Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms

Lifeng Bo¹, Xiaofeng Ren² and Dieter Fox¹
¹University of Washington, ²Intel Labs

This work
- Hierarchical matching pursuit builds a feature hierarchy layer-by-layer using an efficient matching pursuit encoder.
- Hierarchical Matching Pursuit (HMP)
  - Matching pursuit encoder consists of three modules: batch tree orthogonal matching pursuit, spatial pyramid max pooling, and contrast normalization;
  - Recursively run matching pursuit encoder;
  - Extract features from a typical 300×300 image in less than 1 second;
  - Outperform convolutional deep networks and SIFT based single layer sparse coding in terms of accuracy.

K-SVD (Dictionary Learning)
- K-SVD[^10] learns a dictionary D and an associated sparse code matrix X from observations Y by minimizing the following reconstruction error
  \[
  \min_{D, X} \sum_{i} \| Y_i - DX_i \|_2^2 \quad s.t. \forall i, \| x_i \|_0 \leq K
  \]
- The problem can be solved in an alternating manner. In the first stage, D is fixed and only the sparse codes are computed by orthogonal matching pursuit.
  \[
  \min_{x_i} \| Y - DX \|_2^2 \quad s.t. \| x_i \|_0 \leq K
  \]
- In the second stage, each filter in D and its associated sparse codes x are updated simultaneously by Singular Value Decomposition (||D|| = 1)
  \[
  Y = DX + E \quad \text{||D||} = 1
  \]
- When the sparsity level K is set to be 1 and sparse codes are forced to be a binary(0/1), K-Means is exactly reproduced (no constraints on \(d_a\)).

Batch Tree Orthogonal Matching Pursuit

<table>
<thead>
<tr>
<th>Algorithm: Batch Tree Orthogonal Matching Pursuit (BTOMP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Input: Dictionary D, Center C, observation y, and the desired sparsity level K</td>
</tr>
<tr>
<td>2. Output: Sparse code (x) such that (y = Dx)</td>
</tr>
<tr>
<td>3. Initialization: (I = B, r = y, a = 0^T = C^{-1}y, B = C^{-1}D, ) and (v = 0)</td>
</tr>
<tr>
<td>4. For (k = 1, K)</td>
</tr>
<tr>
<td>5. Choosing the sub-dictionary (D_k = \arg\min_{D_k} |D_k</td>
</tr>
<tr>
<td>6. Selecting the new filter (D_k = \arg\min_{D_k} |D_k</td>
</tr>
<tr>
<td>7. (j = j + 1)</td>
</tr>
<tr>
<td>8. Updating the sparse code (x_j = (D_jD_j)^{-1}D_jy)</td>
</tr>
<tr>
<td>9. Updating (r, v = 0^T = D_jr, r)</td>
</tr>
<tr>
<td>10. Computation of residual (r = y - D_jr)</td>
</tr>
<tr>
<td>11. End</td>
</tr>
</tbody>
</table>

K-Means is used to group the whole dictionary into the sub-dictionaries and associate the sub-dictionaries with the learned center matrix C; Line 5 selects the center filter j that best matches the current residual; Line 6 selects the filter within the sub-dictionary associated with the center j; Line 8 updates sparse codes with the incremental Cholesky decomposition; Line 9 computes the correlation between each center and the current residual; If the centers C are set to be the whole dictionary, BTOMP exactly recovers the batch (orthogonal) matching pursuit[^11].


Matching Pursuit Encoder

- Matching pursuit encoder consists of three modules: BTOMP, Spatial Pyramid Max Pooling and Contrast Normalization.

- Spatial Pyramid Max Pooling aggregates the sparse codes which are spatially close, using max pooling in a multi-level patch decomposition.
  \[
  F(P) = \max_{x_i \in P} \{ x_i \} - \max_{x_i \in P} \{ x_i \}
  \]
- Contrast Normalization is helpful since the magnitude of sparse codes varies significantly due to illumination and foreground-background contrast.
  \[
  F(P) = \frac{F(P)}{\sqrt{\|F(P)\|^2 + \epsilon}}
  \]

Scene Recognition (MIT-Scene)

- This dataset contains 15620 images from 67 indoor scene categories;
- Train linear SVM on 80 images and test on 20 images per category;
- The experimental setting is same as with the Caltech-101 dataset except that the filter size is 4×4 (cross validation).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>72.8</td>
<td>68.5</td>
<td>70.7</td>
<td>71.5</td>
<td>73.8</td>
<td>73.5</td>
<td>73.8</td>
<td>77.7</td>
</tr>
</tbody>
</table>

[^1]: Liu, S. and Fox, D., NIPS 2010
[^2]: Quattoni and Torr, P., CVPR 2009
[^3]: Yu, L. and Lazebnik, S., NIPS 2008
[^5]: Yu, L. and Lazebnik, S., NIPS 2008
[^7]: Yue, Y., Gu, H. and Wang, Y., CVPR 2009
[^8]: Horacek, B., LeCun, Y. and Fox, D., NIPS 2010

Event Recognition (UIUC-Sports)

- This dataset consists of 8 sport event categories with 137 to 250 images in each.
- Train linear SVM on 70 images and test on 60 images per category.
- The experimental setting is same as with the MIT-Scene dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>HMP</th>
<th>OB</th>
<th>SIFT+GMM[^1]</th>
<th>SIFT+SC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>85.7</td>
<td>76.3</td>
<td>73.4</td>
<td>82.5</td>
</tr>
</tbody>
</table>

[^1]: Liu and Fox, D., ICCV 2007