Segmentation

Goal: ~~Divi~~ Divide or partition an image into homogeneous pieces based on intensity, color, texture, ~~etc.~~ motion, etc.

A.K.A. - partitioning, grouping, clustering

Dual Problem: finding edges.

Also gives/results in: compact representation of an image emphasizing the properties that make it interesting.

Will discuss/consider three techniques:

1. Region growing
2. Splitting and Merging
3. Mumford-Shah

One algorithmic & computational consideration is how image data will be analyzed for segmentation:

i) local techniques:

a pixel is placed in a partition based on its properties or those of its neighbors.

ii) global techniques:

a pixel is grouped into a partition based on the properties of large numbers of pixels distributed throughout the image.

iii) splitting and merging techniques:

typically utilizes a graph (or some other partitioning structure) to represent regions. local & global properties used to split/merge graph components.

Formalizing the segmentation process:

we have an image defined on the domain D.

a region R is a subset of D, $R \subset D$.

- connectedness: if $x_i, x_j \in R$, then x_i and x_j are connected iff there is a path completely in R going from x_i to x_j.

  * ~~this~~ under discretization, this can vary based on definition of "neighbor".

  * R is a connected region if $\forall x_i, x_j \in R$, x_i and x_j are connected.

Regions can be used to partition D, and hence segment I.

A partition of D is a collection of regions $\{R_i\}$ such that $D = \bigcup_i R_i$ and $R_i \cap R_j = \phi$ for $i \neq j$.

- In segmentation, the goal is to find a partition such that each region R_i in the partition is homogeneous (based on some image property/classification).

Suppose there were a boolean function deciding on the homogeneity of regions:

$$H(R) = \begin{cases} \text{true} & \text{if homogeneous} \\ \text{false} & \text{otherwise.} \end{cases}$$

A restatement of the goal is:

find partition $\{R_i\}$ such that (1) $H(R_i) = \text{true}$ $\forall i$ & (2) $H(R_i \cup R_j) = \text{false}$ $\forall i \neq j$.

→ ~~probably also~~ want to minimize the cardinality of the partition,
e.g., the # regions.

* Already, we have hints of a segmentation method.
- what if boolean function H is replaced by some energy function E whose energy varies monotonically according to degree of homogeneity?

Segmentation via Thresholding:

a very basic/almost primitive approach is thresholding.
assume that image is basically composed of an object (or multiple similar objects)
and a (constant) background.

pick criteria for separating object from background based on image intensity (in scalar imagery).

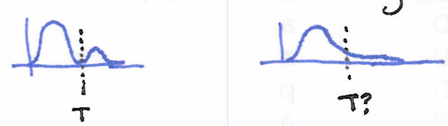
find a T such that $R_1 = \{ \underline{x} : I(\underline{x}) > T \}$

$$R_2 = D \setminus R_1$$

how does one find T? Good question

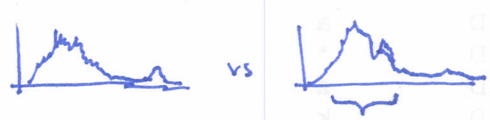
1) subjectively, by testing out various choices.

2) histogram analysis (want bimodal, but uni-modal may suffice)



→ will depend on

1) discretization of histogram (more important for multi-level thresholding)
2) "smoothness" of histogram
   ↳ may have to smooth histogram.
   "how much?" is a good question.



is this one or two peaks?

other problems:

1) objects or background have spatially varying graylevels
2) image is "rich", e.g. fails to be uni/bi-modal.
3) sufficient target noise (imaging ~~[illegible]~~ noise).

possible solutions:

1) high-pass filter the image to de-emphasize low-frequency variation.
2) apply multi-level thresholding strategy.
3) incorporate a probabilistic model.
4) use a spatially varying threshold.

⋮

Last Time:

- introduced segmentation
- basic thresholding for bimodal images
  - pros/cons

Now:

continue thread of segmentation

- spatially varying thresholding
- multi-level thresholding (k-means, Bayesian)
- Splitting & Merging approach.

Spatially varying threshold method:

1) divide image up into rectangular images (size is your discretion)
   - possibly overlapping (slightly), but then may still need to determine dominant region.

2) compute threshold for each subimage
   - should have bimodal histogram
   - if unimodal, then get threshold by interpolating neighbors'.

3) apply appropriate threshold to each region.

K-means multi-level thresholding

- assumes each object has pixel values clustered around a specific mean values

  and that # of clusters/means is known in advance.

  – knowing this quantity, k, is an open question & most likely depends on application.

K-means approach seeks to minimize:

$$\mathcal{E} = \sum_{c \in \mathcal{C}} \int_D \delta_1(c = s(x)) \, \| I(x) - \mu_c \|^2 \, dx$$

when $\delta_1(a) = \begin{cases} 1 & \text{if } a \text{ true} \\ 0 & \text{else} \end{cases}$

least square error of image values $I(x)$ versus segmentation $s(x)$ and associated mean $\mu_{s(x)}$

another way of writing energy is

$$\mathcal{E} = \int_D \| I(x) - \mu_{s(x)} \| \, dx$$

$\mathcal{C}$ – set of classes , $\{1, \ldots, \#\text{ of partitioned regions}\}$

$|\mathcal{C}| = k$, hence k-means.

$s : D \to \mathcal{C}$ – segmentation map

$I : D \to M$ – image function

μ_c for $c \in \mathcal{C}$ – mean associated to region R_c (e.g. class c)

Two ways to start

Ⓐ guess initial means & start segmentation by minimizing $\mathcal{E}$

$\Rightarrow$ results in an initial segmentation

Ⓑ start w/ an initial segmentation

Algorithm proper:

① w/ segmentation compute means

② for each pixel, determine classification

③ goto ① until satisfied or no change occurs.

- can be sensitive to initial segmentation

  $\rightarrow$ but at least small # of attracting solutions

- requires one to know initial means or segmentation

- ~~if done right can be somewhat~~

- there are lots of extensions of this idea.

  - example being non-trivial norm

$$\|x\|_2^2 = x^T \cdot x$$

why not

$$\|x\|_{\Sigma,2}^2 = x^T \Sigma x$$

pos. def. / symmetric

idea: $\Sigma = \mathrm{diag}(\sigma_1^{-2}, \dots, \sigma_k^{-2})$ or $\mathrm{diag}(\sigma_1^{-1}, \dots, \sigma_k^{-1})$

sometimes good, sometimes bad.

Bayesian Segmentation:

Recall that I mentioned energy vs. probability split:

maximizing e^{-x^2} is like minimizing x^2.

Why go to probabilistic version?

- perhaps handles noise better
- can introduce a prior using Bayes' rule.
- can it add flexibility?

first, lets just work out the probabilistic version of k-means,

then see how it can be expanded/improved.

our goal of minimizing $E = \int_D \| I(x) - \mu_{S(x)} \|^2 dx$

is like asking for: $S(x) = \arg\min_{c \in C} \| I(x) - \mu_c \|^2$

as a probability, this is like (not quite equal due to normalization factor):

$$S(x) = \arg\overset{\max}{\cancel{\min}}_{c \in C} \frac{1}{(2\pi)^{n/2} \det(\Sigma_c)^{1/2}} e^{-\frac{1}{2}(I(x)-\mu_c)^T \Sigma_c (I(x)-\mu_c)}$$

(vector case)

$$s(x) = \arg\overset{\max}{\cancel{\min}}_{c \in C} \frac{1}{\sqrt{2\pi}\,\sigma_c} e^{-\frac{(I(x)-\mu_c)^2}{2\sigma_c^2}} \quad \text{(scalar case)}$$

therefore, we can claim that

minimizing energy $\mathcal{E}$ is "like" maximizing probability of different classification outcomes.

let's setup the problem specification and see what it means in practice.

for each class $c \in \mathcal{C}$, there is a mean & standard deviation μ_c, σ_c

from which the following probability may be defined,

$$P_c(\mathbb{I}(x)) = \frac{1}{\sqrt{2\pi}\,\sigma_c} e^{-\frac{(\mathbb{I}(x)-\mu_c)^2}{2\sigma_c^2}}$$

segmentation is class selection at each pixel that maximizes probability

$$s(x) = \arg\max_{c \in \mathcal{C}} \cancel{\frac{1}{\sqrt{2\pi}\,\sigma_c} e^{-(\mathbb{I}(x)-\mu}} \; P_c(x)$$

if $\sigma_c = 1$, then this is the same as k-means.

but, now that we are in probability space, we can do a lot more with what we have.

$\Rightarrow$

$$P(c_i=c \mid v_i=v) = \frac{P(v_i=v \mid c_i=c)\cdot P(c_i=c)}{P(v_i=v)}$$

normalization factor so that ~~prob~~ conditional probability sums to 1 properly.

$\Rightarrow$

$$P(c_i=c \mid v_i=v) \propto P(v_i=v \mid c_i=c)\cdot P(c_i=c)$$

what about this value?
we don't really have it?

$P(c_i=c)$ is assumed to exist & is called the <u>prior</u>:

- $\circledast$ if $P(c_i=c) = \frac{1}{|\mathcal{C}|}$ (homogeneous priors), then this is like running k-means.

- $\circledast$ otherwise $P(c_i=c)$ could represent any prior information about the classification of the image.

for example, if we are iterating to find the solution, $P(c_i=c)$ could be the probabilities from the previous Bayesian step.

It is easier to understand the process from a systems perspective:

Let's change notation to simplify the mathematical expression of our ideas & let's more properly define our quantities.

we have an image $I(x)$

which will be vectorized so that we use i instead of coordinate pair ~~x,y~~ ~~x=(x,y)~~ $x = (x^1, x^2)$ AKA (x, y).

the image value at pixel i is denoted by v_i

the classification of pixel i is denoted by c_i.

our goal is to determine the classification probability given the image value,

$$\cancel{P(v_i = v)} \quad P(c_i = c \mid v_i = v)$$

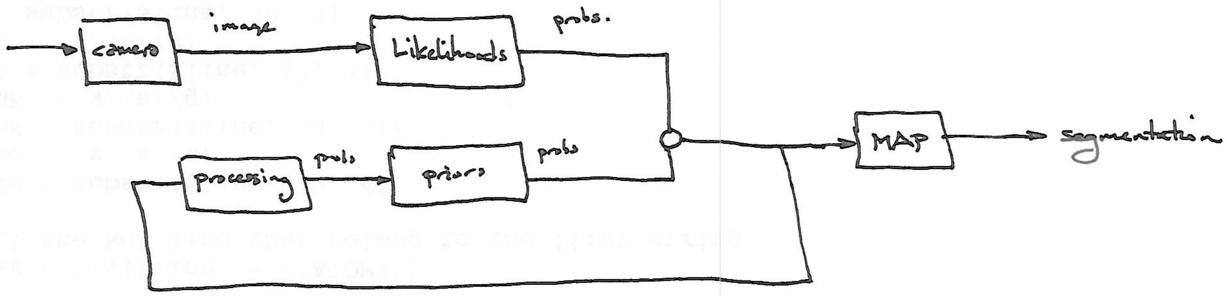
how do we compute this probability given that we don't have an equation for it?

Well, we do have

$$P(v_i = v \mid c_i = c) = \frac{1}{\sqrt{2\pi}\,\mu_c} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

and we know that Bayes' rule is

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$



incomplete/rough flow diagram.

process to steady-state

- steady-state solution is the final segmentation.

basic steps

① start w/ initial guess ~~prob~~ stat. params + prior

② get image and find $P(v|c)$

③ uses Bayes' rule to compute $P(c|v)$

④ segment via MAP.

⑤ set $P(c) = P(c|v)$ (& update parameters)

- repeat ②-⑤ until steady-state

can be modified further.

→ we can do all kinds of stuff in the processing step

- can impose local consistency by smoothing probabilities (different types of smoothers), different support)

⇒ prevents singular outliers from corrupting segmentation.

is actually well founded using Markov Random Field Theory.

- this is for static imagery. what if we have a dynamic image sequence?
  - can expand flow diagram. ← area of research.

Region Growing: Split & Merge

→ more of a computer science technique

often requires specialized data structures to encode the regions

Recall the mostly mathematical definition of segmentation:

find a partition $\{R_i\}$ of D such that:

1. $H(R_i)$ is true
2. $H(R_i \cup R_j)$ is false for $i \neq j$.

Sample homogeneity boolean functions are:

local: $H(R_\ell) = \begin{cases} \text{true} & \text{if all neighboring pixels } \underline{x}_n \text{ of } x \text{ satisfy } |I(\underline{x}) - I(\underline{x}_n)| < \tau \\ \text{false} & \text{otherwise} \end{cases}$

global/non-local: $H(R_g) = \begin{cases} \text{true} & \text{if points in } R \text{ pass unimodal/peak test.} \\ \text{false} & \text{otherwise} \end{cases}$

⇒ Leads to the region growing algorithm, which examines when to split/merge regions

1. if $H(R_i)$ is false, then region not homogeneous → split

   break it up into subregions

2. if $H(R_i \cup R_j)$ is true for an $i \neq j$, then union is homogeneous → merge.

   join them to form one region

1) Locally adaptive threshold

divide image up into rectangular subimages & compute threshold for each subimage

- if local histogram is not bi-modal (e.g. better to say if unimodal), then has no threshold, need to interpolate thresholds from neighboring subimages that are bimodal

final picture is segmented by applying thresholds for the subimages.

2) Pyramiding? Hierarchical Image Representation.

Region growing occurs at coarsest level.

When algorithm has terminated at one resolution level,

pixels at boundaries are dissected w/ their regions.

Region growing applied to these pixels at higher resolution level.

---

split & merge

① $H(R_i) = \text{true}$, if $H(R_i) = \text{false}$, time to split.

② $H(R_i \cup R_j) = \text{false}$, if $H(R_i \cup R_j) = \text{true}$, time to merge.

Region growing via Split & Merge

1. Pick ~~grid structure~~ initial partition & homogeneity property

2. Check regions

   splitting: check for $H(R_i) = \text{false}$

   merging: check for i,j such that $H(R_i \cup R_j) = \text{true}$

3. Repeat until no change in partition

Tough parts:

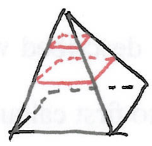
* how to define homogeneity for a given image
* how to split a region
* combinatorial nightmare to check for merging.
  (has high computational complexity, potentially)

- more computer science type of problem since one also has to consider the data structure to use for the region growing procedure.

  ↳ don't want this to be focus of class.

~~[illegible]~~

Can incorporate pyramidal structure into the procedure



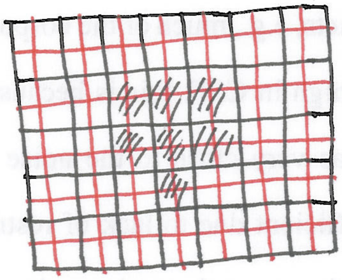
① goto coarsest level & pick ~~post~~ initial partition.

② perform region growing ~~to~~ current level.

③ move up to finer level & repeat

④ stop when at finest level.

→ the region growing is done fully for the coarse level, then only on the boundary regions when moving to a ~~coarser~~ finer level.

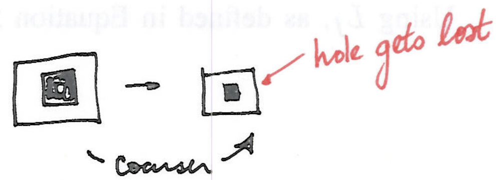


- outer white regions will obviously not change
- inner white regions will not change either

⇒ only boundary areas may change.

→ be careful: coarsest level will determine scale of classification. Some regions smaller than coarsest scale may get lost if only checking boundaries

e.g.



Implementation Issues:

Many of the segmentation algorithms require special data structures in their implementation.

A region can be encoded by

- the collection of pixels that form it, or
- the collection of pixels that form its boundary

(maybe even store both)

Spatial proximity of regions given by adjacency graphs

Segmentation then involves the appropriate manipulation of the data structure elements.

Energy Based Methods:

Mumford-Shah Segmentation & more ...

Probabilistic | Energy-based methods typically have two parts:

① data fitting term

② prior or knowledge-based term

→ imposes certain characteristic onto minimal solution

· can be past history

explicit assumptions

in probabilistic setting, used Bayes' Rule

const factor, doesn't depend on A.

$$P(A|B) = P(B|A)\,P(A)\,/\,P(B)$$

prior info about our choice.

data fitting conditional probability

probability of outcome given data

what about energy setting? well, we can take logarithm to get energy terms

$$\underbrace{-\log(P(A|B))}_{\text{energy of outcome given current observation}} = -\log(P(B|A)) - \log(P(A)) + \cancel{\log(P(B))}$$

energy of observation given an outcome

data based term

energy of outcome (changes w/choice of outcome)

Constant that does not affect minimization. can ignore. given by observation, so does not change.

$$\Rightarrow \quad \mathcal{E} = \overset{\text{data}}{\mathcal{E}_d} + \overset{\text{prior}}{\mathcal{E}_p}$$

where have we seen this before? OPTICAL FLOW!!!

So, I've sort of lied to you, we can add prior knowledge type terms to energies ... but they typically incorporate constraints or regularization and not past history.

⇒ usually ends up making problem well posed e.g., "unique solution".

(more like reduces ambiguity)

So, really we are ~~trying to do~~ going to try out k-means plus regularization, then take it to the next level!

But first start out simple: binary/bimodal version.

Ising Model (Binary/Bimodal Images)

* from statistical physics (describes 2D crystal lattice of iron atoms subject to external magnetic field)

  - first ~~model~~ mathematical model to rigorously proven to model phase transitions.

→ assumes that image is bimodal. ⟹ $P = \{R, D \setminus R\}$

prior model: regions R, of objects minimizes # of blobs

⟹ region R minimizes perimeter length.

e.g. $\underbrace{|\partial R|}_{\text{length of perimeter/boundary } \partial R.}$ is minimal

⟹

$$\mathcal{E}_p(R) = \nu\, |\partial R| , \quad \nu > 0.$$

data model:

define $\chi(x) = \begin{cases} \mu_1 & x \in R \\ \mu_0 & x \notin R \end{cases}$ ← characteristic function.

- Ising really used $\mu_1 = 1,\ \mu_0 = 0$ (but let's generalize!)
- assume that image has sufficient noise to make thresholding ~~differ~~ tough/impossible.

$$\mathcal{E}_d(I, R) = \mu \int_D (I - \chi)^2\, dx$$

⟹

$\mathcal{E} = \mathcal{E}_d + \mathcal{E}_p$ ~~minimize~~ find R minimizing $\mathcal{E}$ given I.

thus, Ising model goes along the following lines:

$$\text{minimize} \quad \mathcal{E} = \mu \int_D (I - X)^2 dx + \nu |\partial R| \quad \text{given } I.$$

$$\text{e.g.} \quad \arg\min_R \mathcal{E}(I, R)$$

how does one implement this procedure? that's a great question!

- let consider other energy functionals first, then get back to this!

Cartoon Model

discrete: Geman & Geman

continuous: Mumford & Shah

assume that image can be described by a collection of shaded regions, within which the intensity changes slowly, but across boundaries the intensity varies quickly (in general abruptly).

$\Rightarrow$

seek a piecewise smooth cartoon $J : D \to \mathbb{R}$ of the image $I : D \to \mathbb{R}$.

J is discontinuous at some parts (boundaries of regions).

$\Rightarrow$

there is a curve $\Gamma = \{x : J(x) \text{ is discontinuous}\}$

on $D \setminus \Gamma$, the cartoo J is smooth.

Prior model: ① want to minimize Γ,

$$\mathcal{E}_p^1 = \nu |\Gamma|$$

② want to be smooth off of Γ,

$$\mathcal{E}_p^2 = \int_{D \setminus \Gamma} \phi(\|\nabla J\|)\,dx$$

where ϕ is convex (& minimal at origin).

$\Rightarrow$

$$\mathcal{E} = \mathcal{E}_p^1 + \mathcal{E}_p^2$$

Data model: $\mathcal{E}_d = \int_D (I-J)^2 dx$

$\Rightarrow$

$$\mathcal{E} = \int_D (I-J)^2 dx + \nu|\Gamma| + \int_{D\setminus\Gamma} \phi(\|\nabla J\|)\,dx$$

recap:

Approximate I by a piecewise smooth function J so ~~that~~ as to minimize

(i) difference between I & J,

(ii) gradient of J away from edges

(iii) length of curve Γ defined by discontinuities of J.

This energy functional attempts to encode for the same qualities as the region growing method

$$\mathcal{E} = \underbrace{\int_D (I-J)^2 dx}_{\text{try to fit data.}} + \int_{D\setminus\Gamma} \phi(\|\nabla J\|)\,dx + \nu|\Gamma|$$

have approximation be smooth $\Rightarrow$ try to have some kind of uniformity / homogeneity (local).

minimize length of boundaries $\Rightarrow$ minimize # distinct regions

If we demand that J be piecewise-constant, then $\|\nabla J\|\big|_{D\setminus\Gamma} = 0$

$\Rightarrow$

$$\mathcal{E}(J,\Gamma) = \int_D (I-J)^2 dx + \nu|\Gamma|$$

When D is discrete, finite domain

⇒

E has a minimum since there is a countable, finite # of possibilities of Γ (may be huge, but it's finite),

~~and~~ it has continuous dependence on J,

and goes to ∞ is any value of J goes to ∞.

In continuous manifestation, well-posedness of solution is a question to consider

- DiGiorgi, Ambrosio (Italian school).

typically, we consider cts version to get idea about behavior of discrete version. if cts well-behaved, discrete typically is.

- want to avoid meaningless zig-zagging.