Course: Model, Learning, and Inference: Lecture 10

Alan Yuille
Department of Statistics, UCLA
Los Angeles, CA 90095
yuille@stat.ucla.edu

Abstract

Clustering Algorithms for Segmentation
NOTE: NOT FOR DISTRIBUTION!!

1 Introduction

We describe two new important approach to image segmentation which are based on clustering algorithms.

SWA is very efficient (takes a few seconds to run on an image) and produces a hierarchical representation by grouping together pixels which have similar *affinities*. It does not give a complete segmentation of the image – instead it gives a set of *salient segments*. Here a segment is a set of pixels and it is salient if the affinity is high between pixels within the segment and low across the boundaries of the segment.

2 Segmentation by Weighted Aggregation (SWA)

SWA is a hierarchical clustering algorithm for detecting salient regions in images. This will not give a complete segmentation. It may produce salient regions at different scales which can partially overlap. For example, at one scale you find the back window of a car and at another scale you find the faces of people looking through the window.

Here is the basic algorithm. We start with a grid Λ^0 which is the image lattice.

Define affinities $w^0_{ij} = \exp\{-\gamma|I_i - I_j|\}$ between neighbouring pixels in Λ^0 where I_i, I_j are the image intensities.

Next select a subset $\Lambda^1 \subset \Lambda^0$ obeying the properties:

$$\forall i \in \Lambda^0/\Lambda^1, \quad \sum_{k\in\Lambda^1} w^0_{ik} \geq \beta \sum_{j\in\Lambda^0} w_{ij}. \tag{1}$$

Intuitively, a pixel i is left behind ($i \in \Lambda^0/\Lambda^1$) if it is represented by nodes in Λ^1 – has sufficiently high affinity to nodes in Λ^1. In short, the nodes Λ^1 are representative of Λ^0.

The algorithm to select Λ^1 proceeds by grouping the pixels $\{i\}$ into blocks $\{C_i\}$ then selecting an element from each block to construct Λ^1. The algorithm scans through all the image pixels sequentially. Let $C^{(i-1)}$ denote the set of blocks before testing pixel i. Check the inequality: $\max_{j\in C^{(i-1)}} w^0_{ij} \geq \tau \sum_l w^0_{il}$. If the inequality is satisfied, set $C^i = C^{(i-1)}$. If not, the set $C^i = C^{(i-1)} \bigcup \{i\}$.

Choose β, τ so that $|\Lambda^1| \approx (1/2)|\Lambda^0|$.

The pixels in Λ^1 represent the pixels in Λ^0. To make this more precise, define an interpolation matrix.

$$p^0_{ik} = \frac{w^0_{ik}}{\sum_{l\in\Lambda^1} w^0_{il}},\ k \in \Lambda^1, i \in \Lambda^0/\Lambda^1,$$
$$p^0_{ii} = 1,\ p^0_{ik} = 0\ \ i \in \Lambda^0 \bigcap \Lambda^1. \tag{2}$$

Intuitively p^0_{ik} is the probability that node $i \in \Lambda^0$ is represented by node $k \in \Lambda^1$. Note $\sum_{k\in\Lambda^1} p^0_{ik} = 1,\ \forall i \in \Lambda^0$.

Now define new affinities of Λ^1 by:

$$w^1_{kl} = \sum_{i,j\in\Lambda^0} p^0_{ik} w^0_{ij} p^0_{jl}. \tag{3}$$

Then proceed as before starting with Λ^1 to create a series of subsets ... $\subset \Lambda^2 \subset \Lambda^1 \subset \Lambda^1 \subset \Lambda^0$. With affinities defined as follows:

$$w^M_{kl} = \sum_{i,j\in\Lambda^0} p^{M-1}_{ik} w^{M-1}_{ij} p^{M-1}_{jl}$$
$$p^{M-1}_{ik} = \frac{w^{M-1}_{ik}}{\sum_{l\in\Lambda^M} w^{M-1}_{il}}. \tag{4}$$

How to justify this coarsening? Original justification is from algebraic multi-grid (AMG) method for implementing differential equations. AMG chooses a lattice adaptively unlike geometric multi-grid which uses a fixed grid. More importantly, it produces *segments* with high *salience*. A segment is a set of nodes defined by a discrete variable $u^0_i = 1$ if i is in this segment and $u^0_i = 0$ otherwise. The salience is defined by:

$$\Gamma[u^0] = \frac{\sum_{i,j\in\Lambda^0} w^0_{ij}(u^0_i - u^0_j)^2}{\sum_{i,j\in\Lambda^0} w^0_{ij} u^0_i u^0_j}. \tag{5}$$

Here $\sum_{i,j\in\Lambda^0} w^0_{ij}(u^0_i - u^0_j)^2$ is the cost of the affinity across the boundaries of the salient while $\sum_{i,j\in\Lambda^0} w^0_{ij} u^0_i u^0_j$ is a measure of the affinity within the segment.

Claim: segments with low Γ correspond to salient regions of the image (i.e., high affinity within the region and low affinity across the boundary).

Intuitively, SWA coarsens the grid so that segments with low salience are preserved. Here is an approximate argument. Define a segment S^1 on Λ^1 by u^1 so that $u^1_k = 1$ if $k \in S^1$ and $u^l_k = 0$ otherwise.

This has salience at level Λ^1:

$$\Gamma^1[u^1] = \frac{\sum_{i,j\in\Lambda^1} w^1_{ij}(u^1_i - u^1_j)^2}{\sum_{i,j\in\Lambda^1} w^1_{ij} u^1_i u^1_j}. \tag{6}$$

It corresponds to a segment S^0 on Λ^0 defined by u^0 given by $u^0_i = \sum_{k\in\Lambda^1} p^0_{ik} u^1_k,\ i \in \Lambda^0$ with saliency:

$$\Gamma^0[u^0] = \frac{\sum_{i,j\in\Lambda^0} w^0_{ij}(u^0_i - u^0_j)^2}{\sum_{i,j\in\Lambda^0} w^0_{ij} u^0_i u^0_j}. \tag{7}$$

Claim: if Γ^0 has low salience, then (i) the u^0_i are close to binary values, (ii) $\Gamma^1[u^1] \approx \Gamma^0[u^0]$. This can be verified algebraically.

So far the affinities have been based purely on local measures such as $\exp\{-\gamma|I_i - I_j|\}$ but at higher levels in the hierarchy we can start using statistics defined on the image regions specified by the pixels. This is performed by multiplying the standard affinities:

$$w_{ij}^l = w_{ij}^l \exp\{-\rho|G_i - G_j|\}, \tag{8}$$

where $|G_i - G_j|$ is a measure of the difference of these statistics. For a node i at level l we use the relationship $u_i^{l-1} = \sum_{k \in \Lambda^l} p_{ik}^{l-1} u_k^{l-1}$ recursively to determine the region in the image associated to i. Note this is soft association, so the statistics may be weighted accordingly.

3 Normalized Cuts

Normalized cuts starts with the saliency measure $\Gamma[u^0]$ given by $\Gamma[u^0] = \frac{\sum_{i,j \in \Lambda^0} w_{ij}^0 (u_i^0 - u_j^0)^2}{\sum_{i,j \in \Lambda^0} w_{ij}^0 u_i^0 u_j^0}$.

This can be related to a continuous optimization problem where the variables $\{ui\}$ are allowed to take continuous values. This continuous problem can be solved by linear algebra:

$$E[\{u_i\}; \lambda] = \sum_{i,j \in \Lambda^0} w_{ij}^0 (u_i^0 - u_j^0)^2 - \lambda \{ \sum_{i,j \in \Lambda^0} w_{ij}^0 u_i^0 u_j^0 - 1\}, \tag{9}$$

where λ is a Lagrange multiplier. This takes advantage of the fact that the saliency measure is independent of the overall magnitude of the $\{u_i\}$ and so is independent of a scaling constant.

$E[\{u_i\}; \lambda]$ can be re-expressed in form $E = \vec{u}^T \mathbf{A} \vec{u} + \lambda \{\vec{u}^T \mathbf{B} \vec{u} - 1\}$ (where $\mathbf{A}$ and $\mathbf{B}$ are computed from the $\{w_{ij}\}$. Hence the solution can be obtained by solving the generalized eigenvalue problem:

$$\mathbf{A}\vec{u} = \lambda \mathbf{B} \vec{u}. \tag{10}$$

with the scale of $\vec{u}$ determined by the constraint $\vec{u}^T \mathbf{B} \vec{u} = 1$.

This solution $\vec{u}^*$ will be continuous-valued. So it can be scaled to lie in the range $[0, 1]$ and then thresholded to give a binary-valued output.

This method will give the "best" segment. Then you will need to re-run it to obtain the second best.

In practice, normalized cuts requires considerable pre-processing – filtering, clustering, and so on – in order to obtain reliable segmentations. See later papers.

4 Partitioning by Clustering and Encoding

Another related method by Yi Ma *et al* is also based on clustering and has been shown to be very effective on labelled datasets (e.g. Berkeley).

The data is $W = \{w_i, i = 1, ..., m\}$ where each $w_i = (w_{i1}, w_{i2}, ..., w_{in})$ is a vector in n-dim space.

We seek the best encoding into k partitions $\{V_i : i = 1, ..., k\}$ with $\bigcup_{i=1}^k V_i = W$, $V_i \bigcap V_j = \emptyset$ $i \neq j$ and each $w_i \in V_j$ for some j. The best encoding is defined to minimize:

$$L(V_1, ..., V_k) = \sum_{i=1}^k L(V_i) + |V_i|(-\log(|V_i|/m)), \tag{11}$$

where the cost of encoding the vectors $v_1, ..., v_{m(k)}$ in a set V is given by:

$$L(V) = \frac{m(k) + n}{2} \log_2 \det\{I + \frac{n}{\epsilon^2 m(k)} V V^T\}, \tag{12}$$

where it is assumed that the set has zero mean VV^T is a measure of the covariance. (The approach can be extended to include a cost for encoding the mean).

The cost for encoding each partition – see equation (12) – assumes that the data is generated by an (unknown) Gaussian and is encoded with tolerance ϵ. This is lossy compression. The cost for encoding all the partitions – see equation (11) – includes a term for the number k of partitions.

The algorithm proceeds as follows:

Input data $W = (w_1, ..., w_m) \in R^{n \times m}$ and distortion $\epsilon > 0$.

Initialize $S = \{S = \{w\} | w \in W\}$.

While $|S| > 1$ do,

Choose distinct sets $S_1, S_2 \in S$ such that $L(S_1 \bigcup S_2) - L(S_1, S_2)$ is maximal.

If $L(S_1 \bigcup S_2) - L(S_1, S_2) \geq 0$ then END.

Else $S = \{S\{S_1, S_2\}\} \bigcup \{S_1 \bigcup S_2\}$.

The algorithm starts by assigning every point to a unique partition. Then it proceeds by merging partitions provided this decreases the overall coding cost (CHECK COST OF NUMBER OF REGIONS). It finishes when it is impossible to merge any partitions without increasing the cost.

The effectiveness of the algorithm depends on the size of the parameter ϵ. The approach also requires obtaining vectors w that are (approximately) Gaussian encoded.

Yi Ma *et al* apply this to image segmentation. They compute 8×8 windows in the images. They select $1,000$ of these windows at random and do PCA to extract the first 8 principal components. Then they construct vectors w for each window by projecting the intensity onto these first 8 components. They run their algorithm and obtain a surprisingly good segmentation.

They improve this method by breaking up the image into small homogeneous regions by using a pre-processing algorithm such as normalized cuts. To each of these *super-pixels* they associate a feature vector as above. Then run the clustering algorithm. This works faster (since the algorithm is only run on super-pixels and not on the entire image) and gives better performance.