Simultaneous Learning and Alignment: Multi-Instance and Multi-Pose Learning

Boris Babenko[1] Piotr Dollár[2] Zhuowen Tu[3] Serge Belongie[1,2]

[1] Computer Science and Engineering
University of California, San Diego
{bbabenko,sjb}@cs.ucsd.edu

[2] Electrical Engineering,
California Institute of Technology
pdollar@caltech.edu

[3] Lab of Neuro Imaging
University of California, Los Angeles
Zhuowen.tu@loni.ucla.edu

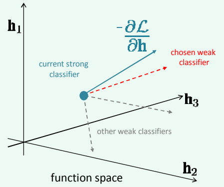
Abstract

In object recognition in general and in face detection in particular, data alignment is necessary to achieve good classification results with certain statistical learning approaches such as Viola-Jones. Data can be aligned in one of two ways: (1) by separating the data into coherent groups and training separate classifiers for each; (2) by adjusting training samples so they lie in correspondence. If done manually, both procedures are labor intensive and can significantly add to the cost of labeling. In this paper we present a unified boosting framework for simultaneous learning and alignment. We present a novel boosting algorithm for Multiple Pose Learning (mpl), where the goal is to simultaneously split data into groups and train classifiers for each. We also review Multiple Instance Learning (mil), and in particular mil-boost, and describe how to use it to simultaneously train a classifier and bring data into correspondence. We show results on variations of LFW and MNIST, demonstrating the potential of these approaches.

Gradient Boosting Review

- Boosting trains a strong classifier of the form $\mathbf{h}(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$
- For a given loss function $\mathcal{L}$, perform gradient descent in function space. Each step is one weak classifier.
- Let $\mathbf{h}_i = \mathbf{h}(x_i)$, and $w_i = -\frac{\partial \mathcal{L}}{\partial \mathbf{h}_i}$.
- At step t we want a weak classifier that is close to the gradient:

$$h_t = \mathrm{argmax}_h \sum_{i=1}^{n} w_i h(x_i)$$



- Log likelihood is a standard loss function that we will use:

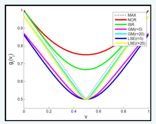
$$\mathcal{L}(\mathbf{h}) = -\sum_{i=1}^{n} \Big(\mathbf{1}(y_i = 1) \log p_i + \mathbf{1}(y_i = -1) \log(1 - p_i) \Big)$$

where $p_i \equiv \mathbb{P}(y_i = 1 | x_i)$.

Approximating Max

		$g_\ell(v_\ell)$	$\partial g_\ell(v_\ell)/\partial v_i$	domain
(Boyd *et al.*)	LSE	$\frac{1}{r} \ln\left(\frac{1}{m}\sum_\ell \exp(r v_\ell)\right)$	$\frac{\exp(r v_i)}{\sum_\ell \exp(r v_\ell)}$	$[-\infty, \infty]$
	GM	$\left(\frac{1}{m}\sum_\ell v_\ell^r\right)^{\frac{1}{r}}$	$g_\ell(v_\ell) \frac{v_i^{r-1}}{\sum_\ell v_\ell^r}$	$[0, \infty]$
(Viola *et al.*)	NOR	$1 - \prod_\ell (1 - v_\ell)$	$\frac{1-g_\ell(v_\ell)}{1-v_i}$	$[0, 1]$
(Keeler *et al.* , Viola *et al.*)	ISR	$\frac{\sum_\ell v'_\ell}{1+\sum_\ell v'_\ell}, v'_\ell = \frac{v_\ell}{1-v_\ell}$	$\left(\frac{1-g_\ell(v_\ell)}{1-v_i}\right)^2$	$[0, 1]$

Table 1: Four softmax approximations $g_\ell(v_\ell) \approx \max_\ell(v_\ell)$.



Multiple Instance Learning (MIL)

Overview

- Goal: simultaneously train classifier and bring data into correspondence.
- Training data consists of:
  bags: $X_i = \{x_{i1}, \ldots, x_{im}\}$
  bag labels: $\{y_1, \ldots, y_n\}$
- Bag labels defined as $y_i = \max_j \{y_{ij}\}$ but instance labels are unknown during training (latent variables).

MIL-BOOST

[Re-derivation of Viola *et al.* 2005]

- Define bag probability as a softmax of instance probabilities:

$$p_i = g_j(p_{ij}) = g_j(\sigma(2\mathbf{h}_{ij}))$$

- Derivative of the loss function gives us the instance weights:

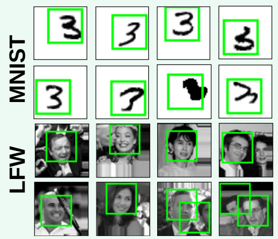
$$\frac{\partial \mathcal{L}}{\partial \mathbf{h}_{ij}} = \frac{\partial \mathcal{L}}{\partial p_i} \frac{\partial p_i}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial \mathbf{h}_{ij}}$$

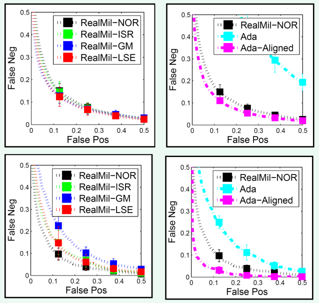
- Algorithm Summary:

```
Input: Dataset {X_1,...,X_n}, {y_1,...,y_n}, y_i ∈ {−1,1}
1: for t = 1 to T do
2:    Compute weights w_ij = −∂L/∂h_ij
3:    Train weak classifier h_t using weights |w_ij|
        h_t = argmin_h Σ_ij 1(h(x_ij) ≠ y_i)|w_ij|
4:    Find α_t via line search to minimize L(h)
        α_t = argmin_α L(h + αh_t)
5:    Update strong classifier h ← h + α_t h_t.
6: end for
```

Experiments

MILBoost Detections:





Multiple Pose Learning (MPL)

Overview

- Goal: simultaneously group the positive data, and train classifiers $h^1, \ldots, h^K$ for each of the K groups.
- Training data given in standard form.
- Instance labels are defined as $y_i = \max_j \{y_i^k\}$ where y_i^k is a latent (unknown) variable, which is positive if example i is in group k.

MPL-BOOST

- Define probability as a softmax of probabilities determined by each classifier:

$$p_i = g_k(p_i^k) = g_k(\sigma(2\mathbf{h}_i^k))$$

- Derivative of the loss function gives us the instance weights for each classifier:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{h}_i^k} = \frac{\partial \mathcal{L}}{\partial p_i} \frac{\partial p_i}{\partial p_i^k} \frac{\partial p_i^k}{\partial \mathbf{h}_i^k}$$

- Algorithm Summary:

```
Input: Dataset {x_1,...,x_n}, {y_1,...,y_n}, y_i ∈ {−1,1}, K
1: for t = 1 to T do
2:    for k = 1 to K do
3:       Compute weights w_i^k = −∂L/∂h_i^k
4:       Train weak classifier h_t^k using weights |w_i^k|
           h_t^k = argmin_h Σ_i 1(h(x_i) ≠ y_i)|w_i^k|
5:       Find α_t via line search to minimize L(·, h^k, ·)
           α_t = argmin_α L(·, h^k + αh_t^k, ·)
6:       Update strong classifier h^k ← h^k + α_t^k h_t^k.
7:    end for
8: end for
```

Experiments