

## CS 534: Computer Vision Model Fitting

Spring 2004  
Ahmed Elgammal  
Dept of Computer Science  
Rutgers University

CS 534 – Model Fitting - 1

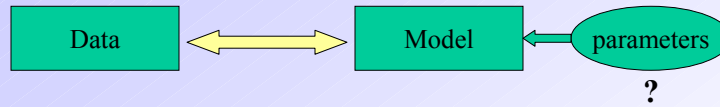
### Outlines

- Model fitting is important
- Least-squares fitting
- Maximum likelihood estimation
- MAP estimation
- Robust estimation
- Missing values problem – EM algorithm

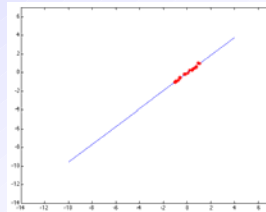
CS 534 – Model Fitting - 2

## Model Fitting

- Model fitting is a fundamental problem in computer vision



- Find the model **parameters** that best fits the data  $\Rightarrow$  Optimization problem
- The model can be as simple as a 2D line or as complex as 3D articulated object.

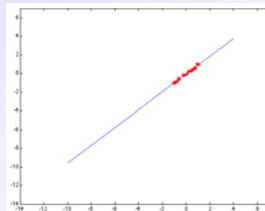


CS 534 – Model Fitting - 3

## Fitting Models

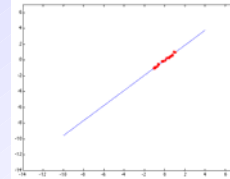
Issues:

- What is the model ?
- How to measure a good fit ? What is your metric ?
- Effect of noise on the fitting
- Multiple instances of the same model (object) – different models
  - Which data points belong to which object ?
  - How many objects are there ?



CS 534 – Model Fitting - 4

- Simple example: Fitting a line in 2D to a set of point.
- The same issues apply to more complex problems
- Given a set of points  $\{(x_i, y_i)\}$  find line parameters
- Least-squares



$$y = ax + b$$

Find  $a, b$  which minimize

$$\sum_i (y_i - ax_i - b)^2 \Rightarrow \text{Residual: measure how far is point } i \text{ from the model}$$

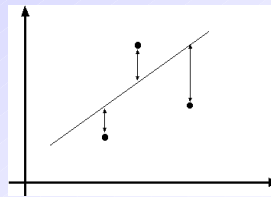
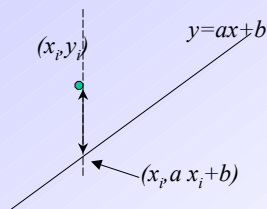
$$\begin{bmatrix} \dots \\ x_i \\ \dots \end{bmatrix} \begin{bmatrix} 1 \\ a \\ b \end{bmatrix} = \begin{bmatrix} \vdots \\ y_i \\ \vdots \end{bmatrix} \Rightarrow A\mathbf{x} = \mathbf{b} \Rightarrow \min_x \|A\mathbf{x} - \mathbf{b}\|^2$$

Linear least-squares  
We have seen an example of this before in calibration

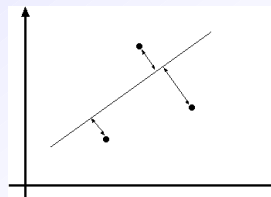
- Question: what does the residual mean ? What are we minimizing ?

Find  $a, b$  which minimize

$$\sum_i (y_i - ax_i - b)^2$$

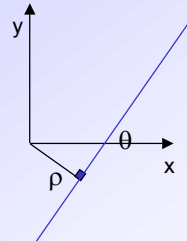


Minimizes vertical distances – Very poor model  
(e.g., We can not fit vertical/near vertical lines)



- Better model  $x \cos \theta + y \sin \theta + \rho = 0$

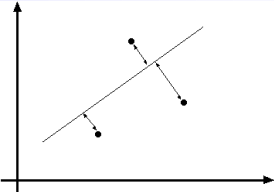
$ax + by + c = 0,$   
 where  $a^2 + b^2 = 1$



- Still we will use least-squares to minimize the residual:

$$\sum_i (ax_i + by_i + c)^2$$

- $|ax_i + by_i + c|$  is the perpendicular distance
- This is called total least-squares



CS 534 – Model Fitting - 7

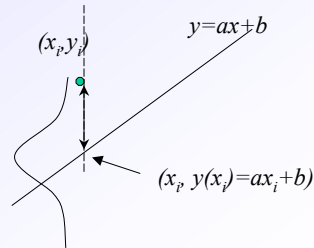
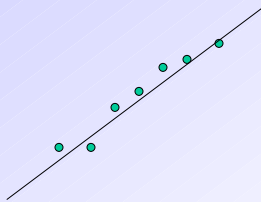
## Maximum Likelihood Parameter Estimation (MLE)

- Fitting as a probabilistic inference problem
- We still using the 2D line example

Assume

- $x_i$  is correct (deterministic – no errors)
- $y_i$  has a measurement error that is Normally distributed around true  $y(x_i)$

$$y_i : N(y(x_i), \sigma^2) \Rightarrow P(y_i) \propto e^{-1/2 \left( \frac{y_i - y(x_i)}{\sigma} \right)^2}$$



## Maximum Likelihood Parameter Estimation (MLE)

Assume

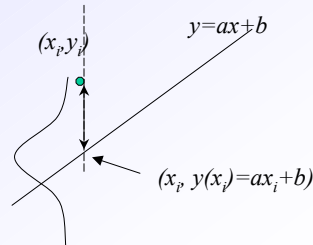
- $x_i$  is correct (deterministic – no errors)
- $y_i$  has a measurement error that is Normally distributed around true  $y(x_i)$

$$y_i : N(y(x_i), \sigma^2) \Rightarrow P(y_i) \propto e^{-1/2 \left( \frac{y_i - y(x_i)}{\sigma} \right)^2}$$

- Assume errors are independent, standard deviations  $\sigma$  for all points
- Probability that the data comes from the model: probability that the data and model prediction are within  $\Delta y$

$$P \propto \prod_{i=1}^N \left( e^{-1/2 \left( \frac{y_i - y(x_i)}{\sigma} \right)^2} \cdot \Delta y \right)$$

Find the parameters (a,b) that maximize P  
Maximize the likelihood  
 $P(\text{measurement} | a, b)$   
 $P(\text{measurement} | \text{model parameters})$



CS 534 – Model Fitting - 9

## Maximum Likelihood = Least squares

$$P \propto \prod_{i=1}^N \left( e^{-1/2 \left( \frac{y_i - y(x_i)}{\sigma} \right)^2} \cdot \Delta y \right)$$

This term is irrelevant  
(doesn't depend on a,b)

$$-\text{Log } P = \sum_{i=1}^N \frac{(y_i - y(x_i))^2}{2\sigma^2} - N \log \Delta y$$

Find the parameters (a,b) that maximize P  $\equiv$   
Find the parameters (a,b) that minimizes  $-\text{Log } P$   
 $\equiv$  Minimize:

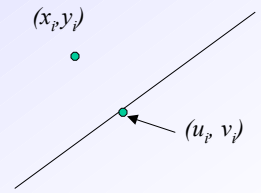
$$\sum_{i=1}^N (y_i - y(x_i))^2 = \sum_{i=1}^N (y_i - ax_i - b)^2 \Rightarrow \text{Least-squares as we know it}$$

CS 534 – Model Fitting - 10

- We can assume that both  $x$  and  $y$  contains measurement errors
- Both  $x, y$  are probabilistic random variables

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} + w$$

$N(0, \Sigma)$



## MAP estimate

Maximum likelihood estimate (MLE)

⇒ Maximize  $P(\text{measurement} | \text{model parameters})$

But we are actually interested in maximizing  $P(\text{model parameters} | \text{measurement})$

So what is the difference ?

Bayes' Rule

$$P(\text{model} | \text{measurement}) = P(\text{measurement} | \text{model}) * P(\text{model}) / P(\text{measurement})$$

Posterior probability    measurement likelihood    model prior    Measurement probability

Maximize  $P(\text{model} | \text{measurement}) \propto P(\text{measurement} | \text{model}) * P(\text{model})$

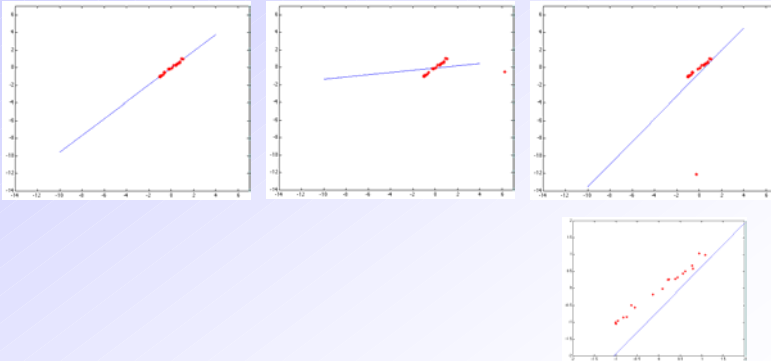
### Maximum A posteriori Estimation (MAP)

MAP will be useful if we have reasons to prefer one model over the others, i.e., if we have prior knowledge about model parameters.

e.g., for line fitting, certain line orientations are most probable than others

## Robustness

- What is the effect of noisy data (outliers)
- Least-squares estimate (similarly MLE) is extremely sensitive to outliers
- The problem is the single point on the right; the error for that point is so large that it drags the line away from the other points

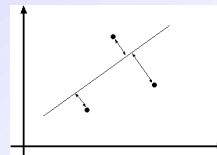


CS 534 – Model Fitting - 13

## M-estimators

- Least-squares minimize the residual: sum of squared distances (residuals) for each data point.

$$\sum_i r(x_i; \theta)^2 \quad \text{e.g.} \quad \sum_i (ax_i + by_i + c)^2$$



- The residual for a far away point (outlier) is huge
- Instead, we want to reduce the effect of the residuals for far away points
- How to do that: replace  $(\text{distance})^2$  with something that looks like  $(\text{distance})^2$  for small distances, and is about constant for large distances

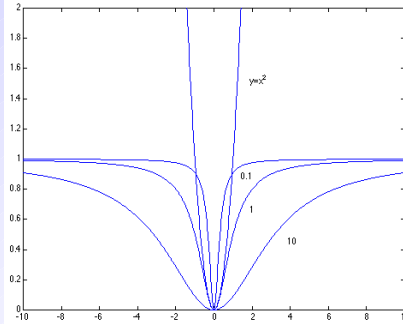
CS 534 – Model Fitting - 14

## M-estimators

- How to do that: replace  $(\text{distance})^2$  with something that looks like  $(\text{distance})^2$  for small distances, and is about constant for large distances

$$\text{minimize } \sum_i \rho(r_i, \sigma)$$

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

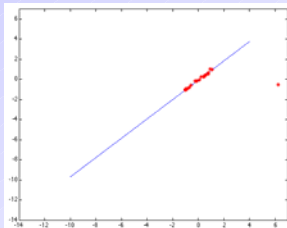
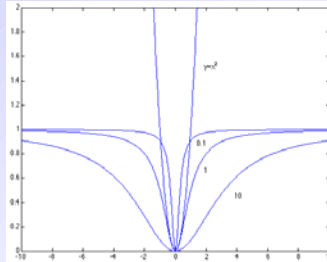


Residual (distance) for each point

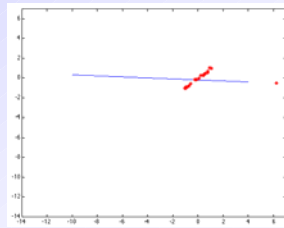
## M-estimators

$$\text{minimize } \sum_i \rho(r_i, \sigma) \quad \rho(r, \sigma) = \frac{r^2}{r^2 + \sigma^2}$$

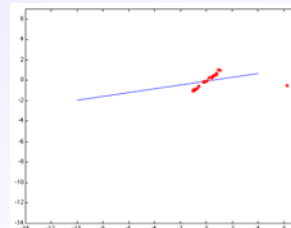
- This leads to a nonlinear optimization problem
- Iterative procedure – given an initial solution
- Can stuck to a local minima – depends on initial guess
- The choice of  $\sigma$  is critical



Right  $\sigma$

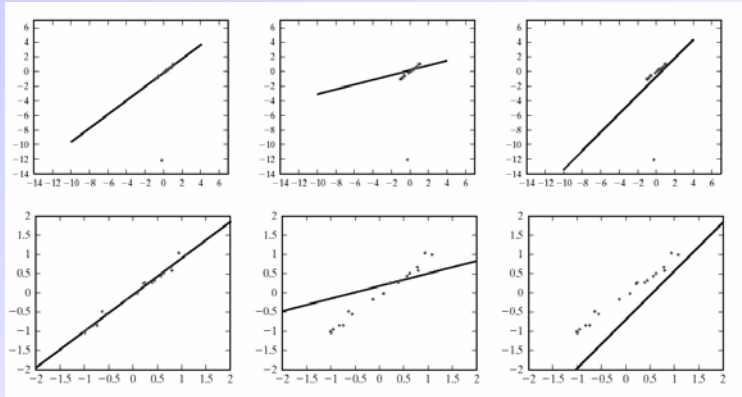
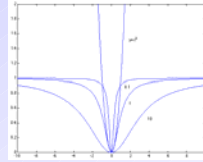


Too small –  
fit is insensitive to all points  
Just stuck to initial guess



Too Large – similar to LS  
Outlier has big contribution

## M-estimators



Right  $\sigma$

Too small –  
fit is insensitive to all points  
Just stuck to initial guess

Too Large – similar to LS  
Outlier has big contribution

## M-estimators

- The choice of  $\sigma$  is critical
- Start with big  $\sigma$  - solution similar to least squares.
- Reduce as  $\sigma$  you go

## RANSAC – RANDOM Sample Consensus

- Searching for a random sample that leads to a fit on which many of the data points agree
- Extremely useful concept
- Can fit models even if up to 50% of the points are outliers.

### Repeat

- Choose a subset of points randomly
- Fit the model to this subset
- See how many points agree on this model (how many points fit that model)
- Use only points which agree to re-fit a better model

Finally choose the best fit

## RANSAC – RANDOM Sample Consensus

- Four parameters  
n : the smallest # of points required  
k : the # of iterations required  
t : the threshold used to identify a point that fits well  
d : the # of nearby points required

Until k iterations have occurred

    Pick n sample points uniformly at random

    Fit to that set of n points

    For each data point outside the sample

        Test distance; if the distance  $< t$ , it is close

    If there are d or more points close, this is a good fit.

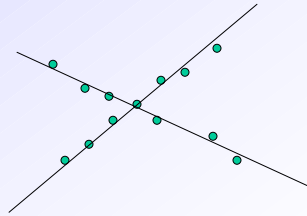
        Refit the line using all these points

End

use the best fit

## 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.
- This sort of thing happens in statistics, too

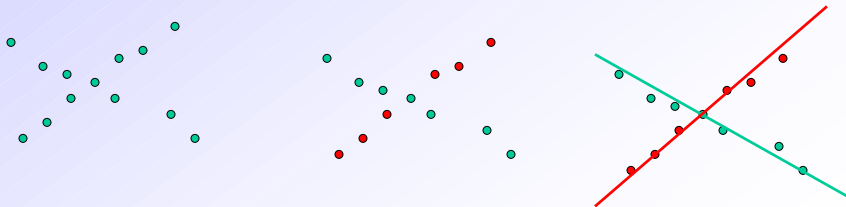


CS 534 – Model Fitting - 21

## Missing variable problems

Consider line fitting:

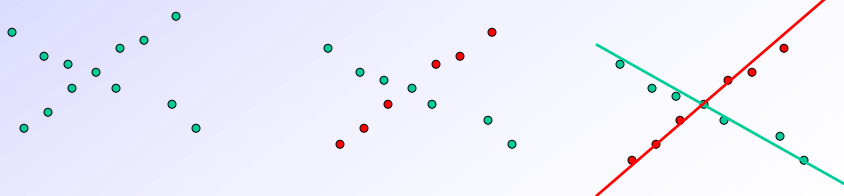
- What is given? point locations
- What is missing? which points belong to which line
- If we know the line assignment for each point  $\Rightarrow$  fit the lines (estimate the parameters) using MLE
- If we know the parameters of the two lines  $\Rightarrow$  we can figure out the line assignment for each point
- Chicken and egg problem



CS 534 – Model Fitting - 22

## Missing variable problems

- Strategy
  - estimate appropriate values for the missing variables
  - plug these in, now estimate parameters
  - re-estimate appropriate values for missing variables, continue
- eg
  - guess which line gets which point
  - now fit the lines
  - now reallocate points to lines, using our knowledge of the lines
  - now refit, etc.
- We've seen this line of thought before (k means)



CS 534 – Model Fitting - 23

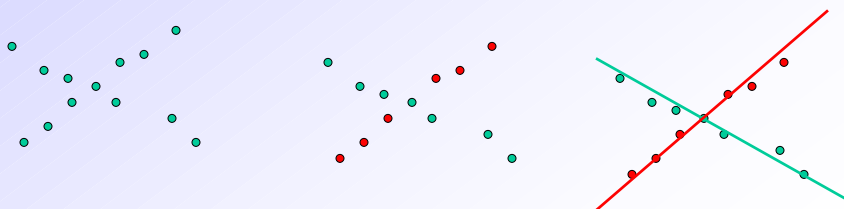
## EM algorithm

Expectation-Maximization

Iterate until convergence:

- replace missing variable with expected values, given **fixed** values of parameters (E-step - expectation)
  - fix missing variables, choose parameters to maximize likelihood given fixed values of missing variables (M-step - maximization)
- e.g., (line fitting) iterate till convergence:
- 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 assignment for each point is assumed to be a missing value (Hidden variable)



CS 534 – Model Fitting - 24

## EM algorithm

- Mixture Model:

Probability of point  $x$  given line  $l$   $p(x | \theta_l)$

Probability of point  $x$  given all lines  $p(x | \Theta) = \sum_l \alpha_l p(x | \theta_l)$

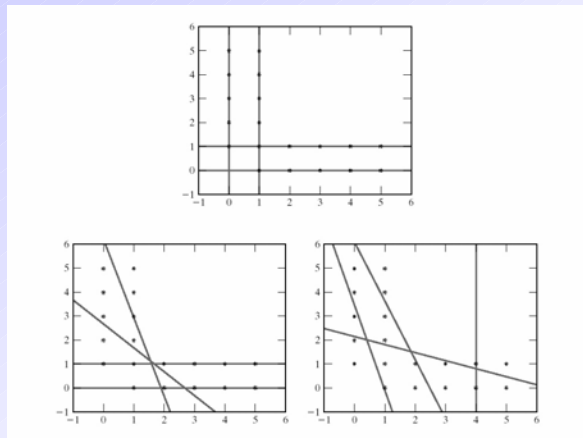
$\alpha_l$  : probability of point  $x$  lies on line  $l$  – *mixture weights*

The whole parameter set  $\Theta = (\alpha_1, \dots, \alpha_N, \theta_1, \dots, \theta_N)$

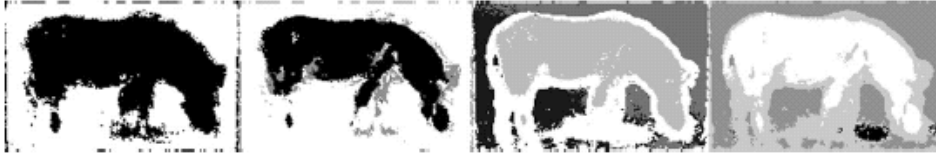
Likelihood for all points

$$p(X | \Theta) = \prod_i \sum_l \alpha_l^i p(x_i | \theta_l)$$

- Example: line fitting using EM.
- Solution is very sensitive to initial guess (assignment) – local minima.



## Segmentation with EM



$K=2,3,4,5$

Figure from "Color and Texture Based Image Segmentation Using EM and Its Application to Content Based Image Retrieval", S.J. Belongie et al., Proc. Int. Conf. Computer Vision, 1998, c1998, IEEE

CS 534 – Model Fitting - 27

## Sources

- Forsyth and Ponce, Computer Vision a Modern approach: chapters 15,16
- R. O. Duda, P. E. Hart, and D. G. Stork. "Pattern Classification." Wiley, New York, 2nd edition, 2000
- Slides by D. Forsyth

CS 534 – Model Fitting - 28