoder are 1560, 1660, 1450 and 660, respectively. We randomly select a mini-batch of P classes and K images per class for training. We set the dimension of the latent space to 64, P = 10 and K = 5. We increase α from epoch 6 to epoch 22 by a rate of 0.54 per epoch and increase λ from epoch 21 to 75 by 0.044 per epoch. The weight of KL-divergence β is increased from epoch 0 to epoch 90 by a rate of 0.0026 per epoch. The L1 norm is used as reconstruction loss.

We evaluate our methods on four benchmark datasets including Animals with Attributes 1 (AWA1 [11]), Animals with Attributes 2 (AWA2 [12]), Caltech-UCSD Birds-200-2011 (CUB) [13] and SUN Attribute (SUN) [14]. The evaluation protocol and the splits are as the same set in [12]. Also, we use the 2048-D image features provided by [12] for all datasets. The performance of ZSL is measured by the accuracy on unseen classes. Otherwise, the performance of GZSL is measured by the harmonic mean H = 2 × S × U/(S + U), where S and U are the accuracy on seen and unseen classes, respectively.

4. Results and Analysis

As shown in Table 1, compared to the state-of-the-art GZSL methods, our methods achieve great improvement on four benchmark datasets. And we explore the imapct of different variants of triple loss on the performance of our mothods by ablation analysis. It should be noted that AVAE represents aligned VAE. BA-AVAE, BH-AVAE, RH-AVAE and SH-AVAE represent the AVAE with all possible triplets, with hardness negative, with random-hard negative and with semi-hard negative, respectively. Compared with other GAN based methods [9], [10], our VAE based methods are relatively efficient to be trained and the performance is close to them. It can be seen that our methods with variants of triple loss are more competitive in GZSL. Our methods achieve the best S of 77.3% and 82.2% on the AWA1 and AWA2 datasets, which exceed other methods by 1.0% and 7.1%, and best U of 48.2% on the SUN dataset, which exceeds other methods by 1.0%. And our methods obtain the second best H of 65.0% and 65.5% on the AWA1 and AWA2 datasets, respectively.

In ablation analysis, we remove the variants of triple loss, and the method without a variant of triple loss is called AVAE. As shown in Table 1, the performance of our methods with variants of triple loss is generally better than AVAE in GZSL scenario. On the CUB and SUN datasets, the performance of all our methods with variants of triple loss is better than AVAE at most 1.5%. In SH-AVAE, the results on all datasets are better than AVAE.

We also report the results of our models in the conventional ZSL scenario in Table 2. In the conventional ZSL scenario, the performance of our methods with variants of triple loss is still more competitive than AVAE. The best results and second best results on all datasets are obtained by methods with variants of triple loss. Specifically, the results of BA-AVAE on all datasets are better than AVAE, the results on the AWA1 and AWA2 datasets are better than AVAE by 2% and 3.2%, respectively. This manifests that variants of triple loss are also applicable to the conventional ZSL. In addition, we can see that different variants of triple losses have different degrees of impact on the performance of the methods. Therefore, selecting proper triplets is the key for better performance.

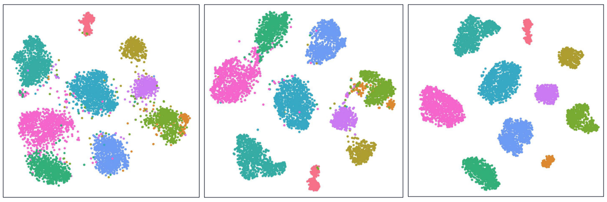
Figure 2 shows the t-SNE visualization of original and reconstructed image features for the AWA2 dataset. In Fig. 2, (a) shows the distribution of original image features, (b) shows the distribution of the reconstructed image fea-

Table 1 Results on GZSL. The best results and the second best results are respectively marked in bold and underlined.

Method	AWA1			AWA2			CUB			SUN		
	U	S	H	U	S	H	U	S	H	U	S	H
CVAE [4]	-	-	47.2	-	-	51.2	-	-	34.5	-	-	26.7
f-VAEGAN-D2 [6]	-	-	-	57.6	70.6	63.5	48.4	60.1	53.6	45.1	38.0	41.3
LisGAN [7]	52.6	76.3	62.3	-	-	-	46.5	57.9	51.6	42.9	37.8	40.2
CADA-VAE [8]	57.3	72.8	64.1	55.8	75.0	63.9	51.6	53.5	52.4	47.2	35.7	40.6
RFF-GZSL [9]	**59.8**	75.1	**66.5**	-	-	-	52.6	56.6	54.6	45.7	38.6	41.9
TF-VAEGAN [10]	-	-	-	**59.8**	75.1	**66.6**	**52.8**	**64.7**	**58.1**	45.6	**40.7**	**43.0**
Ours(AVAE)	55.6	74.6	63.7	53.7	77.1	63.3	52.4	52.6	52.5	45.6	36.5	40.5
Ours(**BA-AVAE**)	57.4	73.0	65.0	51.3	78.9	62.2	49.2	57.4	53.0	**48.2**	35.3	40.7
Ours(**BH-AVAE**)	53.3	**77.3**	63.1	52.8	77.1	62.7	49.7	57.8	53.4	46.3	36.7	41.0
Ours(**RH-AVAE**)	56.5	75.6	64.7	51.8	79.0	62.6	50.0	58.2	53.8	47.6	36.5	41.3
Ours(**SH-AVAE**)	56.9	73.8	64.3	54.5	**82.2**	65.5	50.8	57.7	54.0	47.4	36.3	41.1

Table 2 Results on conventional ZSL. The best results and the second best results are respectively marked in bold and underlined.

Method	AWA1	AWA2	CUB	SUN
AVAE	65.7	63.0	60.3	61.8
BA-AVAE	**67.7**	**66.2**	60.5	**62.4**
BH-AVAE	64.4	63.1	60.8	62.2
RH-AVAE	66.0	62.3	**61.1**	61.7
SH-AVAE	65.8	66.0	60.6	62.0



(*a*) *original image features* (*b*) *AVAE reconstructed image features* (*c*) *SH* − *AVAE reconstructed image features*

Fig. 2 t-SNE visualization for the AwA2 dataset.

tures in AVAE, and (c) shows the reconstructed image features in SH-AVAE. Compared to AVAE, the reconstructed image features of SH-AVAE are similar to the original image features, and almost all reconstructed image features of same class are closer than reconstructed image features of different classes. This shows that our methods with variants of triple loss are able to capture the discriminative information to a good extent, and greatly optimize the latent space.

5. Conclusion

In this work, we learn two VAEs for image features and attributes, respectively. For preserving discriminative information in latent space, we align the corresponding latent distributions of image features and attributes by minimizing the 2-Wasserstein distance and cross-reconstruction loss. And we use some variants of triple loss to optimize the latent space. Then, we train a softmax classifier in the latent space. The proposed methods achieve great improvement on four benchmark datasets. In addition, we explore the impact of different variants of triple loss. How to better apply the triple loss to ZSL and GZSL is worthy of study.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China under Grant No.61906099, No.61906098.

References

[1] D.P. Kingma and M. Welling, "Auto-encoding variational bayes," Proc. ICLR, pp.1–14, 2014.

[2] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, and Y. Bengio, "Generative adversarial networks," CoRR, abs/1406.2661, 2014.

[3] Y.H.H. Tsai, L.K. Huang, and R. Salakhutdinov, "Learning robust visual-semantic embeddings," Proc. ICCV, pp.3591–3600, 2017.

[4] A. Mishra, S.K. Reddy, A. Mittal, and H.A. Murthy, "A generative model for zero shot learning using conditional variational autoencoders," Proc. CVPR, pp.2188–2196, 2018.

[5] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," CoRR, abs/1703.07737, 2017.

[6] Y. Xian, S. Sharma, B. Schiele, and Z. Akata, "F-vaegan-d2: A feature generating framework for any-shot learning," Proc. CVPR, pp.10275–10284, 2019.

[7] J. Li, M. Jin, K. Lu, Z. Ding, L. Zhu, and Z. Huang, "Leveraging the invariant side of generative zero-shot learning," Proc. CVPR, pp.7402–7411, 2019.

[8] E. Schönfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero- and few-shot learning via aligned variational autoencoders," Proc. CVPR, pp.8247–8255, 2019.

[9] Z. Han, Z. Fu, and J. Yang, "Learning the redundancy-free features for generalized zero-shot object recognition," Proc. CVPR, pp.12862–12871, 2020.

[10] S. Narayan, A. Gupta, F.S. Khan, C.G.M. Snoek, and S. Ling, "Latent embedding feedback and discriminative features for zero-shot classification," Proc. ECCV, pp.479–495, 2020.

[11] C.H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," Proc. CVPR, pp.951–958, 2009.

[12] Y. Xian, C.H. Lampert, B. Schiele, and Z. Akata, "Zero-shot learning–a comprehensive evaluation of the good, the bad and the ugly," IEEE Trans. Pattern Anal. Mach. Intell., vol.41, no.9, pp.2251–2265, Sept. 2019.

[13] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[14] G. Patterson and J. Hays, "Sun attribute database: Discovering, annotating, and recognizing scene attributes," Proc. CVPR, pp.2751–2758, 2012.