Energy Minimization for Segmentation in Computer Vision

Meng Tang, Dmitrii Marin, Ismail Ben Ayed, Yuri Boykov
Outline

• Clustering/segmentation methods
  • K-means, GrabCut, Normalized Cut
• K-means for clustering/segmentation
  • Probabilistic K-means
  • Kernel K-means
• Our Kernel Cut: Normalized Cut+ MRF
  • How to optimize

• Applications
  • Motion, RGBD segmentation, image clustering etc.
K-means Clustering

0. Initialize cluster centers
1. Assign points to closest clusters
2. Re-compute means

Repeat (1) and (2) until convergence
K-means Segmentation: Color Quantization
Graph Cut Segmentation

\[ E(S, \theta_0, \theta_1) = \sum_{p \in \Omega} -\log Pr(I_p|\theta_{sp}) + \sum_{pq \in N} w_{pq} \cdot [s_p \neq s_q] \]

GrabCut [Rother et al. 2004]

MRF regularization
Markov Random Field (MRF) (graphical models)

$$\sum_{c \in \mathcal{F}} E_c(S_c)$$

*Potts Model*: edge alignment

interactive segmentation [Boykov & Jolly, 2001]
Markov Random Field (MRF) (graphical models)

\[ \sum_{c \in F} E_c(S_c) \]

Robust \( P^n \) Potts: bin consistency

semantic segmentation [Kohli & Torr 2009] [Gould 2014]
Normalized Cut (NC)

$$NC(S) = -\sum_k \frac{a_{SS}(S_{kk}, S_k)}{a_{SS}(S_k, \Omega)}$$

[Arbelaez, Maire, Fowlkes & Malik, 2010]
K-means minimizes “sum of squared distance (SSD)”

Assume $K=2$

$$SSD = \sum_{p \in S} \| I_p - \mu_S \|^2 + \sum_{p \in \overline{S}} \| I_p - \mu_{\overline{S}} \|^2$$

$\mu_S$, $\mu_{\overline{S}}$ are in color space as $I_p$
“squared distance” as “log-likelihoods”

\[
\sum_{p \in S} \| I_p - \mu_S \|^2 + \sum_{p \in \overline{S}} \| I_p - \mu_{\overline{S}} \|^2
\]

\[
= - \sum_{p \in S} \ln \mathcal{N}(I_p | \mu_S) - \sum_{p \in \overline{S}} \ln \mathcal{N}(I_p | \mu_{\overline{S}})
\]

single Gaussian of \textit{fixed} covariance
Probabilistic K-means with Descriptive Models $\theta$

$$- \sum_{p \in S} \ln \Pr(I_p | \theta_S) - \sum_{p \in \bar{S}} \ln \Pr(I_p | \theta_{\bar{S}})$$

model-fittings: ML estimation of $\theta$

$$\theta_S = \{\mu_S, \sigma_S\}$$
**Probabilistic** K-means with Descriptive Models $\theta$

$$- \sum_{p \in S} \ln \Pr(I_p | \theta_S) - \sum_{p \in \bar{S}} \ln \Pr(I_p | \theta_{\bar{S}})$$

Optimized in GrabCut (with smoothness term)

$\theta_S = \{\ldots, \mu^i_S, \sigma^i_S, \ldots\}$

Gaussian Mixture Models (GMM)
K-means

\[ \sum_{p \in S} \| I_p - \mu_S \|^2 + \sum_{p \in \bar{S}} \| I_p - \mu_{\bar{S}} \|^2 \]

probabilistic K-means
make models more complex

kernel K-means
make data more complex

\[ - \sum_{p \in S} \ln \Pr(I_p | \theta_S) - \sum_{p \in \bar{S}} \ln \Pr(I_p | \theta_{\bar{S}}) \]

\[ \sum_{p \in S} \| \phi(I_p) - \mu_S \|^2 + \sum_{p \in \bar{S}} \| \phi(I_p) - \mu_{\bar{S}} \|^2 \]

kernel K-means = K-means in feature space
Why is it called “kernel” K-means?

• Cluster variance

\[
\sum_{p \in S} \| \phi(I_p) - \mu_S \|^2 = |S| \cdot \text{var}(S)
\]

\[
\sim \frac{1}{|S|} \sum_{pq \in S} \| \phi(I_p) - \phi(I_q) \|^2
\]

• Replace norms by dot products

\[
\sim - \frac{1}{|S|} \sum_{pq \in S} \langle \phi(I_p), \phi(I_q) \rangle
\]

“kernel trick”

\[ k_{pq} \]

kernel, affinity, similarity, dot product

\begin{align*}
\text{explicit kernel } K & \iff \\
\text{implicit embedding } \phi
\end{align*}
Why is it called “kernel” K-means?

- Cluster variance
  \[ \sum_{p \in S} \| \phi(I_p) - \mu_S \|^2 = |S| \cdot \text{var}(S) \]
  \[ \approx \frac{1}{|S|} \sum_{pq \in S} \| \phi(I_p) - \phi(I_q) \|^2 \]
  \[ \approx - \frac{\sum_{pq \in S} k_{pq}}{|S|} \]

- Replace norms by dot products

“kernel trick”  \[ k_{pq} \]

\text{kernel, affinity, similarity, dot product}

\textit{explicit} kernel \( K \)  \( \Leftrightarrow \)  \textit{implicit} embedding \( \phi \)
Normalized Cut is Kernel K-means

\[ N\text{cut}(S, \bar{S}) = \frac{\text{cut}(S, \bar{S})}{\text{cut}(S, \Omega)} + \frac{\text{cut}(S, \bar{S})}{\text{cut}(\bar{S}, \Omega)} \]

\[ = 2 - \frac{\text{cut}(S, S)}{\text{cut}(S, \Omega)} - \frac{\text{cut}(\bar{S}, \bar{S})}{\text{cut}(\bar{S}, \Omega)} \]

\[ = 2 - \frac{\sum_{pq \in S} k_{pq}}{d|S|} - \frac{\sum_{pq \in \bar{S}} k_{pq}}{d|\bar{S}|} \]

(assuming constant degree d)

\[ \sim \frac{1}{d} \left( \sum_{p \in S} \| \phi(I_p) - \mu_S \|^2 + \sum_{p \in \bar{S}} \| \phi(I_p) - \mu_{\bar{S}} \|^2 \right) \]
“World Map” of Segmentation

**Probabilistic K-means**
- (ML model fitting)
  - geometric fitting
  - histogram fitting
  - GMM fitting
  - gamma fitting
  - Gibbs fitting

**kernel K-means**
- (average distortion AD or association AA)
  - Gaussian kernel K-means
  - Normalized Cuts
  - Spectral clustering

**weak kernel clustering**
- (unary) Hilbertian distortion
- K-modes
  - (mean-shift)

**pKM**
- Probabilistic K-means
- (ML model fitting)

**kKM**
- kernel K-means

Distortion formula:
- $D_{pq} = \| I_p - I_q \|_d$

- p.s.d. kernel $\| \|_k^2$
- distortion $\| \|_d$
Probabilistic K-means or Kernel K-means

(log-likelihoods) (normalized cut)
Secrets of GrabCut

model fitting (e.g. GMM)  
GrabCut [Rother, Kolmogorov, Blake, 2004]  
(color space clustering)  
(probablistic k-means [Kearns, Mansour & Ng, UAI’97])

poor clustering 
(overfitting & local minima)

ML term for $\theta^k$

$$\sum_k \sum_{p \in S^k} \ln P(I_p | \theta^k) + \sum_{pq \in \Lambda} w_{pq} [s_p \neq s_q]$$

typical MRF for segmentation:
Normalized Cut clustering

\[ NC(S) + \sum_{pq \in \mathcal{N}} w \cdot [s_p \neq s_q] \]

NC for colors

Normalized Cut

color space clustering

good clustering
Our proposal: **Normalized Cut** + **MRF**

\[
E(S) = \sum_k -\frac{S^k S^k'}{d' S^k} + \gamma \sum_{c \in \mathcal{F}} E_c(S_c)
\]

- **balanced clustering**
- **regularization constraints**
Why MRF for Normalized Cut?

previous NC approach:
weak edge alignment
post-processing (e.g. [Arbelaez et al., 2011])

semi-supervision is challenging
cannot-link
must-link
reformulation of NC constrained eigen problem

Our approach: NC + Potts MRF  NC + Potts + seeds MRF

[Yu & Shi 2004]
[Eriksson et al. 2010]
[Maji et al. 2011]
[Chew et al. 2015]
Why *MRF* for Normalized Cut?

How to incorporate group priors?

- Our approach: \( \text{NC + Robust } P^n \text{ Potts MRF} \)

How many clusters?

- 8 clusters
- 3 clusters

Our approach: \( \text{NC + label costs MRF} \)
Bound optimization, in general

\[ E(S_t) \]

\[ E(S_{t+1}) \]

\[ S_t \]

\[ S_{t+1} \]

guaranteed energy decrease
Bound for our joint energy

\[ E(S') = \sum_k -S^k \frac{A S^k}{d^k} + \gamma \sum_{c \in F} E_c(S_c) \]

\[ \forall \text{ unary bound for NC} \]

\[ A_t(S') = \sum_{p \in \Omega} U_p(S_p) + \gamma \sum_{c \in F} E_c(S_c) \]

we propose *kernel bound* and *spectral bound* for NC
Kernel bound for NC

Lemma 1 (concavity)

$$NC(S) = \sum_k -\frac{S^k A S^k}{d^k S^k}$$

Function $e : \mathbb{R}^{\Omega} \rightarrow \mathbb{R}$ is concave over region $S^k > 0$ given p.s.d. affinity matrix $A := [A_{pq}]$.

First-order Taylor expansion:

$$e(S^k_t) + \nabla e(S^k_t) \cdot (S^k_t - S^k_t)$$

equivalently kernel k-means for NC [Dhillon et al., 2004]
Our Kernel Cut algorithm

\[ E(S) = \sum_{k} - \frac{S^{k'}}{d'} A S^{k} + \gamma \sum_{c \in F} E_c(S_c) \]

\( \land \) unary bound for NC (Kernel Bound or Spectral Bound)

\[ A_t(S) = \sum_{p \in \Omega} U_p(S_p) + \gamma \sum_{c \in F} E_c(S_c) \]

iterate

\[ S_{t+1} = \arg \min_{S} A_t(S) \] (move-making and graph cuts [Boykov, Veksler, Zabih, 2001])
Experiments: MRF helps Normalized Cut using image tags (e.g. beach, car) to help image clustering

NC + robust $P^n$ Potts

+ with $knn$ kernel on deep features
Experiments: Normalized Cut helps MRF

Fig. motion segmentation using RGB, location (XY) and motion (M). “+xy” means with MRF
NC with increasing label cost
Robustness to smoothness term

separating similar objects
More Experiments

**Potts model** improves edge alignment

Spectral Clustering (no Potts)  
Our Kernel Cut (NC with Potts)  
Our Spectral Cut (NC with Potts)

Fig. 1. RGBD segmentation

GrabCut  
Kernel Cut
Medical image segmentation (3D)
Conclusion

- Probabilistic K-means vs Kernel K-means
- NC equivalent to Kernel K-means
- new unary kernel and spectral bounds for NC
- can combine NC with any MRF constraints
- can combine MRF with balanced clustering
- MRF with features of any dimension (RGBD, RGBM, RGBXYM, deep,...)
Take home code for \[ NC(S) = -\sum_{k} \frac{assoc(S^k, S^k)}{assoc(S^k, \Omega)} \]

```python
linearization = S_t' * A * S_t / (d' * S_t)^2 * d - A * S_t * 2 / (d' * S_t);
```