Unsupervised Learning Jointly With Image Clustering

Jianwei Yang
Devi Parikh
Dhruv Batra

Virginia Tech

https://filebox.ece.vt.edu/~jw2yang/
Unsupervised Learning
Huge amount of images!!!

Unsupervised learning
Huge amount of images!!!
Learning without annotation efforts
Unsupervised Learning
Huge amount of images!!!
Learning without annotation efforts
What we need to learn?
Unsupervised learning
Huge amount of images!!!

Learning without annotation efforts

What we need to learn?

Unsupervised learning

An open problem
Huge amount of images!!!
Learning without annotation efforts
What we need to learn?

Unsupervised learning

An open problem
A hot problem
Huge amount of images!!!
Learning without annotation efforts
What we need to learn?

Unsupervised learning

An open problem
A hot problem
Various methodologies
Learning distribution (structure)

Clustering
Learning distribution (structure)

Clustering

K-means (Image Credit: Jesse Johnson)

Learning distribution (structure)

Clustering

K-means (Image Credit: Jesse Johnson)

Hierarchical Clustering
Learning distribution (structure)

Clustering

- K-means (Image Credit: Jesse Johnson)
- Hierarchical Clustering
- Spectral Clustering
  Manor et al, NIPS’04

Learning distribution (structure)

Clustering

K-means (Image Credit: Jesse Johnson)

Hierarchical Clustering

Spectral Clustering
Manor et al, NIPS’04

Graph Cut
Shi et al, TPAMI’00

Learning distribution (structure)

Clustering

K-means (Image Credit: Jesse Johnson)

Hierarchical Clustering

Spectral Clustering
Manor et al, NIPS’04

DBSCAN, Ester et al, KDD’96 (Image Credit: Jesse Johnson)

Graph Cut
Shi et al, TPAMI’00

Learning distribution (structure)

Clustering

K-means (Image Credit: Jesse Johnson)

Hierarchical Clustering

Spectral Clustering
Manor et al, NIPS’04

Graph Cut
Shi et al, TPAMI’00

DBSCAN, Ester et al, KDD’96 (Image Credit: Jesse Johnson)

EM Algorithm, Dempster et al, JRSS’77

Learning distribution (structure)

Clustering

K-means (Image Credit: Jesse Johnson)

Hierarchical Clustering

Spectral Clustering
Manor et al, NIPS’04

Graph Cut
Shi et al, TPAMI’00

DBSCAN, Ester et al, KDD’96 (Image Credit: Jesse Johnson)

EM Algorithm, Dempster et al, JRSS’77

NMF, Xu et al, SIGIR’03 (Image Credit: Conrad Lee)

Learning distribution (structure)

Sub-space Analysis

PCA (Image Credit: Jesse Johnson)

ICA (Image Credit: Shylaja et al)

tSNE, Maaten et al, JMLR’08

Subspace Clustering, Vidal et al.

Sparse coding, Olshausen et al. Vision Research’97
Learning representation (feature)

Learning representation (feature)

VAE, Kingma et al, arXiv’13  
(Image Credit: Fast Forward Labs)

GAN, Goodfellow et al, NIPS’14  
DCGAN, Radford et al, arXiv’15  
(Image Credit: Mike Swarbrick Jones)
Most Recent CV Works

Spatial context, Doersch et al, ICCV’15

Temporal context, Wang et al, ICCV’15

Solving Jigsaw, Noroozi et al, ECCV’16

Ego-motion, Jayaraman et al, ICCV’15

Context Encoder, Deepak et al, CVPR’16
Most Recent CV Works

**Visual concept clustering**, Huang et al, CVPR’16

**TAGnet**, Wang et al, SDM’16

**Deep Embedding**, Xie et al, ICML’16

**Graph constraint**, Li et al, ECCV’16
Our Work

Joint Unsupervised Learning (JULE) of Deep Representations and Image Clusters
Outline

• Intuition
• Approach
• Experiments
• Extensions
Intuition

Meaningful clusters can provide supervisory signals to learn image representations.
Intuition

Meaningful clusters can provide supervisory signals to learn image representations

Good representations help to get meaningful clusters
Intuition

Cluster images first, and then learn representations
Intuition

Cluster images first, and then learn representations

Learn representations first, and then cluster images
Intuition

Cluster images first, and then learn representations.

Learn representations first, and then cluster images.

Cluster images and learn representations progressively.
Intuition

Good clusters

Poor clusters
Poor representations

Good representations

Good cluster
Good representations
Intuition

Good clusters

Good representations

Poor clusters

Poor representations

Good cluster

Good representations
Intuition

Good clusters

Poor clusters

Poor representations

Good representations

Good cluster
Good representations
Intuition

Good clusters

Poor clusters

Poor representations

Good representations

Good cluster

Good representations
Approach

• Framework
• Objective
• Algorithm & Implementation
Approach: Framework

\[ \arg \min_{\theta} L(\theta \mid y, I) \]

Convolutional Neural Network

\[ \arg \min_{y, \theta} L(y, \theta \mid I) \]

Agglomerative Clustering

\[ \arg \min_{y} L(y \mid \theta, I) \]
Approach: Framework

Convolutional Neural Network

\[ \arg \min_{\theta} L(\theta \mid y, I) \]

Agglomerative Clustering

\[ \arg \min_{y} L(y \mid \theta, I) \]
Approach: Recurrent Framework
Approach: Recurrent Framework
Approach: Recurrent Framework
Approach: Recurrent Framework
Approach: Recurrent Framework
Approach: Recurrent Framework
Approach: Recurrent Framework

Backward at each time-step is time-consuming and prone to over-fitting!
Approach: Recurrent Framework

Backward at each time-step is time-consuming and prone to over-fitting!

How about updating once for multiple time-steps?
Approach: Recurrent Framework

Partially Unrolling: divide all T time-steps into P periods

In each period, we merge clusters for multiple times and update CNN parameters at the end of period
Approach: Recurrent Framework

Partially Unrolling: divide all T time-steps into P periods

In each period, we merge clusters for multiple times and update CNN parameters at the end of period
Approach: Recurrent Framework

Partially Unrolling: divide all T time-steps into P periods

In each period, we merge clusters for multiple times and update CNN parameters at the end of period

P is determined by a hyper-parameter will be introduced later
Approach: Objective Function

Overall loss:

\[ L(\{y^1, ..., y^T\}, \{\theta^1, ..., \theta^T\} | I) = \sum_{t=1}^{T} L'(y', \theta' | y'^{-1}, I) \]

\[ \text{sum over all } T \text{ timesteps} \]
Approach: Objective Function

\[ L(\{\{y^1, \ldots, y^T\}, \{\{\theta^1, \ldots, \theta^T\}\} | I) = \sum_{t=1}^{T} L'(y^t, \theta^t | y^{t-1}, I) \]

Loss at time-step t:

\[ \mathcal{L}^t(y^t, \theta^t | y^{t-1}, I) = -\mathcal{A}(C_i^t, \mathcal{N}_c^{K_c}[1]) \]

\[-\frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( \mathcal{A}(C_i^t, \mathcal{N}_c^{K_c}[1]) - \mathcal{A}(C_i^t, \mathcal{N}_c^{K_c}[k]) \right) \]

Conventional Agg. Clustering Strategy

Proposed Agg. Clustering Strategy
Approach: Objective Function

Objective Function

\[ L(\{y^1, \ldots, y^T\}, \{\theta^1, \ldots, \theta^T\} | I) = \sum_{t=1}^{T} L'(y^t, \theta^t | y^{t-1}, I) \]

Loss at time-step t:

\[ L^t(y^t, \theta^t | y^{t-1}, I) = \mathcal{A}(C^t_i, \mathcal{N}^{K_c}_c[i][1]) - \frac{\lambda}{K_c - 1} \sum_{k=2}^{K_c} \left( \mathcal{A}(C^t_i, \mathcal{N}^{K_c}_c[i][1]) - \mathcal{A}(C^t_i, \mathcal{N}^{K_c}_c[k]) \right) \]
Approach: Objective Function

Loss at time-step t:

\[
L^t(y^t, \theta^t | y^{t-1}, I) = -\mathcal{A}(C_i^t, \mathcal{N}_C^{K_c} [1]) - \frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( \mathcal{A}(C_i^t, \mathcal{N}_C^{K_c} [k]) - \mathcal{A}(C_i^t, \mathcal{N}_C^{K_c} [k]) \right)
\]
Approach: Objective Function

Objective Function

\[
L(\{y^1, ..., y^T\}, \{\theta^1, ..., \theta^T\} | I) = \sum_{t=1}^{T} L'(y^t, \theta^t | y^{t-1}, I)
\]

where
- \(L\) is the loss function
- \(\{y^1, ..., y^T\}\) are the cluster labels
- \(\{\theta^1, ..., \theta^T\}\) are the CNN parameters
- \(I\) is the input data
- \(T\) is the number of time steps

Loss at time-step \(t\):

\[
L^t(y^t, \theta^t | y^{t-1}, I) = -A(C^t_i, N^K_{C^t_i}[1])
\]

\[
- \frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( A(C^t_i, N^K_{C^t_i}[k]) - A(C^t_i, N^K_{C^t_i}[k]) \right)
\]

- \(\lambda\) is a hyperparameter
- \(A\) is an activation function
- \(C^t_i\) is the \(i\)-th cluster at time \(t\)
- \(N^K_{C^t_i}\) is the set of \(K\) nearest neighbors of the \(i\)-th cluster

K_c nearest neighbor clusters of \(i\)-th cluster

Conventional Agg. Clustering Strategy

Proposed Agg. Clustering Strategy
Approach: Objective Function

Objective Function

\[
L(\{y^1, \ldots, y^T\}, \{\theta^1, \ldots, \theta^T\} \mid I) = \sum_{t=1}^{T} L'(y^t, \theta^t \mid y^{t-1}, I)
\]

Loss at time-step t:

\[
L^t(y^t, \theta^t \mid y^{t-1}, I) = -A(C^t_i, \mathcal{N}^{K_c}_c[1])
\]

\[
-\frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( A(C^t_i, \mathcal{N}^{K_c}_c[1]) - A(C^t_i, \mathcal{N}^{K_c}_c[k]) \right)
\]

Affinity between i-th cluster and its NN

Conventional Agg. Clustering Strategy

Proposed Agg. Clustering Strategy
Approach: Objective Function

Loss at time-step $t$:

$$
L^t(y^t, \theta^t | y^{t-1}, I) = -\mathcal{A}(C_i^t, \mathcal{N}_{C_i}^{K_c}[1])
$$

$$
-\frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( \mathcal{A}(C_i^t, \mathcal{N}_{C_i}^{K_c}[1]) - \mathcal{A}(C_i^t, \mathcal{N}_{C_i}^{K_c}[k]) \right)
$$

Affinity between $i$-th cluster and its NN

Differences between two cluster affinities

Conventional Agg. Clustering Strategy

Proposed Agg. Clustering Strategy
Approach: Objective Function

Loss at time-step $t$:

$$\mathcal{L}^t(y^t, \theta^t | y^{t-1}, I) = \mathcal{A}(C^t_i, N^{K_c}_{C^t_i}[1])$$

Differences between two cluster affinities:

$$-\frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( \mathcal{A}(C^t_i, N^{K_c}_{C^t_i}[1]) - \mathcal{A}(C^t_i, N^{K_c}_{C^t_i}[k]) \right)$$

Affinity between $i$-th cluster and its NN

Merge these two clusters

Conventional Agg. Clustering Strategy

Proposed Agg. Clustering Strategy
Approach: Objective Function

Loss at time-step $t$:

$$ L^t(y^t, \theta^t | y^{t-1}, I) = -\mathcal{A}(C^t_i, \mathcal{N}^{K_c}_{C_i} [1]) $$

$$ -\frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( \mathcal{A}(C^t_i, \mathcal{N}^{K_c}_{C_i} [1]) - \mathcal{A}(C^t_i, \mathcal{N}^{K_c}_{C_i} [k]) \right) $$

Affinity between $i$-th cluster and its NN

Differences between two cluster affinities

Merge these two clusters

Conventional Agg. Clustering Strategy

Proposed Agg. Clustering Strategy
Approach: Objective Function

\[
L(\{y^1, \ldots, y^T\}, \{\theta^1, \ldots, \theta^T\} | I) = \sum_{t=1}^{T} L'(y^t, \theta^t | y^{t-1}, I)
\]

Loss in forward pass in period \( p \) (merge clusters):

Loss in forward pass in period \( p \) (merge clusters):
Approach: Objective Function

\[ L(\{y^1, ..., y^T\}, \{\theta^1, ..., \theta^T\} | I) = \sum_{t=1}^{T} L'(y^t, \theta^t | y^{t-1}, I) \]

Loss in forward pass in period \( p \) (merge clusters):

\[ L(\{y\}^p | \theta^p, I) = \sum_{t \in p} L'(y^t | \theta^p, y^{t-1}, I) \]

Loss in forward pass in period \( p \) (merge clusters):
Approach: Objective Function

\[
L(\{y^1, \ldots, y^T\}, \{\theta^1, \ldots, \theta^T\} \mid I) = \sum_{t=1}^{T} L'(y^t, \theta^t \mid y^{t-1}, I)
\]

Loss in forward pass in period \(p\) (merge clusters):

\[
L(\{y\}^p \mid \theta^p, I) = \sum_{t \in p} L'(y^t \mid \theta^p, y^{t-1}, I)
\]

Loss in forward pass in period \(p\) (merge clusters):
Approach: Objective Function

\[ L(\{y^1, \ldots, y^T\}, \{\theta^1, \ldots, \theta^T\} \mid I) = \sum_{t=1}^{T} L'(y^t, \theta^t \mid y^{t-1}, I) \]

Loss in forward pass in period \( p \) (merge clusters):

\[ L(\{y\}^p \mid \theta^p, I) = \sum_{t \in p} L'(y^t \mid \theta^p, y^{t-1}, I) \]

Loss in forward pass in period \( p \) (merge clusters):

\[ L(\theta \mid \{y\}_{1}^{1}, \ldots, \{y\}^{p}_{p}, I) = \sum_{k=1}^{p} L^k(\theta \mid \{y\}^{k}_{p}, I) \]
Approach: Objective Function

Forward Pass:

\[ L(\{y\}^P | \theta^P, I) = \sum_{t \in p} L'(y^t | \theta^P, y^{t-1}, I) \]

Simple Greedy Algorithm
Merge two clusters which minimize the loss at each time step
Approach: Objective Function

Forward Pass:

\[ L(\{y\}^p | \theta^p, I) = \sum_{t \in p} L'(y^t | \theta^p, y^{t-1}, I) \]

Simple Greedy Algorithm
Merge two clusters which minimize the loss at each time step
Approach: Objective Function

Forward Pass:

\[ L(\{y\}^p | \theta^p, I) = \sum_{t \in T} L'(y^t | \theta^p, y^{t-1}, I) \]

Simple Greedy Algorithm
Merge two clusters which minimize the loss at each time step
Approach: Objective Function

Forward Pass:

$$L(\{y\}^p | \theta^p, I) = \sum_{t \in p} L'(y^t | \theta^p, y^{t-1}, I)$$

sum over timesteps in unrolling period $p$

Simple Greedy Algorithm
Merge two clusters which minimize the loss at each time step
Approach: Objective

Backward Pass:

\[
L(\theta \mid \{y\}_1^1, \ldots, \{y\}_p^p, I) = \sum_{k=1}^{p} L^k(\theta \mid \{y\}_k^k, I)
\]

optimal solutions

sum over all previous periods
Approach: Objective

Backward Pass:

\[
L(\theta | \{y\}_1^1, \ldots, \{y\}_p^p, I) = \sum_{k=1}^{p} L^k(\theta | \{y\}_k^k, I)
\]

Consider all previous periods
Approach: Objective

Backward Pass:

Consider all previous periods

\[
L(\theta | \{y\}_{1}^{1}, \ldots, \{y\}_{p}^{p}, I) = \sum_{k=1}^{p} L^k(\theta | \{y\}_{1}^{k}, I)
\]

Cluster based loss is not proper for batch optimization!!!
Approach: Objective

Backward Pass:

\[
L(\theta \mid \{y^{1}_*, \ldots, y^{p}_*\}, I) = \sum_{k=1}^{p} L^k(\theta \mid \{y^k_*\}, I)
\]

Consider all previous periods

Cluster based loss is not proper for batch optimization!!!

Approximation:

\[
A(C_m \cup C_n, C_i) = A(C_m \rightarrow C_i) + A(C_n \rightarrow C_i)
\]

\[
+ \frac{|C_m|}{|C_m| + |C_n|} A(C_i \rightarrow C_m)
\]

\[
+ \frac{|C_m|}{|C_m| + |C_n|} A(C_i \rightarrow C_n)
\]

67
Approach: Objective

Backward Pass:

\[ L(\theta \mid \{y\}_1^p, \ldots, \{y\}_p^I) = \sum_{k=1}^{p} L^{k}(\theta \mid \{y\}_*^k, I) \]

Consider all previous periods

Recall cluster-based loss:

\[ \mathcal{L}^t(y^t, \theta^t \mid y^{t-1}, I) = -\mathcal{A}(C_i^t, N_{C_i^t}^{K_c}[1]) \]

\[ -\frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left( \mathcal{A}(C_i^t, N_{C_i^t}^{K_c}[1]) - \mathcal{A}(C_i^t, N_{C_i^t}^{K_c}[k]) \right) \]

Convert to sample-based loss:

\[ L(\theta \mid \{y\}_1^p, \ldots, \{y\}_p^I) = -\sum_{i,j,k} (\gamma \mathcal{A}(x_i, x_j) - \mathcal{A}(x_i, x_k)) \]
Approach: Objective

Backward Pass:

\[
L(\theta \mid \{y\}_{*}^{1}, \ldots, \{y\}_{*}^{p}, I) = \sum_{k=1}^{p} L^{k}(\theta \mid \{y\}_{*}^{k}, I)
\]

Consider all previous periods

Recall cluster-based loss:

\[
\mathcal{L}^{t}(y^{t}, \theta^{t} \mid y^{t-1}, I) = -\mathcal{A}(C_{i}^{t} \mid N_{C_{i}^{t}}^{Kc}[1])
\]

\[
-\frac{\lambda}{(K_{c} - 1)} \sum_{k=2}^{K_{c}} \left( \mathcal{A}(C_{i}^{t} \mid N_{C_{i}^{t}}^{Kc}[k]) - \mathcal{A}(C_{i}^{t} \mid N_{C_{i}^{t}}^{Kc}[1]) \right)
\]

Convert to sample-based loss:

\[
L(\theta \mid \{y\}_{*}^{1}, \ldots, \{y\}_{*}^{p}, I) = -\sum_{i,j,k}^{\gamma} \left( A(x_{i} \mid x_{j}) - A(x_{i} \mid x_{k}) \right)
\]

Weighted triplet loss

Intra-sample affinity

Inter-sample affinity
Approach: Algorithm & Implementation

Algorithm 1 Joint Optimization on $y$ and $\theta$

Input:
- $I$: collection of image data;
- $n_c^*$: target number of clusters;

Output:
- $y^*, \theta^*$: final image labels and CNN parameters;

1: $t \leftarrow 0$; $p \leftarrow 0$
2: Initialize $\theta$ and $y$
3: repeat
4: Update $y^t$ to $y^{t+1}$ by merging two clusters
5: if $t = t_c^*$ then
6: Update $\theta^p$ to $\theta^{p+1}$ by training CNN
7: $p \leftarrow (p + 1)$
8: end if
9: $t \leftarrow t + 1$
10: until Cluster number reaches $n_c^*$
11: $y^* \leftarrow y^t$; $\theta^* \leftarrow \theta^p$
Algorithm 1: Joint Optimization on $y$ and $\theta$

Input:
- $I$: collection of image data;
- $n_c^*$: target number of clusters;

Output:
- $y^*, \theta^*$: final image labels and CNN parameters;

1: $t \leftarrow 0; p \leftarrow 0$
2: Initialize $\theta$ and $y$
3: repeat
4: Update $y^t$ to $y^{t+1}$ by merging two clusters
5: if $t = t_p^c$ then
6: Update $\theta^p$ to $\theta^{p+1}$ by training CNN
7: $p \leftarrow (p + 1)$
8: end if
9: $t \leftarrow t + 1$
10: until Cluster number reaches $n_c^*$
11: $y^* \leftarrow y^t; \theta^* \leftarrow \theta^p$

Raw image data
Algorithm 1 Joint Optimization on $y$ and $\theta$

**Input:**
- $I$: collection of image data;
- $n_c^*$: target number of clusters

**Output:**
- $y^*, \theta^*$: final image labels and CNN parameters;

1: $t \leftarrow 0; p \leftarrow 0$
2: Initialize $\theta$ and $y$
3: repeat
4: Update $y^t$ to $y^{t+1}$ by merging two clusters
5: if $t = t^*_p$ then
6: Update $\theta^p$ to $\theta^{p+1}$ by training CNN
7: $p \leftarrow (p + 1)$
8: end if
9: $t \leftarrow t + 1$
10: until Cluster number reaches $n_c^*$
11: $y^* \leftarrow y^t; \theta^* \leftarrow \theta^p$

Raw image data
Assume it is known
Approach: Algorithm & Implementation

**Algorithm 1** Joint Optimization on $y$ and $\theta$

**Input:**
- $I$: collection of image data;
- $n_c^*$: target number of clusters;

**Output:**
- $y^*, \theta^*$: final image labels and CNN parameters;

1: $t \leftarrow 0; p \leftarrow 0$
2: Initialize $\theta$ and $y^*$
3: repeat
4: Update $y^t$ to $y^{t+1}$ by merging two clusters
5: if $t = t_p^*$ then
6: Update $\theta^p$ to $\theta^{p+1}$ by training CNN
7: $p \leftarrow (p + 1)$
8: end if
9: $t \leftarrow t + 1$
10: until Cluster number reaches $n_c^*$
11: $y^* \leftarrow y^t; \theta^* \leftarrow \theta^p$

- Raw image data
- Assume it is known
- Randomly initialize CNN parameters
- 4 samples in each cluster in average
Approach: Algorithm & Implementation

Algorithm 1 Joint Optimization on $y$ and $\theta$

**Input:**
- $I$: collection of raw image data
- $n_c^*$: target number of clusters

**Output:**
- $y^*, \theta^*$: final image labels and CNN parameters

1. $t \leftarrow 0; p \leftarrow 0$
2. Initialize $\theta$ and $y^*_t$
3. repeat
   4. Update $y^t$ to $y^{t+1}$ by merging two clusters
   5. if $t = t_p^*$ then
   6. Update $\theta^p$ to $\theta^{p+1}$ by training CNN
   7. $p \leftarrow (p + 1)$
5. end if
9. $t \leftarrow t + 1$
10. until Cluster number reaches $n_c^*$
11. $y^* \leftarrow y^t; \theta^* \leftarrow \theta^p$

- Raw image data
- Assume it is known
- Randomly initialize CNN parameters
- 4 samples in each cluster in average
- Train CNN for about 20 epochs
Approach: Algorithm & Implementation

Algorithm 1 Joint Optimization on $y$ and $\theta$

Input:
- $I$: collection of image data
- $n_c^*$: target number of clusters

Output:
- $y^*$, $\theta^*$: final image labels and CNN parameters

1: $t \leftarrow 0$; $p \leftarrow 0$
2: Initialize $\theta$ and $y$
3: repeat
4: Update $y^t$ to $y^{t+1}$ by merging two clusters
5: if $t = t_p^c$ then
6: Update $\theta^p$ to $\theta^{p+1}$ by training CNN
7: $p \leftarrow (p + 1)$
8: end if
9: $t \leftarrow t + 1$
10: until Cluster number reaches $n_c^*$
11: $y^* \leftarrow y^t$; $\theta^* \leftarrow \theta^p$

Raw image data
Assume it is known
Randomly initialize CNN parameters
4 samples in each cluster in average
Train CNN for about 20 epochs
We can go back and retrain the model, but it improve slightly
Experiments

• Datasets
• Network Architecture
• Image Clustering
• Representation Learning
Experiments: Datasets

MNIST (70000, 10, 28x28)
USPS (11000, 10, 16x16)
COIL20 (1440, 20, 128x128)
COIL100 (7200, 100, 128x128)
UMist (575, 20, 112x92)
FRGC (2462, 20, 32x32)
CMU-PIE (2856, 68, 32x32)
Youtube Face (1000, 41, 55x55)
Experiments: Settings

Two important parameters

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>$K_a$</th>
<th>$\alpha$</th>
<th>$K_c$</th>
<th>$\lambda$</th>
<th>$\gamma$</th>
<th>$\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>20</td>
<td>1.0</td>
<td>5</td>
<td>1.0</td>
<td>2.0</td>
<td>0.9 or 0.2</td>
</tr>
</tbody>
</table>

Set the layer numbers so that the Output feature map is about 10x10
Experiments: Clustering: Performance

+6.43% on NMI to best performance of existing approaches averaged over all datasets

Table 3: Quantitative clustering performance (NMI) for different algorithms using image intensities as input.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COIL20</th>
<th>COIL100</th>
<th>USPS</th>
<th>MNIST-test</th>
<th>MNIST-full</th>
<th>UMIST</th>
<th>FRGC</th>
<th>CMU-PIE</th>
<th>YTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means [39]</td>
<td>0.775</td>
<td>0.822</td>
<td>0.447</td>
<td>0.528</td>
<td>0.500</td>
<td>0.609</td>
<td>0.389</td>
<td>0.549</td>
<td>0.761</td>
</tr>
<tr>
<td>SC-NJW [43]</td>
<td>0.860/0.889</td>
<td>0.872/0.854</td>
<td>0.409/0.690</td>
<td>0.528/0.755</td>
<td>0.476</td>
<td>0.727</td>
<td>0.186</td>
<td>0.543</td>
<td>0.752</td>
</tr>
<tr>
<td>SC-ST [67]</td>
<td>0.673/0.895</td>
<td>0.706/0.858</td>
<td>0.342/0.726</td>
<td>0.445/0.756</td>
<td>0.416</td>
<td>0.611</td>
<td>0.431</td>
<td>0.581</td>
<td>0.620</td>
</tr>
<tr>
<td>SC-LS [3]</td>
<td>0.877</td>
<td>0.833</td>
<td>0.681</td>
<td>0.756</td>
<td>0.706</td>
<td>0.810</td>
<td>0.550</td>
<td>0.788</td>
<td>0.759</td>
</tr>
<tr>
<td>N-Cuts [52]</td>
<td>0.768/0.884</td>
<td>0.861/0.823</td>
<td>0.382/0.675</td>
<td>0.386/0.753</td>
<td>0.411</td>
<td>0.782</td>
<td>0.285</td>
<td>0.411</td>
<td>0.742</td>
</tr>
<tr>
<td>AC-Link [25]</td>
<td>0.512</td>
<td>0.711</td>
<td>0.579</td>
<td>0.662</td>
<td>0.686</td>
<td>0.643</td>
<td>0.168</td>
<td>0.545</td>
<td>0.738</td>
</tr>
<tr>
<td>AC-Zell [70]</td>
<td>0.954/0.911</td>
<td>0.963/0.913</td>
<td>0.774/0.799</td>
<td>0.810/0.768</td>
<td>0.017</td>
<td>0.755</td>
<td>0.351</td>
<td>0.910</td>
<td>0.733</td>
</tr>
<tr>
<td>AC-GDL [68]</td>
<td>0.945/0.937</td>
<td>0.954/0.929</td>
<td>0.854/0.824</td>
<td>0.864/0.844</td>
<td>0.017</td>
<td>0.755</td>
<td>0.351</td>
<td>0.934</td>
<td>0.622</td>
</tr>
<tr>
<td>AC-PIC [69]</td>
<td>0.950</td>
<td>0.964</td>
<td>0.840</td>
<td>0.853</td>
<td>0.017</td>
<td>0.750</td>
<td>0.415</td>
<td>0.902</td>
<td>0.697</td>
</tr>
<tr>
<td>NMF-LP [1]</td>
<td>0.720</td>
<td>0.783</td>
<td>0.435</td>
<td>0.467</td>
<td>0.452</td>
<td>0.560</td>
<td>0.346</td>
<td>0.491</td>
<td>0.720</td>
</tr>
<tr>
<td>NMF-D [57]</td>
<td>0.692</td>
<td>0.719</td>
<td>0.286</td>
<td>0.243</td>
<td>0.148</td>
<td>0.500</td>
<td>0.258</td>
<td>0.983</td>
<td>0.569</td>
</tr>
<tr>
<td>TSC-D [61]</td>
<td>-0.928</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-0.651</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OURS-SF</td>
<td>1.000</td>
<td>0.978</td>
<td>0.858</td>
<td>0.876</td>
<td>0.906</td>
<td>0.880</td>
<td>0.566</td>
<td>0.984</td>
<td>0.848</td>
</tr>
<tr>
<td>OURS-RC</td>
<td>1.000</td>
<td>0.985</td>
<td>0.913</td>
<td>0.915</td>
<td>0.913</td>
<td>0.877</td>
<td>0.574</td>
<td>1.000</td>
<td>0.848</td>
</tr>
</tbody>
</table>
Experiments: Clustering: Performance

+12.76% on AC to best performance of existing approaches averaged over all datasets

Table 10: Quantitative clustering performance (AC) for different algorithms using image intensities as input.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COIL20</th>
<th>COIL100</th>
<th>USPS</th>
<th>MNIST-test</th>
<th>MNIST-full</th>
<th>UMist</th>
<th>FRGC</th>
<th>CMU-PIE</th>
<th>YTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means [39]</td>
<td>0.665</td>
<td>0.580</td>
<td>0.467</td>
<td>0.560</td>
<td>0.564</td>
<td>0.419</td>
<td>0.327</td>
<td>0.246</td>
<td>0.548</td>
</tr>
<tr>
<td>SC-NJW [43]</td>
<td>0.641</td>
<td>0.544</td>
<td>0.413</td>
<td>0.220</td>
<td>0.502</td>
<td>0.551</td>
<td>0.178</td>
<td>0.255</td>
<td>0.551</td>
</tr>
<tr>
<td>SC-ST [67]</td>
<td>0.417</td>
<td>0.300</td>
<td>0.308</td>
<td>0.454</td>
<td>0.311</td>
<td>0.411</td>
<td>0.358</td>
<td>0.293</td>
<td>0.290</td>
</tr>
<tr>
<td>SC-LS [3]</td>
<td>0.717</td>
<td>0.609</td>
<td>0.659</td>
<td>0.740</td>
<td>0.714</td>
<td>0.568</td>
<td>0.407</td>
<td>0.549</td>
<td>0.544</td>
</tr>
<tr>
<td>N-Cuts [52]</td>
<td>0.544</td>
<td>0.577</td>
<td>0.314</td>
<td>0.304</td>
<td>0.327</td>
<td>0.550</td>
<td>0.235</td>
<td>0.155</td>
<td>0.536</td>
</tr>
<tr>
<td>AC-Link [25]</td>
<td>0.251</td>
<td>0.269</td>
<td>0.421</td>
<td>0.693</td>
<td>0.657</td>
<td>0.398</td>
<td>0.175</td>
<td>0.201</td>
<td>0.547</td>
</tr>
<tr>
<td>AC-Zell [70]</td>
<td>0.867</td>
<td>0.811</td>
<td>0.575</td>
<td>0.693</td>
<td>0.112</td>
<td>0.517</td>
<td>0.266</td>
<td>0.765</td>
<td>0.519</td>
</tr>
<tr>
<td>AC-GDL [68]</td>
<td>0.865</td>
<td>0.797</td>
<td>0.867</td>
<td>0.933</td>
<td>0.113</td>
<td>0.563</td>
<td>0.266</td>
<td>0.842</td>
<td>0.430</td>
</tr>
<tr>
<td>AC-PIC [69]</td>
<td>0.855</td>
<td>0.840</td>
<td>0.855</td>
<td>0.920</td>
<td>0.115</td>
<td>0.576</td>
<td>0.320</td>
<td>0.797</td>
<td>0.472</td>
</tr>
<tr>
<td>NMF-LP [1]</td>
<td>0.621</td>
<td>0.553</td>
<td>0.522</td>
<td>0.479</td>
<td>0.471</td>
<td>0.365</td>
<td>0.259</td>
<td>0.229</td>
<td>0.546</td>
</tr>
<tr>
<td>OURS-SF</td>
<td>1.000</td>
<td>0.894</td>
<td>0.922</td>
<td>0.940</td>
<td>0.959</td>
<td>0.809</td>
<td>0.461</td>
<td>0.980</td>
<td>0.684</td>
</tr>
<tr>
<td>OURS-RC</td>
<td>1.000</td>
<td>0.916</td>
<td>0.950</td>
<td>0.961</td>
<td>0.964</td>
<td>0.809</td>
<td>0.461</td>
<td>1.000</td>
<td>0.684</td>
</tr>
</tbody>
</table>
Experiments: Clustering: Performance

Average +21.5% on NMI

Table 4: Quantitative clustering performance (NMI) for different algorithms using our learned representations as inputs.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>COIL20</th>
<th>COIL100</th>
<th>USPS</th>
<th>MNIST-test</th>
<th>MNIST-full</th>
<th>UMist</th>
<th>FRGC</th>
<th>CMU-PIE</th>
<th>YTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means [39]</td>
<td>0.926</td>
<td>0.919</td>
<td>0.758</td>
<td>0.908</td>
<td>0.927</td>
<td>0.871</td>
<td>0.636</td>
<td>0.956</td>
<td>0.835</td>
</tr>
<tr>
<td>SC-NJW [43]</td>
<td>0.915</td>
<td>0.898</td>
<td>0.753</td>
<td>0.878</td>
<td>0.931</td>
<td>0.833</td>
<td>0.625</td>
<td>0.957</td>
<td>0.789</td>
</tr>
<tr>
<td>SC-ST [67]</td>
<td>0.959</td>
<td>0.922</td>
<td>0.741</td>
<td>0.911</td>
<td>0.906</td>
<td>0.847</td>
<td>0.651</td>
<td>0.938</td>
<td>0.741</td>
</tr>
<tr>
<td>SC-LS [3]</td>
<td>0.950</td>
<td>0.905</td>
<td>0.780</td>
<td>0.912</td>
<td>0.932</td>
<td>0.879</td>
<td>0.639</td>
<td>0.950</td>
<td>0.802</td>
</tr>
<tr>
<td>N-Cuts [52]</td>
<td>0.963</td>
<td>0.900</td>
<td>0.705</td>
<td>0.910</td>
<td>0.930</td>
<td>0.877</td>
<td>0.640</td>
<td>0.995</td>
<td>0.823</td>
</tr>
<tr>
<td>AC-Link [25]</td>
<td>0.896</td>
<td>0.884</td>
<td>0.783</td>
<td>0.901</td>
<td>0.918</td>
<td>0.872</td>
<td>0.621</td>
<td>0.990</td>
<td>0.803</td>
</tr>
<tr>
<td>AC-Zell [70]</td>
<td>1.000</td>
<td>0.989</td>
<td>0.910</td>
<td>0.893</td>
<td>0.919</td>
<td>0.870</td>
<td>0.551</td>
<td>1.000</td>
<td>0.821</td>
</tr>
<tr>
<td>AC-GDL [68]</td>
<td>1.000</td>
<td>0.985</td>
<td>0.913</td>
<td>0.915</td>
<td>0.913</td>
<td>0.870</td>
<td>0.574</td>
<td>1.000</td>
<td>0.842</td>
</tr>
<tr>
<td>AC-PIC [69]</td>
<td>1.000</td>
<td><strong>0.990</strong></td>
<td><strong>0.914</strong></td>
<td>0.909</td>
<td>0.907</td>
<td>0.870</td>
<td>0.553</td>
<td>1.000</td>
<td>0.829</td>
</tr>
<tr>
<td>NMF-LP [1]</td>
<td>0.855</td>
<td>0.834</td>
<td>0.729</td>
<td>0.905</td>
<td>0.926</td>
<td>0.854</td>
<td>0.575</td>
<td>0.690</td>
<td>0.788</td>
</tr>
</tbody>
</table>
Experiments: Clustering: Performance

Average +25.7% on NMI

Table 11: Quantitative clustering performance (AC) for different algorithms using our learned representations as inputs.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>( COIL20 )</th>
<th>( COIL100 )</th>
<th>( USPS )</th>
<th>( MNIST\text{-test} )</th>
<th>( MNIST\text{-full} )</th>
<th>( UMist )</th>
<th>( FRGC )</th>
<th>( CMU-PIE )</th>
<th>( YTF )</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means [39]</td>
<td>0.821</td>
<td>0.751</td>
<td>0.776</td>
<td>0.957</td>
<td>0.969</td>
<td>0.761</td>
<td>0.476</td>
<td>0.834</td>
<td>0.660</td>
</tr>
<tr>
<td>SC-NJW [43]</td>
<td>0.738</td>
<td>0.659</td>
<td>0.716</td>
<td>0.868</td>
<td>0.972</td>
<td>0.707</td>
<td>0.485</td>
<td>0.776</td>
<td>0.521</td>
</tr>
<tr>
<td>SC-ST [67]</td>
<td>0.851</td>
<td>0.705</td>
<td>0.661</td>
<td>0.960</td>
<td>0.958</td>
<td>0.697</td>
<td>0.496</td>
<td>0.896</td>
<td>0.575</td>
</tr>
<tr>
<td>SC-LS [3]</td>
<td>0.867</td>
<td>0.735</td>
<td>0.792</td>
<td>0.960</td>
<td>\textbf{0.973}</td>
<td>0.733</td>
<td>0.502</td>
<td>0.802</td>
<td>0.571</td>
</tr>
<tr>
<td>N-Cuts [52]</td>
<td>0.888</td>
<td>0.626</td>
<td>0.634</td>
<td>0.959</td>
<td>0.971</td>
<td>\textbf{0.798}</td>
<td>\textbf{0.504}</td>
<td>0.981</td>
<td>0.441</td>
</tr>
<tr>
<td>AC-Link [25]</td>
<td>0.678</td>
<td>0.539</td>
<td>0.773</td>
<td>0.955</td>
<td>0.964</td>
<td>0.795</td>
<td>0.495</td>
<td>0.947</td>
<td>0.602</td>
</tr>
<tr>
<td>AC-Zell [70]</td>
<td>\textbf{1.000}</td>
<td>0.931</td>
<td>0.879</td>
<td>0.879</td>
<td>0.969</td>
<td>0.790</td>
<td>0.449</td>
<td>\textbf{1.000}</td>
<td>0.644</td>
</tr>
<tr>
<td>AC-GDL [68]</td>
<td>\textbf{1.000}</td>
<td>0.920</td>
<td>0.949</td>
<td>\textbf{0.961}</td>
<td>0.878</td>
<td>0.790</td>
<td>0.461</td>
<td>\textbf{1.000}</td>
<td>\textbf{0.677}</td>
</tr>
<tr>
<td>AC-PIC [69]</td>
<td>\textbf{1.000}</td>
<td>\textbf{0.950}</td>
<td>\textbf{0.955}</td>
<td>0.958</td>
<td>0.882</td>
<td>0.790</td>
<td>0.438</td>
<td>\textbf{1.000}</td>
<td>0.652</td>
</tr>
<tr>
<td>NMF-LP [1]</td>
<td>0.769</td>
<td>0.603</td>
<td>0.778</td>
<td>0.955</td>
<td>0.970</td>
<td>0.725</td>
<td>0.481</td>
<td>0.504</td>
<td>0.575</td>
</tr>
</tbody>
</table>
Experiments: Clustering : Performance

Our clustering performance vs. that of existing clustering approaches using raw image data.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. NMI</td>
<td>0.598</td>
<td>0.595</td>
<td>0.751</td>
<td>0.559</td>
<td>0.696</td>
<td>0.699</td>
<td>0.710</td>
<td>0.553</td>
<td>0.489</td>
<td>0.877</td>
<td>0.892</td>
</tr>
<tr>
<td>Avg. AC</td>
<td>0.486</td>
<td>0.428</td>
<td>0.612</td>
<td>0.394</td>
<td>0.569</td>
<td>0.631</td>
<td>0.639</td>
<td>0.449</td>
<td>-</td>
<td>0.850</td>
<td>0.861</td>
</tr>
</tbody>
</table>

Clustering performance using our representation fed to existing clustering algorithms.

<table>
<thead>
<tr>
<th>Method</th>
<th>K-means</th>
<th>SC-NJW</th>
<th>SC-LS</th>
<th>N-Cuts</th>
<th>AC-Zell</th>
<th>AC-GDL</th>
<th>AC-PIC</th>
<th>NMF-LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. NMI</td>
<td>0.860</td>
<td>0.842</td>
<td>0.861</td>
<td>0.860</td>
<td>0.884</td>
<td><strong>0.890</strong></td>
<td>0.886</td>
<td>0.795</td>
</tr>
<tr>
<td>Avg. AC</td>
<td>0.778</td>
<td>0.716</td>
<td>0.771</td>
<td>0.756</td>
<td>0.838</td>
<td><strong>0.848</strong></td>
<td>0.847</td>
<td>0.707</td>
</tr>
</tbody>
</table>
Experiments: Clustering: Visualization

(a) Initial stage (421)  
(b) Middle stage (42)  
(c) Final stage (20)

(d) Initial stage (2162)  
(e) Middle stage (216)  
(f) Final stage (100)
Experiments: Clustering : Visualization

USPS

(g) Initial stage (2232)  (h) Middle stage (22)  (i) Final stage (10)

MNIST-test

(j) Initial stage (1762)  (k) Middle stage (22)  (l) Final stage (10)
Experiments: Clustering: Ablation study

Figure 7: Clustering performance (NMI) with different $\eta$ (left) and $K_s$ (right).
Experiments: Clustering: Verification

Figure 8: Average purity of K-nearest neighbour for varying values of $K$. Left is computed using raw image data, while right is computed using our learned representation.
Experiments: Clustering : Time Cost
Experiments: Representation Learning

Representation transfer

Table 5: NMI performance across COIL20 and COIL100.

<table>
<thead>
<tr>
<th>Layer</th>
<th>data</th>
<th>top(ip)</th>
<th>top-1</th>
<th>top-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>COIL20 → COIL100</td>
<td>0.924</td>
<td>0.927</td>
<td>0.939</td>
<td>0.934</td>
</tr>
<tr>
<td>COIL100 → COIL20</td>
<td>0.944</td>
<td>0.949</td>
<td>0.957</td>
<td>0.951</td>
</tr>
</tbody>
</table>

Table 6: NMI performance across MNIST-test and USPS.

<table>
<thead>
<tr>
<th>Layer</th>
<th>data</th>
<th>top(ip)</th>
<th>top-1</th>
<th>top-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST-test → USPS</td>
<td>0.874</td>
<td>0.892</td>
<td>0.907</td>
<td>0.908</td>
</tr>
<tr>
<td>USPS → MNIST-test</td>
<td>0.872</td>
<td>0.873</td>
<td>0.886</td>
<td>-</td>
</tr>
</tbody>
</table>

Representation learning

Testing generalization of our learnt (unsupervised) representation to LFW face verification.

<table>
<thead>
<tr>
<th>#Samples</th>
<th>10k</th>
<th>20k</th>
<th>30k</th>
<th>50k</th>
<th>100k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised</td>
<td>0.737</td>
<td>0.746</td>
<td>0.748</td>
<td>0.764</td>
<td>0.770</td>
</tr>
<tr>
<td>OURS</td>
<td>0.728</td>
<td>0.743</td>
<td><strong>0.750</strong></td>
<td>0.762</td>
<td>0.767</td>
</tr>
</tbody>
</table>

Evaluation on CIFAR-10 classification

<table>
<thead>
<tr>
<th>#Samples</th>
<th>K-means</th>
<th>conv1</th>
<th>conv2</th>
<th>conv1&amp;2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5k</td>
<td>62.81%</td>
<td>63.05%</td>
<td>63.10%</td>
<td>63.50%</td>
</tr>
<tr>
<td>10k</td>
<td>68.01%</td>
<td>68.30%</td>
<td>68.46%</td>
<td>69.11%</td>
</tr>
<tr>
<td>25k</td>
<td>74.01%</td>
<td>72.83%</td>
<td>72.93%</td>
<td>75.11%</td>
</tr>
<tr>
<td>50k (full set)</td>
<td><strong>76.59%</strong></td>
<td>74.68%</td>
<td>74.68%</td>
<td><strong>78.55%</strong></td>
</tr>
</tbody>
</table>
Extensions: Data Visualization

Figure 9: Visualization of 10,000 MNIST test samples in different embedding spaces.

Table 13: 1-nearest neighbor classification error on low-dimensional embedding of MNIST dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>2D</th>
<th>10D</th>
<th>30D</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [62]</td>
<td>0.782</td>
<td>0.430</td>
<td>0.108</td>
</tr>
<tr>
<td>NCA [48]</td>
<td>0.568</td>
<td>0.088</td>
<td>0.073</td>
</tr>
<tr>
<td>Autoencoder [22]</td>
<td>0.668</td>
<td>0.063</td>
<td><strong>0.027</strong></td>
</tr>
<tr>
<td>Param. t-SNE [38]</td>
<td>0.099</td>
<td>0.046</td>
<td><strong>0.027</strong></td>
</tr>
<tr>
<td>OURS</td>
<td><strong>0.067</strong></td>
<td><strong>0.019</strong></td>
<td><strong>0.027</strong></td>
</tr>
</tbody>
</table>

Table 14: Trustworthiness T(12) on low-dimensional embedding of MNIST dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>2D</th>
<th>10D</th>
<th>30D</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [62]</td>
<td>0.744</td>
<td>0.991</td>
<td>0.998</td>
</tr>
<tr>
<td>NCA [48]</td>
<td>0.721</td>
<td>0.968</td>
<td>0.971</td>
</tr>
<tr>
<td>Autoencoder [22]</td>
<td>0.729</td>
<td>0.996</td>
<td><strong>0.999</strong></td>
</tr>
<tr>
<td>Param. t-SNE [38]</td>
<td><strong>0.927</strong></td>
<td><strong>0.997</strong></td>
<td><strong>0.999</strong></td>
</tr>
<tr>
<td>Ours</td>
<td>0.768</td>
<td>0.936</td>
<td>0.975</td>
</tr>
</tbody>
</table>
Conclusion

• A new unsupervised learning method jointly with image clustering, cast the problem into a recurrent optimization problem;
• In the recurrent framework, clustering is conducted during forward pass, and representation learning is conducted during backward pass;
• A unified loss function in the forward pass and backward pass;
• Performance outperforms the state-of-the-art over a number of datasets;
• It can also learn plausible representations for image recognition.
Thanks!

https://github.com/jwyang/joint-unsupervised-learning