# Contributions to Large-Scale Learning for Image Classification

Zeynep Akata

PhD Defense, January 6th 2014

| | |
|---|---|
| **Rapporteur** | Prof Matthieu Cord |
| **Rapporteur** | Prof Christoph Lampert |
| **Examinateur** | Prof Vittorio Ferrari |
| **Examinateur** | Dr Georges Quénot |
| **Directeur de Thése** | Dr Cordelia Schmid |
| **Co-directeur de Thése** | Dr Florent Perronnin |

# Motivation


Decrease in the cost of digital cameras

- Large personal visual data collections

The Internet and social networking websites

- Visual data shared with public

# Motivation

How to process and access such big data?

- Manual management is impossible
- Classify data automatically for easy access
  - Assign keywords to images

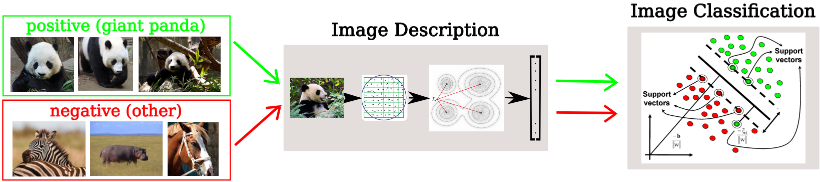
# Motivation

How to process and access such big data?

- Manual management is impossible
- Classify data automatically for easy access
  - Assign keywords to images

Focus of this thesis: image classification in large image sets

# Standard classification pipeline



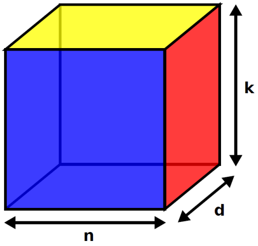

- **Input:** Image descriptors and labels $\{(\mathbf{x}_i, y_i)\}$ where $\mathbf{x}_i \in X$ and $Y = \{1, ..., C\}$
- **Goal:** Learn a prediction function $f : X \rightarrow [0, 1]^C$ that predicts the presence/absence of each label

Description: [Csurka *et al.*'04], [Lazebnik *et al.*'06], [Zhang *et al.*'07]
Classification: [Boser *et al.*'92], [Cortes and Vapnik'95]

# Dimensions of large-scale learning

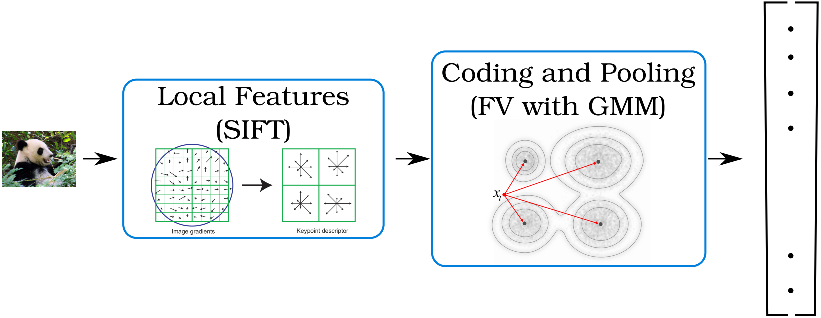
Scale of a learning problem is measured through 3 dimensions:

- Descriptor dimensionality ($d$)
- Number of classes ($k$)
- Number of images ($n$)



# Descriptor dimensionality

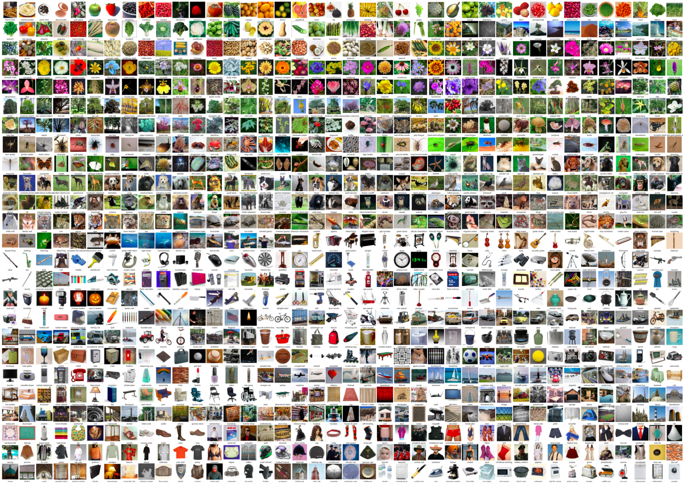
Fisher Vectors (FV) [Perronnin and Dance'07], [Perronnin *et al.*'10] use high order statistics to map images into high dimensional space



# Number of classes and images

ImageNet [Deng *et al.*'09] is an example of large scale datasets

- $k = 21,841$ classes and $n = 14 \times 10^6$ labeled images



poster created by Fengjun Lv using VIPBase

# State-of-the-art for large-scale learning

1. Handling large descriptor dimensionality ($d$):

   Linear classifiers and descriptor compression

   [Perronnin *et al.*'10], [Jégou *et al.*'11], [Sánchez *et al.*'11]

2. Handling large number of classes ($k$):

   Train one classifier at a time with One-vs-Rest SVM

   [Rifkin and Klautau'04]

3. Handling large number of images ($n$):

   Process one sample at a time

   [Bottou and Bousquet'07], [Shalev-Shwartz *et al.*'07]

# Contribution 1

**Good practices in large-scale learning**

- Compare different objective functions for linear SVMs
- Analyze the effects of key parameters

# Scarceness of labeled data

Fine-grained subsets of Imagenet are sparsely populated

- Difficult to harvest images, *e.g.* from the Internet
- Image labeling can only be done by experts which is costly

Spanish Fly:

Jerboa Kangaroo:

Argentinosaur:

(Some of the least populated classes in ImageNet)

# State-of-the-art for learning with scarce labeled data

1. Attributes enable parameter sharing between classes

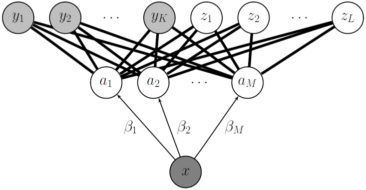
   [Ferrari *et al.*'07], [Lampert *et al.*'09]

2. Zero-shot learning:

   Direct Attribute Prediction (DAP) [Lampert *et al.*'09]

bill shape::dagger
size::small
wing color::rufous
wing color::orange
upperparts color::rufous
underparts color::pink
back color::grey
→
**attributes**
[1 1 1 1 1 1 0 0 1 0 0]
**member class**
Rufous Hummingbird

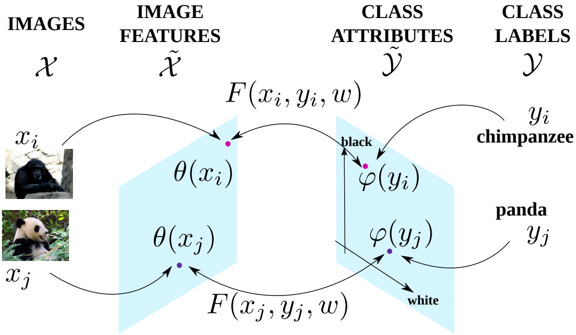
bill shape::dagger
size::small
underparts color::olive
underparts color::green
back color::grey
upper tail color::rufous
upper tail color::grey
→
**attributes**
[1 1 0 0 0 0 1 1 1 1 1]
**member class**
Ruby-throated Hummingbird



# Contribution 2

**Label-embedding for image classification**

- Learning with scarce training data
- Embed classes in a Euclidean space with side information



# Outline

# Outline

1. Good practices in large-scale learning
2. Label-embedding with attributes
3. Conclusion

**Towards Good Practice in Large Scale Learning for Image Classification**
F. Perronnin, Z.Akata, Z.Harchaoui, C.Schmid, *IEEE CVPR, 2012.*

**Good Practice in Large Scale Learning for Image Classification**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *to appear in IEEE TPAMI, 2013.*

# Introduction

Accuracy in ImageNet: top-k accuracy

- Correct if actual label appears in the first top-k labels

| Query | Top-5 |
|---|---|
|  | House<br>Bush<br>Fence<br>Flowerbed<br>Greenhouse |

# Introduction

Accuracy in ImageNet: top-k accuracy

- Correct if actual label appears in the first top-k labels

| Query | Top-5 |
|---|---|
|  | House<br>Bush<br>Fence<br>Flowerbed<br>Greenhouse |

# Introduction

Accuracy in ImageNet: top-k accuracy

- Correct if actual label appears in the first top-k labels

| Query | Top-5 |
|---|---|
|  | House<br>Bush<br>Fence<br>Flowerbed<br>Greenhouse |

Why is top-k accuracy useful?

- Image has multiple objects but a single label is assigned
- k can be adjusted based on the recall target

# Introduction

Accuracy in ImageNet: top-k accuracy

- Correct if actual label appears in the first top-k labels

| Query | Top-5 |
|---|---|
|  | House |
| | Bush |
| | Fence |
| | Flowerbed |
| | Greenhouse |

Why is top-k accuracy useful?

- Image has multiple objects but a single label is assigned
- k can be adjusted based on the recall target

Top-k accuracy $\implies$ rank annotations according to relevance

# Introduction

Alternatives for choosing the objective function

# Introduction

Alternatives for choosing the objective function

- “ Simple one-vs-rest is as accurate as any other approach “
  [Rifkin and Klautau ’04]

# Introduction

Alternatives for choosing the objective function

- “ Simple one-vs-rest is as accurate as any other approach “
  [Rifkin and Klautau ’04]
- “ Ranking (WSABIE) outperforms all competing methods “
  [Weston *et al.*’10]

# Introduction

Alternatives for choosing the objective function

- " Simple one-vs-rest is as accurate as any other approach "
  [Rifkin and Klautau '04]
- " Ranking (WSABIE) outperforms all competing methods "
  [Weston *et al.*'10]

→ Compare one-vs-rest and ranking algorithms on large-scale

# Objective functions

- $S = \{(\mathbf{x}_i, y_i), i = 1 \ldots N\}$, $\mathbf{x}_i \in \mathcal{X}$, $y_i \in \mathcal{Y} = \{1, ..., C\}$

- Supervised learning:

$$\min_{\mathbf{W}} \frac{\lambda}{2}\Omega(\mathbf{W}) + L(S; \mathbf{W})$$

- Empirical risk:

$$L(S; \mathbf{W}) := \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{x}_i, y_i; \mathbf{W})$$

- Regularization:

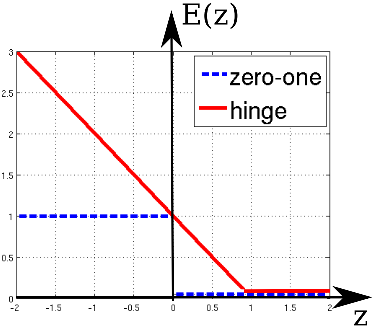
$$\Omega(\mathbf{W}) := \sum_{c=1}^{C} \|\mathbf{w}_c\|^2$$

# One-vs-Rest SVM (OVR)

- Two classes: $\mathcal{Y} = \{-1, +1\}$
- 0/1 loss: $\mathbb{1}(y_i \mathbf{w}^T \mathbf{x}_i < 0)$
- Upper-bounded by:

  $L_{\text{OVR}}(\mathbf{x}_i, y_i; \mathbf{w}) = \max\{0, 1 - y_i \mathbf{w}^T \mathbf{x}_i\}$

- $C$ classes: train $C$ independent classifiers



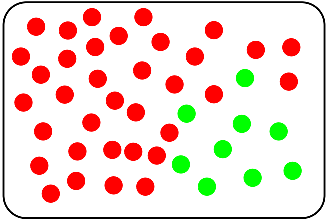

$\rightarrow$ Training time scales linearly with the number of classes

# Sample rebalancing

OVR: many more negative samples than the positives

- Standard formulation of OVR without reweighting

$$\sum_{i \in I_+} L_{\mathsf{OVR}}(\mathbf{x}_i, y_i; \mathbf{w}) + \sum_{i \in I_-} L_{\mathsf{OVR}}(\mathbf{x}_i, y_i; \mathbf{w})$$



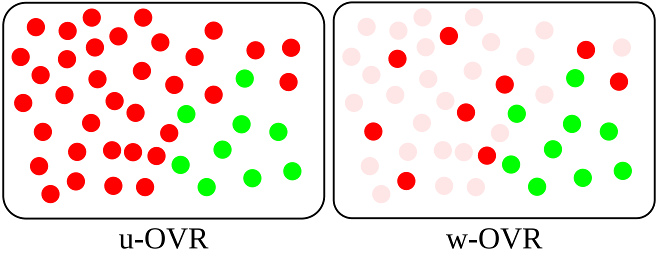
u-OVR

# Sample rebalancing

OVR: many more negative samples than the positives

- Unbalance parameter $\rho$

$$\frac{\rho}{N_+}\sum_{i\in I_+} L_{\text{OVR}}(\mathbf{x}_i, y_i; \mathbf{w}) + \frac{1-\rho}{N_-}\sum_{i\in I_-} L_{\text{OVR}}(\mathbf{x}_i, y_i; \mathbf{w})$$



# Ranking framework

Consider $C$ classes at once: $\mathcal{Y} = \{1, \ldots, C\}$

Goal:

- Enforce $\mathbf{w}_{y_i}^T \mathbf{x}_i > \mathbf{w}_y^T \mathbf{x}_i$ with $y_i =$ correct label and $y \neq y_i$

Define:

- $\alpha_k =$ penalty of going from rank $k$ to $k+1$
- Cumulative penalty $\ell_k = \sum_{j=1}^{k} \alpha_j$ with $\alpha_1 \geq \alpha_2 \geq \ldots \alpha_C \geq 0$

Objective function:

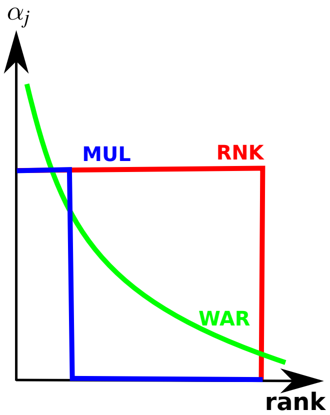
- $\ell_{r(\mathbf{x},y)}$ where $r(\mathbf{x}, y) =$ rank of label $y$ for sample $\mathbf{x}$

[Usunier *et al.*'09]

# Ranking algorithms

Loss: $\ell_k = \sum_{j=1}^{k} \alpha_j$

1. Multiclass SVM (MUL):
   $\alpha_1 = 1$ and $\alpha_j = 0$ for $j \geq 2$
   [Crammer and Singer'01]
2. Ranking SVM (RNK): $\alpha_j = 1$ , $\forall j$
   [Joachims'02]
3. Weighted Approximate Ranking (WAR):
   $\alpha_j = 1/j$ [Weston *et al.*'10]




MUL and RNK use an upper bound of the loss while WAR uses an approximation.

# Optimization

Stochastic Gradient Descent (SGD) for optimization:

1. Choose a sample $z_t$ at random at step $t$
   - OVR & MUL: $z_t$ is a pair $(\mathbf{x}_i, y_i)$
   - RNK & WAR: $z_t$ is a triplet $(\mathbf{x}_i, y_i, \bar{y})$, where $\bar{y} \neq y_i$

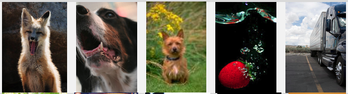
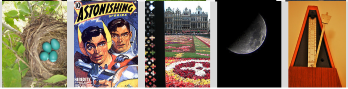
2. Update the parameters $\mathbf{w}$ using a sample-wise estimate of the regularized risk $R(z_t; \mathbf{w})$

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta_t \nabla_{\mathbf{w}=\mathbf{w}^{(t-1)}} R(z_t; \mathbf{w})$$

   where $\eta_t$ is the step size

[Bottou and Bousquet'07], [Shalev-Shwartz *et al.*'07]

# Datasets used in experiments

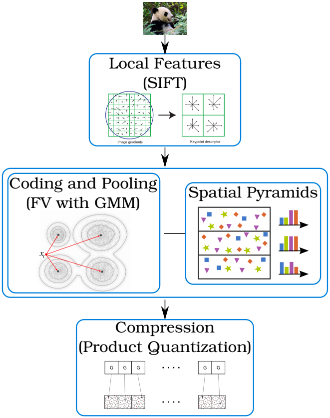
| | # images | # classes | Example Images |
|---|---|---|---|
| ILSVRC10 | 1.4M | 1,000 |  |
| ImageNet10K | 9M | 10,184 |  |

→ We report results with Top-1 accuracy

[Deng *et al.*'09, Deng *et al.*'10]

# Image descriptors used in experiments

- Local features ($D = 128$) with SIFT [Lowe'04] + PCA
- Visual vocabulary with Gaussian Mixture Models ($G = 8, ..., 256$)
- Aggregating features with BOV (4K-dim) [Csurka *et al.*'04] or FV (130K-dim) [Perronnin and Dance '07]
- Spatial Pyramids ($S = 4$) [Lazebnik *et al.*'06]
- Compression with Product Quantization [Jegou *et al.*'11]



# Experiments

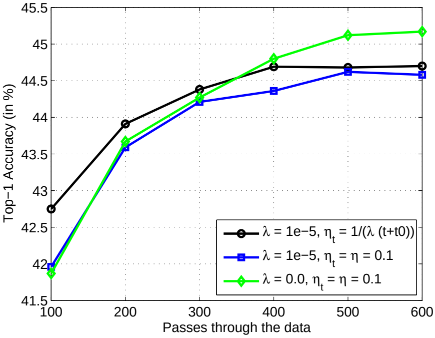
1. Regularization $\lambda$ in $\min_{\mathbf{W}} \frac{\lambda}{2}\Omega(\mathbf{W}) + L(S; \mathbf{W})$
2. Step size $\eta_t$ in $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta_t \nabla_{\mathbf{w}=\mathbf{w}^{(t-1)}} R(z_t; \mathbf{w})$
3. Unbalance parameter $\rho$ in sample rebalancing
4. Descriptor dimensionality $d$
5. Comparison between different objective functions

# Regularization and step size

1. Is explicit regularization better than implicit regularization?
2. Is decreasing step size better than constant step size?

# Regularization and step size

1. Is explicit regularization better than implicit regularization?
2. Is decreasing step size better than constant step size?




ILSVRC10 with w-OVR

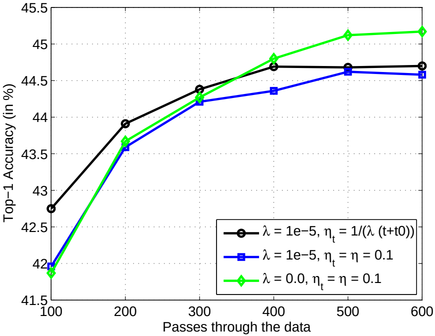
a) $\lambda > 0$ and $\eta_t = 1/(\lambda(t + t_0))$

b) $\lambda > 0$ and $\eta_t = \eta$

c) $\lambda = 0$ and $\eta_t = \eta$

# Regularization and step size

1. Is explicit regularization better than implicit regularization?
2. Is decreasing step size better than constant step size?




ILSVRC10 with w-OVR

a) $\lambda > 0$ and $\eta_t = 1/(\lambda(t + t_0))$
b) $\lambda > 0$ and $\eta_t = \eta$
c) $\lambda = 0$ and $\eta_t = \eta$

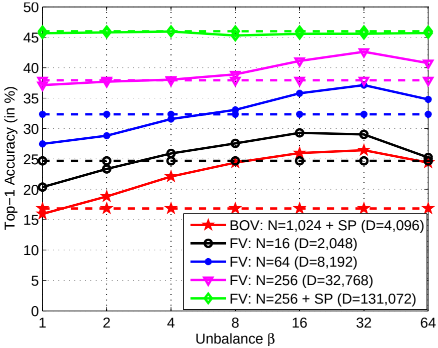
- Implicit regularization with fixed step size is effective
- It requires one less parameter to tune

# Data rebalancing

3. Is data rebalancing beneficial in OVR on large scale?

# Data rebalancing

❸ Is data rebalancing beneficial in OVR on large scale?

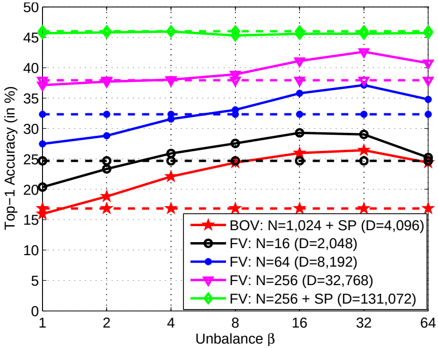



ILSVRC10 with w-OVR

- $\beta = (1-\rho)/\rho$: number of negatives sampled for each positive
- Dashed lines = u-OVR

# Data rebalancing

❸ Is data rebalancing beneficial in OVR on large scale?




ILSVRC10 with w-OVR

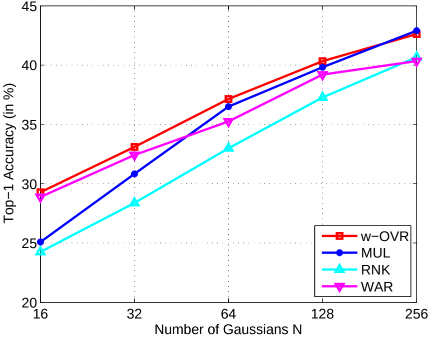
- $\beta = (1-\rho)/\rho$: number of negatives sampled for each positive
- Dashed lines = u-OVR
- Rebalancing is beneficial for small dimensional features [Bartlett *et al.*'03]

# Descriptor dimensionality ($d$)

4. How do different methods behave with increasing descriptor dimensionality on large scale?

# Descriptor dimensionality ($d$)

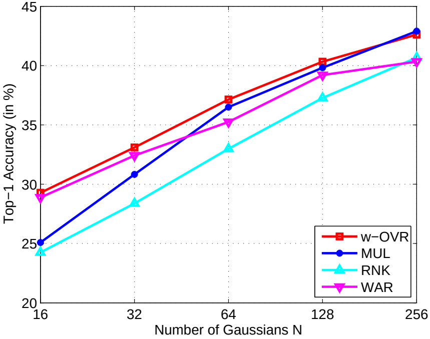
4. How do different methods behave with increasing descriptor dimensionality on large scale?




ILSVRC10

# Descriptor dimensionality ($d$)

4. How do different methods behave with increasing descriptor dimensionality on large scale?




ILSVRC10

- Methods tend to converge
- With the increasing descriptor dimensionality
- Impact of surrogate loss is mitigated as capacity of the classifier increases

# Comparison of all methods

5. Which method works best on large scale?

# Comparison of all methods

5. Which method works best on large scale?

- Comparison between methods on ILSVRC10

| | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|
| BOV 4K | 15.8 | 26.4 | 22.7 | 20.8 | 24.1 |
| FV 130K | 45.9 | 45.7 | 46.2 | 46.1 | 46.1 |

- Comparison between methods on ImageNet10K

| | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|
| BOV 4K | 3.8 | 7.5 | 6.0 | 4.4 | 7.0 |
| FV 130K | - | 19.1 | - | - | 17.9 |

u-OVR: unweighted OVR, w-OVR: weighted OVR

MUL: Multiclass, RNK: Ranking, WAR: Weighted Average Ranking

# Comparison of all methods

5. Which method works best on large scale?

- Comparison between methods on ILSVRC10

| | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|
| BOV 4K | 15.8 | **26.4** | 22.7 | 20.8 | 24.1 |
| FV 130K | 45.9 | 45.7 | **46.2** | 46.1 | 46.1 |

- Comparison between methods on ImageNet10K

| | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|
| BOV 4K | 3.8 | 7.5 | 6.0 | 4.4 | 7.0 |
| FV 130K | - | 19.1 | - | - | 17.9 |

u-OVR: unweighted OVR, w-OVR: weighted OVR

MUL: Multiclass, RNK: Ranking, WAR: Weighted Average Ranking

# Comparison of all methods

5. Which method works best on large scale?

- Comparison between methods on ILSVRC10

| | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|
| BOV 4K | 15.8 | **26.4** | 22.7 | 20.8 | 24.1 |
| FV 130K | 45.9 | 45.7 | **46.2** | 46.1 | 46.1 |

- Comparison between methods on ImageNet10K

| | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|
| BOV 4K | 3.8 | **7.5** | 6.0 | 4.4 | 7.0 |
| FV 130K | - | **19.1** | - | - | 17.9 |

u-OVR: unweighted OVR, w-OVR: weighted OVR

MUL: Multiclass, RNK: Ranking, WAR: Weighted Average Ranking

# Qualitative examples from ImageNet10K

- Some classes with top-1 accuracy higher than 85%

Star Anise (92%)

Nest Egg (87%)

Geyser (86%)

- Some classes with 75% and 50% top-1 accuracy

Traction engine (77 %)

Ready to Wear (76 %)

Stonechat (50%)

# Qualitative examples from ImageNet10K

- Some classes with 25% and 10% top-1 accuracy

Tortrix (25%)

Pyralid (25%)

Egyptian cobra (10%)

- Some classes with 5% and 0% top-1 accuracy

Hare (5%)

Weasel (5%)

Felt fungus (0%)

# Good practices for large-scale image classification

1. Early stopping: fast training and good generalization
2. Step-size: small constant step-size is sufficient
3. Sample rebalancing: a must in OVR
4. Sufficiently large descriptors: all methods tend to converge
5. OVR: efficient for large-scale classification

**Towards Good Practice in Large Scale Learning for Image Classification**
F. Perronnin, Z.Akata, Z.Harchaoui, C.Schmid, *IEEE CVPR, 2012.*

**Good Practice in Large Scale Learning for Image Classification**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *to appear in IEEE TPAMI, 2013.*

# Outline

1. Good practices in large-scale learning
2. Label-embedding with attributes
3. Conclusion

**Label-Embedding with Attributes**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *IEEE CVPR, 2013.*

**Attribute-Based Classification with Label-Embedding**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *ORL Workshop at NIPS, 2013.*

**Label-Embedding with Attributes for Image Classification**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *Submitted to IEEE TPAMI.*

# Introduction

Large-scale datasets have fine-grained subsets

Hummingbirds

# Introduction

Large-scale datasets have fine-grained subsets

Hummingbirds

Fine-grained images can only be distinguished by experts


Due to cost of image labeling: scarce labeled data

# Attributes

Visual qualities of objects such as red or striped [Ferrari *et al.*'07]

- Understandable by humans and interpretable by computers

Human-specified high-level description of objects [Lampert *et al.*'09]

- Enable parameter sharing between classes

Rufous Hummingbird

bill shape::dagger
size::small
wing color::rufous
wing color::orange
upperparts color::rufous
underparts color::pink
back color::grey

Ruby-throated Hummingbird

bill shape::dagger
size::small
underparts color::olive
underparts color::green
back color::grey
upper tail color::rufous
upper tail color::grey

# Attributes

Visual qualities of objects such as red or striped [Ferrari *et al.*'07]

- Understandable by humans and interpretable by computers

Human-specified high-level description of objects [Lampert *et al.*'09]

- Enable parameter sharing between classes

Rufous Hummingbird

**bill shape::dagger**
**size::small**
wing color::rufous
wing color::orange
upperparts color::rufous
underparts color::pink
**back color::grey**

class attributes

[**1** **1** 1 1 1 1 0 0 **1** 0 0]

Ruby-throated Hummingbird

**bill shape::dagger**
**size::small**
underparts color::olive
underparts color::green
**back color::grey**
upper tail color::rufous
upper tail color::grey

[**1** **1** 0 0 0 0 1 1 **1** 1 1]

# Attributes

Visual qualities of objects such as red or striped [Ferrari *et al.*'07]

- Understandable by humans and interpretable by computers

Human-specified high-level description of objects [Lampert *et al.*'09]

- Enable parameter sharing between classes

Rufous Hummingbird

bill shape::dagger
size::small
**wing color::rufous**
**wing color::orange**
**upperparts color::rufous**
**underparts color::pink**

class attributes
[1 1 **1** **1** **1** **1** 0 0 1 0 0]

Ruby-throated Hummingbird

bill shape::dagger
size::small
underparts color::olive
underparts color::green
back color::grey
upper tail color::rufous
upper tail color::grey

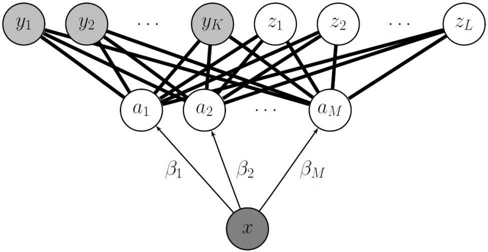
[1 1 **0** **0** **0** **0** 1 1 1 1 1]

# Direct Attribute Prediction (DAP)

Image $x$ is assigned to the class $y$ with maximum

$$p(y|x) \propto \prod_{e=1}^{E} p(a_e = \rho_{y,e}|x)$$

where $\rho_{y,e}$ associates an attribute $a_e$ and a class $y$




[Lampert *et al.*'09]

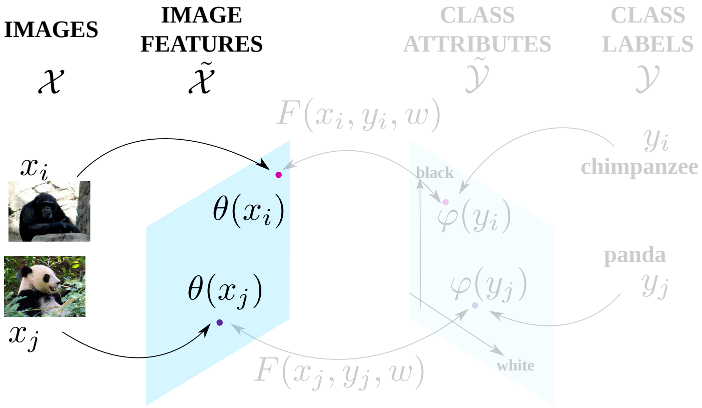
# Direct Attribute Prediction (DAP)

State-of-the-art DAP has 3 potential shortcomings

1. Two-step procedure:
   - Learn attribute classifiers & combine attribute scores
2. Attributes are costly to obtain:
   - Not clear how to integrate other sources of side information
3. Difficult to leverage few additional labeled samples

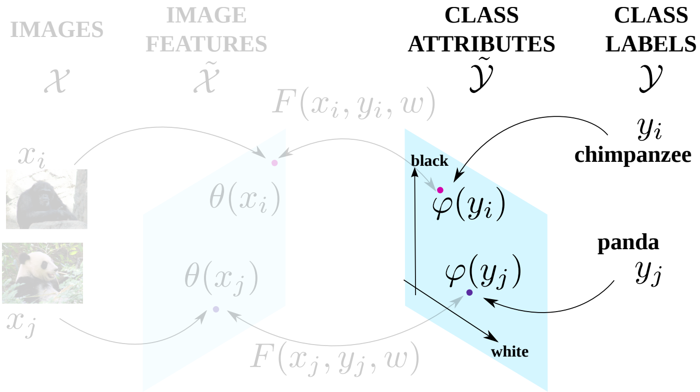
# Attribute Label Embedding (ALE)

ALE is a novel algorithm that overcomes these problems.



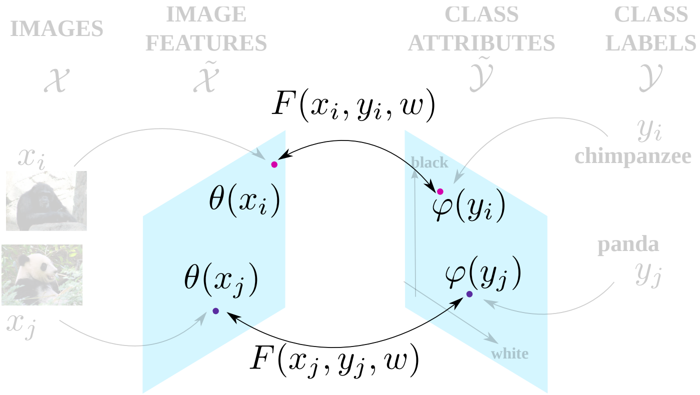
# Attribute Label Embedding (ALE)

ALE is a novel algorithm that overcomes these problems.



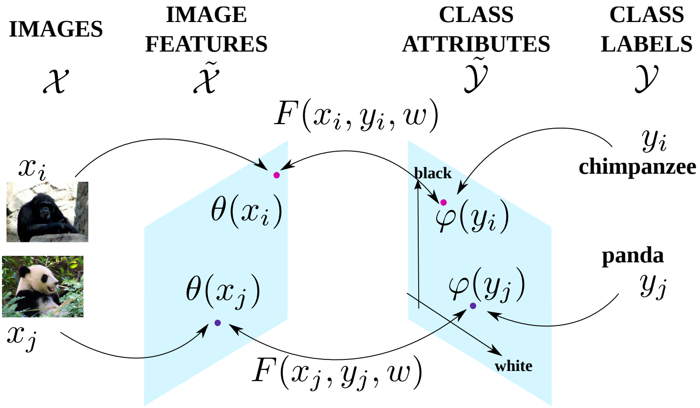
# Attribute Label Embedding (ALE)

ALE is a novel algorithm that overcomes these problems.



# Attribute Label Embedding (ALE)

ALE is a novel algorithm that overcomes these problems.



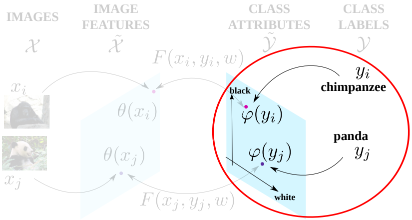
# Attribute Label Embedding (ALE)

Define: $\mathcal{Y} = \{1, \ldots, C\}$ and $\mathcal{A} = \{a_i, i = 1 \ldots E\}$

Association between a class $y$

and an attribute $a_i$: $\rho_{y,i}$

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




$\varphi^{\mathcal{A}}(y)$ models

- Presence/absence of each attribute: $\rho_{y,i} \in \{0, 1\}$ or $\{-1, 1\}$
- Confidence level of each attribute: $\rho_{y,i} \in \mathcal{R}$

# Attribute Label Embedding (ALE)

1. Optimizes directly the classification objective

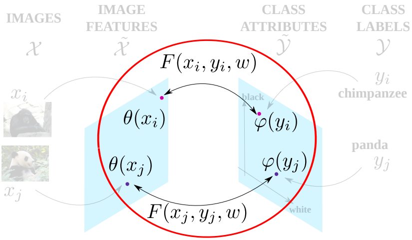
Structured output learning

[Tsochantaridis *et al.*'05]

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

Compatibility function:

$$F(x,y;W) = \theta(x)^T W \varphi(y)$$




Input: $\theta(x)$ = image features and $\varphi(y)$ = class attributes

Output: $W$ = mapping between $\theta(x)$ and $\varphi(y)$

# Parameter learning

Strategies for optimization

a) Maximize correlation between input and output
[Palatucci *et al.*'09, Socher *et al.*'13]

$$\frac{1}{N}\sum_{i=1}^{N} F(x_i, y_i; W)$$

- Does not directly optimize object classification

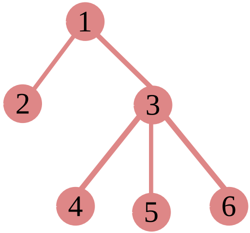
b) Maximize the ranking of the correct label
- Use any ranking method
[Joachims'02], [Crammer and Singer'02], [Weston *et al.*'10]

# Attribute Label Embedding (ALE)

❷ Other sources of side information easily integrated

HLE: Hierarchy Label-Embedding
[Tsochantaridis *et al.*'05]

$$\Phi^{\mathcal{H}}(6) = [1\ 0\ 1\ 0\ 0\ 1]$$



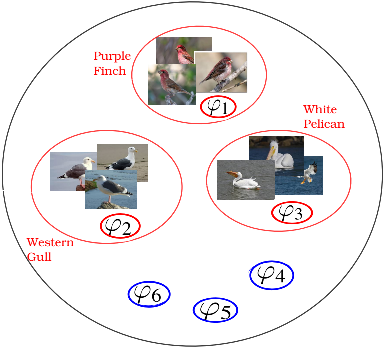

Different sources can be combined

- Early fusion of output embeddings
- Late fusion of scores

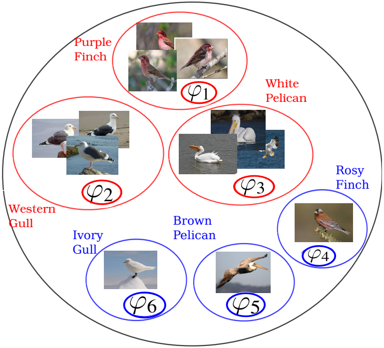
# Attribute Label Embedding (ALE)

3. Easy to leverage few additional labeled samples

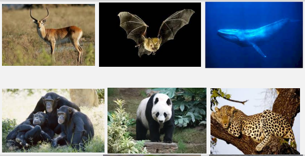
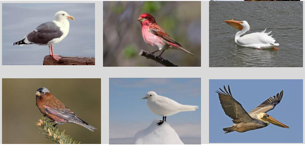
**Zero-shot objective:**
$\Phi$ is fixed, $W$ is learned




**Few-shots objective:**
$\Phi$ and $W$ are learned



# Datasets used in experiments

| | # classes | # attributes | Example images |
|---|---|---|---|
| Animals with Attributes (AWA) [Lampert *et al.*'09] | 50 | 85 |  |
| Caltech UCSD Birds (CUB) [Wah *et al.*'11] | 200 | 312 |  |

# Input and output embeddings

Input embeddings

- 128-dim SIFT and 96-dim color $\rightarrow$ 64-dim PCA
- GMM with 16 or 256 Gaussians $\rightarrow$ FV(4K or 64K)

Output embeddings

1. Baselines: No side information
   - OVR: $\Phi = C \times C$ identity matrix
   - Gaussian LE: $\Phi$ is drawn from $\mathcal{N}(\mu, \sigma^2)$ [Hsu *et al.*'09]
   - WSABIE [Weston *et al.*'10]: $\Phi$ and $W$ are learned

# Input and output embeddings

Input embeddings

- 128-dim SIFT and 96-dim color $\rightarrow$ 64-dim PCA
- GMM with 16 or 256 Gaussians $\rightarrow$ FV(4K or 64K)

Output embeddings

1. Baselines: No side information
   - OVR: $\Phi = C \times C$ identity matrix
   - Gaussian LE: $\Phi$ is drawn from $\mathcal{N}(\mu, \sigma^2)$ [Hsu *et al.*'09]
   - WSABIE [Weston *et al.*'10]: $\Phi$ and $W$ are learned
2. Using side information:
   - ALE: continuous and discrete attributes ($\Phi^{\mathcal{A}}$)
   - HLE: hierarchical label embedding ($\Phi^{\mathcal{H}}$)
   - AHLE: ALE and HLE concatenated ($\Phi^{\mathcal{A}}$ and $\Phi^{\mathcal{H}}$)

# Experiments

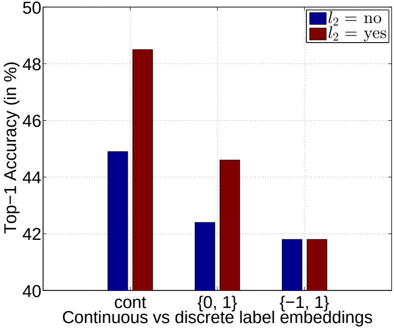
1. Discrete vs continuous embeddings
2. Different objectives for learning in ALE
3. ALE vs DAP for object prediction
4. Attributes and Hierarchies for label embedding
5. Determine if side information is beneficial in few-shots

# Discrete vs continuous embeddings

1. In zero-shot learning with ALE, how do discrete and continuous embeddings compare?

# Discrete vs continuous embeddings

1. In zero-shot learning with ALE, how do discrete and continuous embeddings compare?

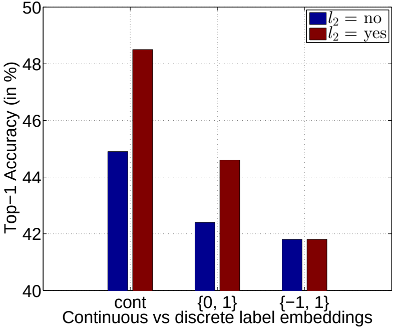



AWA (FV=64K)

- $\ell_2$ norm: each class is closest to itself $\rightarrow$ dot product similarity

# Discrete vs continuous embeddings

1. In zero-shot learning with ALE, how do discrete and continuous embeddings compare?




AWA (FV=64K)

- $\ell_2$ norm: each class is closest to itself $\rightarrow$ dot product similarity
- Continuous embedding outperforms discrete embeddings

# Learning framework in ALE

2. Does learning framework make a difference in ALE for zero-shot learning?

# Learning framework in ALE

2. Does learning framework make a difference in ALE for zero-shot learning?

| | RR | MUL | WAR |
|---|---|---|---|
| AWA dataset | 44.5 | 47.9 | 48.5 |
| CUB dataset | 21.6 | 26.3 | 26.3 |

RR: Ridge Regression [Hoerl and Kennard'70], MUL: Multiclass [Crammer and Singer'02], WAR: Weighted Average Ranking [Weston *et al.*'10]

# Learning framework in ALE

2. Does learning framework make a difference in ALE for zero-shot learning?

| | RR | MUL | WAR |
|---|---|---|---|
| AWA dataset | 44.5 | 47.9 | **48.5** |
| CUB dataset | 21.6 | **26.3** | **26.3** |

- ALE: Ranking objective performs the best

RR: Ridge Regression [Hoerl and Kennard'70], MUL: Multiclass [Crammer and Singer'02], WAR: Weighted Average Ranking [Weston *et al.*'10]

# ALE vs DAP

3. How do ALE and DAP compare for object prediction in zero-shot learning?

# ALE vs DAP

3. How do ALE and DAP compare for object prediction in zero-shot learning?

| | DAP | ALE cont | ALE $\{0, 1\}$ |
|---|---|---|---|
| AWA dataset | 41.0 | 48.5 | 44.6 |
| CUB dataset | 12.3 | 26.3 | 22.3 |

- DAP: OVR with $\log$ loss for each attribute
- DAP [Lampert *et al.*'09]: different features + nonlinear kernels

# ALE vs DAP

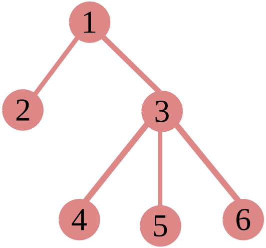
3. How do ALE and DAP compare for object prediction in zero-shot learning?

| | DAP | ALE cont | ALE $\{0, 1\}$ |
|---|---|---|---|
| AWA dataset | 41.0 | **48.5** | 44.6 |
| CUB dataset | 12.3 | **26.3** | 22.3 |

- DAP: OVR with $\log$ loss for each attribute
- DAP [Lampert *et al.*'09]: different features + nonlinear kernels
- ALE: with continuous attributes performs the best

# ALE vs HLE

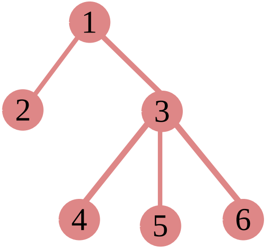
4. How do ALE and HLE compare for zero-shot learning and do they contain complementary information?




- HLE: Hierarchy Label-Embedding
- AHLE early: $\Phi^{\mathcal{H}}$ & $\Phi^{\mathcal{A}}$ concatenated
- AHLE late: ALE & HLE scores combined

# ALE vs HLE

4. How do ALE and HLE compare for zero-shot learning and do they contain complementary information?

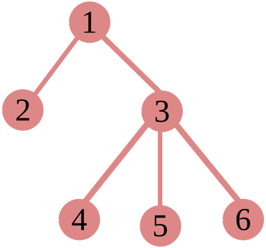



- HLE: Hierarchy Label-Embedding
- AHLE early: $\Phi^{\mathcal{H}}$ & $\Phi^{\mathcal{A}}$ concatenated
- AHLE late: ALE & HLE scores combined

| | ALE | HLE | AHLE early | AHLE late |
|---|---|---|---|---|
| AWA dataset | 48.5 | 40.4 | 46.8 | 49.4 |
| CUB dataset | 26.9 | 18.5 | 27.1 | 27.3 |

# ALE vs HLE

4. How do ALE and HLE compare for zero-shot learning and do they contain complementary information?




- HLE: Hierarchy Label-Embedding
- AHLE early: $\Phi^{\mathcal{H}}$ & $\Phi^{\mathcal{A}}$ concatenated
- AHLE late: ALE & HLE scores combined

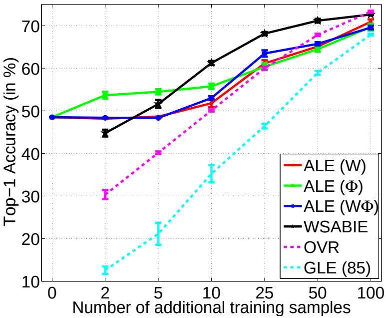
| | ALE | HLE | AHLE early | AHLE late |
|---|---|---|---|---|
| AWA dataset | 48.5 | 40.4 | 46.8 | **49.4** |
| CUB dataset | 26.9 | 18.5 | 27.1 | **27.3** |

# Side information in few-shots

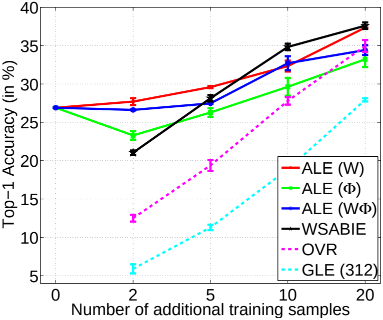
5. Is side information beneficial for few-shots learning?

# Side information in few-shots

5. Is side information beneficial for few-shots learning?



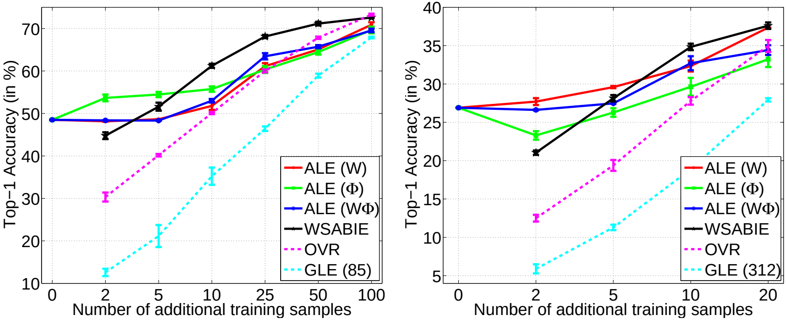

AWA (FV=64K)




CUB (FV=64K)

# Side information in few-shots

5. Is side information beneficial for few-shots learning?




AWA (FV=64K) CUB (FV=64K)

- Side information is beneficial with scarce training data
- All methods converge with more training data

# Advantages of ALE over DAP

1. Solves directly image classification problem
2. Accommodates other sources of side information
   - Improves zero-shot learning with continuous attributes
3. Leverages few additional labeled training data

**Label-Embedding with Attributes**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *IEEE CVPR, 2013.*

**Attribute-Based Classification with Label-Embedding**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *ORL Workshop at NIPS, 2013.*

**Label-Embedding with Attributes for Image Classification**
Z.Akata, F. Perronnin, Z.Harchaoui, C.Schmid, *Submitted to IEEE TPAMI.*

# Outline

# Large-scale image classification

**Conclusions**

- Comparison of objective functions in large-scale learning
- Set of good practices for large-scale learning

# Large-scale image classification

**Conclusions**

- Comparison of objective functions in large-scale learning
- Set of good practices for large-scale learning

**Future work**

- Hierarchical loss [Tsochantaridis *et al.*'05]
- ASGD [Polyak and Juditsky'92], [Bach and Moulines'13]
- Sampling [Loosli *et al.*'05], [Mineiro and Karampatziakis'13]

# Label-embedding with attributes

**Conclusions**

- Novel approach for zero-shot learning using attributes
- Several improvements over the state of the art

# Label-embedding with attributes

**Conclusions**

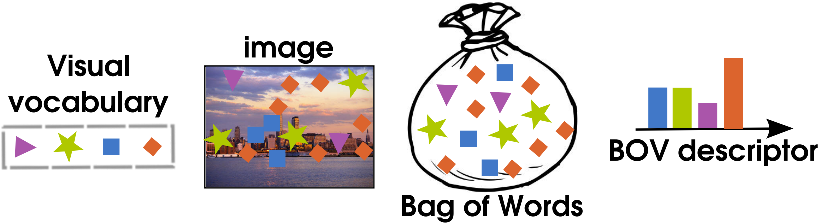
- Novel approach for zero-shot learning using attributes
- Several improvements over the state of the art

**Future work**

- Deep Embedding of ALE and HLE
- Beyond label trees in HLE [Yen *et al.*'08]
- Text from textual resources [Rohrbach'10], [Frome *et al.*'13]

Thank you for your attention!

# Bag of Visual Words (BOV) [Csurka *et al.*'04]




- Dense sampling of local features from an image
- Create visual vocabulary with k-means
- Assign each local feature to a visual word
- Calculate frequency of each visual word

# Fisher Vectors (FVs) [Perronnin and Dance'07]

- Dense sampling of local features from an image
- Create visual vocabulary with GMMs

$$p(\mathbf{x}|\lambda) \text{ with } \lambda = \{\pi_i, \boldsymbol{\mu}_i, \Sigma_i, i = 1, ..., k\}$$

- Take gradients w.r.t mixture weight, mean and variance

$$G_\lambda^{\mathbf{x}} = \frac{1}{N} \nabla_\lambda \log p(\mathbf{x}|\lambda)$$

- Improves with power, $\ell_2$ normalization and SPM
- PQ compressed FVs have small memory fooprint

# Multiclass SVM (MUL) [Crammer and Singer'04]

- Convex surrogate loss to $\Delta(y, \hat{y})$:

$$\hat{y}_i = \arg\max_y \mathbf{w}_y^T x_i$$

- Upper bound to misclassification loss:

$$L_{\mathsf{MUL}}(\mathbf{x}_i, y_i; \mathbf{w}) = \max_y \left\{\Delta(y_i, y) + \mathbf{w}_y^T x_i\right\} - \mathbf{w}_{y_i}^T x_i$$

# Ranking SVM (RNK) [Joachims'02]

- Ordering pairs of documents
- Sample $(\mathbf{x}_i, y_i)$ and label $y \neq y_i$: enforce $\mathbf{w}_{y_i}\mathbf{x}_i > \mathbf{w}_y^T\mathbf{x}_i$
- Rank of label $y$ for sample $\mathbf{x}$:

$$r(\mathbf{x}, y) = \sum_{c=1}^{C} \mathbb{1}(\mathbf{w}_c^T\mathbf{x} \geq \mathbf{w}_y^T\mathbf{x})$$

- $\mathbb{1}(\mathbf{w}_c^T\mathbf{x} \geq \mathbf{w}_y^T\mathbf{x})$ is upper-bounded by:

$$L_{\text{tri}}(\mathbf{x}_i, y_i, y; \mathbf{w}) = \max\{0, \Delta(y_i, y) - \mathbf{w}_{y_i}^T\mathbf{x}_i + \mathbf{w}_y^T\mathbf{x}_i\}$$

- Overall loss of $(\mathbf{x}_i, y_i)$:

$$L_{\text{RNK}}(\mathbf{x}_i, y_i; \mathbf{w}) = \sum_{y=1}^{C} \max\{0, \Delta(y_i, y) - (\mathbf{w}_{y_i} - \mathbf{w}_y)^T\mathbf{x}_i\}$$

# Weighted Average Ranking (WAR) [Weston *et al.*'10]

- Give more weight to the top of the ranking list
- Ranking loss $\ell_{r(\mathbf{x}_i,y_i)}$: $\ell_k = \sum_{j=1}^{k} \frac{1}{j}$
- Regularized rank:

$$r_\Delta(\mathbf{x}, y) = \sum_{c=1}^{C} \mathbb{1}(\mathbf{w}_c^T x + \Delta(y, c) \geq \mathbf{w}_y^T x)$$

- Approximated upper bound to the loss:

$$L_{\text{WAR}}(\mathbf{x}_i, y_i; \mathbf{w}) = \sum_{y=1}^{C} \ell_{r_\Delta(\mathbf{x}_i,y_i)} \frac{L_{\text{tri}}(\mathbf{x}_i,y_i,y;\mathbf{w})}{r_\Delta(\mathbf{x}_i,y_i)}$$

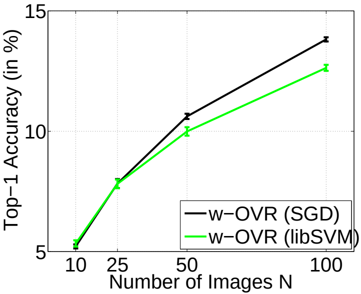
# Sampling and update equations

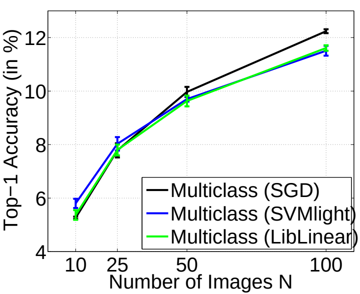
| | Sampling | Update |
|---|---|---|
| $R_{\text{OVR}}$ | Draw $(\mathbf{x}_i, y_i)$ from $S$. | $\delta_i = 1$ if $L_{\text{OVR}}(\mathbf{x}_i, y_i; \mathbf{w}) > 0, 0$ otherwise.<br>$\mathbf{w}^{(t)} = (1 - \eta_t \lambda)\mathbf{w}^{(t-1)} + \eta_t \delta_i \mathbf{x}_i y_i$ |
| $R_{\text{MUL}}$ | Draw $(\mathbf{x}_i, y_i)$ from $S$. | $\bar{y} = \arg\max_y \Delta(y_i, y) + \mathbf{w}_y' \mathbf{x}_i$ and $\delta_i = \begin{cases} 1 & \text{if } \bar{y} \neq y_i \\ 0 & \text{otherwise.} \end{cases}$<br>$\mathbf{w}_y^{(t)} = \begin{cases} \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) + \delta_i \eta_t \mathbf{x}_i & \text{if } y = y_i \\ \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) - \delta_i \eta_t \mathbf{x}_i & \text{if } y = \bar{y} \\ \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) & \text{otherwise.} \end{cases}$ |
| $R_{\text{RNK}}$ | Draw $(\mathbf{x}_i, y_i)$ from $S$.<br>Draw $\bar{y} \neq y_i$ from $\mathcal{Y}$. | $\delta_i = 1$ if $L_{\text{tri}}(\mathbf{x}_i, y_i, \bar{y}; \mathbf{w}) > 0, 0$ otherwise.<br>$\mathbf{w}_y^{(t)} = \begin{cases} \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) + \delta_i \eta_t \mathbf{x}_i & \text{if } y = y_i \\ \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) - \delta_i \eta_t \mathbf{x}_i & \text{if } y = \bar{y} \\ \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) & \text{otherwise.} \end{cases}$ |
| $R_{\text{WAR}}$ | Draw $(\mathbf{x}_i, y_i)$ from $S$.<br>For $k = 1, 2, \ldots, C-1$ do:<br>$\begin{cases} \text{Draw } \bar{y} \neq y_i \text{ from } \mathcal{Y}. \\ \text{If } L_{\text{tri}}(\mathbf{x}_i, y_i, \bar{y}; \mathbf{w}) > 0, \text{ break.} \end{cases}$ | $\delta_i = 1$ if $\bar{y}$ s.t. $L_{\text{tri}}(\mathbf{x}_i, y_i, \bar{y}; \mathbf{w}) > 0$ was sampled, $0$ otherwise.<br>$\mathbf{w}_y^{(t)} = \begin{cases} \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) + \delta_i \ell_{\lfloor \frac{C-1}{k} \rfloor} \eta_t \mathbf{x}_i & \text{if } y = y_i \\ \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) - \delta_i \ell_{\lfloor \frac{C-1}{k} \rfloor} \eta_t \mathbf{x}_i & \text{if } y = \bar{y} \\ \mathbf{w}_y^{(t-1)}(1 - \eta_t \lambda) & \text{otherwise.} \end{cases}$ |

# SGD vs Batch

- Is SGD better than Batch in large scale classification?

SGD vs batch experiments on Ungulate183




| N | LibSVM / OVR SGD (sec) | $SVM^{light}$ / MUL SGD (sec) |
|---|---|---|
| 10 | 31 / 18 | 324 / 81 |
| 25 | 175 / 36 | 441 / 198 |

# Comparison between methods on ILSVRC10

| | | u-OVR | w-OVR | MUL | RNK | WAR |
|---|---|---|---|---|---|---|
| Top-1 | BOV 4K | 15.8 | 26.4 | 22.7 | 20.8 | 24.1 |
| | FV 130K | 45.9 | 45.7 | 46.2 | 46.1 | 46.1 |
| Top-5 | BOV 4K | 28.8 | 46.4 | 38.4 | 41.2 | 44.2 |
| | FV 130K | 63.7 | 65.9 | 64.8 | 65.8 | 66.5 |

- Despite its simplicity and suboptimality in theory, OVR performs the best

# Attribute Label Embedding

- $\mathcal{S} = \{(x_n, y_n), n = 1 \ldots N\}$: $x_n \in \mathcal{X}$ and $y_n \in \mathcal{Y}$
- Learn $f : \mathcal{X} \to \mathcal{Y}$ with $\frac{1}{N}\sum_{n=1}^{N} \Delta(y_n, f(x_n))$
- 0/1 loss: $\Delta(y, z) = 0$ if $y = z$, 1 otherwise
- Compatibility function: $f(x; w) = \arg\max_{y \in \mathcal{Y}} F(x, y; w)$
- Rewrite in bilinear form: $F(x, y; W) = \theta(x)' W \varphi(y)$
- Attribute Label-Embedding with Attributes (ALE):
  - $\mathcal{Y} = \{1, \ldots, C\}$, $\mathcal{A} = \{a_i, i = 1 \ldots E\}$
  - association measure between $y$ and $a_i$: $\rho_{y,i}$
  - embed class $y$ in attribute space:

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

# Zero-Shot Objective

- $\Phi$ fixed, $W$ learned

$$\frac{1}{N}\sum_{n=1}^{N} \max_{y \in \mathcal{Y}} \ell(x_n, y_n, y)$$

- where $\ell(x_n, y_n, y)$ is defined as:

$$\Delta(y_n, y) + \theta(x)' W[\varphi(y) - \varphi(y_n)]$$

# Few-Shots Objective

- $\Phi$ and $W$ learned using $\Phi^{\mathcal{A}}$

$$R(\mathcal{S}; W, \Phi) + \frac{\mu}{2}||\Phi - \Phi^{\mathcal{A}}||^2$$

- where $R(\mathcal{S}; W, \Phi)$ is defined as:

$$\frac{1}{N}\sum_{n=1}^{N} \frac{\beta_{r_\Delta(x_n, y_n)}}{r_\Delta(x_n, y_n)} \sum_{y \in \mathcal{Y}} \max\{0, \ell(x_n, y_n, y)\}$$

- upper-bound on rank of label $y_n$ for image $x_n$:

$$r_\Delta(x_n, y_n) = \sum_{y \in \mathcal{Y}} \mathbb{1}(\ell(x_n, y_n, y) > 0)$$

# SGD optimization for ALE

- Intitialize $W^{(0)}$ randomly.
- Draw ($x$,$y$) randomly from $\mathcal{S}$
- Draw $\bar{y} \neq y$ from $\mathcal{Y}$
- If $\ell(x, y, \bar{y}) > 0$
  - Update $W$

$$W^{(t)} \quad = \quad W^{(t-1)} + \eta_t \beta_{\lfloor \frac{C-1}{k} \rfloor} \theta(x)[\varphi(y) - \varphi(\bar{y})]'$$

  - Update $\Phi$ (not applicable to zero-shot)

$$\varphi^{(t)}(y) = (1 - \eta_t \mu)\varphi^{(t-1)}(y) + \eta_t \mu \varphi^{\mathcal{A}}(y) + \eta_t \beta_{\lfloor \frac{C-1}{k} \rfloor} W' \theta(x)$$

$$\varphi^{(t)}(\bar{y}) = (1 - \eta_t \mu)\varphi^{(t-1)}(\bar{y}) + \eta_t \mu \varphi^{\mathcal{A}}(\bar{y}) - \eta_t \beta_{\lfloor \frac{C-1}{k} \rfloor} W' \theta(x)$$

# Attribute prediction

- Are the attributes still interpretable for ALE?

$\theta(x)'W$ can be interpreted as a vector of attribute scores of $\mathbf{x}$

| | Attribute prediction | |
|---|---|---|
| | DAP | ALE |
| AWA | **72.7** | **72.7** |
| CUB | **64.8** | 59.4 |

Attribute interpretability:

is quadrapedal

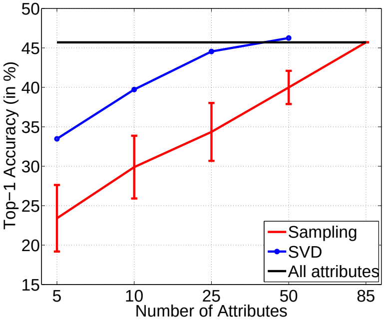
lives in ocean

hibernates

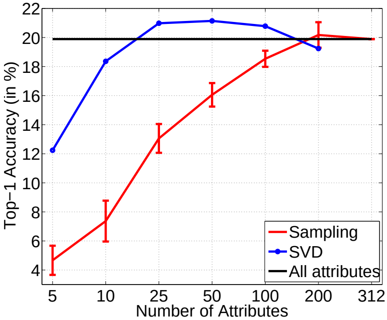
is weak

# Attribute Correlation

- Are the attributes correlated for zero-shot learning?




AWA (FV=4K)




CUB (FV=4K)

- SVD vs random attribute sampling
- Significant correlation in output space