Multiview Feature Learning

Roland Memisevic

Uni Frankfurt

Tutorial at CVPR 2012
Outline

1 Introduction
   - Feature Learning
   - Correspondence in Computer Vision
   - Relational feature learning

2 Learning relational features
   - Sparse Coding Review
   - Encoding relations
   - Inference
   - Learning

3 Factorization, eigen-spaces and complex cells
   - Factorization
   - Eigen-spaces, energy models, complex cells

4 Applications
   - Applications
   - Conclusions
1 Introduction
   • Feature Learning
   • Correspondence in Computer Vision
   • Relational feature learning

2 Learning relational features
   • Sparse Coding Review
   • Encoding relations
   • Inference
   • Learning

3 Factorization, eigen-spaces and complex cells
   • Factorization
   • Eigen-spaces, energy models, complex cells

4 Applications
   • Applications
   • Conclusions
What this is about

- Extending feature learning to model relations.
Recognition tasks

- Recognition has become a focus of interest in Computer Vision.
- Recognition of static objects started to work very well. In fact –
- Recognition is getting quite serious.

(PASCAL challenge)
"It’s the feature, stupid!"

- A main reason is the use of the *right representation*:
- Recognition started to work after the community converged on *local features*, like SIFT.
- With the right representation, the choice of top level classifier (SVM, logreg, NN) doesn’t matter all that much.

(PASCAL challenge)
Task: Recognize the building.

Two approaches:
1. Bag-Of-Features
2. Convolution
1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** ("features").
4. **Add** all local descriptors to obtain a global descriptor for the image.
Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** (“features”).
4. Add all local descriptors to obtain a global descriptor for the image.
Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** ("features").
4. Add all local descriptors to obtain a global descriptor for the image.

\[
\begin{pmatrix}
  f_1 \\
  \vdots \\
  \vdots \\
  f_n
\end{pmatrix}
\]
Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** ("features").
4. **Add** all local descriptors to obtain a global descriptor for the image.
Bag-Of-Features

1. Find *interest points* (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a *sparse local descriptor* ("features").
4. Add all local descriptors to obtain a global descriptor for the image.
1. Crop patches along a regular grid (dense or not).
2. Represent each patch with a local descriptor.
3. Concatenate all descriptors into a very large vector.
1. Crop patches along a regular grid (dense or not).
2. Represent each patch with a local descriptor.
3. Concatenate all descriptors into a very large vector.
Convolutional

1. Crop patches along a regular grid (dense or not).
2. Represent each patch with a local descriptor.
3. **Concatenate** all descriptors into a very large vector.
When images are represented as points in $\mathbb{R}^n$, we can use a simple classifier to do recognition.

Eg., Logistic regression, SVM, NN, ...

(There are various extensions, like fancy pooling, etc.)
How do we get good features?

- Option B: Engineer them. SIFT, HOG, LBP, etc.
- Natural Images are not random.
- Option A: *Learn* the representation from image data.
How do we get good features?

Option B: Engineer them. SIFT, HOG, LBP, etc.

Natural Images are not random.

Option A: Learn the representation from image data.
- How do we get good features?
- Option B: Engineer them. SIFT, HOG, LBP, etc.
- Natural Images are not random.
- Option A: Learn the representation from image data.
How do we get good features?
Option B: Engineer them. SIFT, HOG, LBP, etc.

Natural Images are not random.

Option A: *Learn* the representation from image data.
Why Feature Learning

- Feature Learning, Dictionary Learning, Receptive Field Learning, Sparse Coding, etc.
  - Helps overcome tedious engineering.
  - Helps adapt models to different data domains (including a model’s own representations! – “deep learning”)
  - Biologically consistent.
  - Brings us closer to end-to-end learning of vision systems.
More importantly... it works well

See, eg., (Coates, et al., 2011)
Feature Learning

- Encode patch $y$ using latent variables $z$.
- Learn weights $W$ from training data of image patches.

Feature Learning works well for recognition, so...
Feature Learning

- Encode patch $y$ using latent variables $z$.
- Learn weights $W$ from training data of image patches.

Feature Learning works well for recognition, so...
Outline

1 Introduction
   - Feature Learning
   - Correspondence in Computer Vision
   - Relational feature learning

2 Learning relational features
   - Sparse Coding Review
   - Encoding relations
   - Inference
   - Learning

3 Factorization, eigen-spaces and complex cells
   - Factorization
   - Eigen-spaces, energy models, complex cells

4 Applications
   - Applications
   - Conclusions
Beyond object recognition

Can we do more with Feature Learning than recognize *things*?

- Good features work well for object recognition.
- But brains can do much more than recognize objects.
- A large number of vision tasks goes beyond object recognition.
- In surprisingly many vision tasks, the relationship *between* images carries the relevant information.
Beyond object recognition

Can we do more with Feature Learning than recognize *things*?

- Good features work well for object recognition.
- But brains can do much more than recognize objects.
- A large number of vision tasks goes beyond object recognition.
- In surprisingly many vision tasks, the relationship *between* images carries the relevant information.
Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure
Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision
- Tracking
- Stereo
  - Geometry
  - Optical Flow
  - Invariant Recognition
  - Odometry
  - Action Recognition
  - Contours, Within-image structure
Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
  - Optical Flow
  - Invariant Recognition
  - Odometry
  - Action Recognition
  - Contours, Within-image structure
Correspondences in Computer Vision

Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure
Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure
Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

### Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- **Odometry**
- Action Recognition
- Contours, Within-image structure
Correspondences in Computer Vision

Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure
Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure
Correspondence is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure
Adding frames is not just about adding proportionally more information.

The relationships between frames contain additional information, that is not present in any single frame.

See *Heider and Simmel, 1944*: Any single frame shows a bunch of geometric figures. The motions reveal the story.
Outline

1. **Introduction**
   - Feature Learning
   - Correspondence in Computer Vision
   - Relational feature learning

2. **Learning relational features**
   - Sparse Coding Review
   - Encoding relations
   - Inference
   - Learning

3. **Factorization, eigen-spaces and complex cells**
   - Factorization
   - Eigen-spaces, energy models, complex cells

4. **Applications**
   - Applications
   - Conclusions
If correspondences matter in vision, can we learn them?
It turns out that this requires latent variables to act like *gates*, that dynamically change the connections between fellow variables.
Learning features to model correspondences

- This amounts to letting variables multiply connections between other variables.
- And it is equivalent to having \textit{three-way multiplicative interactions}. 

\[
x_i \rightarrow z_k \rightarrow y_j
\]
Learning and inference in the presence of gating variables is (slightly) different from learning without.

We can set things up, such that inference is almost unchanged. Yet the meaning of the latent variables will be entirely different.
Learning features to model correspondences

- Multiplicative interactions allow hidden variables to *blend in a whole “sub”-network*.
- This leads to a qualitatively quite different behaviour from all common bi-partite feature learning models.
### Brief history of multiplication

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...
Multiplicative interactions

Brief history of multiplication

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...
Multiplicative interactions

Brief history of multiplication

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
  - Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
  - Subspace SOM (Kohonen, 1996)
  - ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
  - (2006 –) GBM, mcRBM, RAE, convISA, applications...
**Multiplicative interactions**

**Brief history of multiplication**

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...
Multiplicative interactions

Brief history of multiplication

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
  - ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
  - (2006 –) GBM, mcRBM, RAE, convISA, applications...
Multiplicative interactions

Brief history of multiplication

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...
Multiplicative interactions

Brief history of multiplication

- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...
(Hinton, 1981)
Mapping units 1981

(Hinton, 1981)
Example application: Action recognition

(Hollywood 2)

(Marszałek et al., 2009)
ISA applied to action recognition

(Le, et al., 2011)

<table>
<thead>
<tr>
<th></th>
<th>KTH</th>
<th>Hollywood2</th>
<th>UCF</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>until 2011</td>
<td>92.1</td>
<td>50.9</td>
<td>85.6</td>
<td>71.2</td>
</tr>
<tr>
<td>hierarchical ISA</td>
<td>93.9</td>
<td>53.3</td>
<td>86.5</td>
<td>75.8</td>
</tr>
</tbody>
</table>
Learning higher-order features

(Ranzato et al., 2010)
Let *labels act like gates*. (Nair et al., 2009; Memisevic et al., 2010)

<table>
<thead>
<tr>
<th>dataset/model:</th>
<th>SVMs</th>
<th>NNet</th>
<th>RBM</th>
<th>DEEP</th>
<th>GSM</th>
<th>GSM (unfact)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rectangles</td>
<td>2.15</td>
<td>7.16</td>
<td>4.71</td>
<td>2.60</td>
<td>2.41</td>
<td>0.83</td>
</tr>
<tr>
<td>rect.-images</td>
<td>24.04</td>
<td>33.20</td>
<td>23.69</td>
<td>22.50</td>
<td>24.05</td>
<td>22.51</td>
</tr>
<tr>
<td>mnistplain</td>
<td>3.03</td>
<td>4.69</td>
<td>3.94</td>
<td>3.11</td>
<td>3.46</td>
<td>3.70</td>
</tr>
<tr>
<td>convexshapes</td>
<td>19.13</td>
<td>32.25</td>
<td>19.92</td>
<td>18.63</td>
<td>18.41</td>
<td>17.08</td>
</tr>
<tr>
<td>mnistbackrand</td>
<td>14.58</td>
<td>20.04</td>
<td>9.80</td>
<td>6.73</td>
<td>11.28</td>
<td>10.48</td>
</tr>
<tr>
<td>mnistbackimg</td>
<td>22.61</td>
<td>27.41</td>
<td>16.15</td>
<td>16.31</td>
<td>23.00</td>
<td>23.65</td>
</tr>
<tr>
<td>mnistrotbackimg</td>
<td>55.18</td>
<td>62.16</td>
<td>52.21</td>
<td>47.39</td>
<td>51.93</td>
<td>55.82</td>
</tr>
<tr>
<td>mnistrot</td>
<td>11.11</td>
<td>18.11</td>
<td>14.69</td>
<td>10.30</td>
<td>10.30</td>
<td>11.75</td>
</tr>
</tbody>
</table>
Tracking

- (Bazzani et al.), (Larochelle, Hinton, 2011)