Discriminatively Activated Sparselets

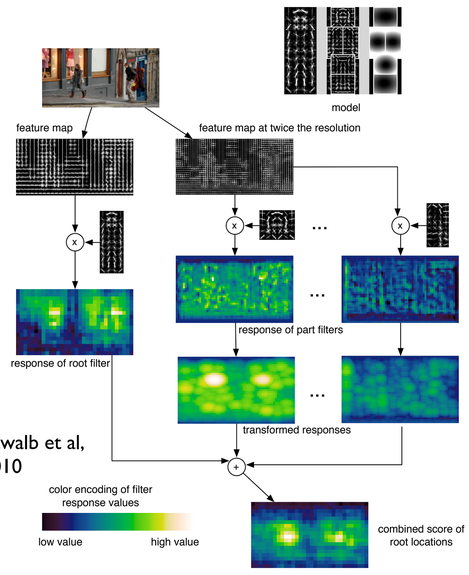
Ross Girshick*, **Hyun Oh Song***, Trevor Darrell

UC Berkeley, *indicates equal contribution

Goal

- Shared predictive model with sparse activation vectors
- Efficient inference for linear structured output predictors
- Example application: realtime object recognition in CV, faster retrieval in IR, etc.

Object recognition



Felzenszwalb et al, PAMI 2010

Sparselet review

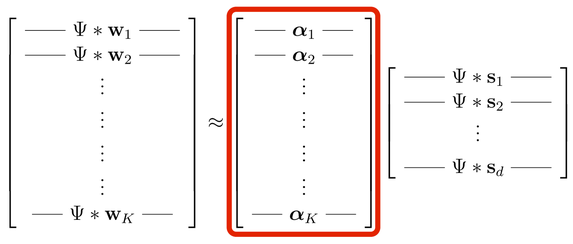
Set of model filters $\mathcal{W} = \{\mathbf{w_1}, ..., \mathbf{w_K}\}$

Set of sparselet filters $\mathcal{S} = \{\mathbf{s_1}, ..., \mathbf{s_d}\}$

$$\min_{\alpha_{ij}, s_j} \sum_{i=1}^{K} ||\mathbf{w}_i - \sum_{j=1}^{d} \alpha_{ij}\mathbf{s}_j||_2^2$$

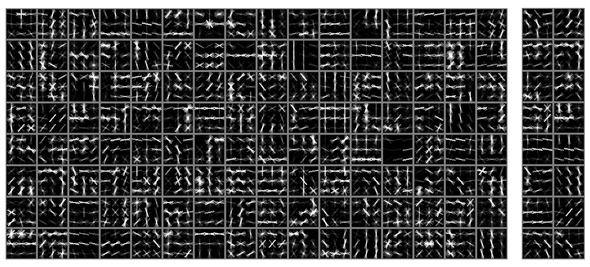
$$\text{subject to} \quad ||\boldsymbol{\alpha_i}||_0 \leq \epsilon \quad \forall i = 1, ..., K$$
$$||\mathbf{s}_j||_2^2 \leq 1 \quad \forall j = 1, ..., d$$

Matrix factorization point of view



80 ~ 99 % Sparse

Visualized sparselet blocks on HOG



(Left) Sparselet dictionary of size 128
(Right) Top 16 activated sparselets for PASCAL motorcycle class

Blocked representation

- Intuition: model weights might be composed of smaller building blocks/tiles
- For a matrix X with k number of elements, reshape the matrix such that the ratio between the full rank and actual rank of the reshaped matrix is maximized

$$\underset{m:\ \mathrm{mod}(k,m)=0}{\operatorname{argmax}} \frac{\min(m, k/m)}{\operatorname{rank}(\operatorname{reshape}(X, m, k/m))}$$

Blocked parameterization

$$f_{\mathbf{w}}(\mathbf{x}) = \underset{k \in \{1,...,K\}}{\operatorname{argmax}} \mathbf{w}_k^\intercal \mathbf{x}$$

Model parameterization $\mathbf{w}_k = (\mathbf{b}_{k1}^\intercal, \ldots, \mathbf{b}_{kp}^\intercal)^\intercal$

Data parameterization $\mathbf{x} = (\mathbf{c}_1^\intercal, \ldots, \mathbf{c}_p^\intercal)^\intercal$

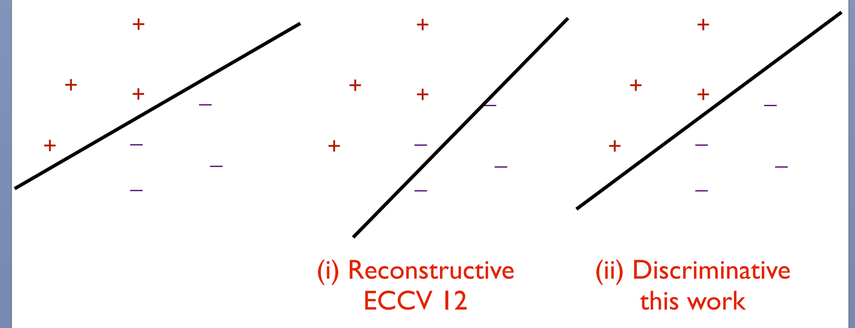
Sparselets approximation of model blocks

$$\mathbf{b} \approx \mathbf{S}\boldsymbol{\alpha} = \sum_{\substack{i=1 \\ \alpha_i \neq 0}}^{d} \alpha_i \mathbf{s}_i$$

Sparselets: $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_d]$

Discriminative sparselet activation

Original w_k — Sparselet approximation w_k



(i) Reconstructive ECCV 12

(ii) Discriminative this work

Learning parameterization

$$\mathbf{w}_k^\intercal \mathbf{x} = (\mathbf{b}_{k1}^\intercal, \ldots, \mathbf{b}_{kp}^\intercal)(\mathbf{c}_1^\intercal, \ldots, \mathbf{c}_p^\intercal)^\intercal$$
$$= \sum_{i=1}^{p} \mathbf{b}_{ki}^\intercal \mathbf{c}_i \approx \sum_{i=1}^{p} (\mathbf{S}\boldsymbol{\alpha}_{ki})^\intercal \mathbf{c}_i = \sum_{i=1}^{p} \boldsymbol{\alpha}_{ki}^\intercal (\mathbf{S}^\intercal \mathbf{c}_i)$$

Model parameter: sparse activation vector

Feature: sparselet response

Structural SVM for DAS

Parameter vector $\boldsymbol{\beta} = (\boldsymbol{\alpha}^\intercal, \tilde{\mathbf{w}}^\intercal)^\intercal$

Transformed features $\tilde{\boldsymbol{\Phi}}_k(x, y) = \left(\mathbf{c}_1^\intercal S, \ldots, \mathbf{c}_{p_k}^\intercal S\right)^\intercal$

Aggregate feature vector

$$\tilde{\boldsymbol{\Phi}}(x, y) = (\tilde{\boldsymbol{\Phi}}_1^\intercal(x, y), \ldots, \tilde{\boldsymbol{\Phi}}_s^\intercal(x, y), \boldsymbol{\Phi}_{s+1}^\intercal(x, y), \ldots, \boldsymbol{\Phi}_K^\intercal(x, y))^\intercal$$

projected feature slot — remainder feature slot

Training

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} R(\boldsymbol{\alpha}) + \frac{\lambda}{2}\|\tilde{\mathbf{w}}\|_2^2 + \frac{1}{M}\sum_{i=1}^{M} \max_{\hat{y} \in \mathcal{Y}} \left(\boldsymbol{\beta}^\intercal \tilde{\boldsymbol{\Phi}}(x_i, \hat{y}) + \Delta(y_i, \hat{y})\right) - \boldsymbol{\beta}^\intercal \tilde{\boldsymbol{\Phi}}(x_i, y_i)$$

Sparsity inducing norm

Sparsity enforcing norms

I. **Lasso penalty** $R_{\text{Lasso}}(\boldsymbol{\alpha}) = \lambda_1 \|\boldsymbol{\alpha}\|_1$

II. **Elastic net penalty** $R_{\text{EN}}(\boldsymbol{\alpha}) = \lambda_1 \|\boldsymbol{\alpha}\|_1 + \lambda_2 \|\boldsymbol{\alpha}\|_2^2$

III. **Combined ℓ_0 and ℓ_2 penalty** $R_{0,2}(\boldsymbol{\alpha}) = \lambda_2 \|\boldsymbol{\alpha}\|_2^2$ subject to $\|\boldsymbol{\alpha}\|_0 \leq \lambda_0$

III-A. **Overshoot, rank, and threshold (ORT)**

III-B. **Orthogonal matching pursuit (OMP)**

Joint feature map: multiclass classification

$$\mathbf{w} = (\mathbf{w}_1^\intercal, \ldots, \mathbf{w}_K^\intercal)^\intercal$$
$$\boldsymbol{\Phi}(\mathbf{x}, k) = (0, \ldots, 0, \mathbf{x}^\intercal, 0, \ldots, 0)^\intercal$$

feature installed in slot k

class index

feature

Inference $f_{\mathbf{w}}(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\ \mathbf{w}^\intercal \boldsymbol{\Phi}(\mathbf{x}, k)$

$$\boldsymbol{\beta} = (\boldsymbol{\alpha}_1^\intercal, \ldots, \boldsymbol{\alpha}_K^\intercal, \tilde{\mathbf{w}}_1^\intercal, \ldots, \tilde{\mathbf{w}}_K^\intercal)^\intercal$$
$$\tilde{\boldsymbol{\Phi}}(\mathbf{x}, k) = \left(0, \ldots, 0, \left(\mathbf{c}_1^\intercal S, \ldots, \mathbf{c}_{p_k}^\intercal S\right)^\intercal, 0, \ldots, 0, 0, \ldots, 0, 1, 0, \ldots, 0\right)^\intercal$$

per-class bias

projected feature blocks installed in slot k

class index

feature

Inference $f_{\boldsymbol{\beta}}(\mathbf{x}) = \underset{k}{\operatorname{argmax}}\ \boldsymbol{\beta}^\intercal \tilde{\boldsymbol{\Phi}}(\mathbf{x}, k)$

Joint feature map: object detection

$$\mathbf{w} = (\mathbf{w}_1^\intercal, \ldots, \mathbf{w}_K^\intercal)^\intercal$$
$$\boldsymbol{\Phi}(\mathbf{x}, (k, y)) = \left(0, \ldots, 0, \mathbf{x}_{y:n}^\intercal, 0, \ldots, 0\right)^\intercal$$

length n window at position y in slot k

position in the pyramid

class index

feature pyramid

Inference $f_{\mathbf{w}}(\mathbf{x}) = \underset{k,y}{\operatorname{argmax}}\ \mathbf{w}^\intercal \boldsymbol{\Phi}(\mathbf{x}, (k, y))$

$$\boldsymbol{\beta} = (\boldsymbol{\alpha}_1^\intercal, \ldots, \boldsymbol{\alpha}_K^\intercal, \tilde{\mathbf{w}}_1^\intercal, \ldots, \tilde{\mathbf{w}}_K^\intercal)^\intercal$$
$$\tilde{\boldsymbol{\Phi}}(\mathbf{x}, (k, y)) = \left(0, \ldots, 0, \left(\mathbf{c}_{y,1}^\intercal S, \ldots, \mathbf{c}_{y,p_k}^\intercal S\right)^\intercal, 0, \ldots, 0, 0, \ldots, 0, \mathbf{1}, 0, \ldots, 0\right)^\intercal$$

per-class bias

projected feature window from position y installed in slot k

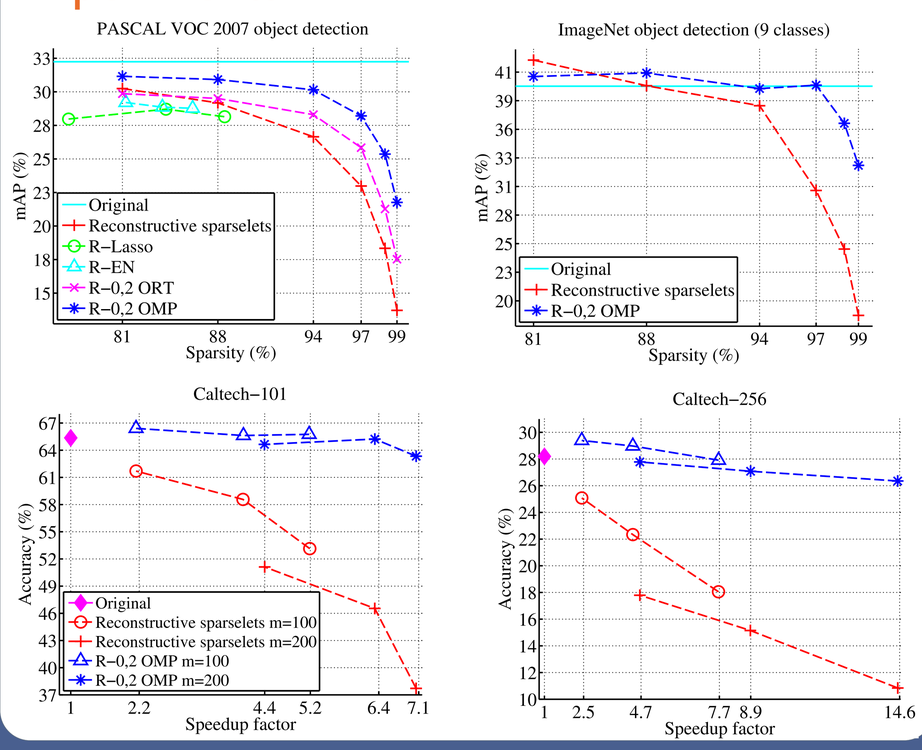
position in the pyramid

class index

feature pyramid

Inference $f_{\boldsymbol{\beta}}(\mathbf{x}) = \underset{k,y}{\operatorname{argmax}}\ \boldsymbol{\beta}^\intercal \tilde{\boldsymbol{\Phi}}(\mathbf{x}, (k, y))$

Experiments



Conclusion

- Training activation vectors discriminatively significantly outperforms reconstructive training
- Generalized framework for training activation vectors discriminatively using SSVM with sparsity constraints