Algorithm for line fitting (review)

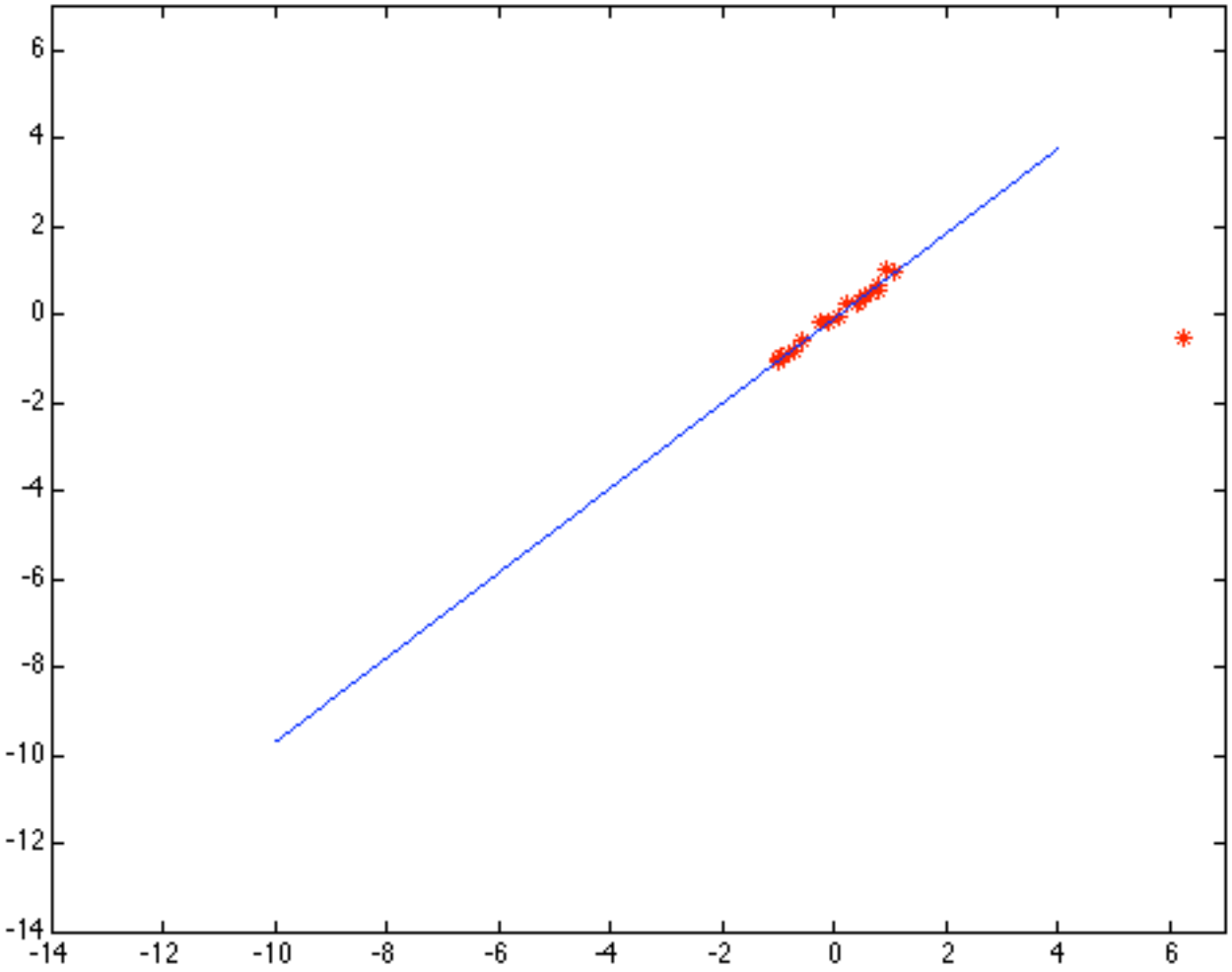
- Obtain some start point (can either estimate the missing values or parameters, which is the choice here)

$$\theta^{(0)} = \left(\phi^{(0)}, c^{(0)}, \sigma^{(0)}\right)$$

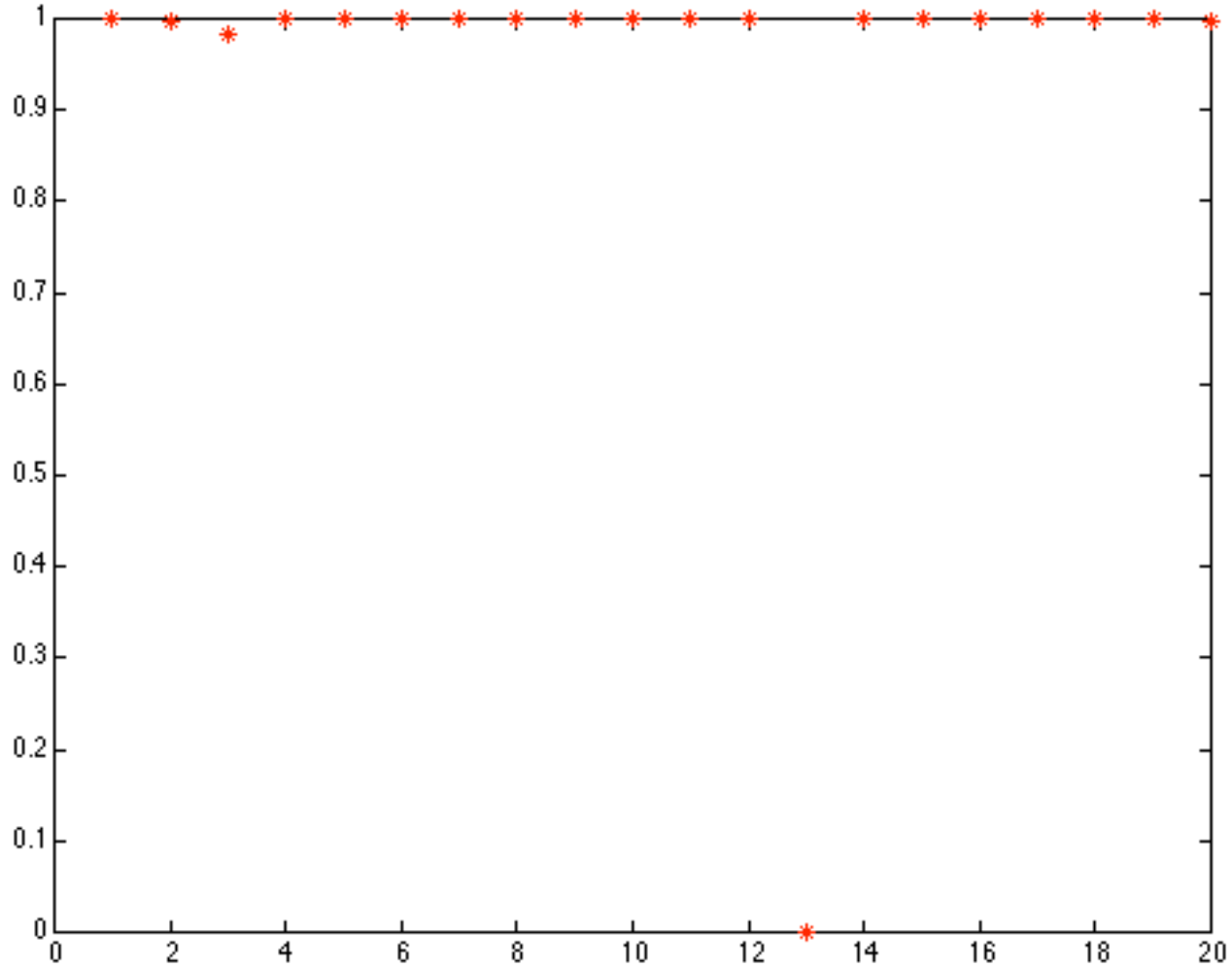
- Now compute γ's (expected value of missing values) using formula above

- Now compute maximum likelihood estimate of $\theta^{(1)}$
  - ϕ, c come from fitting to weighted points (γ's are the weights)
  - σ comes by counting (summing γ's)

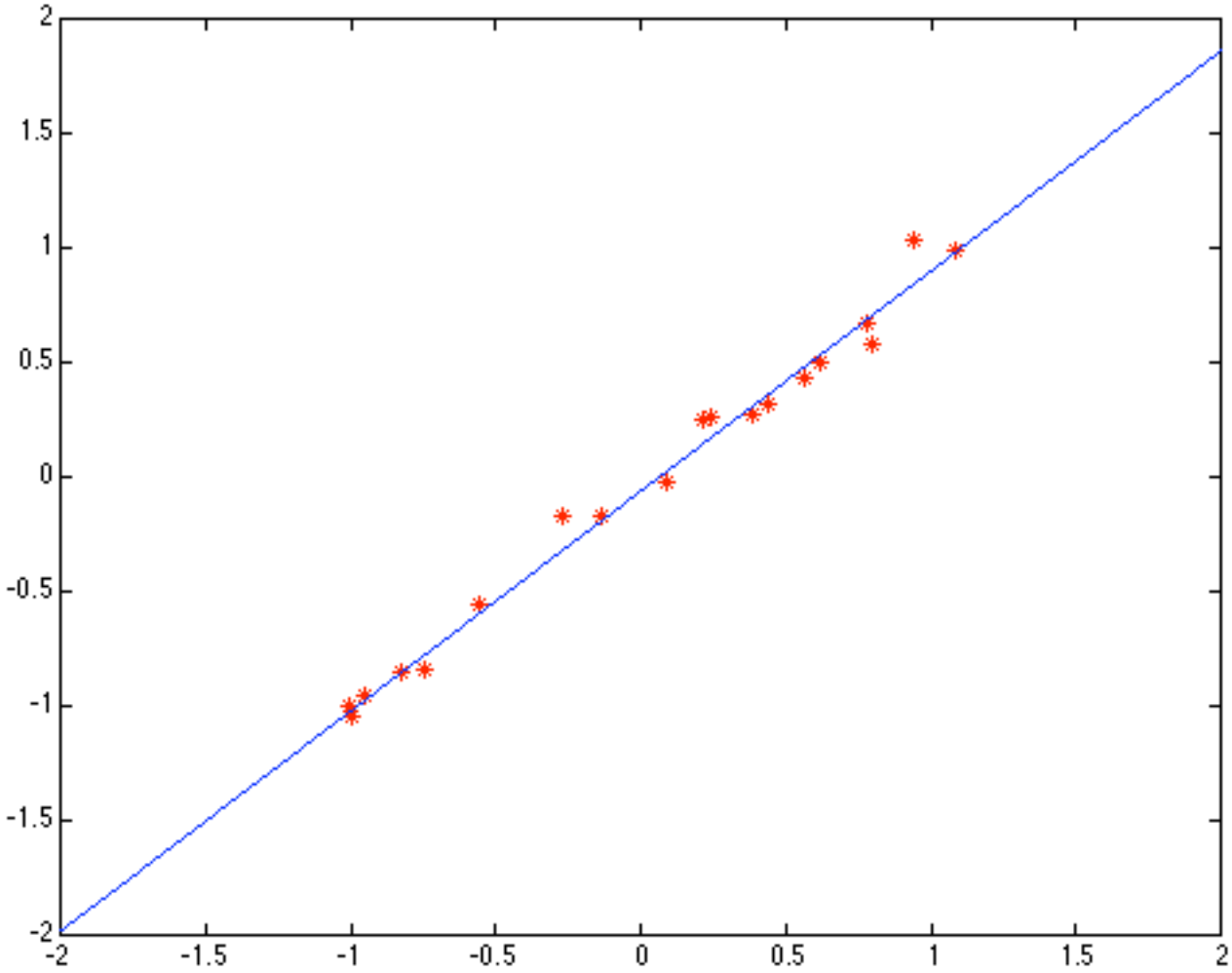
- Iterate to convergence


6
4
2
0
-2
-4
-6
-8
-10
-12
-14
-14
-12
-10
-8
-6
-4
-2
0
2
4
6

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



Closeup of the fit



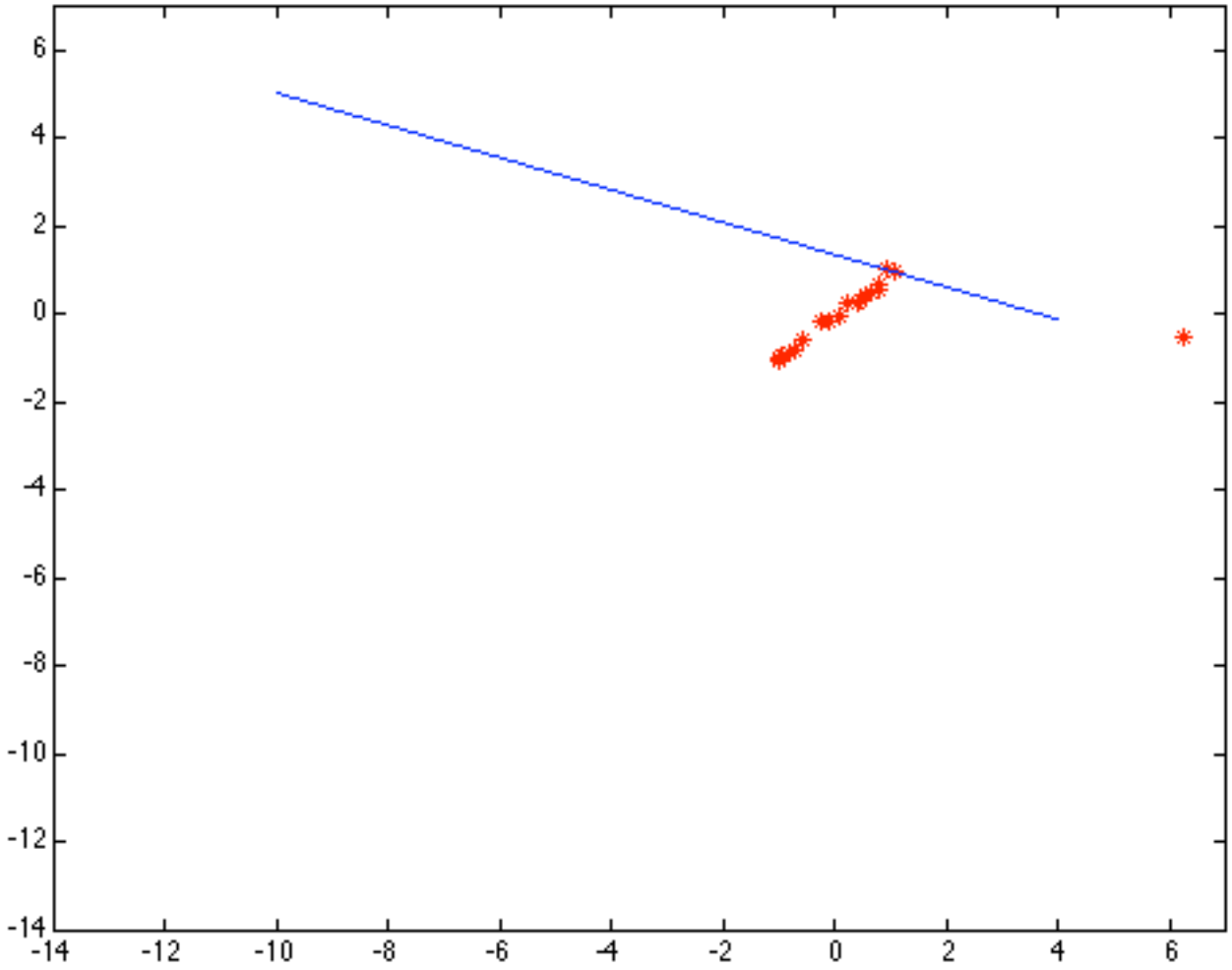
Choosing parameters

- 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)
    - more precisely, do "model selection"
  - notice that if kn is small, 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; its better to fit to the better fitting points, within reason; if this is hard to do, then the model could be a problem

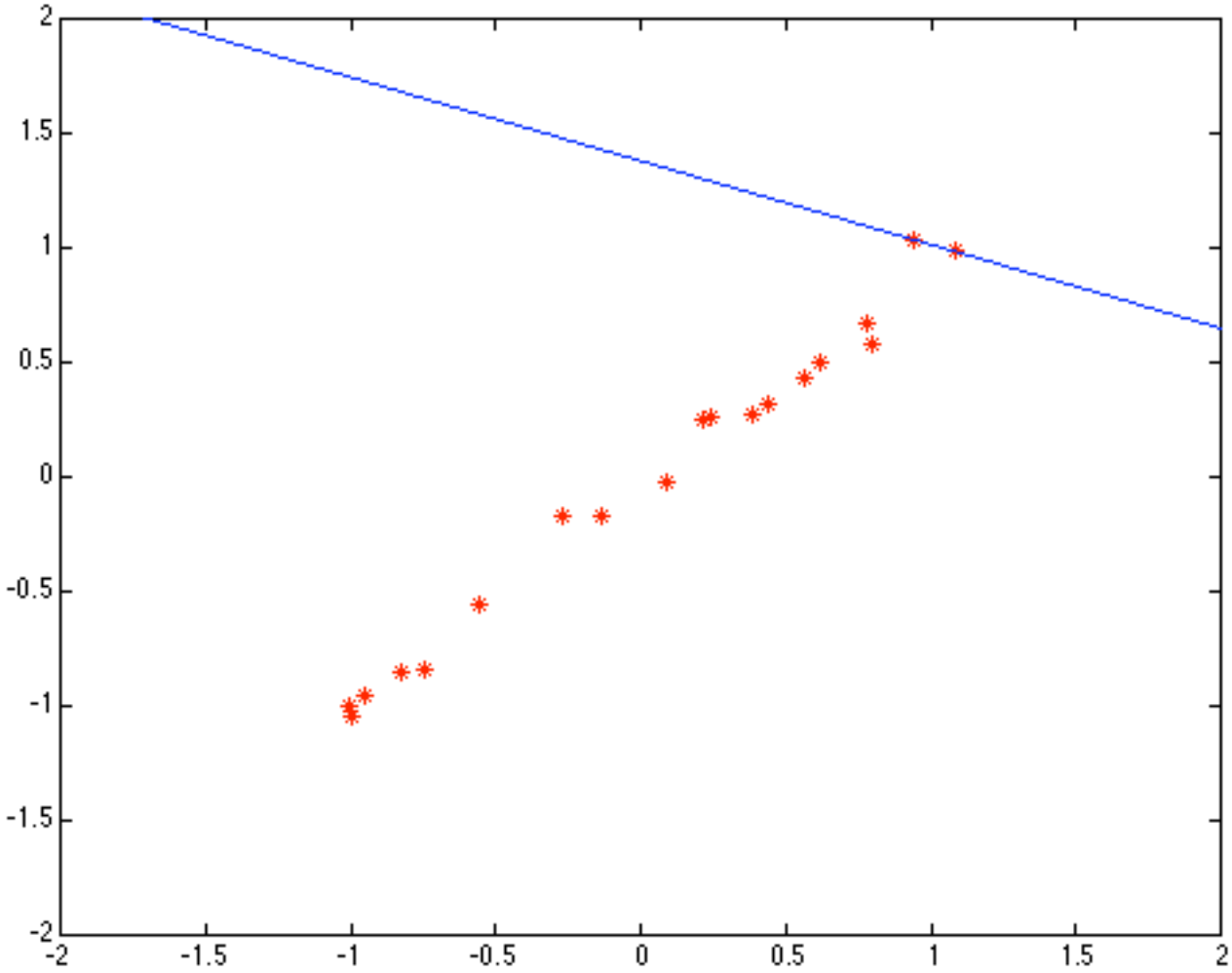
Issues with EM

- 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

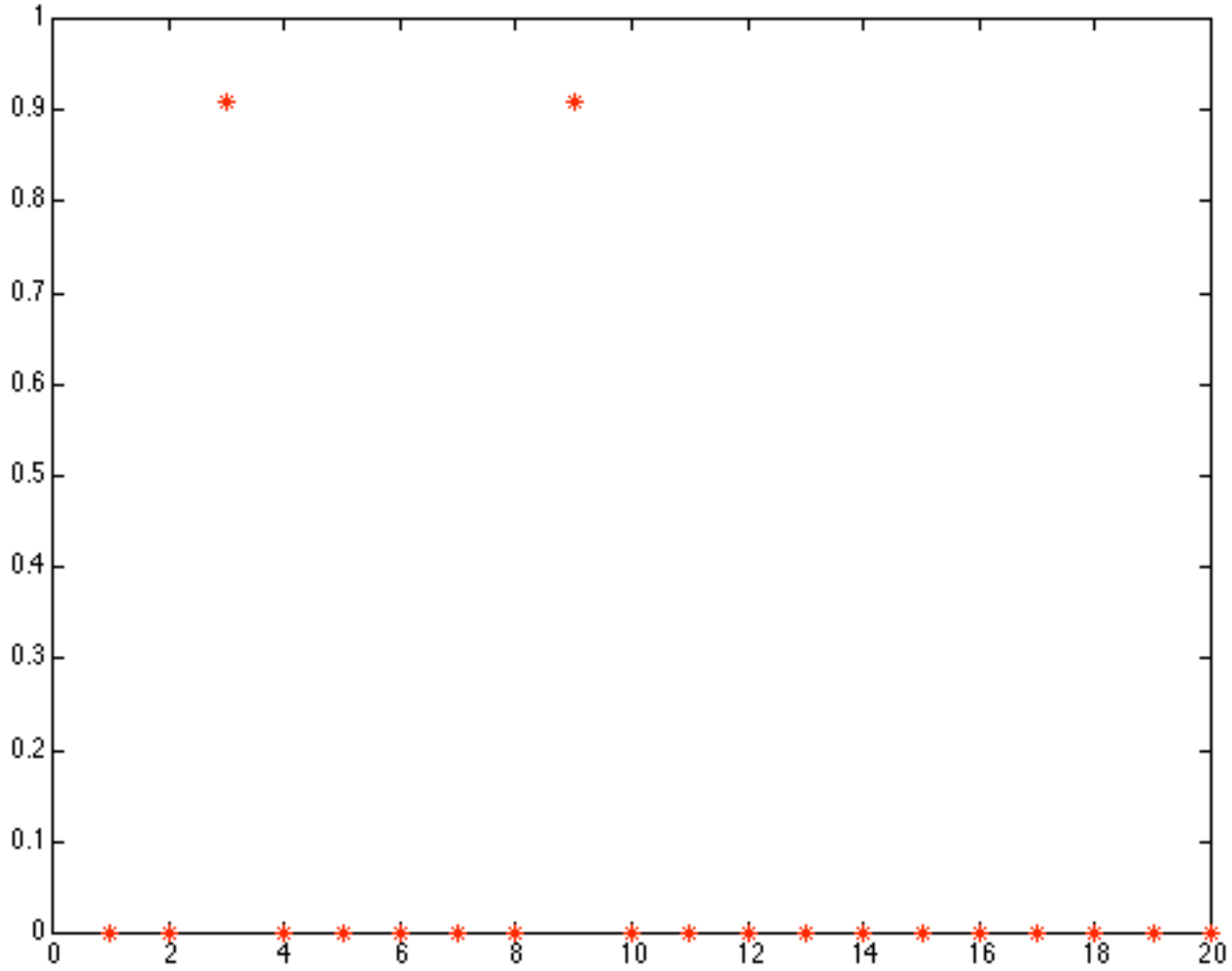
Local maximum



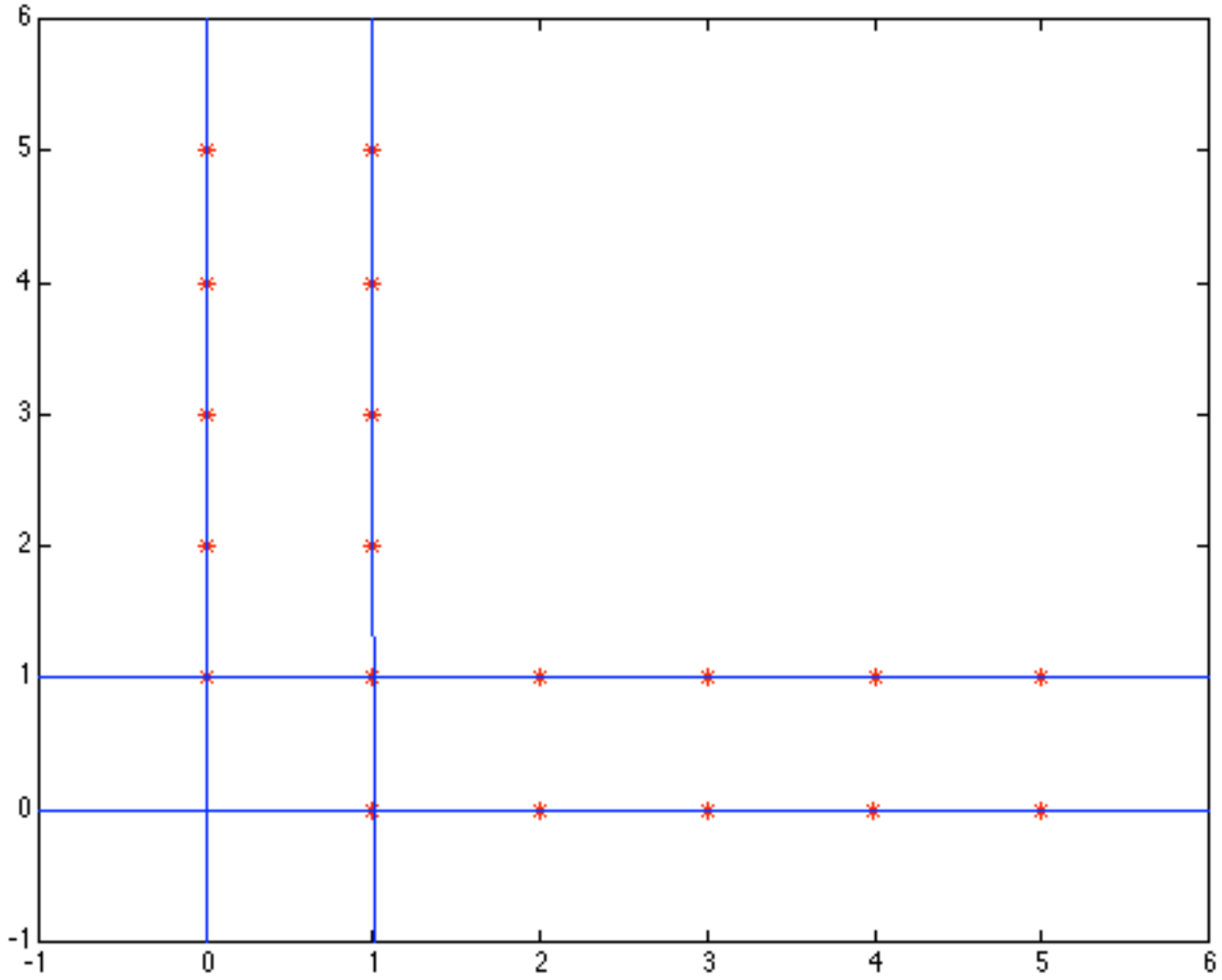
which is an excellent fit to some points



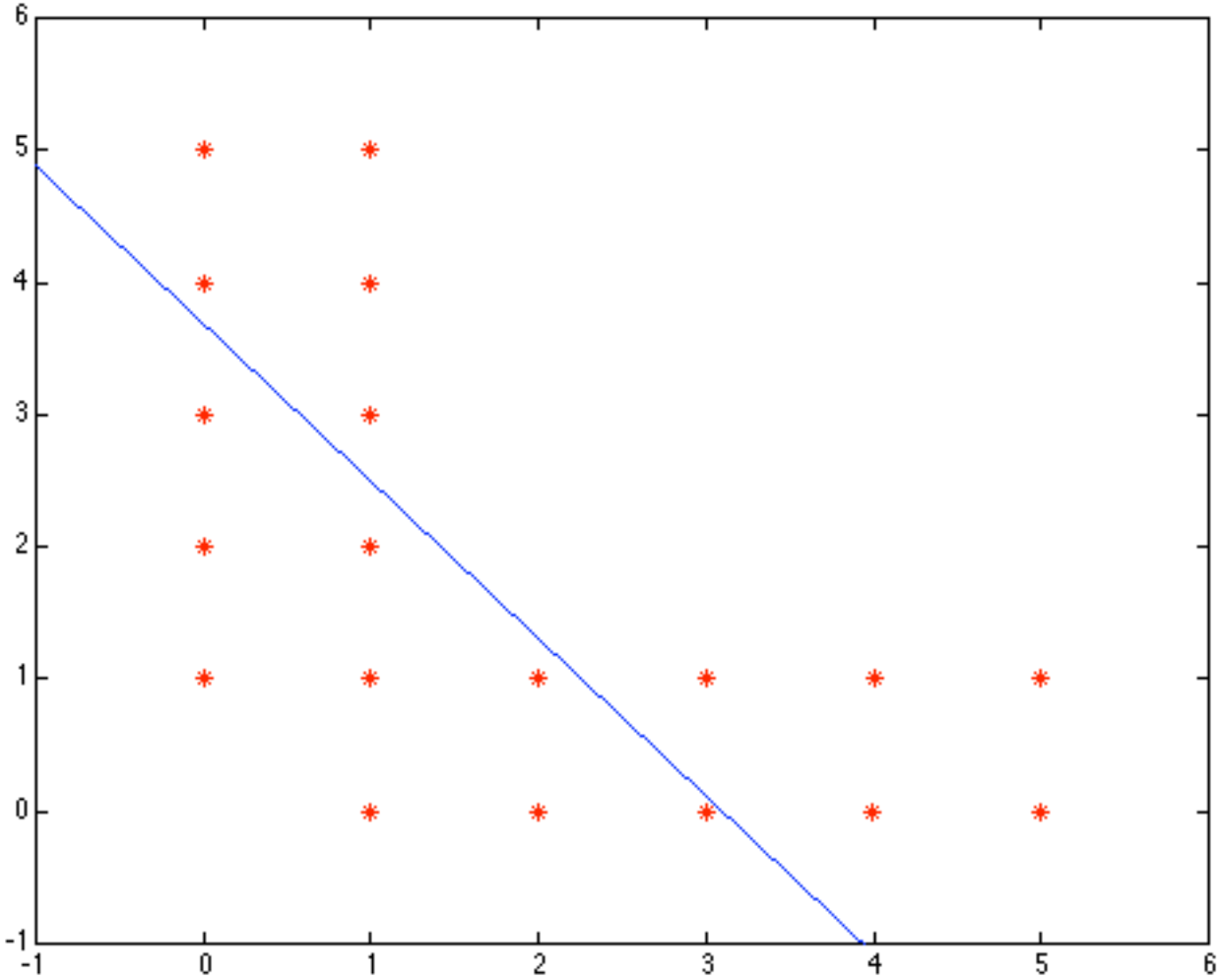
and the deltas for this maximum



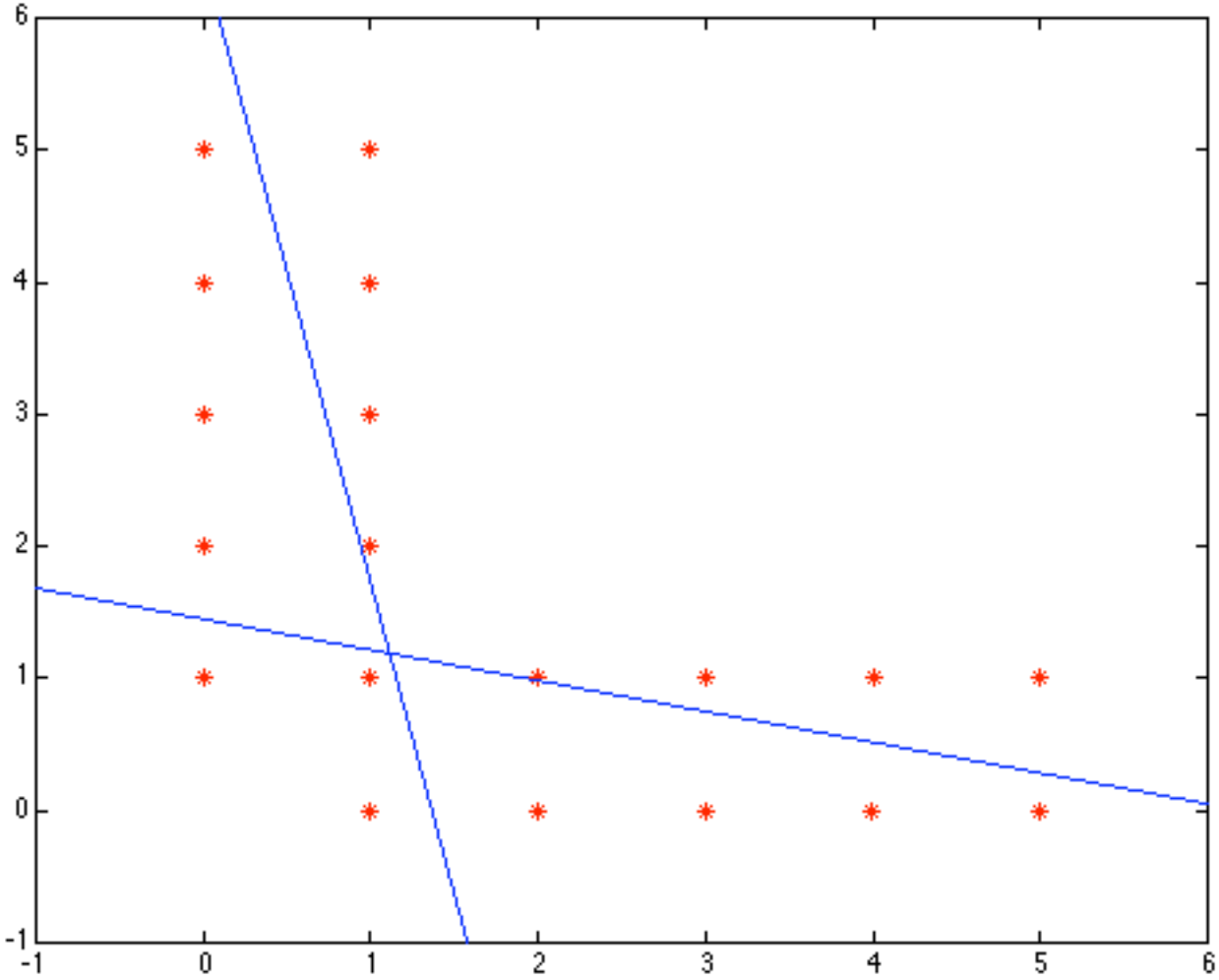
A dataset that is well fitted by four lines



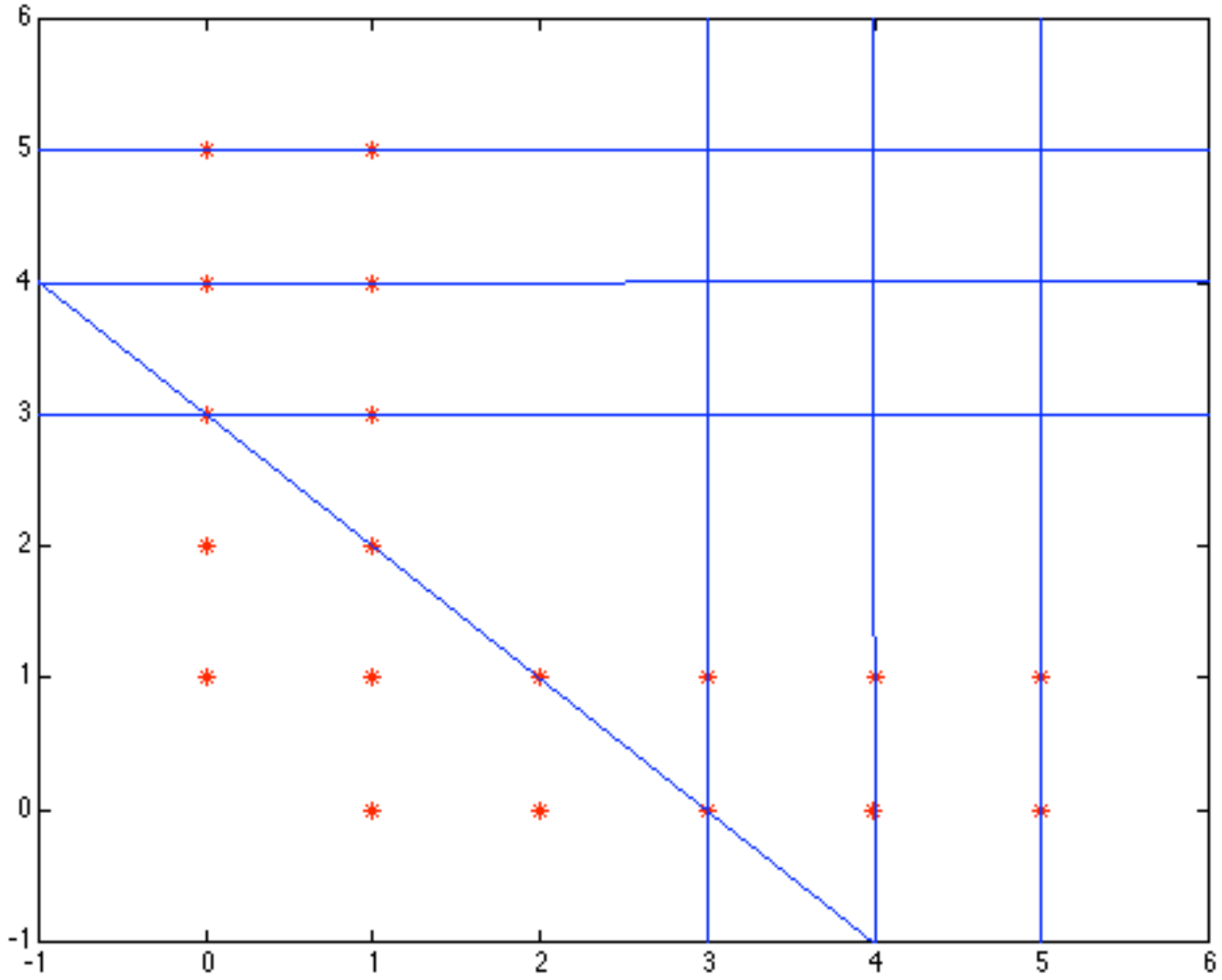
Result of EM fitting, with one line (or at least, one available local maximum).



Result of EM fitting, with two lines (or at least, one available local maximum).



Seven lines can produce a rather logical answer



Fitting multiple lines

- Rather like fitting one line, except there are more hidden variables
- Easiest is to encode as an array of hidden variables, which represent a table with a one where the i'th point comes from the j'th line, zeros otherwise
- Likely want to include a noise source as well
- Rest is similar to single line case

Segmentation/Grouping by EM

- See §16.2.1--note that we assume Gaussian for p().
- A segment is modeled as a Gaussian process that emits feature vectors (which could contain color; or color and position; or colour, texture and position).
- Segment parameters are mean and (perhaps) variance or covariance, and prior probability (was λ in the line fitting example).
- If we knew which segment each point belonged to, estimating these parameters would be easy (this point should be familiar!)

Segmentation/Grouping by EM

- Since we don't know which point comes from which segment, we have to use an **estimate** of the **probabilities** that a given point belongs to a given segment. We thus compute means and variances by **weighting** the standard formulas by these probabilities.
- Formally, these probabilities can be denoted

$$p(m \mid \mathbf{x}_l, \theta^{(s)})$$

- This gives an expression for estimates for the means of the Gaussians for each segment.

Segmentation/Grouping by EM

- We estimate the mean for each segment by:

$$\mu_m^{(s+1)} = \frac{\sum_{l=1}^{r} \mathbf{x}_l \, p(m \mid \mathbf{x}_l, \theta^{(s)})}{\sum_{l=1}^{r} p(m \mid \mathbf{x}_l, \theta^{(s)})}$$

- Variances/covariances work similarly

Segmentation/Grouping--E Step

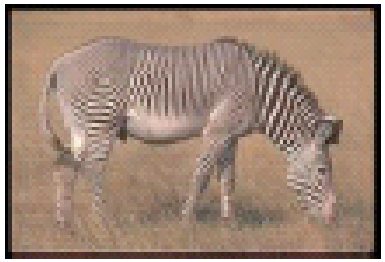
- Given parameters, the probability that a given point is associated with each cluster is easy to compute. For example

$$p(m \mid \mathbf{x}_l, \theta^{(s)}) = \frac{\pi_m^{(s)} p(\mathbf{x}_l \mid \theta_m^{(s)})}{\sum_{k=1}^{M} \pi_k^{(s)} p(\mathbf{x}_l \mid \theta_k^{(s)})}$$

- The book uses $\overline{I_{lm}}$ for $p(m \mid \mathbf{x}_i, \theta^{(s)})$ (I suggests "indicator variable")

- (Also, my copy of the book's version of the above equation looks wrong to me--the index l applies to points and the index for theta should refer to groups)

Segmentation with EM



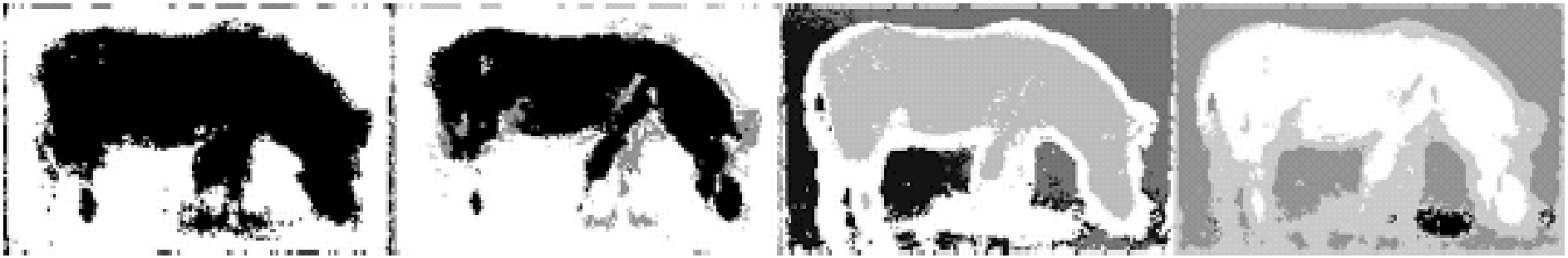


Figure from “Color and Texture Based Image Segmentation Using EM and Its Application to Content Based Image Retrieval”,S.J. Belongie et al., Proc. Int. Conf. Computer Vision, 1998, c1998, IEEE

Skip

Motion segmentation with EM

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

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- Now we need to
  - determine which pixels belong to which region
  - estimate parameters

- Likelihood
  - assume

$$I(x, y, t) = I(x + v_x, y + v_y, t + 1) + noise$$

- Straightforward missing variable problem, rest is calculation

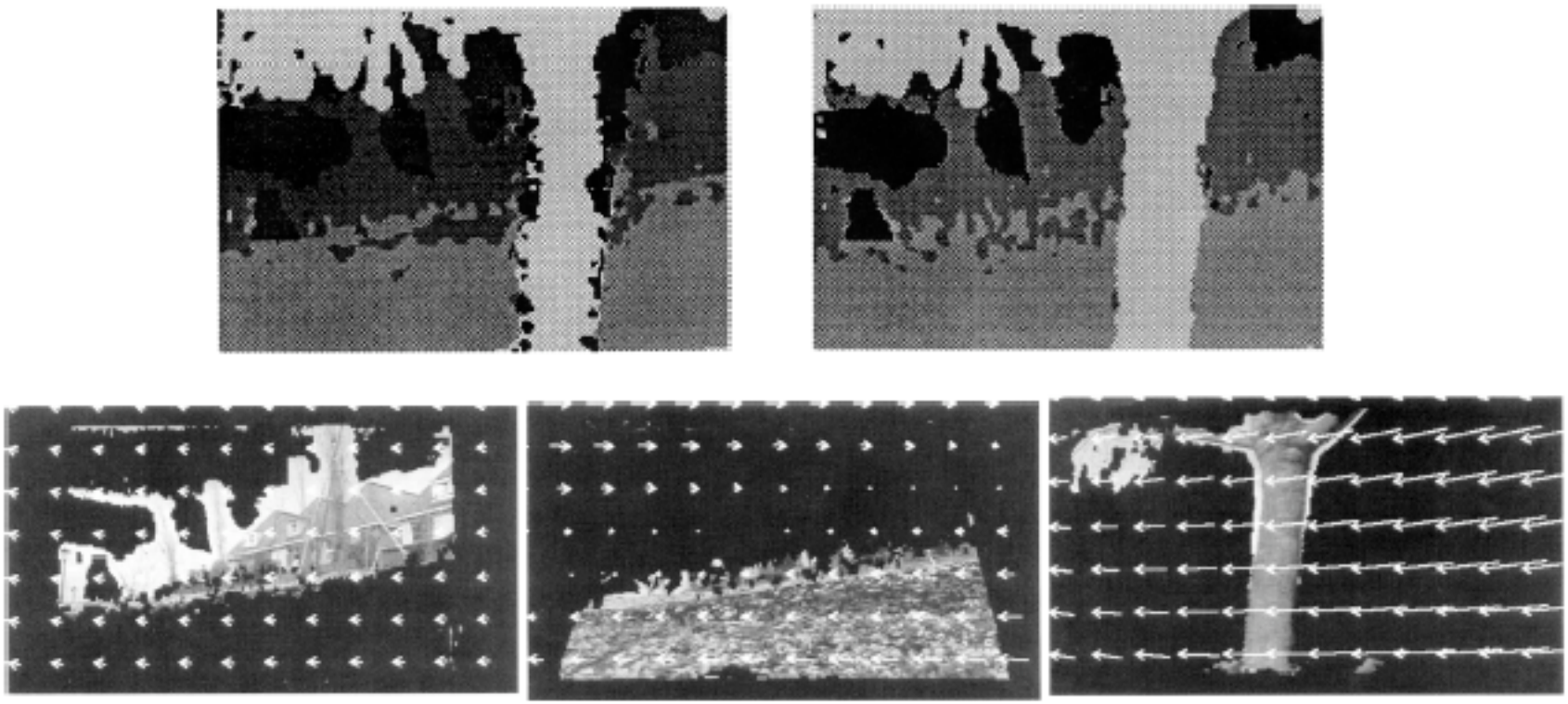
Skip

Three frames from the MPEG “flower garden” sequence

Figure from “Representing Images with layers,”, by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

Skip

Grey level shows region no. with highest probability



Segments and motion fields associated with them

Figure from “Representing Images with layers,”, by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

Skip

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.

Figure from “Representing Images with layers,”, by J. Wang and E.H. Adelson, IEEE Transactions on Image Processing, 1994, c 1994, IEEE

Skip

Some generalities

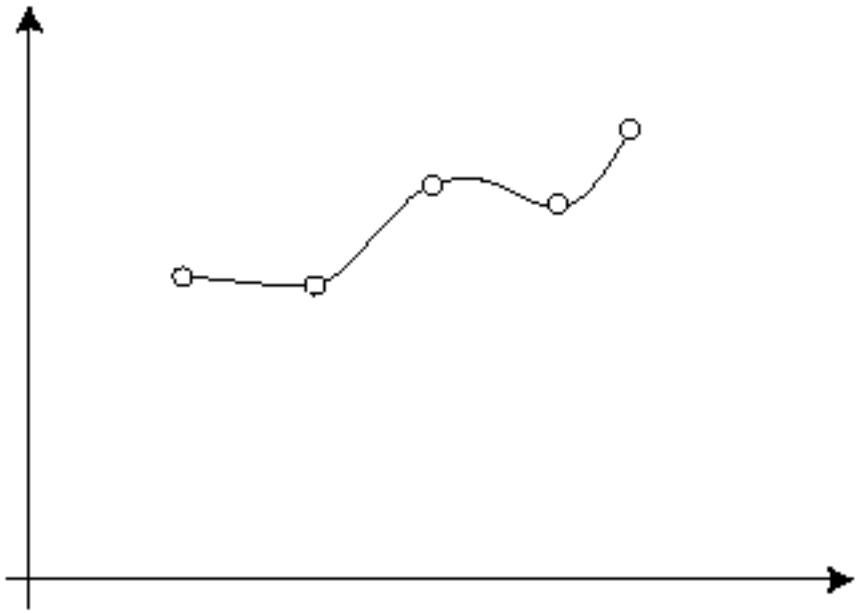
- Many, but not all problems that can be attacked with EM can also be attacked with RANSAC
  - need to be able to get a parameter estimate with a manageably small number of random choices.
  - RANSAC is usually better
- Didn't present in the most general form
  - in the general form, the likelihood may not be a linear function of the missing variables
  - in this case, one takes an expectation of the likelihood, rather than substituting expected values of missing variables
  - Issue doesn't seem to arise in vision applications.

Important

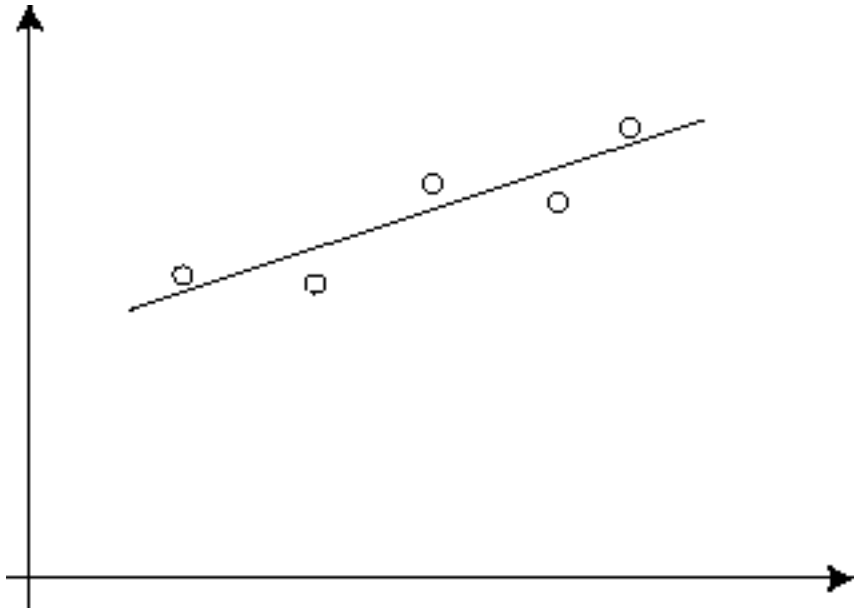
Model Selection

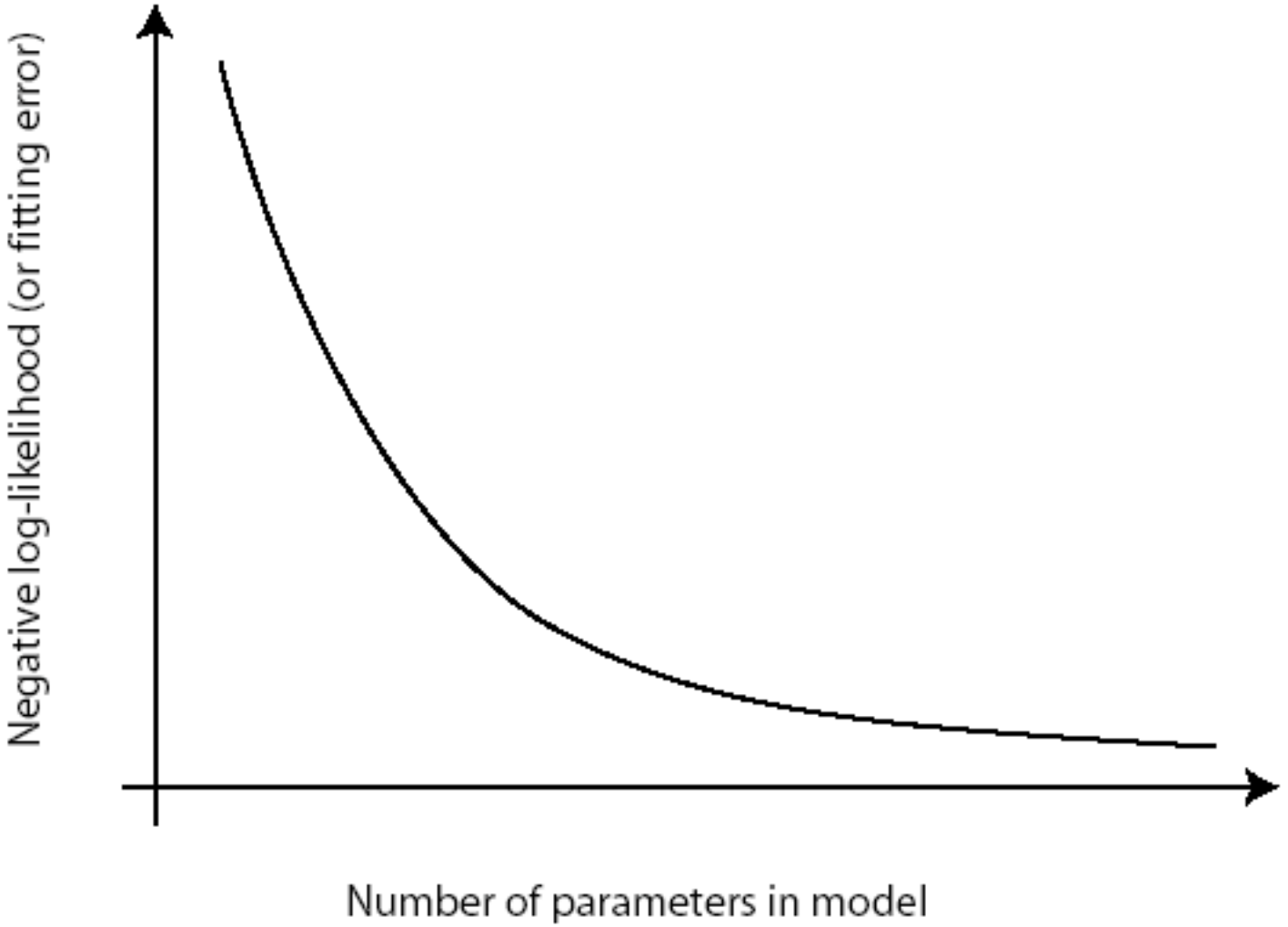
- In general, models with more parameters will fit a dataset better, but are poorer at prediction
- This means we can't simply look at the negative log-likelihood (or fitting error)

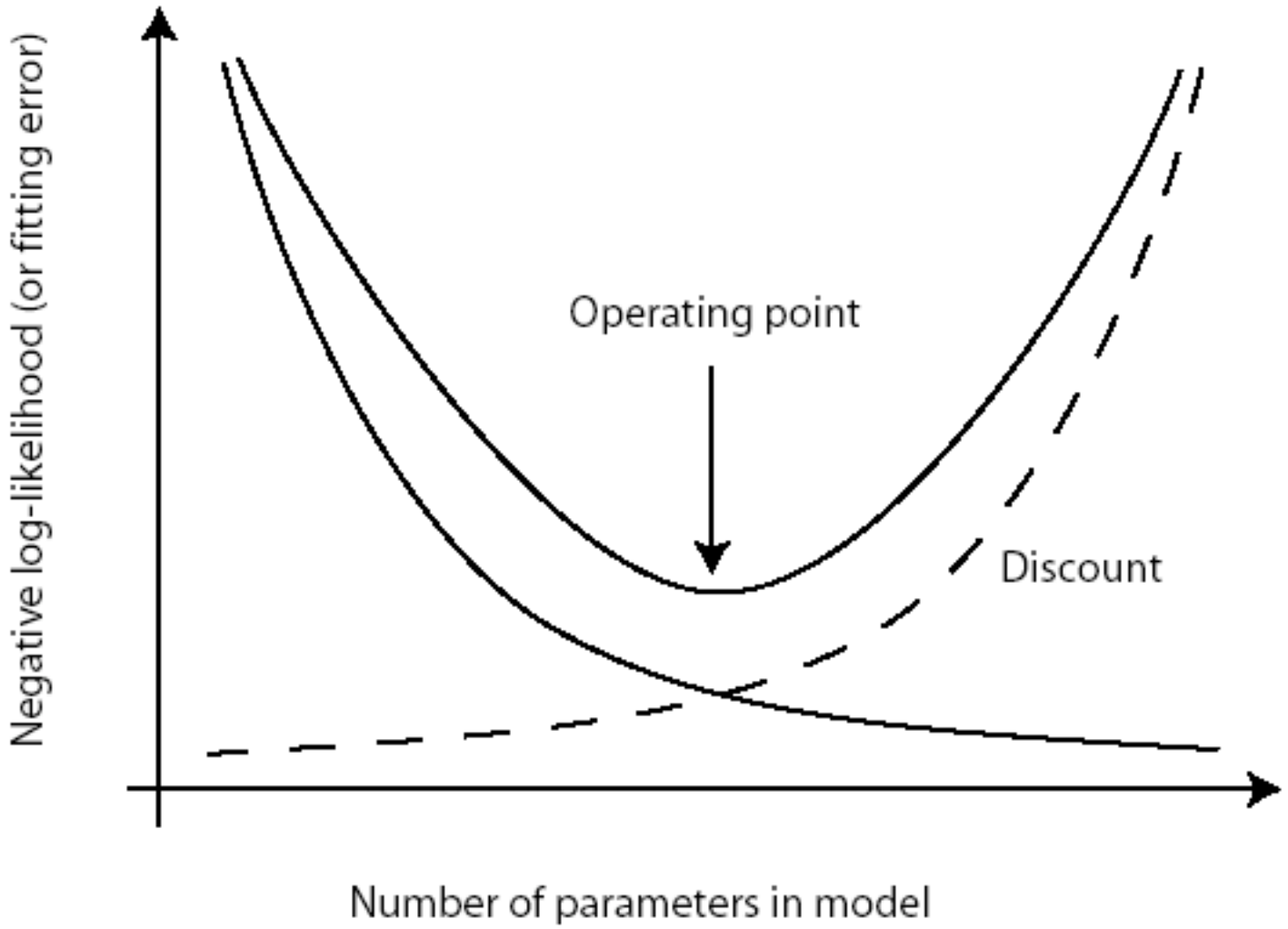
Important



Top is not necessarily a better fit than bottom
(actually, almost always worse)




Negative log-likelihood (or fitting error)
Number of parameters in model



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

Important

Cross-validation

- Split data set into two pieces, fit to one, and compute negative log-likelihood on the other
- Set used for fitting is “training data”, other set is “testing data” or “held out data”
- Average over multiple different splits
- Choose the model with the smallest value of this average
- (Optional point) 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

Skip

Model averaging

- Very often, it is smarter to use multiple models for prediction than just one
- e.g. motion capture data
  - there are a small number of schemes that are used to put markers on the body
  - given we know the scheme S and the measurements D, we can estimate the configuration of the body X

- We want

$$P(X \mid D) = P(X \mid S_1, D)P(S_1 \mid D) + P(X \mid S_2, D)P(S_2 \mid D) + P(X \mid S_3, D)P(S_3 \mid D)$$

- If it is obvious what the scheme is from the data, then averaging makes little difference
- If it isn't, then not averaging underestimates the variance of X --- we think we have a more precise estimate than we do.