Lecture 15: Fitting and Segmentation

Readings: F&P Ch 15.3-15.5, 16

- Supervised→Unsupervised Category Learning needs segmentation
- K-Means
- Mean Shift
- Graph cuts
- Hough transform

The shape model. The mean location is indicated by the cross, with the ellipse showing the uncertainty in location. The number by each part is the probability of that part being present.

Background Techniques Compared

From the Wallflower Paper
Mean Shift Algorithm

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.

The mean shift algorithm seeks the “mode” or point of highest density of a data distribution.

Graph-Theoretic Image Segmentation

Build a weighted graph $G=(V,E)$ from image

$V$: image pixels
$E$: connections between pairs of nearby pixels

Eigenvectors and affinity clusters

- Simplest idea: we want a vector $a$ giving the association between each element and a cluster.
- We want elements within this cluster to, on the whole, have strong affinity with one another.
- We could maximize
- But need the constraint $a^T A a = 1$

- This is an eigenvalue problem - choose the eigenvector of $A$ with largest eigenvalue

- Shi/Malik, Scott/Longuet-Higgins, Ng/Jordan/Weiss, etc.

Hough transform

Robustness

- Squared error can be a source of bias in the presence of noise points
  - One fix is EM - we’ll do this shortly
  - Another is an M-estimator
    - Square nearby, threshold far away
  - A third is RANSAC
    - Search for good points

Today “Fitting and Segmentation (Ch. 15)”

- Robust estimation
- EM
- Model Selection
- RANSAC

(Maybe “Segmentation I” and “Segmentation II” would be a better way to split these two lectures!)
Robust Statistics

- Recover the best fit to the majority of the data.
- Detect and reject outliers.

Estimating the mean

$$\mu = \frac{1}{N} \sum_{i=1}^{N} d_i$$

Mean is the optimal solution to:

$$\min_{\mu} \sum_{i=1}^{N} (d_i - \mu)^2$$

residual
Estimating the Mean

The mean maximizes this likelihood:
\[
\max_{\mu} p(d_i | \mu) = \frac{1}{\sqrt{2\pi}\sigma} \prod_{i=1}^{N} \exp\left(-\frac{1}{2} \frac{(d_i - \mu)^2}{\sigma^2}\right)
\]
The negative log gives (with \(\sigma = 1\)):
\[
\min_{\mu} \sum_{i=1}^{N} (d_i - \mu)^2
\]
“least squares” estimate

Estimating the mean

What happens if we change just one measurement?

\[
\mu' = \mu + \frac{\Delta}{N}
\]
With a single “bad” data point I can move the mean arbitrarily far.

Influence

Breakdown point
* percentage of outliers required to make the solution arbitrarily bad.

Least squares:
* influence of an outlier is linear (\(\Delta/N\))
* breakdown point is 0% -- not robust!

What about the median?

Influence is proportional to the derivative of the \(\rho\) function.

What’s Wrong?

\[
\min_{\rho} \sum_{i=1}^{N} (d_i - \mu)^2
\]
Outliers (large residuals) have too much influence.

\[
\rho(x) = x^2 \quad \psi(x) = 2x
\]
Approach

\[
\min_{\mu} \sum_{i=1}^{N} \rho(d_i - \mu, \sigma)
\]

Robust error function

Scale parameter

Replace

\[ \rho(x, \sigma) = \left( \frac{x}{\sigma} \right)^2 \]

with something that gives less influence to outliers.

No closed form solutions!

- Iteratively Reweighted Least Squares
- Gradient Descent

L1 Norm

\[ \rho(x) = |x| \]

\[ \psi(x) = \text{sign}(x) \]

Redescending Function

Tukey’s biweight.

Beyond a point, the influence begins to decrease.

Beyond where the second derivative is zero – outlier points

Robust Estimation

Geman-McClure function works well.

Twice differentiable, redescending.

Influence function (d/dr of norm):

\[ \rho(r, \sigma) = \frac{r^2}{\sigma^2 + r^2} \]

\[ \psi(r, \sigma) = \frac{2r\sigma^2}{(\sigma^2 + r^2)} \]
Robust scale

Scale is critical!

Popular choice:

$$\sigma^{(n)} = 1.4826 \text{ median, } \tilde{r}^{(n)}(x_i, \theta^{n-1})$$

Example: Motion

Assumption: Within a finite image region, there is only a single motion present.

Violated by: motion discontinuities, shadows, transparency, specular reflections…

Violations of brightness constancy result in large residuals.

Estimating Flow

Minimize:

$$E(a) = \sum_{x \in \mathbb{R}} \rho(I_x u(x; a) + I_y v(x; a) + I_z, \sigma)$$

Parameterized models provide strong constraints:

* Hundred, or thousands, of constraints.
* Handful (e.g. six) unknowns.

Can be very accurate (when the model is good)!
Deterministic Annealing
Start with a “quadratic” optimization problem and gradually reduce outliers.

Continuation method
GNC: Graduated Non-Convexity

Fragmented Occlusion

Results

Multiple Motions, again
Find the dominant motion while rejecting outliers.

Black & Anandan; Black & Jepson
Robust estimation models only a single process explicitly

Robust norm:

$E(a) = \sum_{s,y \in R} \rho(\nabla I^T u(x; a) + I_t; \sigma)$

Assumption:
Constraints that don’t fit the dominant motion are treated as “outliers” (noise).

Problem?
They aren’t noise!

Alternative View

* There are two things going on simultaneously.
* We don’t know which constraint lines correspond to which motion.
* If we knew this we could estimate the multiple motions.
  - a type of “segmentation” problem
* If we knew the segmentation then estimating the motion would be easy.

EM General framework

Estimate parameters from segmented data.

Consider segmentation labels to be missing data.

Missing variable problems

A missing data problem is a statistical problem where some data is missing.

There are two natural contexts in which missing data are important:

• terms in a data vector are missing for some instances and present for other (perhaps someone responding to a survey was embarrassed by a question)
• an inference problem can be made very much simpler by rewriting it using some variables whose values are unknown.

Missing variable problems

In many vision problems, if some variables were known the maximum likelihood inference problem would be easy

– fitting; if we knew which line each token came from, it would be easy to determine line parameters
– segmentation; if we knew the segment each pixel came from, it would be easy to determine the segment parameters
– fundamental matrix estimation; if we knew which feature corresponded to which, it would be easy to determine the fundamental matrix
– etc.
**Strategy**

For each of our examples, if we knew the missing data we could estimate the parameters effectively. If we knew the parameters, the missing data would follow.

This suggests an iterative algorithm:
1. obtain some estimate of the missing data, using a guess at the parameters;
2. now form a maximum likelihood estimate of the free parameters using the estimate of the missing data.

**Motion Segmentation**

“What goes with what?”

The constraints at these pixels all “go together.”

**Smoothness in layers**

**Layered Representation**

[Adelson]

**EM in Pictures**

Given images at times \( t \) and \( t+1 \) containing two motions.

**EM in Pictures**

Assume we know the segmentation of pixels into “layers”

\[
0 \leq w_i(x, y) \leq 1
\]

\[
\sum_i w_i(x, y) = 1
\]
Then estimating the motion of each “layer” is easy.

\[ E(\mathbf{a}_j) = \sum_{x,y \in \mathbb{R}} w_j(x)(\nabla I^T \mathbf{u}(\mathbf{x}; \mathbf{a}_j) + I_j)^2 \]

Ok. So where do we get the weights?

The weights represent the probability that the constraint “belongs” to a particular layer.

Assume we know the motion of the layers but not the ownership probabilities of the pixels (weights).
Assume we know the motion of the layers but not the ownership probabilities of the pixels (weights).

Also assume we have a likelihood at each pixel:

\[
p(I(t), I(t+1) | a) = \frac{1}{\sqrt{2 \pi \sigma}} \exp \left( -\frac{1}{2} (\nabla I^T u(a) + I)^2 / \sigma^2 \right)
\]

Given the flow, warp the first image towards the second.

Look at the residual error \( (I) \) (since the flow is now zero).

\[
p(W(I(t), a_i), I(t+1) | 0) = \frac{1}{\sqrt{2 \pi \sigma}} \exp \left( -\frac{1}{2} (I)^2 / \sigma^2 \right)
\]

Two “explanations” for each pixel.

Two likelihoods:

\[
p(I(x, t+1) | u(a_i))
\]

\[
p(I(x, t+1) | u(a_j))
\]

Motion segmentation Example

- Model image pair (or video sequence) as consisting of regions of parametric motion
  - affine motion is popular

\[
\begin{pmatrix} v_z \end{pmatrix} = \begin{pmatrix} a & b & c & d \\ x \\ y \\ z \end{pmatrix} \begin{pmatrix} s_z \end{pmatrix}
\]

- iterate E/M...
  - determine which pixels belong to which region
  - estimate parameters

Compute total likelihood and normalize:

\[
w_i(x) = \frac{p(I(x, t+1) | u(a_i))}{\sum_i p(I(x, t+1) | u(a_i))}
\]
If we use multiple frames to estimate the appearance of a segment, we can fill in occlusions; so we can re-render the sequence with some segments removed.

Grey level shows region no. with highest probability

Segments and motion fields associated with them

Lines
- Simple case: we have one line, and \( n \) points
- Some come from the line, some from “noise”
- This is a mixture model:
  \[
  P(\text{point} | \text{line and noise params}) = P(\text{point} | \text{line})P(\text{comes from line}) + P(\text{point} | \text{noise})P(\text{comes from noise})
  \]
  \[
  = P(\text{point} | \text{line})\lambda + P(\text{point} | \text{noise})(1 - \lambda)
  \]
- e.g.,
  - allocate each point to a line with a weight, which is the probability of the point given the line
  - refit lines to the weighted set of points

Line fitting review
- In case of single line and normal i.i.d. errors, maximum likelihood estimation reduces to least-squares:
  \[
  \min_{a,b} \sum (ax_i + b - y_i)^2 = \min_{a,b} \sum r_i^2
  \]
- The line parameters \((a, b)\) are solutions to the system:
  \[
  \begin{pmatrix}
  \sum x_i^2 & \sum x_i \\
  \sum x_i & \sum 1
  \end{pmatrix}
  \begin{pmatrix}
  a \\
  b
  \end{pmatrix}
  = \begin{pmatrix}
  \sum x_i y_i \\
  \sum y_i
  \end{pmatrix}
  \]

The E Step
- Compute residuals:
  \[
  r_i(1) = ax_i + b - y_i
  \]
  \[
  r_i(2) = k
  \]
  (uniform noise model)
- Compute soft assignments:
  \[
  w_i(1) = \frac{e^{-\frac{r_i(1)^2}{2\sigma_1^2}}}{e^{-\frac{r_i(1)^2}{2\sigma_1^2}} + e^{-\frac{r_i(2)^2}{2\sigma_2^2}}}
  \]
  \[
  w_i(2) = \frac{e^{-\frac{r_i(2)^2}{2\sigma_2^2}}}{e^{-\frac{r_i(1)^2}{2\sigma_1^2}} + e^{-\frac{r_i(2)^2}{2\sigma_2^2}}}
  \]
The M Step

Weighted least squares system is solved for \((a_1, b_1)\)

\[
\begin{pmatrix}
\sum w_i(i)x_i^2 & \sum w_i(i)x_i \\
\sum w_i(i)x_i & \sum w_i(i)
\end{pmatrix}
\begin{pmatrix}
a_1 \\
b_1
\end{pmatrix}
=
\begin{pmatrix}
\sum w_i(i)x_iy_i \\
\sum w_i(i)y_i
\end{pmatrix}
\]

The expected values of the deltas at the maximum (notice the one value close to zero).

Issues with EM

• Local maxima
  – can be a serious nuisance in some problems
  – no guarantee that we have reached the “right” maximum
• Starting
  – k means to cluster the points is often a good idea

Closeup of the fit
Choosing parameters

- What about the noise parameter, and the sigma for the line?
  - several methods
    - from first principles knowledge of the problem (seldom really possible)
    - play around with a few examples and choose (usually quite effective, as precise choice doesn’t matter much)
  - notice that if $k_n$ is large, this says that points very seldom come from noise, however far from the line they lie
  - usually biases the fit, by pushing outliers into the line
  - rule of thumb, it’s better to fit to the better fitting points, within reason; if this is hard to do, then the model could be a problem

Estimating the number of models

- In weighted scenario, additional models will not necessarily reduce the total error.
- The optimal number of models is a function of the $\sigma$ parameter – how well we expect the model to fit the data.
- Algorithm: start with many models, redundant models will collapse.

Fitting 2 lines to data points

- Input:
  - Data points that were generated by 2 lines with Gaussian noise.
- Output:
  - The parameters of the 2 lines.
  - The assignment of each point to its line.

The E Step

- Compute residuals assuming known lines:
  \[ r_j(i) = a_jx_i + b_i - y_i \]
  \[ r_j(i) = a_jx_i + b_i - y_i \]
- Compute soft assignments:
  \[ w_j(i) = \frac{e^{-r_j(i)/\sigma^2}}{e^{-r_1(i)/\sigma^2} + e^{-r_2(i)/\sigma^2}} \]
  \[ w_2(i) = \frac{e^{-r_2(i)/\sigma^2}}{e^{-r_1(i)/\sigma^2} + e^{-r_2(i)/\sigma^2}} \]
The M Step

- In the weighted case we find

\[
\min_{a,b} \sum w_i \log \left( \frac{1}{2\pi \det(\Sigma)} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\} \right)
\]

- Weighted least squares system is solved twice for \((a_1,b_1)\) and \((a_2,b_2)\).

\[
\begin{align*}
\sum w_i x_i^2 \sum w_i x_i \left( \begin{array}{c}
a_1 \\
b_1
\end{array} \right) &= \sum w_i x_i y_i \\
\sum w_i x_i^2 \sum w_i x_i \left( \begin{array}{c}
a_2 \\
b_2
\end{array} \right) &= \sum w_i x_i y_i
\end{align*}
\]

Illustrations

Color segmentation Example

Parameters include mixing weights and means/covariances:

\[\Theta = (\omega_1, \ldots, \omega_k, \theta_1, \ldots, \theta_k) \quad \theta_i = (\mu_i, \Sigma_i)\]

yielding

\[p(x|\Theta) = \sum_{i=1}^{k} \omega_i p(x|\theta_i)\]

with

\[p_i(x|\theta_i) = \frac{1}{(2\pi)^{d/2} \det(\Sigma_i)^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right\}\]

EM for Mixture models

If log-likelihood is linear in missing variables we can replace missing variables with expectations. E.g.,

\[
p(y) = \sum_{i=1}^{K} \pi_i p(y|x_i) \quad \sum_{i=1}^{K} \pi_i \sum_{j=1}^{N} \sum_{m=1}^{M} \log p(z_{ij} y|x_i)
\]

1. (E-step) estimate complete data (e.g., \(z_{ij}\)'s) using previous parameters
2. (M-step) maximize complete log-likelihood using estimated complete data

\[
u^{t+1} = \arg \max_{\mu} L_c([\mu, \Sigma]; u)
\]

\[
u^{t+1} = \arg \max_{\mu} L_c([\mu, \Sigma]; u)
\]
Color segmentation with EM

Algorithm 17.1: Colour and texture segmentation with EM

Choose a number of segments
Construct a set of support maps, one per segment, containing one element per pixel. These support maps will contain the weight associated with a pixel.
Initialize the support maps by either:
- Estimating segment parameters from small blocks of pixels, and then computing weights using the E-step.
- Randomly allocating values to the support maps.
Until convergence
Update the support maps with an E-step
Update the segment parameters with an M-step
end

Color segmentation

- At each pixel in an image, we compute a \(d\)-dimensional feature vector \(x\), which encapsulates position, colour and texture information.
- Pixel is generated by one of \(G\) segments, each Gaussian, chosen with probability \(\pi\):

\[
p(x) = \sum_{i=1}^{G} p(x|\theta_i)\pi_i
\]

Initialize

E-step

Estimate support maps:

\[
p(x|\theta, \Theta) = \frac{c_i(p(x|\theta_i)\theta_i)}{\sum_{j=1}^{G} C_j(p(x|\theta_j)\theta_j)}
\]

Algorithm 17.2: Colour and texture segmentation with EM - the E-step

For each pixel location \(i\)
Initialize support map \(s(p(x|\theta_i))\)
In pixel location \(i\), the support map \(s\)
end
Add the support map values to obtain
\[
\sum_{j=1}^{G} s(p(x|\theta_j)\theta_j)
\]
and divide the value in location \(i\) in each support map by this term
end
M-step

Update mean’s, covar’s, and mixing coef.’s using support map:

### Algorithm 17.8: Color and texture segmentation with EM - M-step

For each segment m:
1. Form new value of the segment parameter using the expression:
   \[ \mu_m = \frac{1}{N_m} \sum_{i \in S_m} x_i \]
   \[ \Sigma_m = \frac{1}{N_m} \sum_{i \in S_m} (x_i - \mu_m)(x_i - \mu_m)^T \]
   \[ w_m = \frac{1}{N_m} \sum_{i \in S_m} \pi_i \phi_i(x_i) \]
   Where \( \phi_i(x) \) is the value in the \( i \)-th dimension of the feature vector.

### Figure 17.1: The images in the top row are used to fit a model. The top image shows the original image, and the bottom image shows the segmented image. The segmentation process identifies different classes of pixels, as indicated by the color coding. This example illustrates the application of the EM algorithm for image segmentation, where the mixture model is used to estimate the parameters of different classes present in the image.

### Figure 17.2: Both plots of the negative log-likelihood in the same or in different dimensions are shown. The top plot shows the negative log-likelihood as a function of the number of parameters in a model. The bottom plot shows the top plot with a logarithmic y-axis. These plots are used to illustrate the trade-off between model complexity and goodness of fit, as well as the concept of overfitting and underfitting.

### Segmentation with EM

- **Example Image:**
  - The top row shows the original image.
  - The bottom row shows the segmented image.

### Model Selection

- **We wish to choose a model to fit to data**
  - E.g. is it a line or a circle?
  - E.g. is this a perspective or an orthographic camera?
  - E.g. is there an aeroplane there or is it noise?

- **Issue**
  - In general, models with more parameters will fit a dataset better, but are poorer at prediction.
  - This means we can’t simply look at the negative log-likelihood (or fitting error).

- **Graph:**
  - The top plot shows the negative log-likelihood as a function of the number of parameters in a model, with a logarithmic y-axis. The bottom plot shows the top plot with a linear y-axis. The graphs are used to illustrate the concept of model complexity vs. goodness of fit, as well as the trade-off between bias and variance.

- **Graph:**
  - The top plot shows the negative log-likelihood as a function of the number of parameters in a model, with a logarithmic y-axis. The bottom plot shows the top plot with a linear y-axis. The graphs are used to illustrate the concept of model complexity vs. goodness of fit, as well as the trade-off between bias and variance.
We can discount the fitting error with some term in the number of parameters in the model.

Discounts

- **AIC (an information criterion)**
  - choose model with smallest value of $-2L(D; \theta^*) + 2p$
  - $p$ is the number of parameters

- **BIC (Bayes information criterion)**
  - choose model with smallest value of $-2L(D; \theta^*) + p \log N$
  - $N$ is the number of data points

- **Minimum description length**
  - same criterion as BIC, but derived in a completely different way

Cross-validation

- Split data set into two pieces, fit to one, and compute negative log-likelihood on the other
- Average over multiple different splits
- Choose the model with the smallest value of this average

The difference in averages for two different models is an estimate of the difference in KL divergence of the models from the source of the data.

Extreme segmentation

What if more than half the points are noise?

RANSAC

- Iterate:
  - Sample
  - Fit
  - Test
- Keep best estimate; refit on inliers

- Choose a small subset uniformly at random
- Fit to that
- Anything that is close to result is signal; all others are noise
- Refit
- Do this many times and choose the best

RANSAC

- Issues
  - How many times?
    - Often enough that we are likely to have a good line
  - How big a subset?
    - Smallest possible
    - What does close mean?
    - Depends on the problem
    - What is a good line?
    - One where the number of nearby points is so big it is unlikely to be all outliers
Algorithm 15.4: RANSAC fitting from random sample consensus

Determine:
- $m$ — the smallest number of points required
- $k$ — the number of iterations required
- $t$ — the threshold used to identify a point that fits well
- $d$ — the number of nearby points required to assert a model fits well

If $k$ iterations have occurred:
- Draw a sample of $m$ points from the data
- Fit to that set of $m$ points
- For each data point outside the sample:
  - Test the distance from the point to the line
  - If the distance from the point to the line is less than $t$, the point is close
- If there are $d$ or more points close to the line:
  - There is a good fit. Refit the line using all these points.
- Else
  - Use the best fit from this collection, using the fitting error as a criterion.

RANSAC applications

- Fundamental Matrices
  - Estimate $F$ from 7 points
  - Test agreement with all other points
- Direct motion
  - Estimate affine (or rigid motion) from small match
  - See what other parts of image are consistent
- ...

Fitting and Probabilistic Segmentation

- Robust estimation
- EM
- Model Selection
- RANSAC

[Slides from Michael Black and F&P]