DISCRIMINATIVE DECORELATION FOR CLUSTERING AND CLASSIFICATION

ECCV 12

Bharath Hariharan, Jitandra Malik, and Deva Ramanan
MOTIVATION

- State-of-the-art Object Detection
  - HOG
  - Linear SVM

*Dalal & Triggs Histograms of Oriented Gradients for Human Detection* CVPR05
MOTIVATION

- State of the art Object Detection
  - Mixture models
  - Parts Based Models
- Train one linear for each class/mixture/part
  - Hundreds of objects of interest
  - Hard negative mining
  - Increasing number of mixtures (exemplar SVMs)
  - Increasing number of parts

Can we make the training fast (closed form, no iteration) without losing much in performance?

P. Ott and M. Everingham. "Shared Parts for Deformable Part-based Models" CVPR11
Lubomir Bourdev, et al., "Detecting People Using Mutually Consistent Poselet Activations" ECCV10
MOTIVATION

► What does linear SVMs do?
► Discriminative Feature selection!
► Can we do it closed form?
► Seems possible with LDA
CONTENTS

► Motivation
► LDA
► Closed form solution
► Parameters reduction/Regularization
► Properties of the model
► Clustering
► Results
LINEAR DISCRIMINANT ANALYSIS (LDA) – BINARY CASE

- Training set \((X,Y)\)
- Assumes normal \(N(x; \mu_y, \Sigma)\) conditional distributions \(P(x | y=0)\) and \(P(x | y=1)\)
- Assumes equal class covariances
- From log of likelihood ratios give
  - \(w = \Sigma^{-1}(\mu_1 - \mu_0)\)
  - Dot product decision function: \(f(x) = w \cdot x < c\)
CLOSED FORM SOLUTION

- Calculate *object independent* $\mu_c$ and $\Sigma$
  - Assume we want the same $\Sigma$ across different classes
    - MLE of $\Sigma$ becomes the covariance computed over all training samples
  - Let's assume number of negative samples are very much larger than positives of each class
    - $\mu_c$ can be computed once on all training samples including positives

- Even so, Number of parameters in a HOG template can be very large making estimation of $\Sigma$ very hard
REGULARIZATION

► Translation and scale invariance
  ► Take all the negative windows from each scale and position

► Stationary process
  ► Mean will be computed for all HOG cells the same
  ► Covariance will only model relative offsets
  ► $N_0d$ parameters only

► Covariance only on different cells offsets

► Still low rank and not invertible
  ► Add a small amount to the diagonal (0.01)
PROPERTIES OF THE COVARIANCE

- Covariance of 9 orientation dimensions of one cell to 4 horizontal neighbors
  - Dark is negative and light is positive
  - Precision matrix is sparse
  - Fades after a few cells away

- Structure of $\Sigma$
  - encodes spatial correlations between oriented gradients

- Sparsity
PROPERTIES OF THE COVARIANCE

- Structure of $\Sigma$
  - encodes spatial correlations between oriented gradients

- Sparsity
  - Covariance of 9 orientation dimensions of one cell to 4 horizontal neighbors
  - Dark is negative and light is positive
  - Precision matrix is sparse
  - Fades after a few cells away
PROPERTIES OF THE COVARIANCE

- Transformed (Whitened) HOG Feature (WHO)
  - $x' = \Sigma^{-\frac{1}{2}}(x - \mu_0)$
  - More meaningful distances

- Comparison to PCA
  - PCA keeps the dimensions with most variance among training samples
    - Removes the discriminative information!
RESULTS – PEDESTRIAN DETECTION

- Comparing to Dalal Triggs HOG-SVM
  - With hard negative mining
- Method
  - Averaging: Compute positive average HOG ($\mu_1$)
  - Centering: Subtract $\mu_0$
  - Whitening: Multiply by $\Sigma^{-1}$
- To check the whitening effect use the centered version only
- Dalal&Triggs: 79.66%
- LDA: 75.10%
- Centered: 8%
- Pedestrians are well aligned
CLUSTERING

- Use transformed feature
- Recursive normalized cuts
WHO CLUSTERING RESULTS

(a) horse

(b) aeroplane
SOME MINOR RESULTS

- Average over all clusters performance
- Airplane, bus, horse
- Correlation of SVM and LDA
- Low training computation
- Still slow in testing phase

<table>
<thead>
<tr>
<th></th>
<th>LDA on cluster</th>
<th>SVM on cluster</th>
<th>LDA on medoid</th>
<th>SVM on medoid</th>
<th>Centered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean AP</td>
<td>7.59 ± 4.86</td>
<td>6.75 ± 4.80</td>
<td>4.84 ± 4.13</td>
<td>4.05 ± 4.12</td>
<td>0.74 ± 2.02</td>
</tr>
<tr>
<td>Median AP</td>
<td>9.25 ± 3.86</td>
<td>9.16 ± 4.04</td>
<td>4.65 ± 3.71</td>
<td>2 ± 3.6</td>
<td>0.06 ± 0.7</td>
</tr>
</tbody>
</table>

Table 1. Mean and median AP (in %) of the different models.
COMBINING ACROSS CLUSTERS

1. Use Exemplar SVMs method
2. Change the SVM to an LDA with object independent background model
3. To make it fast in test
   - Use clustering to train a classifier for each cluster
     - Train an LDA for each cluster
     - Produce Feature Vector 1: Dot product of a window WHO feature to all exemplars WHO
     - Produce Feature Vector 2: Dot product of a window WHO to all clusters WHO
     - Train a linear SVM on the concatenated feature
<table>
<thead>
<tr>
<th>Object</th>
<th>ESVM + Calibr</th>
<th>ESVM + Co-occ</th>
<th>ELDA + Calibr</th>
<th>Ours-only 1</th>
<th>Ours-only 2</th>
<th>Ours-full</th>
</tr>
</thead>
<tbody>
<tr>
<td>aeroplane</td>
<td>20.4</td>
<td>20.8</td>
<td>18.4</td>
<td>17.4</td>
<td>22.1</td>
<td>23.3</td>
</tr>
<tr>
<td>bicycle</td>
<td>40.7</td>
<td>48.0</td>
<td>39.9</td>
<td>35.5</td>
<td>37.4</td>
<td>41.0</td>
</tr>
<tr>
<td>bird</td>
<td>9.3</td>
<td>7.7</td>
<td>9.6</td>
<td>9.7</td>
<td>9.8</td>
<td>9.9</td>
</tr>
<tr>
<td>boat</td>
<td>10.0</td>
<td>14.3</td>
<td>10.0</td>
<td>10.9</td>
<td>11.1</td>
<td>11.0</td>
</tr>
<tr>
<td>bottle</td>
<td>10.3</td>
<td>13.1</td>
<td>11.3</td>
<td>15.4</td>
<td>14.0</td>
<td>17.0</td>
</tr>
<tr>
<td>bus</td>
<td>31.0</td>
<td>39.7</td>
<td>39.6</td>
<td>17.2</td>
<td>18.0</td>
<td>37.8</td>
</tr>
<tr>
<td>car</td>
<td>40.1</td>
<td>41.1</td>
<td>42.1</td>
<td>40.3</td>
<td>36.8</td>
<td>38.4</td>
</tr>
<tr>
<td>cat</td>
<td>9.6</td>
<td>5.2</td>
<td>10.7</td>
<td>10.6</td>
<td>6.5</td>
<td>11.5</td>
</tr>
<tr>
<td>chair</td>
<td>10.4</td>
<td>11.6</td>
<td>6.1</td>
<td>10.3</td>
<td>11.2</td>
<td>11.8</td>
</tr>
<tr>
<td>cow</td>
<td>14.7</td>
<td>18.6</td>
<td>12.1</td>
<td>14.3</td>
<td>13.5</td>
<td>14.5</td>
</tr>
<tr>
<td>diningtable</td>
<td>2.3</td>
<td>11.1</td>
<td>3</td>
<td>4.1</td>
<td>12.1</td>
<td>12.2</td>
</tr>
<tr>
<td>dog</td>
<td>9.7</td>
<td>3.1</td>
<td>10.6</td>
<td>1.8</td>
<td>10.5</td>
<td>10.2</td>
</tr>
<tr>
<td>horse</td>
<td>38.4</td>
<td>44.7</td>
<td>38.1</td>
<td>39.7</td>
<td>43.1</td>
<td>44.8</td>
</tr>
<tr>
<td>motorbike</td>
<td>32.0</td>
<td>39.4</td>
<td>30.7</td>
<td>26.0</td>
<td>25.8</td>
<td>27.9</td>
</tr>
<tr>
<td>person</td>
<td>19.2</td>
<td>16.9</td>
<td>18.2</td>
<td>23.1</td>
<td>21.3</td>
<td>22.4</td>
</tr>
<tr>
<td>pottedplant</td>
<td>9.6</td>
<td>11.2</td>
<td>1.4</td>
<td>4.9</td>
<td>5.1</td>
<td>3.1</td>
</tr>
<tr>
<td>sheep</td>
<td>16.7</td>
<td>22.6</td>
<td>12.2</td>
<td>14.1</td>
<td>13.8</td>
<td>16.3</td>
</tr>
<tr>
<td>sofa</td>
<td>11.0</td>
<td>17.0</td>
<td>11.1</td>
<td>8.7</td>
<td>12.2</td>
<td>8.9</td>
</tr>
<tr>
<td>train</td>
<td>29.1</td>
<td>36.9</td>
<td>27.6</td>
<td>22.1</td>
<td>30.6</td>
<td>30.3</td>
</tr>
<tr>
<td>tvmonitor</td>
<td>31.5</td>
<td>30.0</td>
<td>30.2</td>
<td>15.2</td>
<td>12.8</td>
<td>28.8</td>
</tr>
<tr>
<td>Mean</td>
<td>19.8</td>
<td>22.6</td>
<td>19.1</td>
<td>17.0</td>
<td>18.3</td>
<td>21.0</td>
</tr>
</tbody>
</table>
WHAT WE WANTED! TO DO

► Train exemplar SVMs for each sample
► Take the selected feature as the new transformed feature
  ► Do Image Classification (combined features and nonlinear SVMs)
  ► Clustering and Linear SVMs
  ► New similarity measure for verification (like face verification in LFW)
  ► Model Natural images with some distribution to make training Exemplar SVMs fast
  ► One can use more sophisticated modelling of natural patches (GMM)
► Make a pyramid of second, 3rd, … order discriminant features