Zero-shot Learning

What is zero-shot learning

- The aim of zero-shot learning is to classify instances belonging to the classes that have no labeled instances
- Examples:
    1. The number of target classes is large – manually labeling a large number of images
    2. Target classes are rare – collect sufficient images for rare breeds of flower
    3. Target classes change over time – image style changes overtime
    4. In some particular tasks, it is expensive to obtain labeled instances – image semantic segmentation problem, the images used as training data should be labeled at the pixel level

Zero-shot learning definition

In zero-shot learning, there are some labeled training instances in the feature space. The classes covered by these training instances are referred to as the *seen classes*. In the feature space, there are also some unlabeled testing instances, which belong to another set of classes. These classes are referred to as the *unseen classes*. The feature space is usually a real number space, and each instance is represented as a vector within it. Each instance is usually assumed to belong to one class.[1] Now, we give the definition of zero-shot learning. Denote $\mathcal{S} = \{c_i^s | i = 1, \ldots, N_s\}$ as the set of seen classes, where each c_i^s is a seen class. Denote $\mathcal{U} = \{c_i^u | i = 1, \ldots, N_u\}$ as the set of unseen classes, where each c_i^u is an unseen class. Note that $\mathcal{S} \cap \mathcal{U} = \emptyset$. Denote $\mathcal{X}$ as the feature space, which is D-dimensional; usually it is a real number space $\mathbb{R}^D$. Denote $D^{tr} = \{(\mathbf{x}_i^{tr}, y_i^{tr}) \in \mathcal{X} \times \mathcal{S}\}_{i=1}^{N_{tr}}$ as the set of labeled training instances belonging to seen classes; for each labeled instance $(\mathbf{x}_i^{tr}, y_i^{tr})$, $\mathbf{x}_i^{tr}$ is the instance in the feature space, and y_i^{tr} is the corresponding class label. Denote $X^{te} = \{\mathbf{x}_i^{te} \in \mathcal{X}\}_{i=1}^{N_{te}}$ as the set of testing instances, where each $\mathbf{x}_i^{te}$ is a testing instance in the feature space. Denote $Y^{te} = \{y_i^{te} \in \mathcal{U}\}_{i=1}^{N_{te}}$ as the corresponding class labels for X^{te}, which are to be predicted.

Zero-shot learning definition cont.

Definition 1.1 (Zero-Shot Learning). Given labeled training instances D^{tr} belonging to the seen classes $\mathcal{S}$, zero-shot learning aims to learn a classifier $f^{u}(\cdot) : \mathcal{X} \rightarrow \mathcal{U}$ that can classify testing instances X^{te} (i.e., to predict Y^{te}) belonging to the unseen classes $\mathcal{U}$.

Learning settings

Definition 1.2 (Class-Inductive Instance-Inductive (CIII) Setting). Only labeled training instances D^{tr} and seen class prototypes T^s are used in model learning.

Definition 1.3 (Class-Transductive Instance-Inductive (CTII) Setting). Labeled training instances D^{tr}, seen class prototypes T^s, and unseen class prototypes T^u are used in model learning.

Definition 1.4 (Class-Transductive Instance-Transductive (CTIT) Setting). Labeled training instances D^{tr}, seen class prototypes T^s, unlabeled testing instances X^{te}, and unseen class prototypes T^u are used in model learning.

Class-Inductive Instance-Inductive (CIII)

Definition 1.2 (Class-Inductive Instance-Inductive (CIII) Setting). Only labeled training instances D^{tr} and seen class prototypes T^s are used in model learning.

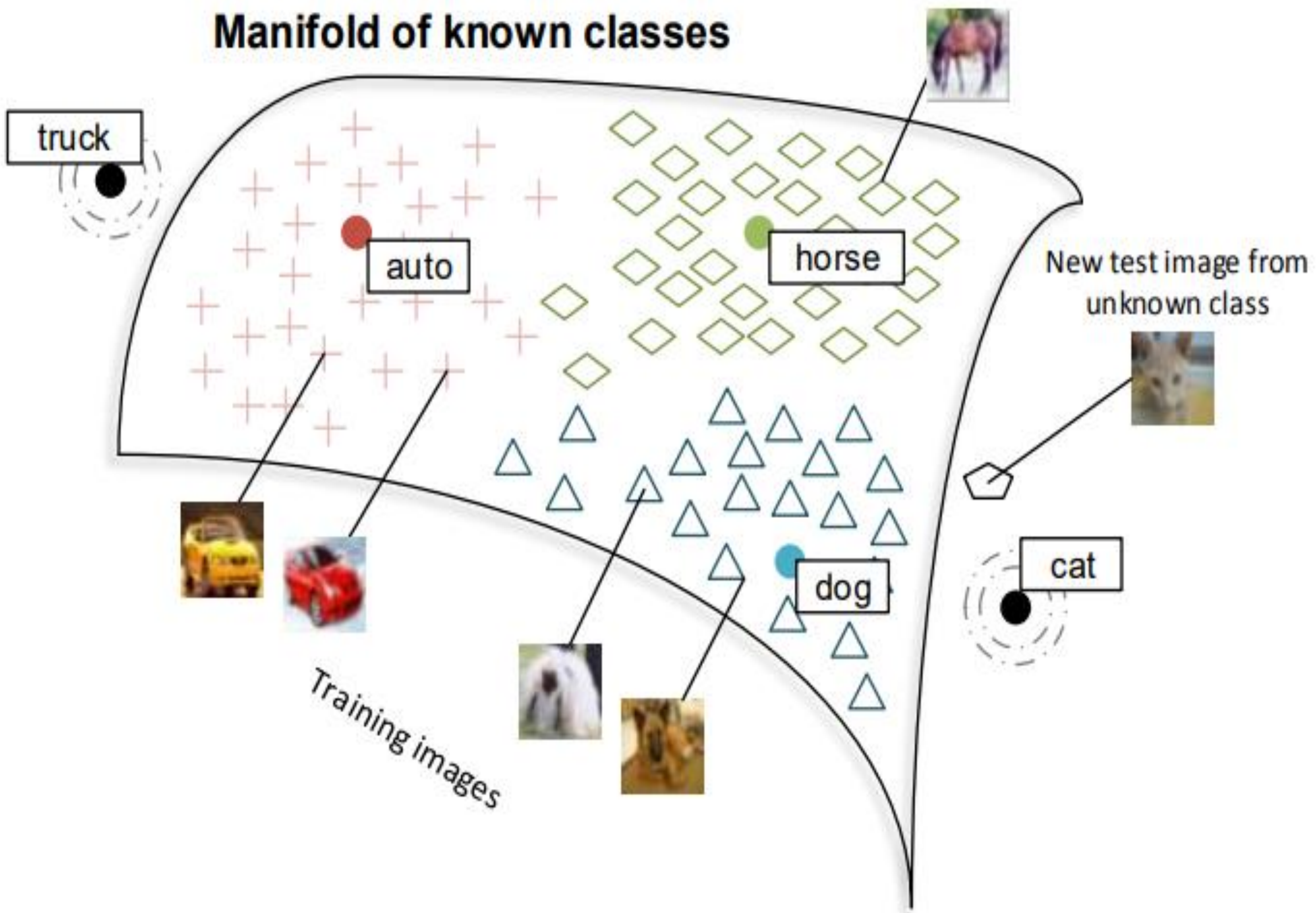
- *Typical approach*: In the training phase, with the training instances D^{tr} and the seen class prototypes T^s, the projection functions θ and ξ are learned. In the testing phase, with the learned projection functions, testing instances X^{te} and unseen class prototypes T^u are projected into the projection space $\mathcal{P}$. Then, each projected testing instance is classified to the nearest unseen class prototype via 1NN classification.

Table 1. Key Notations Used in This Article

Notation	Description
$\mathcal{X}$	Feature space, which is D-dimensional
$\mathcal{T}$	Semantic space, which is M-dimensional
$\mathcal{S}, \mathcal{U}$	Set of seen classes and set of unseen classes, respectively
N_{tr}, N_{te}	Number of training instances and number of testing instances, respectively
N_s, N_u	Number of seen classes and number of unseen classes, respectively
D^{tr}	The set of labeled training data from seen classes
X^{te}	The set of testing instances from unseen classes
Y^{te}	Labels for testing instances
$(\mathbf{x}_i^{tr}, y_i^{tr})$	The ith labeled training instance: features $\mathbf{x}_i^{tr} \in \mathcal{X}$ and label $y_i^{tr} \in \mathcal{S}$
$\mathbf{x}_i^{te}$	The ith unlabeled testing instance: features $\mathbf{x}_i^{te} \in \mathcal{X}$
T^s, T^u	The set of prototypes for seen classes and unseen classes, respectively
$(c_i^s, \mathbf{t}_i^s)$	The ith seen class $c_i^s \in \mathcal{S}$ and its class prototype $\mathbf{t}_i^s \in \mathcal{T}$
$(c_i^u, \mathbf{t}_i^u)$	The ith unseen class $c_i^u \in \mathcal{U}$ and its class prototype $\mathbf{t}_i^u \in \mathcal{T}$
$\pi(\cdot)$	A class prototyping function $\pi(\cdot) : \mathcal{S} \cup \mathcal{U} \to \mathcal{T}$
$f^u(\cdot)$	A zero-shot classifier $f^u(\cdot) : \mathcal{X} \to \mathcal{U}$

CIII cont.

- In the training stage, seen instances and seen prototypes are given
- In the test stage, unseen instances and unseen prototypes are given
- Classify instances based on 1NN



Class-Transductive Instance-Inductive (CTII)

Definition 1.3 (Class-Transductive Instance-Inductive (CTII) Setting). Labeled training instances D^{tr}, seen class prototypes T^s, and unseen class prototypes T^u are used in model learning.

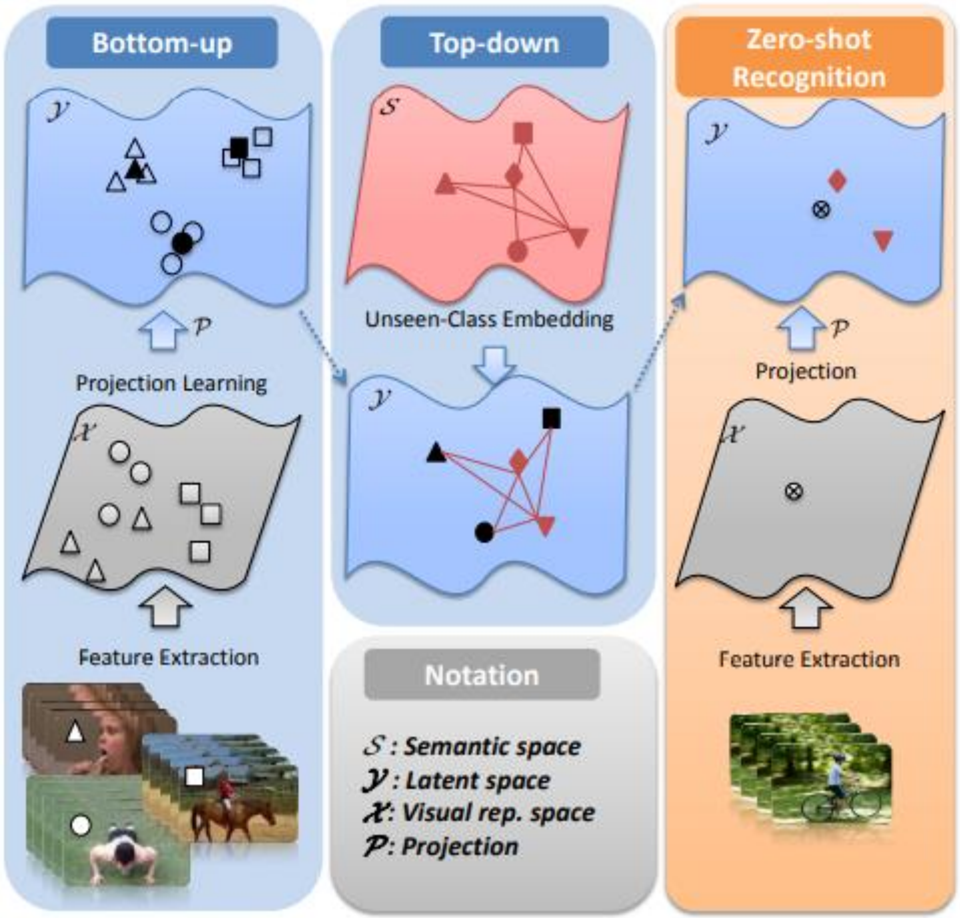
- *Typical approach*: In the training phase, with the training instances D^{tr}, binary classifiers $\{f_i^s(\cdot)\}_{i=1}^{N_s}$ for the seen classes are first learned. Then, with the prototypes T^s of the seen classes and the prototypes T^u of the unseen classes, a graph $\mathcal{G}$ is constructed by taking these classes as nodes. In this way, relationships δ among classes can be obtained via this graph. With the relationships δ in $\mathcal{G}$ and the learned binary seen class classifiers $\{f_j^s(\cdot)\}_{j=1}^{N_s}$, binary classifiers $\{f_i^u(\cdot)\}_{i=1}^{N_u}$ for the unseen classes $\{c_i^u\}_{i=1}^{N_u}$ can be obtained by

$$\{f_i^u(\cdot)\}_{i=1}^{N_u} = g(f_1^s(\cdot), f_2^s(\cdot), \ldots, f_{N_s}^s(\cdot), \delta), \tag{10}$$

where g is the function generating these classifiers. In the testing phase, with the obtained binary unseen class classifiers, classification of the testing instances X^{te} is achieved.

CTII cont.

- In the training stage, seen instances and seen prototypes are given to learn the projection from samples to latent space embeddings
- In the mapping stage, semantic embedding space with the relationship of seen and unseen classes are given to map the unseen class prototypes into the latent space
- In the test stage, unseen instances are given
- Classify instances based on 1NN



Class-Transductive Instance-Transductive (CTIT)

Definition 1.4 (Class-Transductive Instance-Transductive (CTIT) Setting). Labeled training instances D^{tr}, seen class prototypes T^s, unlabeled testing instances X^{te}, and unseen class prototypes T^u are used in model learning.

- *Typical approaches*: With D^{tr}, T^s, X^{te}, and T^u, the projection functions θ and ξ are learned. With these learned projection functions, unseen class prototypes T^u and testing instances X^{te} are projected into the projection space and classification is performed in it.

- Kind of the same as the CTII, except we want to use unseen instance to get a better projection function for unseen prototypes by minimize the distance between projected unseen prototypes and mean of projected unseen instance with pseudo labels.