Image Analysis

7. Image Segmentation II

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Business Economics and Information Systems
& Institute for Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

1. Image Segmentation as Clustering

2. Multivariate Kernel Density Estimation

3. Mean Shift Segmentation

Neighborhoods

For a discrete image

$$f \in \mathbb{R}^{n \times m}$$

we call

$$I := \{1, \ldots, n\} \times \{1, \ldots, m\}$$

the **set of pixels** (also **grid**) and each $(x, y) \in I$ a **pixel**.

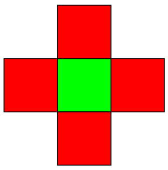
There are two different neighborhood systems in use:
Two different pixels (x, y) and (x', y') are called **neighbors**

$$(x, y) \sim (x', y')$$

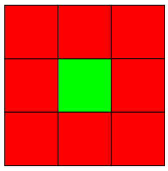
if

$$(|x - x'| = 1 \text{ and } y = y')$$
$$\text{or } (x = x' \text{ and } |x - x'| = 1)$$

$$|x - x'| \leq 1 \text{ and } |y - y'| \leq 1$$



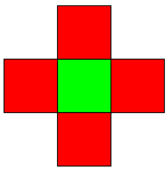
(**4-neighborhood**)

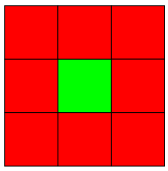


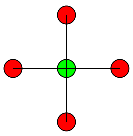
(**8-neighborhood**)

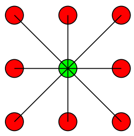
Neighborhoods (2/2)

The neighborhoods define the neighbor graph on the pixels I:









Regions

A **region** of an image $f \in \mathbb{R}^I$ is a subset of its pixels

$$R \subseteq I$$

that is connected w.r.t. the neighborhood graph.

This means: For each two pixels $(x, y), (x', y') \in R$ of the region there is a sequence

$$(x_1, y_1), \ldots, (x_T, y_T)$$

of pixels in R that

- starts in (x, y): $(x_1, y_1) = (x, y)$,
- ends in (x', y'): $(x_T, y_T) = (x', y')$,
- and where two consecutive pixels in the sequence are neighbors:

$$(x_t, y_t) \sim (x_{t+1}, y_{t+1}) \quad \forall t = 1, \ldots, T-1$$

The (Unsupervised) Image Segmentation Problem

Given an image $f \in \mathbb{R}^{n \times m}$, find regions

$$R_1, R_2, \ldots, R_k \subseteq \{1, \ldots, n\} \times \{1, \ldots, m\}$$

such that

- the intensities in these regions

$$f_{|R_i}$$

  are homogenuous, i.e., have similar values or vary only slowly,

  and/or

- there are edges at the borders of the region.

The Supervised Image Segmentation Problem

Given a set of segmented images, i.e., images with attached segments

$$(f^1, s^1), (f^2, s^2), \dots, (f^L, s^L)$$

with images $f^\ell \in \mathbb{R}^{I^\ell}$ and a set of segments

$$s^\ell = \{S_1^\ell, S_2^\ell, \dots, S_{K^\ell}^\ell\}$$

with regions $S_k^\ell \subseteq I^\ell$ (for $\ell = 1, \dots, L$, $k = 1, \dots, K^\ell$).

Learn a **segmentation model**

$$\hat{s} : \bigcup_{n,m} \mathbb{R}^{n \times m} \to \mathcal{P}(\text{regions}(\mathbb{N} \times \mathbb{N}))$$

that assigns to each image f a set of regions $\hat{s}(f)$, s.t. for the given images as well as for new images (from the same distribution) a suitable error measure between the true segments s^ℓ and the predicted segments $\hat{s}(f^\ell)$ is minimal.

Image Segmentation as Clustering

The (unsupervised) image segmentation problem can be viewed as a clustering problem on the pixels of the image.

To do so, one has to describe pixels (x, y) by **pixel feature vectors** $\phi(x, y)$.

Simple feature vector: **pixel position**:

$$\phi(x, y) := (x, y)$$

This obviously does not describe pixels accurately as it does not depend on the intensities at all.

Another simple feature vector: **pixel intensity**:

$$\phi(x, y) := f(x, y)$$

This obviously does not describe pixels accurately as it does not depend on the pixel position, so it does not take any spatial relationships into account.

Pixel Features

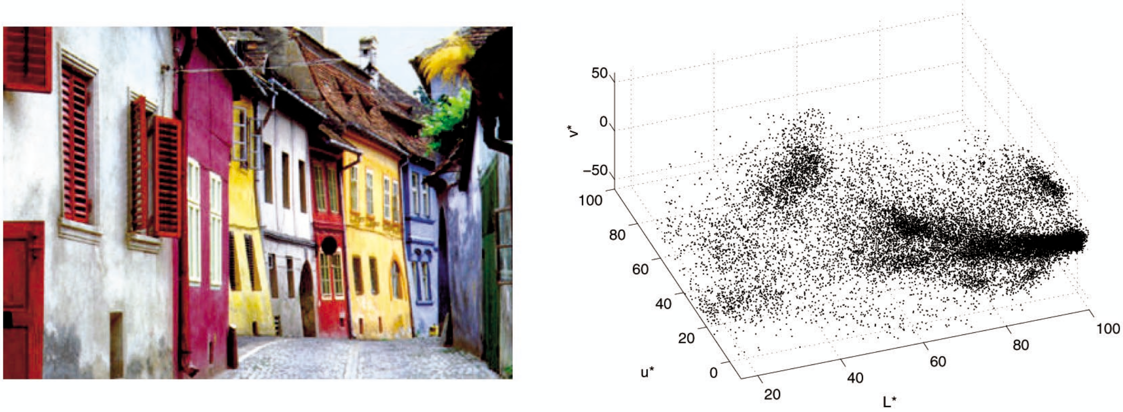
A simple useful pixel feature vector can be made from **a combination of position and intensity**:

$$\phi(x, y) := (x, y, f(x, y))$$

It also makes sense to include information about the intensities of neighborhood pixels.

Now any clustering algorithm can be applied to cluster the pixel feature vectors.

Pixel Features / Example



(a) (b)

Example of a feature space. (a) A 400×276 color image. (b) Corresponding L*u*v* color space with $110,400$ data points.

(from [CM02])

Distance-based Clustering

For example, with a simple distance measure on pixel features such as

$$d(\begin{pmatrix} x \\ y \\ f(x,y) \end{pmatrix}, \begin{pmatrix} x' \\ y' \\ f(x',y') \end{pmatrix}) := \sqrt{(x-x')^2 + (y-y')^2 + \lambda(f(x,y) - f(x',y'))^2}$$

with a suitable weight $\lambda \in \mathbb{R}^+$, one can apply standard distance-based clustering algorithms such as **k-means** and **hierarchical clustering algorithms**.

This approach will join pixels in a cluster that are both, close to each other and have similar intensity values. But it does not guarantee that the clusters actually form regions.

1. Image Segmentation as Clustering

2. Multivariate Kernel Density Estimation

3. Mean Shift Segmentation

Multivariate Density Estimation / Empirical Distribution

Assume, we have N data points

$$x_1, x_2, \ldots, x_N$$

in $\mathbb{R}^d$ and want to estimate the density $p(x)$ of their underlying distribution.

Simple estimator:

$$\hat{p}(x) := \begin{cases} \frac{1}{N}, & \text{if } x \in \{x_1, \ldots, x_N\} \\ 0, & \text{else} \end{cases}$$

– Assumes we have seen all possible points already.

– Assumes all points are equally likely.

– In general, too close to training data.

Note: for the general case where there could be duplicate points, i.e., $i \neq j$ with $x_i = x_j$, the formula is

$$\hat{p}(x) := |\{x_i \,|\, x = x_i, i = 1, \ldots, N\}|/N$$

Multivariate Density Estimation / Uniform Distribution

Another simple estimator:

$$\hat{p}(x) := \begin{cases} \frac{1}{\text{vol}(R)}, & \text{if } x \in R \\ 0, & \text{else} \end{cases}$$

where $R \subseteq \mathbb{R}^d$ is a region that contains $x_1, \ldots, x_N$, e.g.,

$$R := [\min_i x_{i,1}, \max_i x_{i,1}] \times \cdots \times [\min_i x_{i,d}, \max_i x_{i,d}]$$

$$\text{vol}(R) = (\max_i x_{i,1} - \min_i x_{i,1}) \cdots (\max_i x_{i,d} - \min_i x_{i,d})$$

– Assumes all points are equally likely.

– In general, does not take into account training data sufficiently, e.g., cannot distinguish between dense and sparse regions,

Multivariate Density Estimation / Frequency Count

A more appropriate estimator:

$$\hat{p}(x) := c \cdot |\{x_i \,|\, d(x, x_i) \leq d_0\}|$$

where

- $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a distance measure on $\mathbb{R}^d$,
- d_0 is a constant distant threshold and
- c is a constant that makes $\hat{p}$ to integrate to 1.

The frequeny count estimator

- can distinguish between dense and sparse regions,
- equals the empirical distribution for $d_0 = 0$,
- gets smoother for increasing d_0,
- is a step function.

The distance measure defines the shape of the neighborhood regions (euclidean $\rightsquigarrow$ ball, L_1 $\rightsquigarrow$ cube, etc.).

Multivariate Density Estimation / Parzen Window

To avoid the steps whenever an example enters or leaves the neighborhood region, one can weight the contribution of the points by their distance. Such a weight is called a kernel (aka similarity measure):

$$K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$$

Parzen window (aka **kernel density estimator**):

$$\hat{p}(x) := c \frac{1}{N} \sum_i K(x, x_i)$$

where

- c is a constant that makes $\hat{p}$ to integrate to 1.

The simplest kernel is the scalar product (called **linear kernel**):

$$K(x, y) := \langle x, y \rangle = \sum_{j=1}^{d} x_j y_j$$

Multivariate Kernel Density Estimation / Kernel Properties

Most kernels are **shift-invariant**, i.e., they can be written as

$$K(x, y) = k(x - y)$$

for a kernel norm

$$k : \mathbb{R}^d \to \mathbb{R}$$

that is maximal at 0.

Many kernels are **radially symmetric**, i.e., they can be written as

$$K(x, y) = k(\langle x - y, x - y \rangle) = k(||x - y||^2)$$

with a 1-dimensional kernel norm $k : \mathbb{R} \to \mathbb{R}$.

Often kernel norms k with unbounded support are **truncated**:

$$k^{\text{truncated}}(x) := \begin{cases} k(x), & \text{if } ||x|| < 1 \\ 0, & \text{else} \end{cases}$$

Multivariate Kernel Density Estimation / Kernel Properties (2/2)

For a kernel norm k, the maximal absolute radius with a non-vanishing weight

$$\text{bandwidth}(k) := \sup\{||x|| \mid x \in \mathbb{R}^d, k(x) \neq 0\}$$

is called **bandwidth of kernel norm k**.

If k is a kernel norm with bandwidth 1 and $h \in \mathbb{R}^+$, then

$$k'(x) := k(\frac{x}{h})$$

is a kernel norm with bandwidth h.

This way, kernels can be made more narrow or wider.

Multivariate Kernel Density Estimation / Kernel Examples

Examples:

Normal Kernel Norm:

$$k(x) := c\, e^{-\frac{1}{2}x^2}$$

Epanechnikov Kernel Norm:

$$k(x) := \begin{cases} c\,(1 - x^2), & \text{if } |x| < 1 \\ 0, & \text{else} \end{cases}$$

1. Image Segmentation as Clustering

2. Multivariate Kernel Density Estimation

3. Mean Shift Segmentation

Density Gradient Decent

For clustering a set of points, one has to identify dense regions, i.e., local maxima (called **modes**) of the estimated density

$$\hat{p}(x) := c\frac{1}{N}\sum_i k(||x - x_i||^2)$$

For this, one looks for zeros of the density gradient :

$$\frac{\partial}{\partial x}\hat{p}(x) \stackrel{!}{=} 0$$

Density Gradient Decent

$$\begin{aligned}\frac{\partial}{\partial x}\hat{p}(x) &= c\frac{1}{N}\sum_i (\frac{\partial}{\partial x}k)(||x - x_i||^2)2(x - x_i)\\ &= 2c\frac{1}{N}\sum_i \alpha_i(x - x_i)\\ &= 2c\frac{1}{N}((\sum_i \alpha_i)x - \sum_i \alpha_i x_i) \stackrel{!}{=} 0\end{aligned}$$

with

$$\alpha_i := (\frac{\partial}{\partial x}k)(||x - x_i||^2)$$

and if $\sum_i \alpha_i \neq 0$ we get the gradient descent iteration

$$x^{(t+1)} := \frac{\sum_i x_i\,(\frac{\partial}{\partial x}k)(||x^{(t)} - x_i||^2)}{\sum_i (\frac{\partial}{\partial x}k)(||x^{(t)} - x_i||^2)}, \quad t = 0, 1, 2, \ldots$$

Density Gradient Decent

For the normal kernel $k(x) := e^{-\frac{1}{2}x^2}$, we have

$$\frac{\partial}{\partial x} k(x) = -e^{-\frac{1}{2}x^2} x$$

and thus $\sum_i \alpha_i < 0$.

For the Epanechnikov kernel

$$k(x) := \begin{cases} 1 - x^2, & \text{if } |x| < 1 \\ 0, & \text{else} \end{cases}$$

we get

$$\frac{\partial}{\partial x} k(x) = \begin{cases} -2x, & \text{if } |x| < 1 \\ 0, & \text{else} \end{cases}$$

and thus $\sum_i \alpha_i < 0$.

For the proof of convergence of the density gradient descent see [CM02].

Density Gradient Decent / Mode of attraction

All modes can be identified by starting from different starting positions x and following the gradient

$$\begin{aligned} x^{(0)} &:= x \\ x^{(t+1)} &:= \frac{\sum_i x_i \, (\frac{\partial}{\partial x} k)(||x^{(t)} - x_i||^2)}{\sum_i (\frac{\partial}{\partial x} k)(||x^{(t)} - x_i||^2)}, \quad t = 0, 1, 2, \ldots \end{aligned}$$

The iteration stops when the difference between $x^{(t+1)}$ and $x^{(t)}$ becomes sufficient small.

The resulting local maximum $x^{(T)}$ is called **mode of attraction of x**:

$$\text{mode}(x) := x^{(T)}$$

Mean Shift Segmentation

Mean shift segmentation with parameters h_s, h_r and M proceeds as follows:

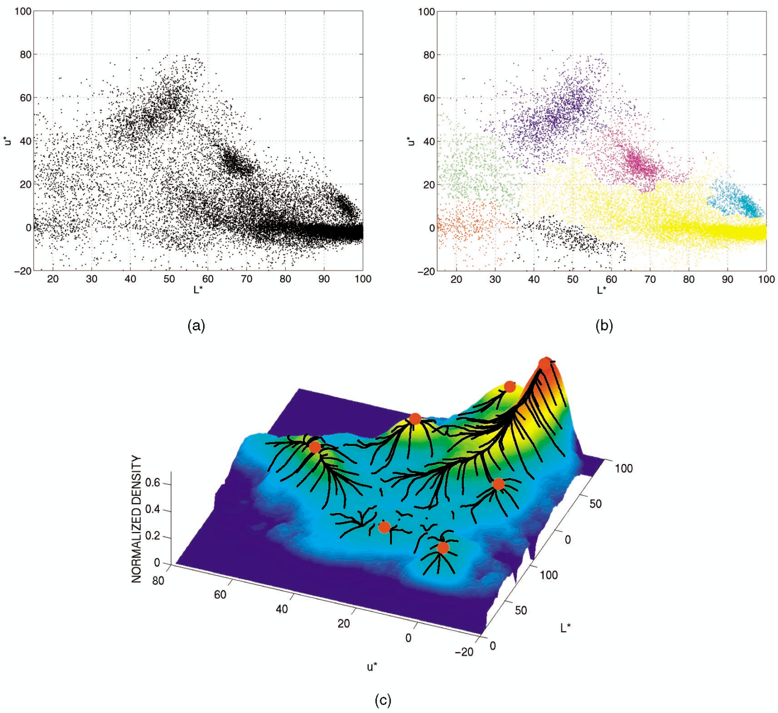
1. For each pixel $x \in I$, compute its mode of attraction mode(x) with kernel
$$k(x) := \frac{c}{h_s^2\, h_r^p}\, k(||\frac{x^s}{h_s}||^2)\, k(||\frac{x^r}{h_r}||^2)$$

2. Group all pixels with the same mode of attraction in the same segment. So let
$$z_1, z_2, \ldots, z_K$$
be all modes, then form
$$S_k := \{x \in I \,|\, \text{mode}(x) = z_k\}, \quad k = 1, \ldots, K$$
(called **basins of attraction**).

3. Iteratively join each two segments that contain two pixels with distance less than h_s and intensity difference of their modes of attraction less than h_r.

4. Eliminate segments having less than M pixels.

Mean Shift Segmentation



Example of a 2D feature space analysis. (a) Two-dimensional data set of 110,400 points representing the first two components of the L*u*v*

(from [CM02])

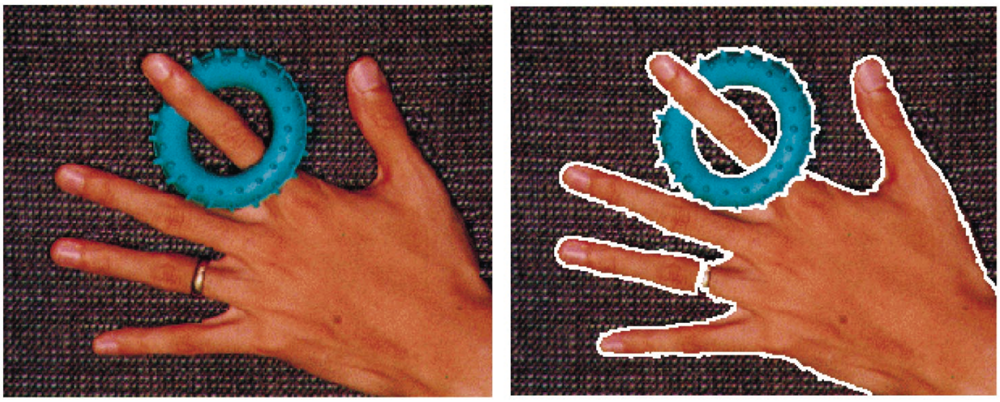
Example

(a) (b) (c)

MIT image. (a) Original. (b) Segmented $(h_s, h_r, M) = (8, 7, 20)$. (c) Region boundaries.

(from [CM02])

Example



(a) (b)

Hand image. (a) Original. (b) Region boundaries delineated with $(h_s, h_r, M) = (16, 19, 40)$ drawn over the input.

(from [CM02])

Example

Landscape images. All the region boundaries were delineated with $(h_s, h_r, M) = (8, 7, 100)$ and are drawn over the original image.

(from [CM02])

Summary

- **(Unsupervised) Image Segmentation** is a clustering problem of the pixels of an image.

- For this one needs **pixel feature vectors**, e.g., a joint vector of pixel position and intensity.

- In principle, any clustering algorithm can be applied to the problem.

- **Mean shift segmentation** is a clustering procedure that (i) identifies the local maxima (modes) of an estimated density and (ii) assigns each point to a cluster described by its **mode of attraction**.

- Mean shift segmentation is a rather simple image segmentation procedure that has given quite reasonable results in practice.

- Besides unsupervised image segmentation, image segmentation also can be treated as **supervised problem** if already segmented images are available.

References

[CM02] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell*, 24:603–619, 2002.