Zero-Shot Learning with Partial Attributes

Matías D. Molina
matias.molina@unc.edu.ar

Universidad Nacional de Córdoba, Argentina

March, 2018

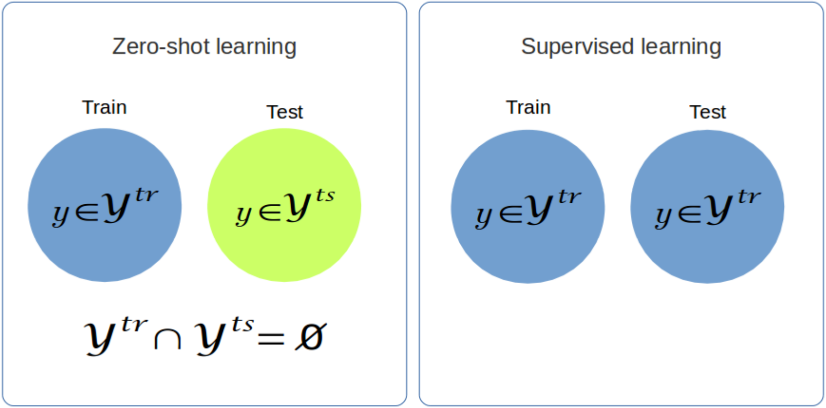
UNC
Universidad
Nacional
de Córdoba

CONICET

FAMAF
Facultad
de Matemática,
Astronomía, Física
y Computación

Outline

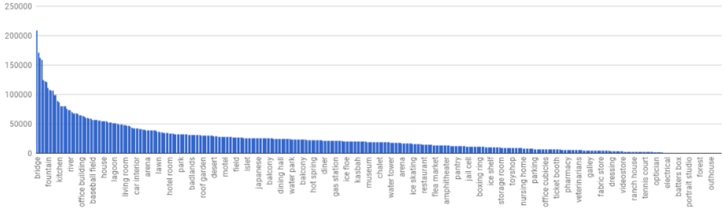
The Problem: Zero-Shot Learning



The Problem: Zero-Shot Learning

Why zero-shot learning?

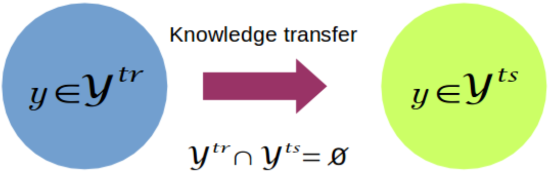
- New categories are always emerging
- Great annotation effort for fine-grained problems
- Long tail distribution



MIT Place2 dataset (10M images)

The Problem: Zero-Shot Learning

How to transfer knowledge?



Subsidiary information

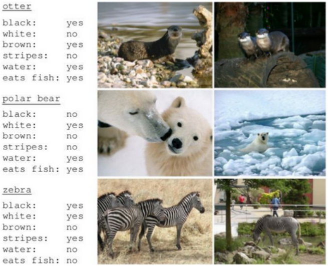
To represent all the (training and testing) categories:

- Human defined **visual attributes.**
  - Good performance
  - High cost (crowdsourcing techniques, human expert)
- Text-based **word embeddings**
  - Scalable
  - Inferior performance

The Problem: Zero-Shot Learning

How to transfer knowledge?

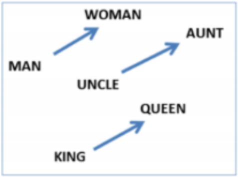
Visual attributes



Lampert et.al. CVPR'09.

$$otter = (1, 0, 1, 0, 1, 1)$$
$$polar\_bear = (0, 1, 0, 0, 1, 1)$$
$$zebra = (1, 1, 0, 1, 0, 0)$$

Word embeddings



Mikolov et.al. NIPS'13.

$$\text{v(queen)} \approx \text{v(king)} + \text{v(woman)} - \text{v(man)}$$

Zero-Shot Learning Models

Notation

- Training inputs (labels): $\mathcal{X}^{tr}(\mathcal{Y}^{tr})$
- Testing inputs (labels): $\mathcal{X}^{ts}(\mathcal{Y}^{ts})$
- Attribute set $\mathcal{A} = \{a_1, ..., a_E\}$
- Attribute vector for class y: $a_y = (a_{y,1}, ..., a_{y,E})$
- Word embedding for class y: ω_y

Zero-Shot Learning Models

Direct Attribute Prediction (Lampert *et. al.*)

Training:

- Learn attribute predictors using image-attribute pairs from the training set: $p(a_i|x), i = 1, .., E. x \in \mathcal{X}^{tr}$
- Define attribute-image predictor $p(a|x) = \prod_i p(a_i|x)$

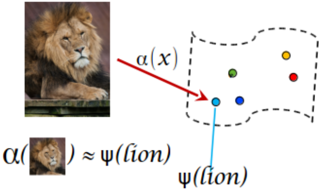
Testing:

- Define the class-attribute predictor: $p(y|a) = \frac{\mathbb{I}[a=a_y]p(y)}{p(a_y)}$
- Combine both predictors to obtain image-class predictor $p(y|x) = \sum p(y|a)p(a|x)$
- Predict according to $\arg\max_{y \in \mathcal{Y}^{ts}} p(y|x)$

Zero-Shot Learning Models

Convex Combination of Semantic Embeddings (Norouzi *et. al.*)

Standard classification problem adapted to the zsl problem by using the auxiliary information.



- Learn a supervised classifier: $p(y|x), (x, y) \in \{(\mathcal{X}^{tr}, \mathcal{Y}^{tr})\}$
- Predict the semantic embedding $\alpha(x)$ by combining the output embeddings of the t-th most likely labels:

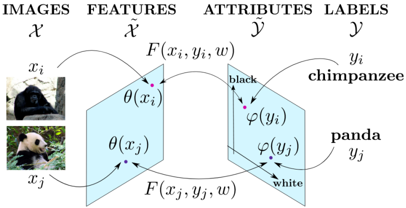
$$\alpha(x) = \sum_{t=1}^{T} \frac{1}{Z} p(\hat{y}_t(x)|x)\psi(\hat{y}_t(x))$$

- Predict using cosine similarity $\arg\max_{y \in \mathcal{Y}^{ts}} cos(\alpha(x), \psi(y))$

Zero-Shot learning models

Structured Joint Embedding (SJE, Akata *et. al.*)

Based on a bilinear compatibility function:



$$F(x, y; W) = \phi(x)^T W \psi(y)$$

- Training (by SGD) based on SSVM (Joachims, 2005): $F(x_n, y_n; W) - F(x_n, y; W) \geq \Delta(y_n, y), \forall y \in \mathcal{Y}^{tr} - \{y_n\}$
- $\ell(x_n, f(x_n; w)) = [\max_{y \in \mathcal{Y}} \Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)]_+$
- Testing: $\arg\max_{y \in \mathcal{Y}^{ts}} F(x, y; W)$

Zero-Shot learning models

Embarrassingly simple approach to zero shot learning (Romera Paredes *et. al.*)

Starting from the general formulation:

$$\min_W L(X, Y; W) + \Omega(W)$$

Defining the following regularizer:

$$\Omega(W) = \gamma||W\psi(Y)||_F^2 + \lambda||\phi(X)^T W||_F^2 + \beta||W||_F^2$$

And taking

$$L(X, Y, W) = ||\phi(X)^T W - \psi(Y)||_F^2, \beta = \lambda\gamma$$

The problem can be solved by a closed form.

Our Work: An improved variant of the SJE method.

- PCA projection step followed by an L2-normalization on the inputs and the outputs.
- ESZSL closed form solution as initialization for the weight matrix W when learning the objective.

		SJE(R)		SJE++		Improvement	
		attr.	w2v	attr.	w2v	attr.	w2v
CUB	GoogLeNet	49.81	28.40	54.69	34.47	+4.88	+6.07
	ResNet	56.02	30.96	59.94	36.82	+3.92	+5.86
	VGG19	49.13	25.43	49.98	34.47	+0.85	+9.04
AWA	GoogLeNet	69.03	49.24	65.66	54.14	-3.37	+4.90
	VGG19	81.32	61.62	81.02	68.40	-0.30	+6.78

Our Work: An improved variant of the SJE method.

		SJE(R)		SJE++		Improvement	
		attr.	w2v	attr.	w2v	attr.	w2v
CUB	GoogLeNet	49.81	28.40	54.69	34.47	+4.88	+6.07
	ResNet	56.02	30.96	59.94	36.82	+3.92	+5.86
	VGG19	49.13	25.43	49.98	34.47	+0.85	+9.04
AWA	GoogLeNet	69.03	49.24	65.66	54.14	-3.37	+4.90
	VGG19	81.32	61.62	81.02	68.40	-0.30	+6.78

- PCA improvement can be explained by considering the granularity of the visual concepts (classes).
- CUB (fine-grained): the PCA projection step helps in disentangling the subtle differences between the representations on both the visual and semantic spaces.

Our Work: An improved variant of the SJE method.

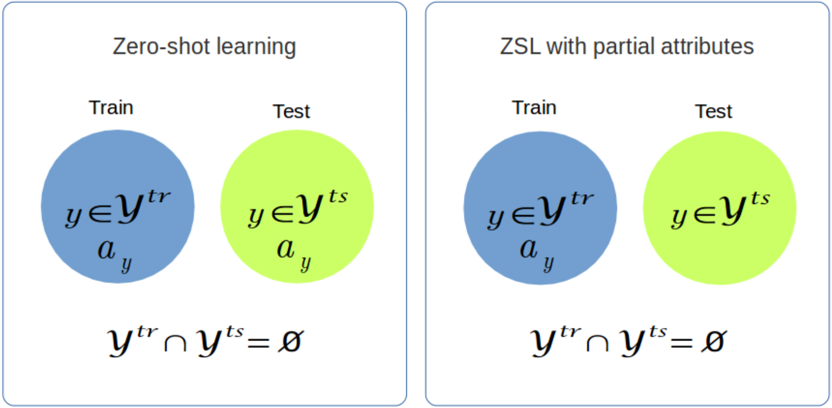
Comparison of SJE++ against the state-of-the-art methods for attribute-based zero-shot learning.

	ResNet		VGG19		GoogLeNet	
	AWA	CUB	AWA	CUB	AWA	CUB
DAP	57.1	37.5	57.23	-	-	-
SSE	68.8	43.3	76.33	30.41	-	-
LATEM	74.8	49.4	-	-	-	-
SYNC	72.2	54.1	-	-	-	-
ConSe	63.6	36.7	-	-	-	-
ALE	78.6	53.2	-	-	-	-
SJE	76.2	55.3	-	-	66.7	50.1
ESZSL	74.7	55.1	75.32	-	-	-
SJE++	-	59.94	81.02	49.98	65.66	54.69

Our Work: Zero-Shot Learning with Partial Attributes

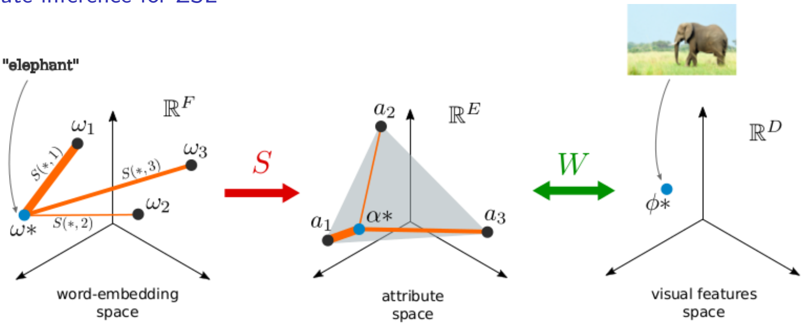
The Partial Attributes Problem

Original ZSL problem vs. Our proposed problem



Zero-Shot Learning with Partial Attributes

Attribute inference for ZSL



Attributes for the testing classes are inferred by using the similarity between the word embedding for each label name:

$$\alpha(y) = \alpha_y = \frac{1}{Z(y)} \sum_{y' \in \mathcal{Y}^{tr}} S(y, y') a_{y'}$$

$$S(y, y') = s(\omega_y, \omega_{y'}) = exp\{-\tau||\omega_y - \omega_{y'}||^2\}$$

Using the attributes inference with SJE

Following the SJE formulation we define the compatibility function and the objective function using the new (transformed) attributes:

- Compatibility function: $F(x, y; W) = \phi(x)^T W \alpha(y)$
- Learning by minimizing the objective:

$$\frac{1}{M} \sum_{1}^{M} \max_{y \in \mathcal{Y}^{tr}} \{0, \Delta(y_n, y) + \phi(x)^T W(\alpha_y - \alpha_{y_n})\}$$

Using the attribute inference with SJE

Performance of our attribute inference approach

	CUB			AWA				
	SJE w2v	SJE++ w2v	α-attr. ω =w2v	SJE w2v	SJE++ w2v	α-attr. ω=w2v	SJE++ GloVe	α-attr. ω =GloVe
GoogLeNet	28.40	34.47	34.57	49.24	54.14	51.17	65.05	64.92
ResNet	30.96	36.82	36.75	-	-	-	-	-
VGG19	25.43	34.47	33.99	61.62	68.40	64.11	68.38	73.33

Conclusions

- We added two simple steps of the SJE model with significant improvements.
  - Applying PCA projection on the inputs and the outputs improves the performance significantly.
  - PCA helps to disentangle fine-grained datasets.
  - Changing the random normal initialization of SJE, the method improves at most 1 point.
- We proposed a variant of the attribute-based zero-shot classification problem where the class attributes are not available at test time.
  - We solved this problem by inferring the attributes deterministically.

Future Works

- Evaluation with different splits (proposed split by Xian et al. (2018)).
- Experiments with more {fine,coarse}-grained datasets.
- Add visual information to create a better attribute inference.
- Learn an function to infer the attributes instead to define it deterministically.
- ...?

Thanks :)