Data Science Project Three Report

Hang Zheng 520021911347

May 21, 2023

Contents

1 Introduction

In this project, I mainly utilize three different zero-shot classification methods (semantic relatedness, semantic embedding, and synthetic method) to perform zero-shot image classification tasks based on the manually annotated semantic information of the categories.

The dataset used in this experiment is the Animals with Attributes (AwA2) dataset, which contains 37322 images of 2048 dim pre-extracted deep learning features and manually annotated semantic information for fifty different animal categories. In the experiment, forty categories of samples were used as the training set, while the remaining ten categories of samples (which were not visible during the training process) were used as the test set.

Furthermore, I do experiment on the parameters of each method trying to find out the optimal hyper parameters for each method to reach the optimal performance and record the best classification accuracy.

2 Experimental Principle

In this section, I will give a brief introduction to the principle of the three zero-shot classification methods I mentioned above.

2.1 Semantic relatedness

In traditional zero-shot classification, semantic relatedness methods are commonly used. This method is based on the semantic relationship between categories to implement the image classification task. It assumes that the semantic relatedness between categories can be used to infer the attributes and features of each category, thereby achieving zero-shot classification.

The implementation of this method usually involves two steps.

1. First, it is necessary to obtain the **semantic relatedness scores** between each category, which can be understood as the degree of similarity or correlation between categories.

2. Second, these semantic relatedness scores need to be combined with image features to achieve zero-shot classification.

In the implementation of this project, the semantic information of the categories (including binary and continuous) is already included in the dataset. I calculated the similarity matrix between categories using various methods(section 2.1.1) based on this semantic information, and trained a classifier (SVM or KNN) on the 40 categories of samples in the training set. The formula for calculating the prediction score of an unknown sample is defined as follows:

$$f_{c_s+k}(\cdot) = \sum_{i=1}^{c_s} (s_{c_s+k,i}) f_i(\cdot)$$

where $\{f_1(\cdot), \cdots, f_{c_s}(\cdot)\}$ are the prediction score functions of the classifier for the c_s known categories, $s_{i,j}$ is the i, j element of the similarity matrix S, representing the semantic similarity between the i-th and j-th categories.

This classification method is intuitive, that is, when predicting unknown category samples, the prediction score function of the known categories with closer semantic similarity will have a higher proportion in the prediction, indicating that the prediction score function of similar categories is more trustworthy.

The principle of the algorithm is shown in Fig 1.

2.1.1 Similarity matrix

When calculating the similarity matrix of semantic information between categories, there are many commonly used methods. In this project, I implemented the following five methods to calculate the similarity between sample points:

1. Cosine distance: $d_{cos}(u, v) = \frac{u \cdot v}{||u||_2 \cdot ||v||_2}$

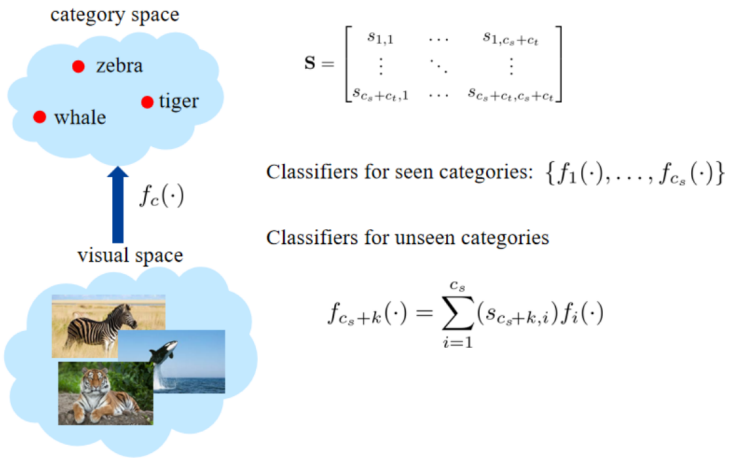


Figure 1: Semantic relatedness method

2. Correlation: $d_{correlation}(u,v) = 1 - \frac{(u-\bar{u})\cdot(v-\bar{v})}{||(u-\bar{u})||_2||(v-\bar{v})||_2}$
3. Euclidean distance: $d_{euclidean}(u,v) = |u-v|^2 = \sqrt{\sum_{k=1}^{n}(x_{ik}-x_{jk})^2}$
4. Manhattan distance: $d_{manhattan}(u,v) = |u-v| = \sum_{k=1}^{n}|x_{ik}-x_{jk}|$
5. Chebyshev distance: $d_{chebyshev}(u,v) = \max_{k=1}^{n}|x_{ik}-x_{jk}|$

When using Euclidean, Cityblock, and Chebyshev distances, a smaller distance indicates a closer semantic similarity between categories. However, when using these distances, we consider that categories with higher similarity values are closer to each other. Therefore, a conversion from distance to similarity is required.

I used the Gaussian kernel function, that is,

$$\text{similarity}(c_i, c_j) = \exp\left(-\frac{d(c_i,c_j)}{2\sigma^2}\right)$$

to perform the conversion. Here, $d(c_i, c_j)$ is the distance between the semantic information of category i and category j, σ is a positive real number, called the bandwidth parameter of the Gaussian kernel function. The bandwidth parameter controls the width of the similarity function, that is, it determines that the similarity between samples that are farther apart will be smaller.

2.2 Semantic embedding

In traditional zero-shot classification, semantic embedding methods are also a common approach. This method embeds image features into a low-dimensional vector with the same dimensionality as the class semantic information and learns the mapping from image features to semantic information by optimizing the embedding model to minimize the distance between the embedding vector and the semantic information vector. During zero-shot classification, the features of unknown category samples are mapped to low-dimensional space, and the closest category is selected as the prediction by comparing with the semantic information vector of unknown categories.

The implementation of this method is usually divided into two steps.

1. First, an embedding model needs to be trained using the features and semantic information vectors of visible categories in the training set;
2. Second, the features of unseen samples in the test set are embedded, and the category corresponding to the closest semantic information vector is selected as the predicted category.

In this project, I chose a simple linear layer model as the embedding model, and five different distance metrics(as mentioned in section 2.1.1) were used for distance calculation during prediction.

The principle of the algorithm is shown in Fig 2.

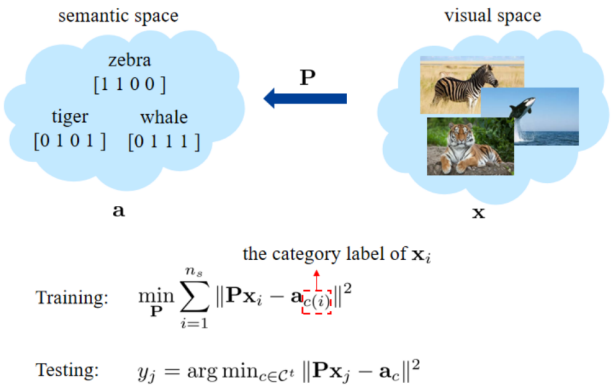


Figure 2: Semantic embedding method

2.3 Synthetic method

The synthetic method in Zero-shot Classification task generates synthetic training examples for unseen categories by combining semantic information with existing samples from seen categories. It relies on the fact that images from different categories **share some common visual patterns**. Given a set of seen categories and their corresponding image features, a semantic embedding model is trained to **map the image features to a semantic space**. Then, for each unseen category, its semantic information is provided, which is typically in the form of a vector of attribute values. The semantic description is also mapped to the same semantic space as the seen categories.

Once the seen and unseen categories are mapped to the same semantic space, synthetic samples for the unseen categories can be generated by combining the semantic description of the unseen category with the image features of the seen categories.

After generating synthetic samples, a classifier is trained on the synthetic data to classify the unseen categories. The classifier can be any traditional supervised learning model, such as SVM, logistic regression, or deep neural networks.

In this project, we implemented the synthetic method by employing the principles of generative adversarial networks (GANs). We trained a generator and a discriminator for this task. The generator takes the existing attribute features (semantic information) as input and generates a 2048-dimensional visual feature vector (training sample). The discriminator, on the other hand, takes both the attribute features and the visual features as input and determines whether they are real or synthetic.

The principle of the algorithm is shown in Fig 3.

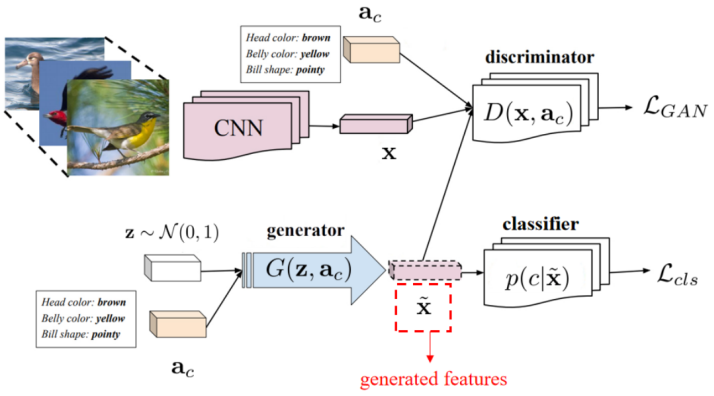


Figure 3: Synthetic method

class preservation loss It is worth noting that we added a class preservation term to the loss function when training the generator of the GAN, which encourages the generator to produce images

consistent with the given class label. The expression of the class preservation term is as follows:

$$L_{cls}(G) = -E_{z,y}[y \log(D(G(z))) + (1-y) \log(1 - D(G(z)))]$$

Here, $D(\cdot)$ is the discriminator model, which outputs a real number between 0 and 1, indicating the probability that the input image is a real image. y is the given class label, where $y = 1$ indicates that the real image belongs to the given class, and $y = 0$ indicates that it does not belong to the given class. $G(z)$ is the synthetic image generated by the generator. $E_{z,y}$ denotes the expectation over both the noise and the class label.

$L_{cls}(G)$ represents the loss of the class preservation term, which aims to encourage the generator to produce images consistent with the given label.

- When the generator outputs an image consistent with the given label, the value of $D(G(z))$ should be close to 1, and $L_{cls}(G)$ should be small.
- When the generator outputs an image inconsistent with the given label, the value of $D(G(z))$ should be close to 0, and $L_{cls}(G)$ should be large. Therefore, the class preservation term can guide the generator to learn how to generate images consistent with the given label.

3 Experimental Result

3.1 Semantic relatedness

3.1.1 Experiment on σ in similarity matrix calculation

As mentioned in section 2.1.1, when calculating the similarity matrix using distance method, a conversion from distance to similarity is required. In the Gaussian kernel function, there's a hyper parameter σ called bandwidth parameter which controls the width of the similarity function. The suitable value of σ varies with the semantic information data and so in this experiment, I first conducted an experiment on the value of σ.

In this experiment, semantic information can be either binary or continuous and can be subject to normalization or not, resulting in four possible compositions of semantic information: binary, binary-normed, continuous, and continuous-normed. The classifier used is SVM.

The result is as shown in Table 1, Table 2 and Table 3.

σ	binary	binary-normed	continuous	continuous-normed
1	3.46	6.57	**17.95**	3.08
5	38.72	21.29	6.57	28.66
10	**44.86**	31.80	2.97	28.84
15	40.95	**43.13**	2.97	32.67
20	40.01	42.41	2.97	41.41
25	39.64	42.82	2.97	46.25
30	39.40	41.63	2.97	47.59
35	39.23	39.82	2.97	48.15
40	39.13	38.56	2.97	**48.36**
45	39.05	37.29	2.97	48.30
50	38.99	36.18	2.97	48.25

Table 1: Experiment on σ when using Euclidean distance

From the result, we can see that:

1. Best setting for the three distance metric:
   - Euclidean: $\sigma = 40$, with normalized continuous semantic information;
   - Cityblock: $\sigma = 45$, with un-normalized binary semantic information;
   - Chebyshev: $\sigma = 5$, with normalized continuous semantic information.

σ	binary	binary-normed	continuous	continuous-normed
1	2.97	6.57	**17.95**	2.91
5	2.97	2.98	17.95	2.91
10	20.62	7.06	17.95	2.91
15	27.75	7.28	17.95	2.91
20	31.04	11.53	17.95	2.94
25	29.71	24.90	6.57	17.65
30	30.66	26.92	6.57	26.25
35	34.25	27.21	6.57	29.22
40	37.22	27.56	6.57	32.09
45	**37.61**	29.69	2.97	**34.26**
50	37.57	**30.56**	2.97	31.43

Table 2: Experiment on σ when using Cityblock distance

σ	binary	binary-normed	continuous	continuous-normed
1	17.95	17.95	**17.95**	25.34
5	**17.96**	**22.20**	9.74	**27.98**
10	17.95	20.21	10.15	27.50
15	12.73	19.13	10.24	20.09
20	17.95	18.59	11.82	13.85
25	17.95	18.41	13.76	10.65
30	12.64	18.21	12.88	9.25
35	17.95	18.06	8.58	8.62
40	17.95	18.00	5.19	8.43
45	17.96	18.00	4.12	8.30
50	17.95	18.00	3.97	8.20

Table 3: Experiment on σ when using Chebyshev distance

2. Continuous semantic information is usually better than the binary one, for that continuous semantic carries more information.

3. Normalization is of great use when using the continuous semantic information, but brings no obvious improvement when using the binary semantic information.

3.1.2 Experiment on different distance metrics

Using the result in the last section 3.1.1, We conducted experiments for each distance metric, with the parameters of each distance metric set to its optimal value (refer to Section 3.1.1), and used two different classifiers, SVM and KNN, to conduct experiments.

The experimental results are shown in Table 4 and Table 5.

Distance metric	binary	binary-normed	continuous	continuous-normed
Cosine	34.50	**43.13**	33.01	44.07
Correlation	34.77	40.71	**36.65**	44.33
Euclidean	**44.86**	37.55	17.95	**48.36**
Cityblock	37.61	30.56	17.95	34.26
Chebyshev	17.96	22.20	17.95	27.98

Table 4: Experiment on different distance metrics using SVM

Distance metric	binary	binary-normed	continuous	continuous-normed
Cosine	**38.25**	33.15	**39.30**	36.81
Correlation	32.04	**33.20**	35.85	35.71
Euclidean	30.32	33.00	17.95	**37.17**
Cityblock	27.06	30.23	17.95	29.52
Chebyshev	17.95	23.90	17.95	27.90

Table 5: Experiment on different distance metrics using KNN

From the result, we can see that:

1. The best performance using Semantic Relatedness method with SVM is 48.36%, achieved using normalized continuous semantic information with Euclidean distance metric.

2. The best performance using Semantic Relatedness method with KNN is 39.30%, achieved using non-normalized continuous semantic information with Cosine distance metric.

3. Cosine, Correlation and Euclidean distance are more suitable in this task than Cityblock and Chebyshev distance.

4. The performance of KNN is more stable compared to SVM, but at the same time, it is also more difficult to achieve high accuracy.

3.2 Semantic embedding

In this experiment, semantic information can be either binary or continuous and can be subject to normalization or not, resulting in four possible compositions of semantic information: binary, binary-normed, continuous, and continuous-normed.

When making final category predictions, it is necessary to calculate the distance between the embedding vector and the semantic vectors of each unknown category. Distance calculation is involved at this stage, and the following five distance metrics were experimentally evaluated: Cosine distance, Correlation, Euclidean distance, Manhattan distance and Chebyshev distance.

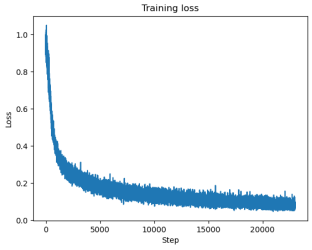
The result is as shown in Table 6.

The training loss of using different distance metric is shown in Fig 4.

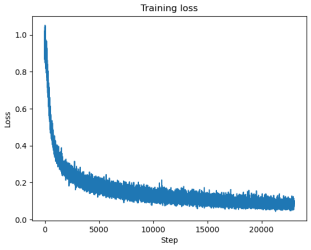
From the result, we can see that:

distance	binary	binary-normed	continuous	continuous-normed
Cosine	38.63	41.65	41.17	43.25
Correlation	42.12	39.66	44.65	43.62
Euclidean	42.55	**48.63**	48.11	**51.47**
Manhattan	**46.86**	45.85	46.51	45.87
Chebyshev	36.43	31.43	**49.26**	49.27

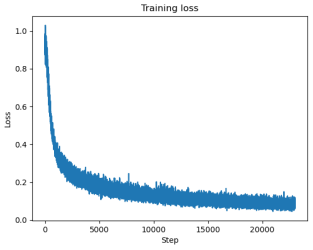
Table 6: Experiment on distance metric in Semantic embedding



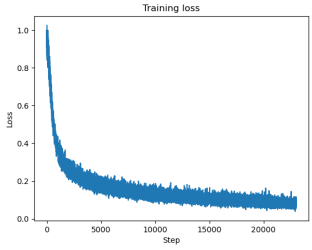
(a) Cosine



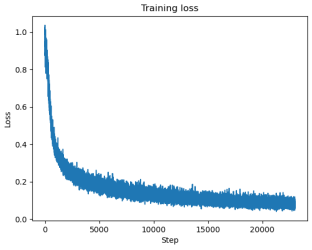
(b) Correlation



(c) Euclidean



(d) Manhattan



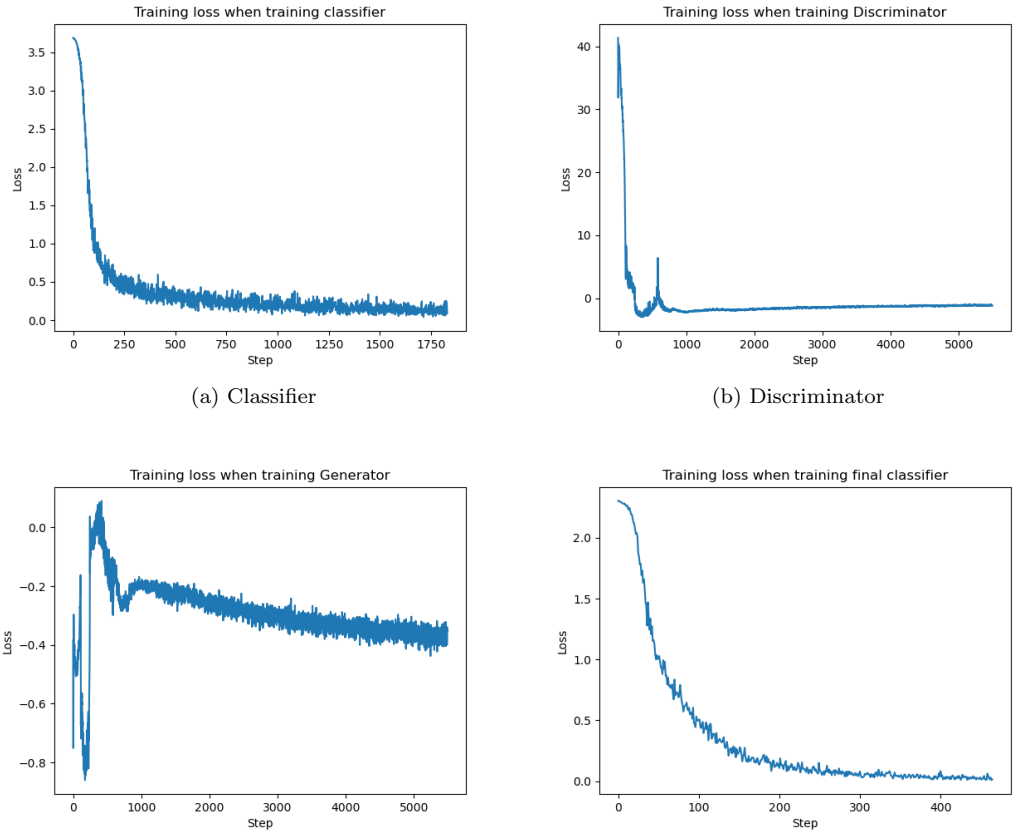
(e) Chebyshev

Figure 4: Training loss with different distance metric in Semantic embedding

1. The best performance using Semantic Embedding method is 51.47%, achieved using normalized continuous semantic information with Euclidean distance metric.
2. Distance metric Euclidean, Manhattan and Chebyshev are more useful in Semantic Embedding method than Cosine distance and Correlation.
3. Continuous semantic information is more useful than binary semantic information here.
4. Normalization is of use and could bring improvement in performance.

3.3 Synthetic method

In this experiment, we conducted the synthetic method on zero-shot classification task. We set the latent dimentionality as 64 and train the model with and without cls loss, where cls loss is a loss function term used to encourage the model to maintain category consistency in the generated features as much as possible. The training loss of the GAN models are shown in Fig 5 and Fig 6.



(a) Classifier

(b) Discriminator

(c) Generator

(d) Final classifier

Figure 5: Training loss of the GAN models using cls loss.

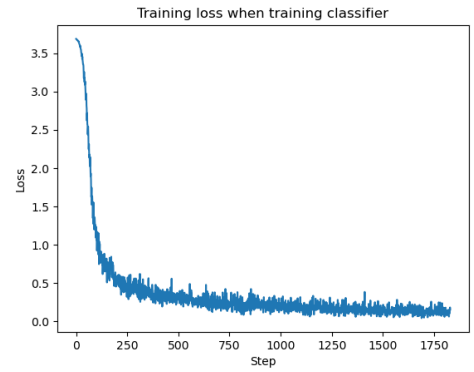
The final result is shown in Table 7.

with cls loss	without cls loss
63.102	17.085

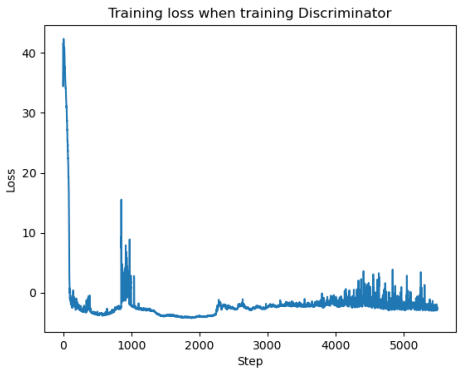
Table 7: Accuracy using synthetic method

From the result, we can see that:

1. The best performance using Synthetic Method is 63.102%, achieved with the use of cls loss.

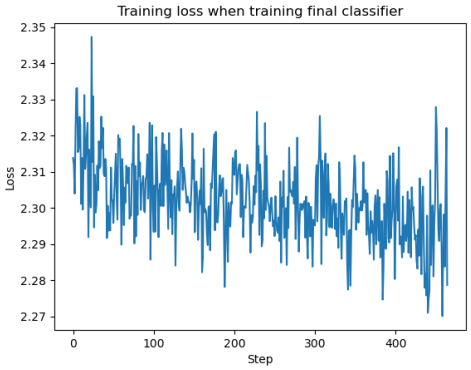


(a) Classifier



(b) Discriminator

(c) Generator



(d) Final classifier

Figure 6: Training loss of the GAN models without using cls loss.

2. Without the cls loss term, the accuracy of the model is significantly reduced and from Fig 6 we can also see that the training process of the Discriminator and Generator are completely out of control. We can infer that without the cls loss term, the generated samples will go much far off the mark.

4 Conclusion

In this paper, we introduced three basic zero-shot classification methods. The best performance comes from Synthetic method using cls loss, achieving 63.102% on the test set.