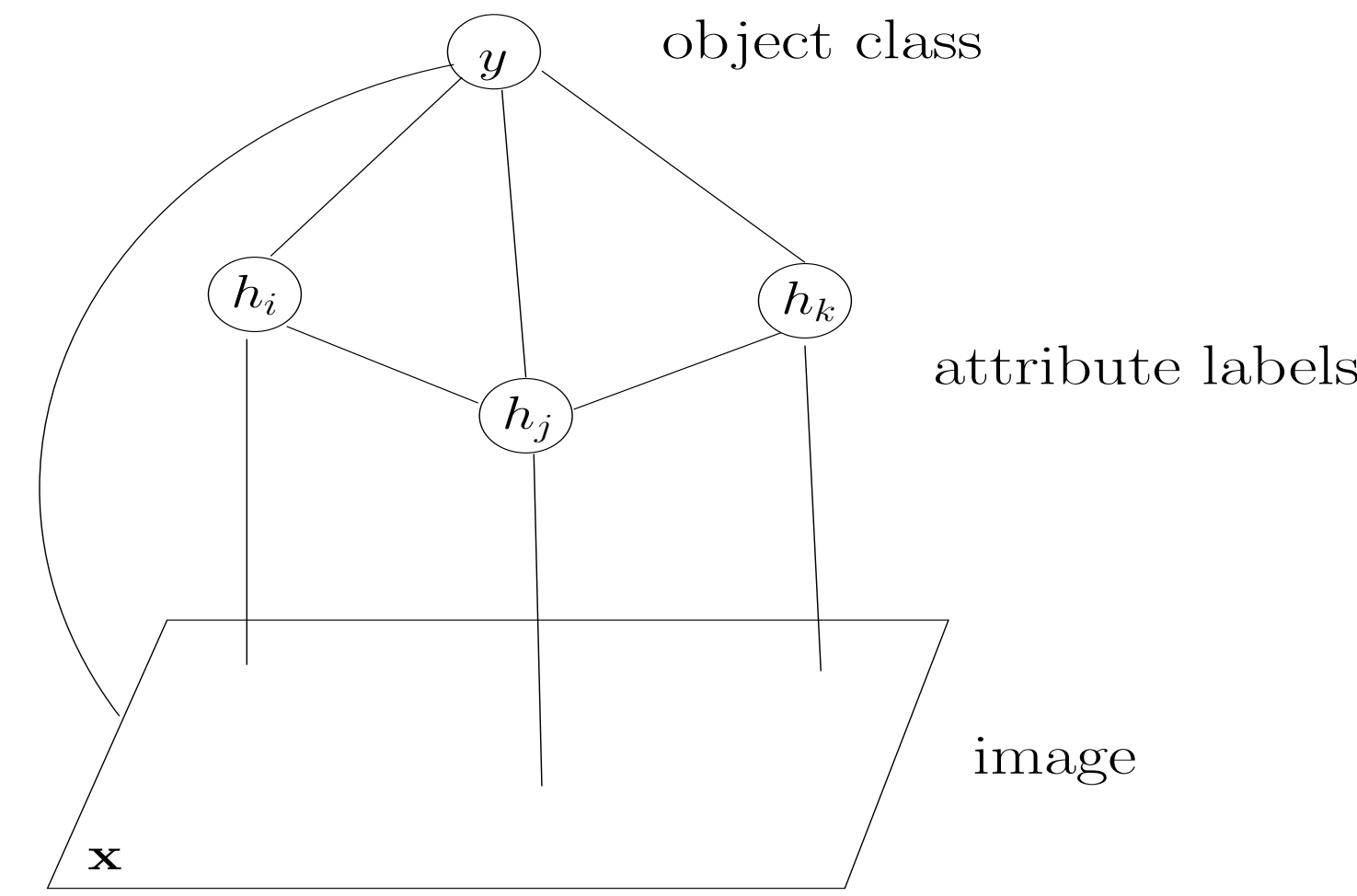


Overview

- Joint modeling of object classes and (correlated) attributes
- Demonstrating attributes can help object recognition
- A general learning framework for classification with auxiliary labels

Why attributes alone are not enough?

Model Formulation



Training data: $\{(\mathbf{x}^{(n)}, \mathbf{h}^{(n)}, y^{(n)})\}_{n=1}^N$

Scoring function: $f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y)$,
where

$$\begin{aligned}
 \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y) = & \mathbf{w}_y^T \phi(\mathbf{x}) \quad \text{object class model} \\
 & + \sum_{j \in \mathcal{V}} \mathbf{w}_{h_j}^T \varphi(\mathbf{x}) \quad \text{global attribute model} \\
 & + \sum_{j \in \mathcal{V}} \mathbf{w}_{y, h_j}^T \omega(\mathbf{x}) \quad \text{class-specific attribute model} \\
 & + \sum_{(j, k) \in \mathcal{E}} \mathbf{w}_{j, k}^T \psi(h_j, h_k) \quad \text{attribute-attribute interaction} \\
 & + \sum_{j \in \mathcal{V}} v_{y, h_j} \quad \text{object-attribute interaction}
 \end{aligned}$$

Learning and Inference

Inference: tractable if attributes \mathbf{h} have special structures (e.g. trees)

$$\mathbf{h}^* = \arg \max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{h}, y) \quad \forall y \in \mathcal{Y}$$

Learning with latent attributes:

$$\begin{aligned}
 \min_{\mathbf{w}, \xi} & \beta \|\mathbf{w}\|^2 + \sum_{n=1}^N \xi^{(n)} \\
 \text{s.t.} & \max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y^{(n)}) - \max_{\mathbf{h}} \mathbf{w}^T \Phi(\mathbf{x}^{(n)}, \mathbf{h}, y) \\
 & \geq \Delta(y, y^{(n)}) - \xi^{(n)}, \forall n, \forall y
 \end{aligned}$$

Another choice is to use the ground-truth attribute labels \mathbf{h}^n (i.e. learning with observed attributes).

Attribute Relation Graph

Running minimum spanning tree with $\text{NormMI}(j, k)$ as the weight on the edge (j, k) .

Other Loss Functions

A simple modification of Δ will optimize different (training) errors.

Overall accuracy:

$$\Delta_{0/1}(y, y^{(n)}) = \begin{cases} 1 & \text{if } y \neq y^{(n)} \\ 0 & \text{otherwise} \end{cases}$$

Mean per-class accuracy:

$$\Delta_{\text{new}}(y, y^{(n)}) = \begin{cases} \frac{1}{m_p} & \text{if } y \neq y^{(n)} \text{ and } y^{(n)} = p \\ 0 & \text{otherwise} \end{cases}$$

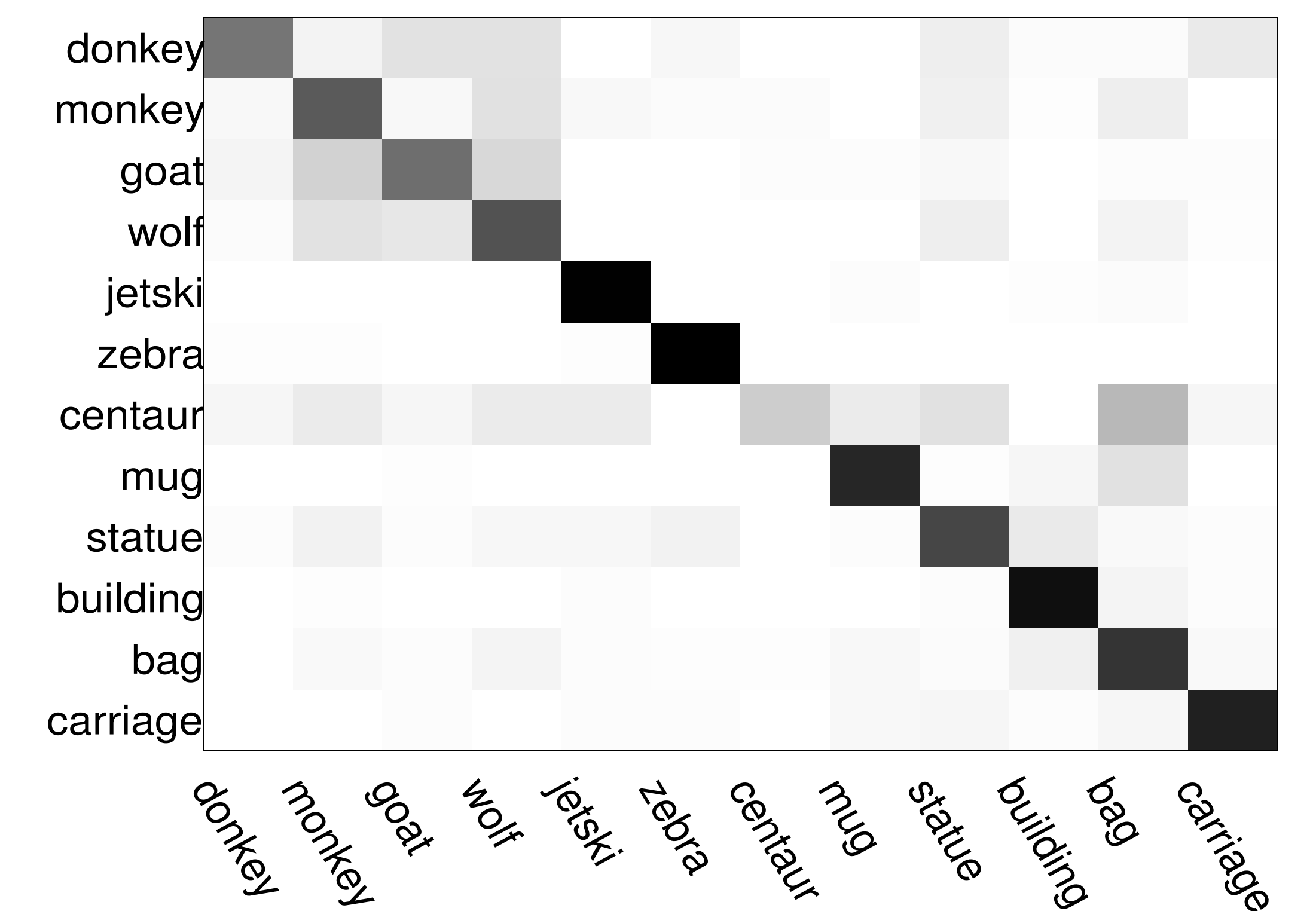
where m_p is the number of training examples with class p .

Experiments

a-Pascal dataset:

method	overall	mean per-class
Our approach with $\Delta_{0/1}$	62.16	46.25
Our approach with Δ_{new}	59.15	50.84
SVM with $\Delta_{0/1}$	58.77	38.52
SVM with Δ_{new}	53.74	44.04
Farhadi et al. CVPR09 (base features+SVM)	58.5	34.3
Farhadi et al. CVPR09 (best result)	59.4	37.7

a-Yahoo dataset:



method	overall	mean per-class
Our approach with $\Delta_{0/1}$	78.67	71.45
Our approach with Δ_{new}	79.88	73.31
SVM with $\Delta_{0/1}$	74.43	65.96
SVM with Δ_{new}	74.51	66.74