Fine-grained Categorization using Attributes

Zhenyang Li, Ran Tao, Silvia Pintea

March 28, 2014

1 Introduction

In this practical assignment, we differentiate between:

- **Analyze:** Just do it, no report required.
- **Assignment:** Experiment and report about it. We expect a report in PDF format and seek for condensed answers.

Download the code from `http://staff.science.uva.nl/~zli2/a25/attributes_handout.zip`.

1. Read the exercises carefully (it really helps).
2. Form teams of two students. It is preferred when for all pairs one student has some experience with Matlab.
3. We will use the same guest account to log into the machines, which will give you the same network disc. In order to avoid chaos, please save your files in the local disc (C: disc) instead of the network disc.

2 Fine-grained Categorization

Figure 1: The Caltech-UCSD Birds 200-2011 dataset.

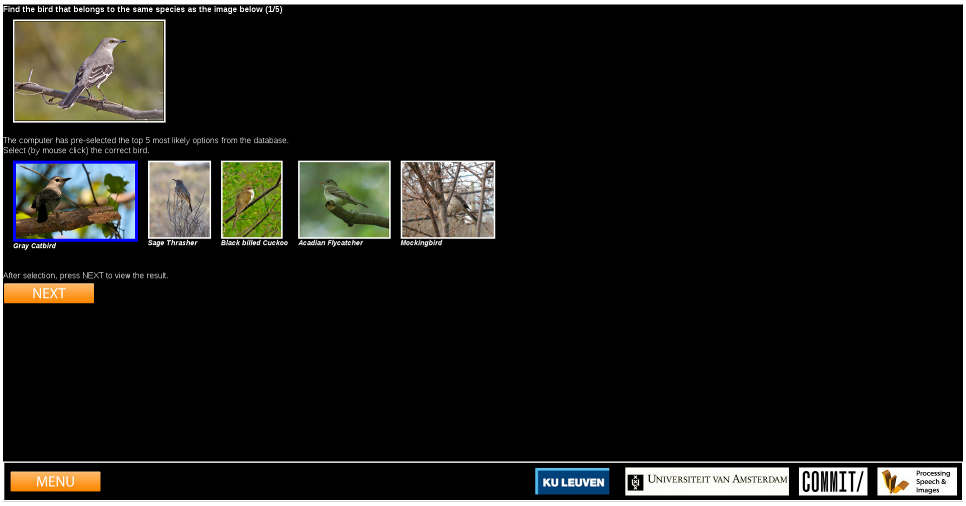
This lab focuses on fine-grained categorization, thus, the used dataset is the Caltech-UCSD Birds 200-2011 dataset (`http://www.vision.caltech.edu/visipedia/CUB-200-2011.html`).

- It contains 200 bird species (mostly North American).
- Total number of images: 11,788.
- Annotations per image: 15 Part Locations, 312 Binary Attributes, 1 Bounding Box.

You can also browse some of the images and annotations from this dataset using the link: `http://www.vision.caltech.edu/visipedia-data/CUB-200-2011/browse/index.html`.

This first task is to analyze the dataset and understand why fine-grained categorization is a challenging problem. Try with your team-mate to see if your skills at distinguishing birds can exceed the performance of the supervised model proposed in [2].

Game link: `http://homes.esat.kuleuven.be/~rompay/BirdCategorization/EN/`.



Find the bird that belongs to the same species as the image below (1/5)

The computer has pre-selected the top 5 most likely options from the database.
Select (by mouse click) the correct bird.

After selection, press NEXT to view the result.

Figure 2: Automatic Recognition of Bird Species.

Analyze:

1. Is your performance better than the one of the trained system?
2. What cues do you use when recognizing a bird type?
3. What caused confusion in the cases in which you have mistakenly guessed?

3 Attribute Label Embedding

From the previous task most of you have probably reached the conclusion that attributes (characteristics) of the birds are really helpful when trying to differentiate very similar categories.

The code you will use in this lab employs Attribute Label Embedding (ALE) [1] to solve the fine grained categorization problem. Figure 3 illustrates this model.

Let $\theta : \mathcal{X} \to \widetilde{\mathcal{X}}$ be the image embedding (mathematical space in which the images reside) $\varphi : \mathcal{Y} \to \widetilde{\mathcal{Y}}$ be the embedding of class labels. We define a compatibility function from the image and labels space to real numbers, $F(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, which measures how compatible image x and class label y are. This function operates in the attribute space and is defined as follows:

$$F(x, y) = \theta(x)' W \varphi(y),$$

where W is the feature-to-attribute mapping matrix (linear classifiers), that we want to learn.

Given an input image x, the prediction function f is the defined as the maximum over labels of the compatibility function:

$$f(x) = \arg\max_{y \in \mathcal{Y}} F(x, y).$$

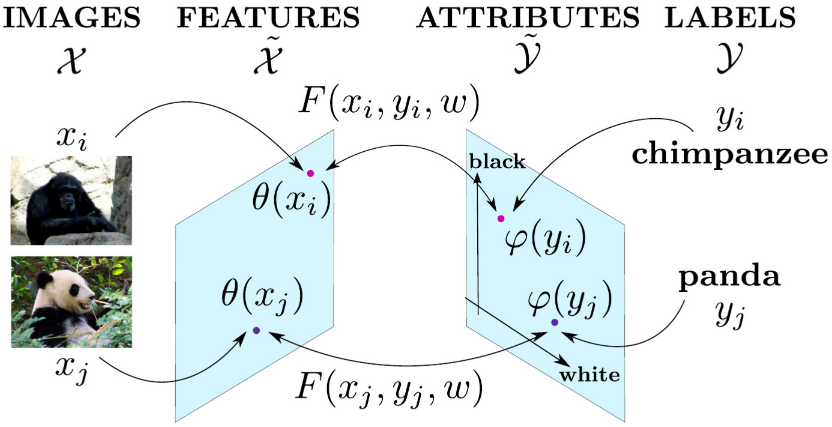


Figure 3: The Attribute Label Embedding model.

The provided code uses state-of-the-art Fisher vectors [3] for defining the image embedding, θ. We now consider the problem of computing label embeddings φ from attributes. Assuming that we have C classes $\mathcal{Y} = \{1, \ldots, C\}$ and a set of E attributes $\mathcal{A} = \{a_i, i = 1 \ldots E\}$, we embed each class y in the E-dimensional attribute space as follows:

$$\varphi(y) = [\rho_{y,1}, \ldots, \rho_{y,E}] \, ,$$

where $\rho_{y,i}, i = 1 \ldots E$ indicates an association between each class y and each attribute a_i. These associations can be binary or real-valued. We stack the individual $\varphi(y)$ into a $C \times E$ matrix of attribute embeddings V for all the classes.

We consider in the assignments two separate cases: fully supervised setting (where we have training examples for all the classes) and the zero-shot learning (where some classes lack training example).

- In the fully supervised setting, where visual examples for each class are provided, next to learning W we also want to learn the attribute embeddings V.
- In the case of zero-shot learning, where we have no training examples for some of the classes (unknown classes), we learn W on classes having training examples (known classes) but use a fixed, human annotated mapping V_0.

4 Supervised Learning

Given a set of training examples $S = \{(x_i, y_i), n = 1 \ldots N\}$, the goal is to learn the W and attribute embedding V for the compatibility function F using the following loss function which is similar to Structured SVM:

$$\ell(x_i, y_i, y) = \Delta(y_i, y) + F(x_i, y) - F(x_i, y_i)$$

where the Δ function represent the loss associated with prediction y when the true label is y_i, i.e. equal to 0 when $y = y_i$, otherwise equal to 1. The objective function is defined over W and V and this needs to be minimized:

$$R(W, V) = \frac{1}{N} \sum_{i=1}^{N} \max_{y \in \mathcal{Y}} \ell(x_i, y_i, y)$$

Finally, the quantity to be minimized adds a constraint over W (in green) thus keeping it bounded and another constraint over V (in blue) which enforces that the learned attribute embedding should be as close as possible to the prior information V_0:

$$\min_{W,V} \frac{\lambda}{2}||W||^2 + \frac{\mu}{2}||V - V_0||^2 + R(W, V)$$

This minimization is performed by the function ale_sgd.m using stochastic gradient descent (SGD).

We select a subset of the Caltech-UCSD Birds 200-2011 dataset for the following assignments:

- 50 classes in total, the list of selected classes is contained in the file imageset/classes.txt, with each line corresponding to one class: <class_id> <class_name>.
- For each class, we have 20 images for training and 5 images for testing, totally 1250 images. The list of training images and test images is contained in file imageset/train.txt and imageset/test.txt, with each line corresponding to one image: <image_name> <image_label>. All the images are located in images/<image_name>.jpg.
- For feature embeddings, we have pre-computed Fisher vectors (2560 dimensional using GMM codebook size of 16) for each image, which are located in features/<image_name>.mat.
- The list of 312 attributes is contained in the file attributes/attributes.txt, with each line corresponding to one attribute: <attribute_id> <attribute_name>. The prior binary attribute embedding matrix V_0 for the 50 classes is stored in the file attributes/attributes_binary.mat.

Assignment:

1.1 Read and run the scripts cub_experiment.m to train the model parameters: W, V, and apply the learned model on the test images. Evaluate the accuracy.

1.2 Investigate which bird categories are more likely to be misclassified (e.g. compute confusion matrix), and visualize a few bird images from these categories.

1.3 Analyze why these categories are most often confused and given a few suggestions as to how these issues could be overcome.

5 Zero-shot Learning

In the case of zero-shot learning, we cannot learn the attribute embedding V from the data due to the lack of labels for some classes. In this situation the prior information, V_0, is used instead and the quantity to be minimized only contains the regularization over the W and the objective function $R(W, V_0)$:

$$\min_{W} \frac{\lambda}{2}||W||^2 + R(W, V_0)$$

This minimization is performed by the function ale_zeroshot_sgd.m using SGD.

For zero-shot learning experiments, we split the 50 classes into 40 known classes for training and 10 unknown classes for testing:

- Known classes: the first 40 of the 50 classes, i.e. Black_Footed_Albatross, . . . , Least_Tern. The list of training images from these 40 classes is in the file imageset/zeroshot_train.txt.
- Unknown classes: the last 10 of the 50 classes, i.e. White_Eyed_Vireo, Cape_May_Warbler, . . . , Marsh_Wren. The list of testing images from these 10 classes is in the file imageset/zeroshot_test.txt.

Assignment:

2.1 Read and run the scripts cub_zeroshot_experiment.m to train the model parameters W on the known classes with fixed V_0, and apply the learned W on unknown classes. Evaluate the accuracy.

2.2 Apply the previously learned model W, V (we only use the submatrix of V that corresponding to the 10 unknown classes) with supervised learning (i.e. the model trained in Assignment 1.1 on all the 50 classes) on test images of the 10 unkown classes only. Compare the difference in accuracy with Assignment 2.1, which one is much better and why?

2.3 Since each column of W can be interpreted as an attribute classifier, $\theta(x)'W$ is then a vector of attribute scores of image x. Compute the attribute scores for each image using learned W from Assignment 2.1, and visualize bird images with their corresponding top scoring attributes:

```
>> for i = 1:numel(img_test)
>>    % compute the prediction
>>    [score, pred_label] = max(V_unknown * W' * feat_test(:, i));
>>    % the id of unknown classes start from 41
>>    pred_label = pred_label+40;

>>    % compute attributes scores and sort them
>>    attr_scores = W' * feat_test(:, i);
>>    [sorted_attr_scores, sorted_attr_ids] = sort(attr_scores, 'descend');

>>    % visualize top (e.g. 5) scoring attributes
>>    plot_cub_image(img_test{i}, label_test(i), pred_label, sorted_attr_ids(1:5));
>>    pause
>> end
```

Do the top scoring attributes correctly describe the bird image? Do you have any idea how to improve it?

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2013.

[2] Efstratios Gavves, Basura Fernando, CGM Snoek, AWM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In *Proc. IEEE Int'l. Conf. Comput. Vision*, 2013.

[3] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *Int. Journal of Computer Vision*, 2013.