Discriminative Clustering for Image Co-segmentation

Armand Joulin    Francis Bach    Jean Ponce

INRIA    Ecole Normale Supérieure, Paris

March 2010
Introduction
Introduction

- **Task**: dividing simultaneously $q$ images in $k$ different segments
  - When $k = 2$, this reduces to dividing images into **foreground** and **background** regions.

- Our approach considers simultaneously the object recognition and the segmentation problems
  - **Semi-supervised discriminative clustering**

- Well-adapted to segmentation problems for 2 reasons:
  - Re-use existing features for supervised classification
  - Introduce spatial and local color-consistency constraints.
Prior work

- Rother et al. (2006), Hochbaum and Singh (2009)
- Identical or similar objects

Goal: objects are different instances from same object class
Outline

- Problem formulation
- Local consistency through Laplacian matrices
- Discriminative clustering
- Efficient optimization
- Results
Problem Notations

- Input: $q$ images.
  - Each image $i$ is reduced to a subsampled grid of $n_i$ pixels
  - For the $j$-th pixel (among the $\sum_{i=1}^{q} n_i$ pixels), we denote by:
    - $c^j \in \mathbb{R}^3$ its color,
    - $p^j \in \mathbb{R}^2$ its position within the corresponding image,
    - $x^j$ an additional $k$-dimensional feature vector.
Problem Notations

- **Input**: \( q \) images.
  - Each image \( i \) is reduced to a subsampled grid of \( n_i \) pixels
  - For the \( j \)-th pixel (among the \( \sum_{i=1}^{q} n_i \) pixels), we denote by :
    - \( c^j \in \mathbb{R}^3 \) its color,
    - \( p^j \in \mathbb{R}^2 \) its position within the corresponding image,
    - \( x^j \) an additional \( k \)-dimensional feature vector.

- **Goal**: find \( y = \) vector of size \( \sum_{i=1}^{q} n_i \) such that
  - \( y_j = 1 \) if the \( i \)-th pixel is in the foreground
  - \( -1 \) otherwise.
Problem Notations

feature $x_i$

color $c_i$

position $p_i$

spatial consistency based on $\Delta c$ and $\Delta p$
Co-segmenting images relies on two tasks:

1. **Within an image**: maximize local spatial and appearance consistency (**normalized cuts**)
2. **Over all images**: maximize the separability of two classes between different images (**semi-supervised SVMs**)
Local consistency through Laplacian matrices
Local consistency through Laplacian matrices
(Shi and Malik, 2000)

Spatial consistency within an image $i$ is enforced through a similarity matrix $W^i$

- $W^i$ is based on color features ($c^j$) and spatial position ($p^j$)
- Similarity between two pixels $l$ and $m$ within an image $i$:

$$W^i_{lm} = \exp(-\lambda_p \|p^m - p^l\|^2 - \lambda_c \|c^m - c^l\|^2),$$  \hspace{1cm} (1)
Local consistency through Laplacian matrices

(Shi and Malik, 2000)

Spatial consistency within an image $i$ is enforced through a similarity matrix $W^i$

- $W^i$ is based on color features ($c^j$) and spatial position ($p^j$)
- Similarity between two pixels $l$ and $m$ within an image $i$:

$$W^i_{lm} = \exp(-\lambda_p \Vert p^m - p^l \Vert^2 - \lambda_c \Vert c^m - c^l \Vert^2),$$

(1)

- Concatenate all similarity matrices into a block-diagonal matrix $W$ (with $W_i$ on its diagonal)
- Normalized Laplacian matrix $L = I - D^{-1/2}WD^{-1/2}$
Local consistency through Laplacian matrices

(Shi and Malik, 2000)

- Concatenate all similarity matrices into a block-diagonal matrix $W$ (with $W_i$ on its diagonal)
- Normalized Laplacian matrix $L = I_n - D^{-1/2}WD^{-1/2}$
- Minimizing $y^T Ly$ segments all images independently
Discriminative clustering

- Generative clustering (e.g., K-means)
Discriminative clustering

- Generative clustering (e.g., K-means)

- Discriminative clustering (Xu et al., 2002, Bach and Harchaoui, 2007)
Discriminative clustering

- Discriminative clustering framework based on positive definite kernels

- Histograms of features $\Rightarrow$ kernel matrix $K$ based on the $\chi^2$-distance:

$$K_{lm} = \exp \left( -\lambda_h \sum_{d=1}^{k} \frac{(x^l_d - x^m_d)^2}{x^l_d + x^m_d} \right),$$  \hspace{1cm} (2)

- Equivalent to mapping each of our $n$ $k$-dimensional vectors $x^j$, $j = 1, \ldots, n$ into a high-dimensional Hilbert space $\mathcal{F}$ through a feature map $\Phi$, so that $K_{ml} = \Phi(x^m) \top \Phi(x^l)$
Discriminative clustering

- Minimize with respect to both the predictor $f$ and the labels $y$ (Xu et al., 2002):

$$\frac{1}{n} \sum_{j=1}^{n} \ell(y_j, f^\top \Phi(x^j)) + \lambda_k \| f \|^2,$$  \hspace{1cm} (3)

where $\ell$ is a loss function.
Discriminative clustering

- Minimize with respect to both the predictor $f$ and the labels $y$ (Xu et al., 2002):

\[
\frac{1}{n} \sum_{j=1}^{n} \ell(y_j, f^\top \Phi(x_j)) + \lambda_k \|f\|^2,
\]

where $\ell$ is a loss function.

- Square loss function: $\ell(a, b) = (a - b)^2$, solution $f$ in closed form (Bach and Harchaoui, 2007)

\[
g(y) = \min_f \frac{1}{n} \sum_{j=1}^{n} \ell(y_j, f^\top \Phi(x_j)) + \lambda_k \|f\|^2 = \text{tr}(Ayy^\top)
\]

where $A = \lambda_k (I - \frac{1}{n}11^\top)(n\lambda_k I + K)^{-1}(I - \frac{1}{n}11^\top)$.

- **Linear in** $Y = yy^\top \in \mathbb{R}^{n \times n}$
Minimize with respect to the labels $y$:

$$g(y) = \min_{f} \frac{1}{n} \sum_{j=1}^{n} \ell(y_j, f^\top \Phi(x^j)) + \lambda_k \| f \|^2 = \text{tr}(Ay y^\top)$$

where $A = \lambda_k (I - \frac{1}{n}11^\top)(n\lambda_k I + K)^{-1}(I - \frac{1}{n}11^\top)$.

Linear in $Y = yy^\top \in \mathbb{R}^{n \times n}$
Discriminative semi-supervised clustering
Diffrac (Bach and Harchaoui, 2007)

Minimize with respect to the labels $y$:

$$g(y) = \min_{f} \frac{1}{n} \sum_{j=1}^{n} \ell(y_j, f^\top \Phi(x^j)) + \lambda_k \|f\|^2 = \text{tr}(Ayy^\top)$$

where $A = \lambda_k (I - \frac{1}{n} 11^\top)(n\lambda_k I + K)^{-1}(I - \frac{1}{n} 11^\top)$.

- **Linear in** $Y = yy^\top \in \mathbb{R}^{n \times n}$
- **Adding supervision on** $Y$ (positive and negative constraints)
- **Semi-supervised method** that is applicable to
  - High supervision (close to regular supervised learning)
  - Low supervision (close to clustering)
Diffrac - Semi-supervised classification

- Equivalence matrices $Y$ allow simple inclusion of prior knowledge (Xu et al., 2004, De Bie and Cristianini, 2006)
  - “must-link” constraints (positive constraints): $Y_{ij} = 1$
  - “must-not-link” constraints (negative constraints): $Y_{ij} = -1$
- Diffrac “works” with any amount of supervision
- Comparison with LDS (Chapelle & Zien, 2004)
Cluster size constraints

- Putting all pixels into a single class leads to perfect separation.
- Constrain the number of elements in each class (Xu et al., 2002).
Cluster size constraints

- Putting all pixels into a single class leads to perfect separation
  - Constrain the number of elements in each class (Xu et al., 2002)
- Multiple images:
  - Constrain the number of elements of each class in each image to be upper bounded by $\lambda_1$ and lower bounded by $\lambda_0$.
  - Denote $\delta_i \in \mathbb{R}^n$ the indicator vector of the $i$-th image
Problem formulation

- Combining:
  - spatial consistency through Laplacian matrix $L$
  - discriminative cost through matrix $A$ and cluster size constraints

\[
\min_{y \in \{-1,1\}^n} y^\top (A + \frac{\mu}{n} L) y,
\]

subject to $\forall i$, $\lambda_0 1 \leq (yy^\top + 11^\top) \delta_i \leq \lambda_1 1$.

- Combinatorial optimization problem
  - Convex relaxation with semi-definite programming (Goemans and Williamson, 1995)
Optimization - Convex Relaxation

$$\min_{y \in \{-1,1\}^n} \text{tr}\left((A + \frac{\mu}{n}L)yy^\top\right),$$
subject to \quad \forall i, \quad \lambda_0 1 \leq (yy^\top + 11^\top)\delta_i \leq \lambda_1 1.$$

- Reparameterize problem with $Y = yy^\top$
- $Y$ referred to as the equivalence matrix
  - $Y_{ij} = 1$ if points $i$ and $j$ belong to the same cluster
  - $Y_{ij} = -1$ if points $i$ and $j$ do not belong to the same cluster
- $Y$ is symmetric, positive semidefinite, with diagonal equal to one, and unit rank.
Denote by $\mathcal{E}$ the *elliptope*, i.e., the convex set defined by:

$$\mathcal{E} = \{ Y \in \mathbb{R}^{n \times n} , \ Y = Y^\top , \ \text{diag}(Y) = 1 , \ Y \succeq 0 \} ,$$

Reformulated optimization problem :

$$\min_{Y \in \mathcal{E}} \text{tr}( Y ( A + \frac{\mu}{n} L ) ) ,$$

subject to \ 
$$\forall i , \ \lambda_0 1 \leq ( Y + 11^\top ) \delta_i \leq \lambda_1 1 \ \ \text{rank}(Y) = 1$$

Rank constraint is not convex

**Convex relaxation by removing the rank constraint**
Optimization

\[
\min_{Y \in \mathcal{E}} \text{tr}(Y(A + \frac{\mu}{n}L)),
\]

subject to \( \forall i, \lambda_0 1 \leq (Y + 11^T)\delta_i \leq \lambda_1 1 \)

- SDP: semidefinite program (Boyd and Vandenberghe, 2002)
- General purpose toolboxes would solve this problem in \( O(n^7) \)
- Bach and Harchaoui (2007) considers a partial dualization technique that scales up to thousands of data points.
- To gain another order of magnitude: optimization through low-rank matrices (Journée et al, 2008)
Efficient low-rank optimization (Journée et al, 2008)

- Replace constraints by penalization $\Rightarrow$ optimization of a convex function $f(Y)$ on the elliptope $\mathcal{E}$.
- Empirically: global solution has low rank $r$
- Property: a local minimum of $f(Y)$ over the rank constrained elliptope
  \[ \mathcal{E}_d = \{ Y \in \mathcal{E}, \text{rank}(Y) = d \} \]
  is a global minimum of $f(Y)$ over $\mathcal{E}$, if $d > r$. 
Efficient low-rank optimization (Journée et al, 2008)

- Replace constraints by penalization ⇒ optimization of a convex function $f(Y)$ on the elliptope $\mathcal{E}$.
- Empirically: global solution has low rank $r$
- Property: a local minimum of $f(Y)$ over the rank constrained elliptope
  
  $$\mathcal{E}_d = \{ Y \in \mathcal{E}, \text{rank}(Y) = d \}$$

  is a global minimum of $f(Y)$ over $\mathcal{E}$, if $d > r$.
- Adaptive procedure to automatically find $r$
- Manifold-based trust-region method for a given $d$ (Absil et al., 2008)
Low-rank optimization (Journée et al., 2008)

- **Final (combinatorial) goal**: minimize $f(Y)$ over the rank-one constrained elliptope $\mathcal{E}_1 = \{ Y \in \mathcal{E}, \text{rank}(Y) = 1 \}$

- **Convex relaxation**: minimize $f(Y)$ over the unconstrained elliptope $\mathcal{E}$

- **Subproblems**: minimize $f(Y)$ over the rank-$d$ constrained elliptope $\mathcal{E}_d = \{ Y \in \mathcal{E}, \text{rank}(Y) = d \}$ for $d \geq 2$
  - It is a Riemannian manifold for $d \geq 2$
  - If $d$ is large enough, there is no local minima
  - Find a local minimum with trust-region method

- **Adaptive procedure**:
  - Start with $d = 2$
  - Find local minimum over $\mathcal{E}_d = \{ Y \in \mathcal{E}, \text{rank}(Y) = d \}$
  - Check global optimality condition
  - Stop or augment $d$
Preclustering

- Cost function $f$ uses a full $n \times n$ matrix $A + (\mu/n)L$
  $\Rightarrow$ memory issues
- To reduce the total number of pixels
  - superpixels obtained from an oversegmentation of our images
    (watershed, Meyer, 2001)
In order to retrieve \( y \in \{-1, 1\} \) from our relaxed solution \( Y \), we compute the largest eigenvector \( e \in \mathbb{R}^n \) of \( Y \).

Final clustering is \( y = \text{sign}(e) \).

Other techniques could be used (e.g., randomized rounding).

Additional post-processing to remove some artefacts
Method overview (co-segmentation on two bear images)

From left to right: input images, over-segmentations, scores obtained by our algorithm and co-segmentations.
Results

Results on two different problems:

- **Simple problems**: images with foreground objects which are identical or very similar in appearance and with few images to co-segment.

- **Hard problems**: images whose foreground objects exhibit higher appearance variations and with more images to co-segment (up to 30).
Results - similar objects
Results - similar objects
Results - similar objects
Results - similar classes - Faces
Results - similar classes - Cows
Results - similar classes - Horses
Results - similar classes - Cats
Results - similar classes - Bikes
Results - similar classes - Planes
Comparison with MN-cut (Cour, Bénédit, and Shi, 2005)

- Segmentation accuracies on the Weizman horses and MSRC databases.

<table>
<thead>
<tr>
<th>class</th>
<th>#</th>
<th>cosegm.</th>
<th>independent</th>
<th>Ncut</th>
<th>uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cars (front)</td>
<td>6</td>
<td>87.65 ± 0.1</td>
<td>89.6 ± 0.1</td>
<td>51.4 ± 1.8</td>
<td>64.0 ± 0.1</td>
</tr>
<tr>
<td>Cars (back)</td>
<td>6</td>
<td>85.1 ± 0.2</td>
<td>83.7 ± 0.5</td>
<td>54.1 ± 0.8</td>
<td>71.3 ± 0.2</td>
</tr>
<tr>
<td>Face</td>
<td>30</td>
<td>84.3 ± 0.7</td>
<td>72.4 ± 1.3</td>
<td>67.7 ± 1.2</td>
<td>60.4 ± 0.7</td>
</tr>
<tr>
<td>Cow</td>
<td>30</td>
<td>81.6 ± 1.4</td>
<td>78.5 ± 1.8</td>
<td>60.1 ± 2.6</td>
<td>66.3 ± 1.7</td>
</tr>
<tr>
<td>Horse</td>
<td>30</td>
<td>80.1 ± 0.7</td>
<td>77.5 ± 1.9</td>
<td>50.1 ± 0.9</td>
<td>68.6 ± 1.9</td>
</tr>
<tr>
<td>Cat</td>
<td>24</td>
<td>74.4 ± 2.8</td>
<td>71.3 ± 1.3</td>
<td>59.8 ± 2.0</td>
<td>59.2 ± 2.0</td>
</tr>
<tr>
<td>Plane</td>
<td>30</td>
<td>73.8 ± 0.9</td>
<td>62.5 ± 1.9</td>
<td>51.9 ± 0.5</td>
<td>75.9 ± 2.0</td>
</tr>
<tr>
<td>Bike</td>
<td>30</td>
<td>63.3 ± 0.5</td>
<td>61.1 ± 0.4</td>
<td>60.7 ± 2.6</td>
<td>59.0 ± 0.6</td>
</tr>
</tbody>
</table>
Comparing co-segmentation with independent segmentations

From left to right: original image, multiscale normalized cut, our algorithm on a single image, our algorithm on 30 images.
Conclusion

- Co-segmentation through semi-supervised discriminative clustering
  1. Within an image: maximize local spatial and appearance consistency (*normalized cuts*)
  2. Over all images: maximize the separability of two classes between different images (*semi-supervised SVMs*)
Conclusion

- Co-segmentation through semi-supervised discriminative clustering
  1. Within an image: maximize local spatial and appearance consistency \textit{(normalized cuts)}
  2. Over all images: maximize the separability of two classes between different images \textit{(semi-supervised SVMs)}

- Future work
  - Add negative images
  - More than 2 classes
  - Feature selection
  - Scale up to hundred of thousands
  - Change the loss function