Bayes Theorem again

- Bayes's Theorem begins with a statement of knowledge prior to getting the data (called the *prior)*
- The prior is in the form of a probability density. It can be based on physics, on the results of other experiments, on expert opinion, or any other source of relevant information.
- To improve this state of knowledge (hypothesis or prior distribution), a measurement is taken (data)
- Bayes's Theorem is the mechanism used to update the state of knowledge to provide a *posterior* distribution.

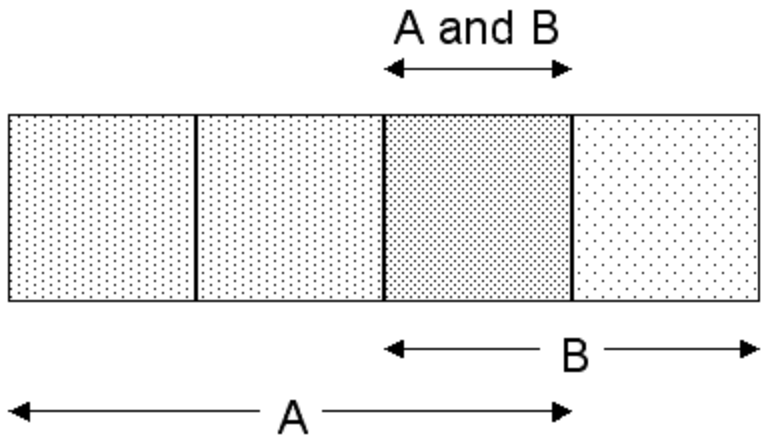
- Both the prior and the measurement have a joint distribution
- Let the measurement be A and the prior be B.
- Since both have occurred, the event AB has taken place.
- The probability of both A and B happening together is P(AB).
- This probability can be found as the product of the conditional probability of one, given the other, times the probability of the other.
- $P(A|B) \cdot P(B) = \mathbf{P(AB)} = P(B|A) \cdot P(A)$

  (assuming both P(A) and P(B) are non zero)

Bayes's Theorem

- $P(B|A) = \frac{P(A|B)\ P(B)}{P(A)}$
- Posterior probability of B (the updated prediction) is the product of the conditional probability of the data, given the influence of the parameters being investigated, times the prior probability of those parameters.
- Division by the total probability of A assures that the resulting quotient falls on the [0, 1] interval, as all probabilities must.

Venn Diagram illustrating absolute, conditional and joint probabilities.



P(A) = 3/4

P(B) = 2/4

P(A and B) = P(AB) = 1/4

P(A|B) = P(AB) / P(B) = (1/4) / (2/4) = 1/2

P(B|A) = P(AB) / P(A) = (1/4) / (3/4) = 1/3

Bayesian inference

- Bayes' rule: $$P(H \mid D) = \frac{P(H)P(D \mid H)}{P(D)}$$
- What makes a good scientific argument?
- $P(H|D)$ is high if:
  - Hypothesis is plausible: $P(H)$ is high
  - Hypothesis strongly predicts the observed data:

    $P(D|H)$ is high
  - Data are surprising: $P(D)$ is low

K-means clustering

- Brute force difficult because many spheres, many pixels.
- Assume all spheres same radius; just need sphere centers.
- Iterative method.
  - If we knew centers, it would be easy to assign pixels to clusters.
  - If we knew which pixels in each cluster, it would be easy to find centers.
  - So guess centers, assign pixels to clusters, pick centers for clusters, assign pixels to clusters, ….

K-means Algorithm

1. Initialize – Pick k random cluster centers
   - Pick centers *near* data. Heuristics: uniform distribution in range of data; randomly select data points.
2. Assign each point to nearest center.
3. Make each center average of pts assigned to it.
4. Go to step 2.

simple example.

Suppose we want to cluster black and white intensities, and we have the intensities: 1 3 8 11.

start with centers c1 = 7 and c2=10.

assign 1, 3, 8 to c1, 11 to c2.

update c1 = (1+3+8)/3 = 4, c2 = 11.

assign 1,3 to c1 and 8 and 11 to c2.

update c1 = 2, c2 = 9 ½.

converged.

No assignments change, so the centers don’t change.

K-means Properties

- Think of this as trying to find the optimal solution to:
  - Given points p1… pn, find centers c1…ck
  - and find mapping f:{p1…pn}->{c1…ck}

  that minimizes

  $$C = (p1-f(p1))^2 + \ldots + (pn-f(pn))^2.$$
- Every step reduces C.
  - The mean is the pt that minimizes sum of squared distance to a set of points. So changing the center to be the mean reduces this distance.
  - When we reassign a point to a closer center, we reduce its distance to its cluster center.
- Convergence: since there are only a finite set of possible assignments.

Local Minima

- However, algorithm might not find the best possible assignments and centers.
- Consider points 0, 20, 32.
  - K-means can converge to centers at 10, 32.
  - Or to centers at 0, 26.

E-M (Expectation maximization)

- Like K-means but with soft assignment.
  - Assign point partly to all clusters based on probability it belongs to each.
  - Compute weighted averages (c_j) and variance (σ).

$$f_j(p_i) = \frac{e^{-\|p_i - c_j\|^2 / \sigma^2}}{\sum_j e^{-\|p_i - c_j\|^2 / \sigma^2}}$$

Cluster centers are c_j.

Example

- *Matlab: tutorial2*
- Fuzzy assignment allows cluster to creep towards nearby points and capture them.

E-M/K-Means domains

- Used color/intensity as example.
- But same methods can be applied whenever a group is described by parameters and distances.
- Lines (circles, ellipses); independent motions; textures (a little harder).

E-M

- Reading:
  - Forsyth & Ponce 16.1, 16.2
  - Forsyth & Ponce 16.3
  - Yair Weiss: *Motion Segmentation using EM – a short tutorial.* (1st 2 pages.)

Parametric Methods

- We discussed Ransac, Hough Transform.
- The have some limitations
  - Object must have few parameters.
  - Finds an answer, but is it the best answer?
  - Hard to say because problem definition a bit vague.

Expectation-Maximization (EM)

- Can handle more parameters.
- Uses probabilistic model of all the data.
  - Good: we are solving a well defined problem.
  - Bad: Often we don’t have a good model of the data, especially noise.
  - Bad: Computations only feasible with pretty simple models, which can be unrealistic.
- Finds (locally) optimal solution.

E-M Definitions

- Models have parameters: u
  - Examples: line has slope/intercept; Gaussian has mean and variance.
- Data is what we know from image: y
  - Examples: Points we want to fit a line to.
- Assignment of data to models: z
  - Eg., which points came from line 1.
  - z(i,j) = 1 means data i came from model j.
- Data and assignments (y & z): x.

E-M Definitions

- Missing Data: We know y. Missing values are u and z.
- Mixture model: The data is a mixture of more than one model.

E-M Quick Overview

- We know data (y).
- Want to find assignments (z) and parameters (u).
- If we know y & u, we can find z more easily.
- If we know y & z, we can find u more easily.
- Algorithm:
  1. Guess u.
  2. Given current guess of u and y, find z. (E)
  3. Given current guess of z and y, find u. (M)
  4. Go to 2.

Example: Histograms

- Histogram gives 1D clustering problem.
- Constant regions + noise = Gaussians.
- Guess mean and variance of pixel intensities.
- Compute membership for each pixel.
- Compute means as weighted average.
- Compute variance as weighted sample variance.
- Details: *whiteboard;* Also, Matlab and Weiss.

More subtle points

- Guess must be reasonable, or we won't converge to anything reasonable.
  - Seems good to start with high variance.
- How do we stop.
  - When things don't change much.
  - Could look at parameters (u).
  - Or likelihood of data.

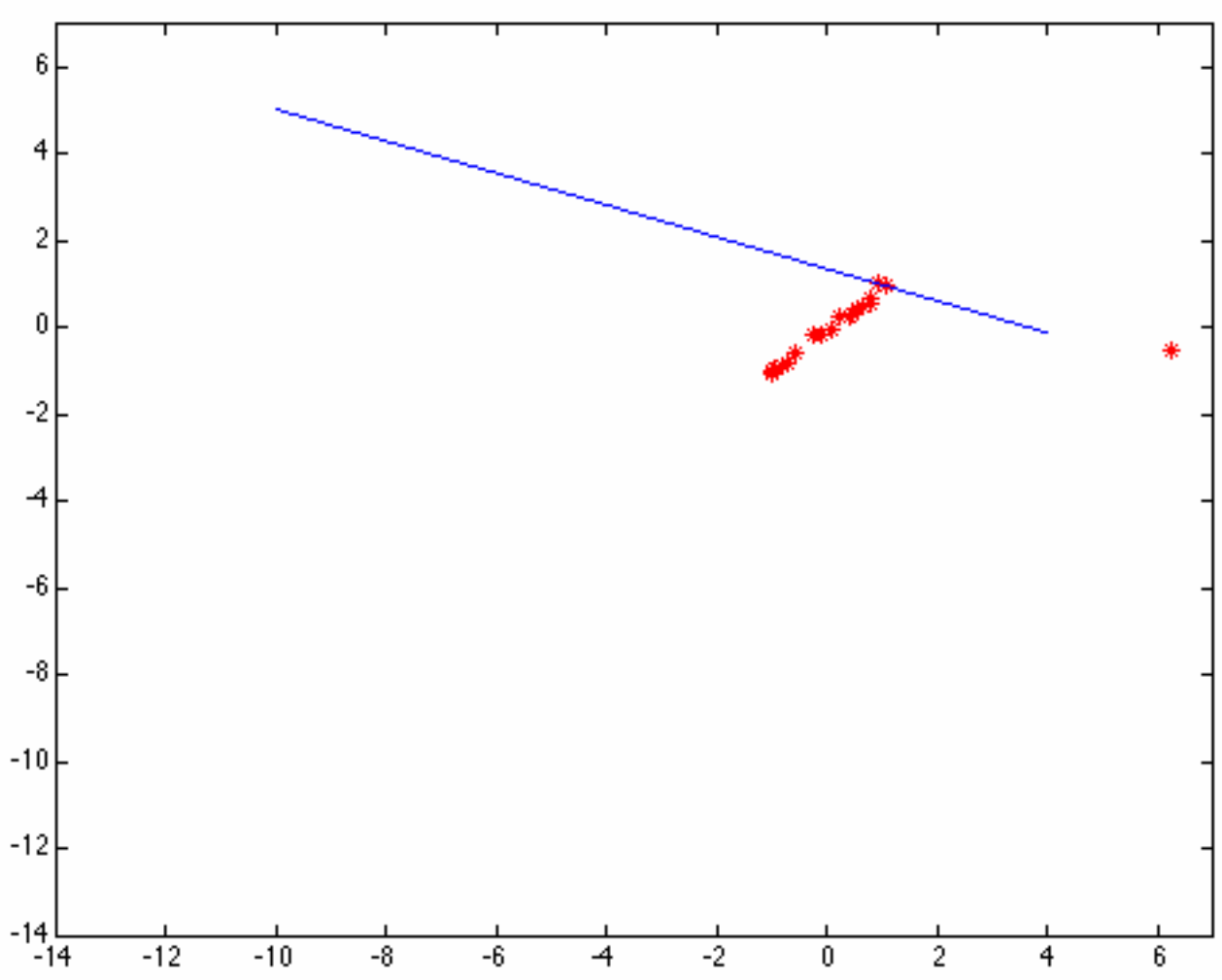
Overview again

- Break unknowns into pieces. If we know one piece, other is solvable.
- Guess piece 1.
- Solve for piece 2. Then solve for 1. ….
- Very useful strategy for solving problems.

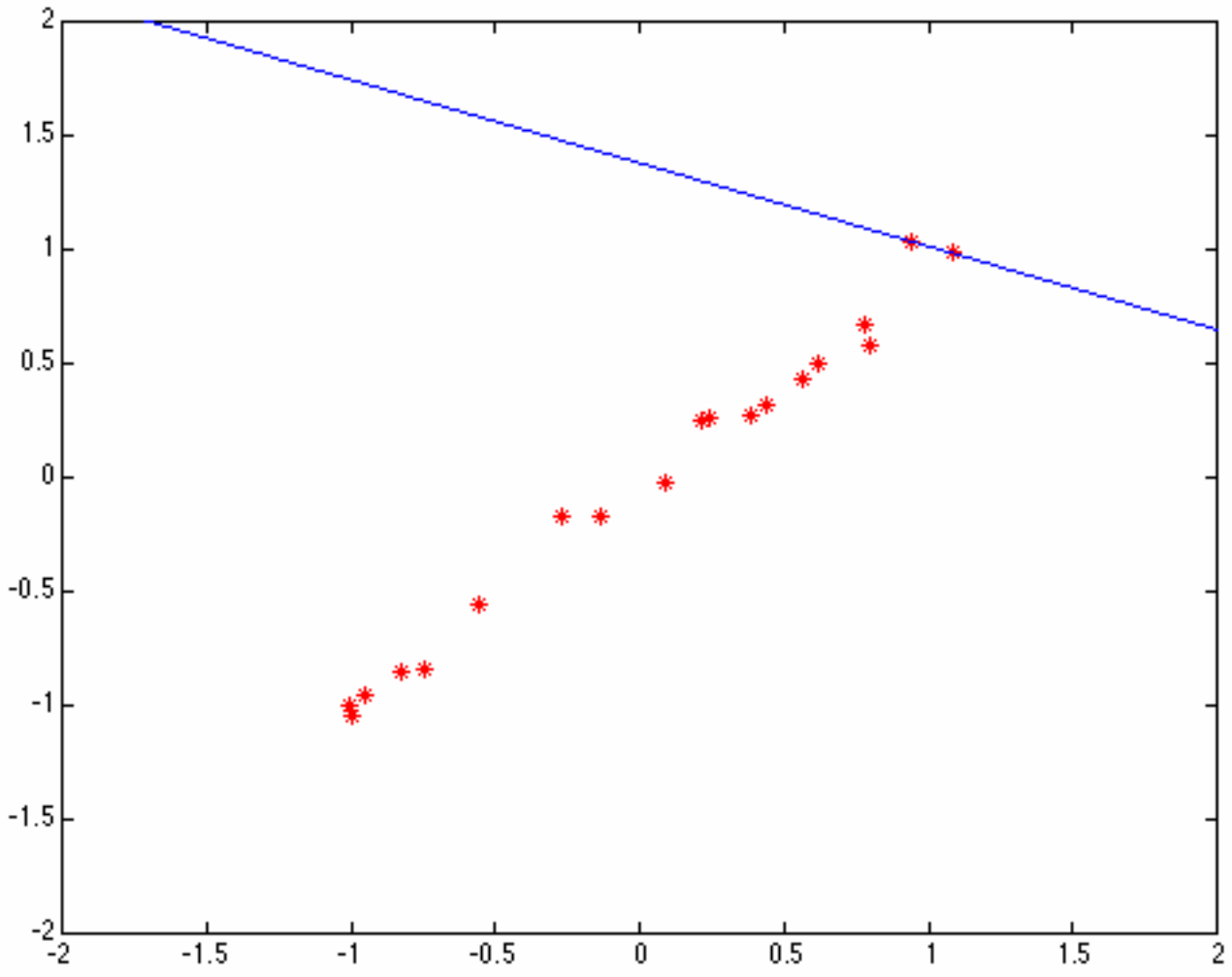
Drawbacks

- Local optimum.
- Optimization: we take steps that make the solution better and better, and stop when next step doesn’t improve.
- But, we don’t try all possible steps.

Local maximum



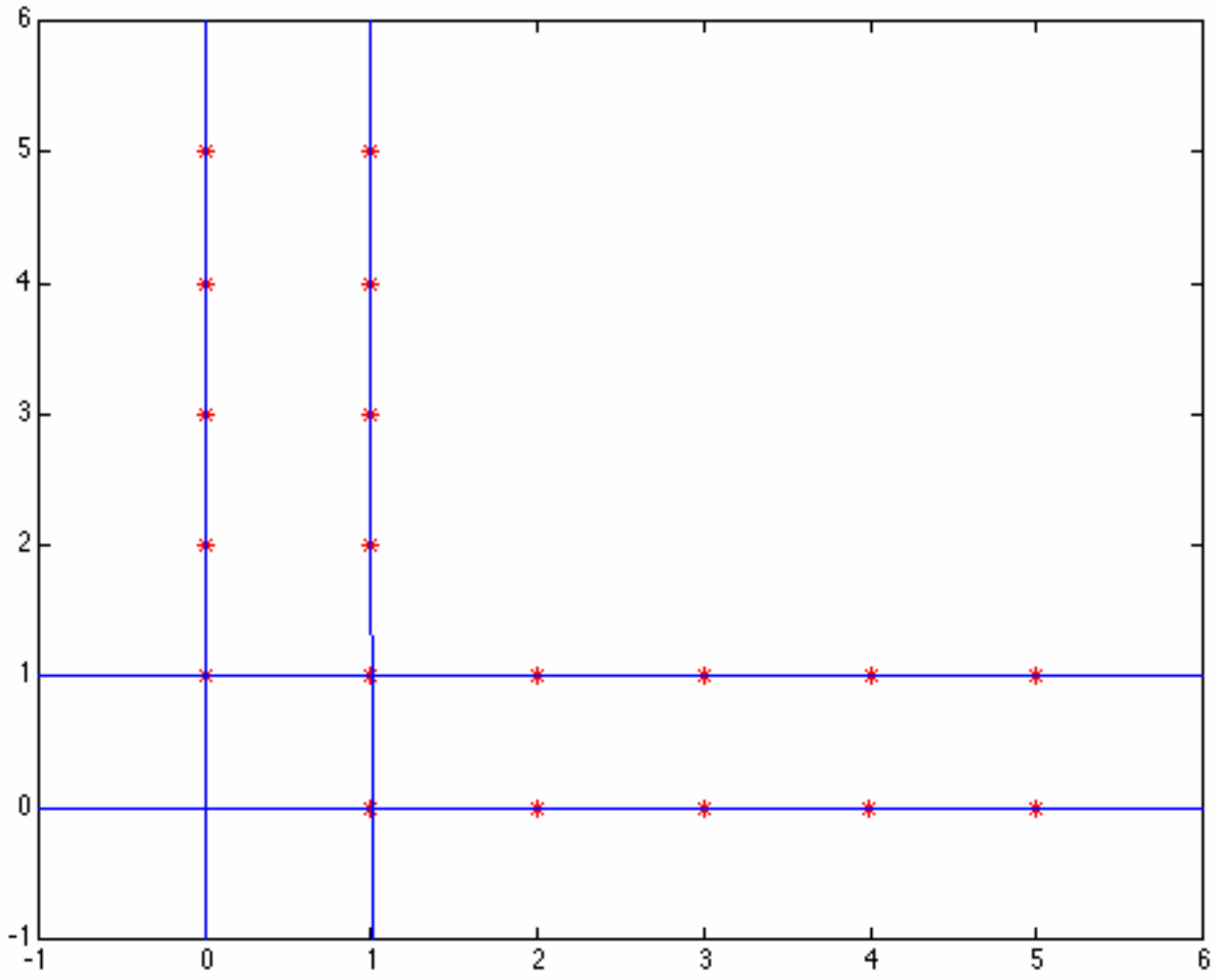
which is an excellent fit to some points



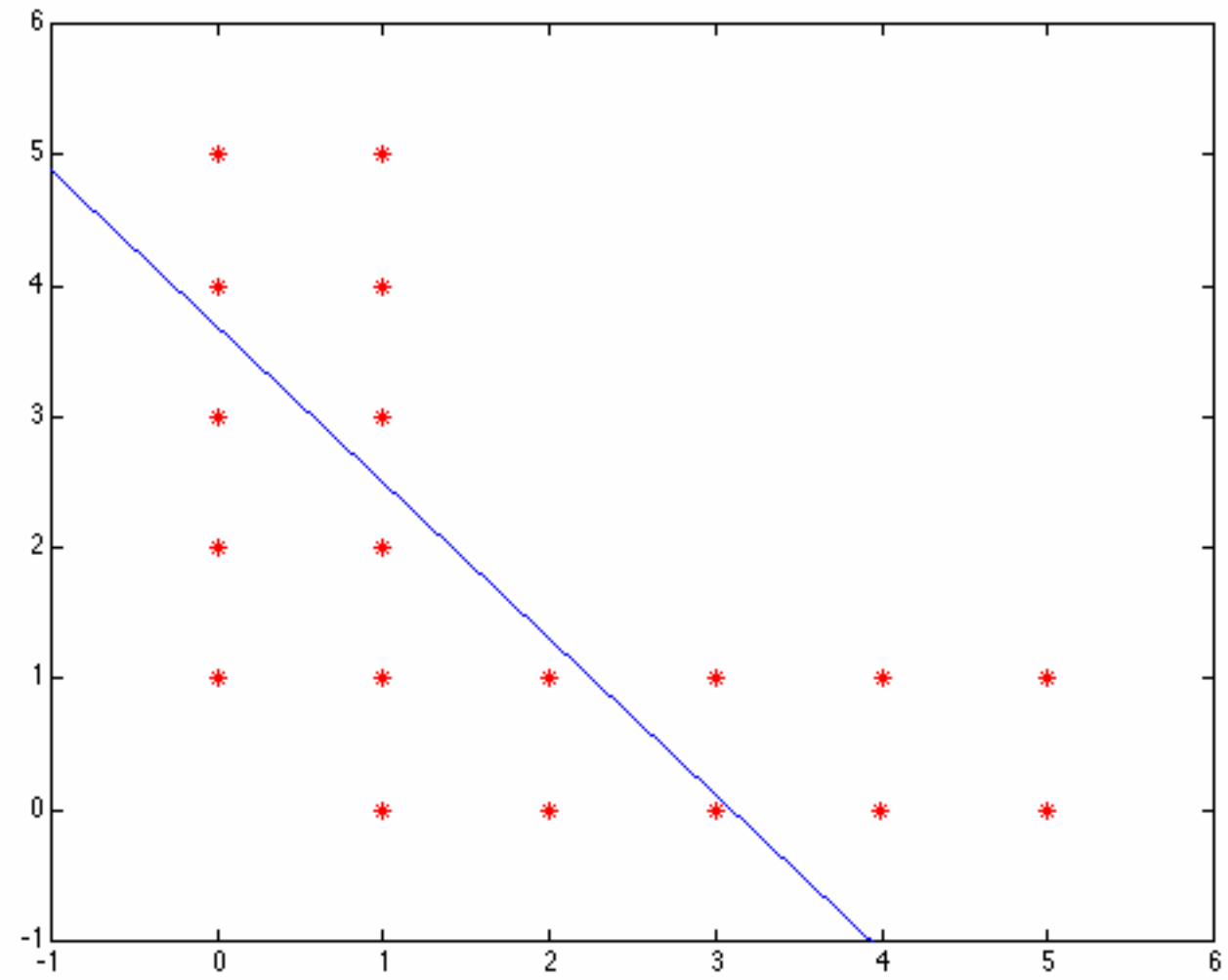
Drawbacks

- How do we get models?
  - But if we don’t know models, in some sense we just don’t understand the problem.
- Starting point? Use non-parametric method.
- How many models?

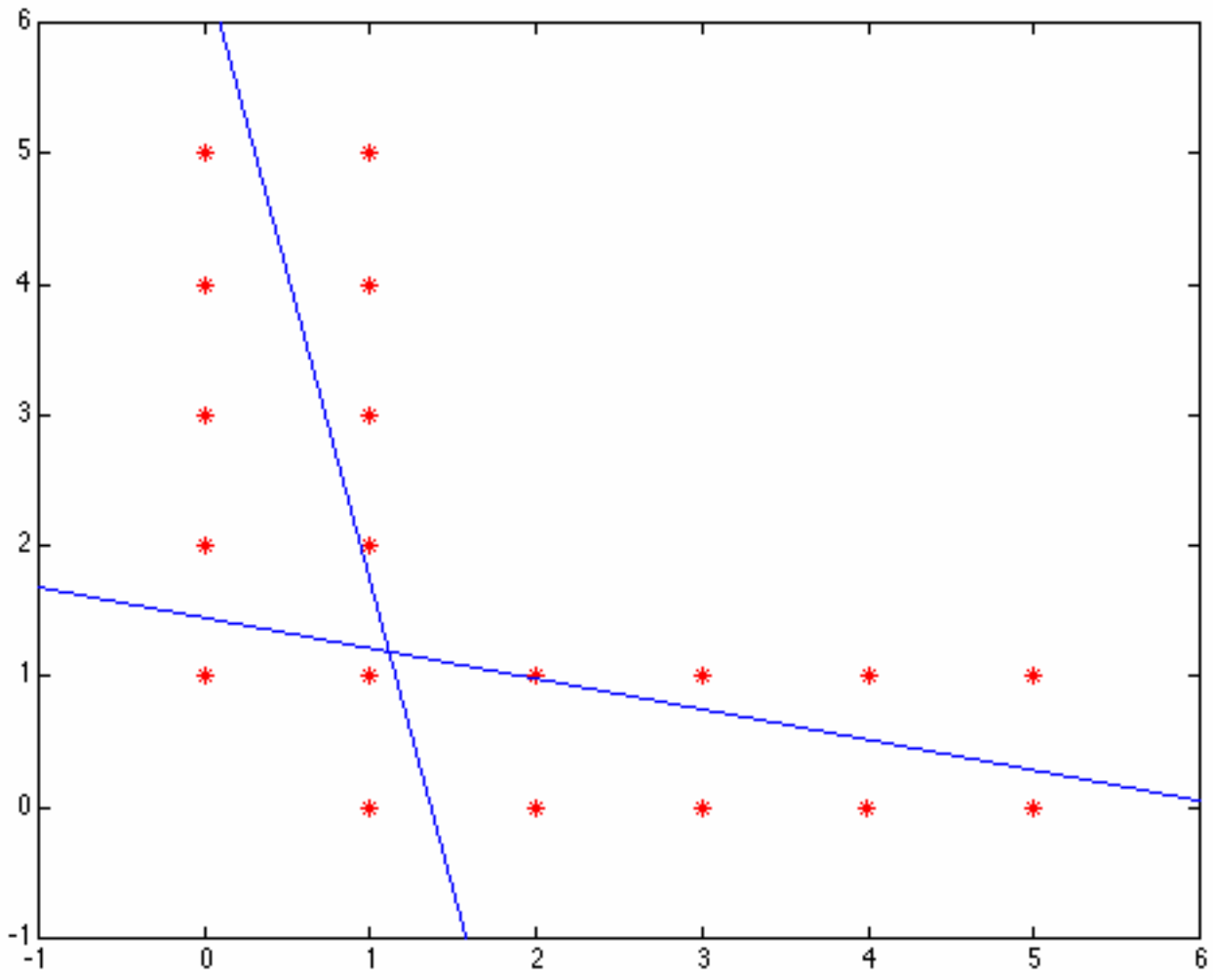
A dataset that is well fitted by four lines



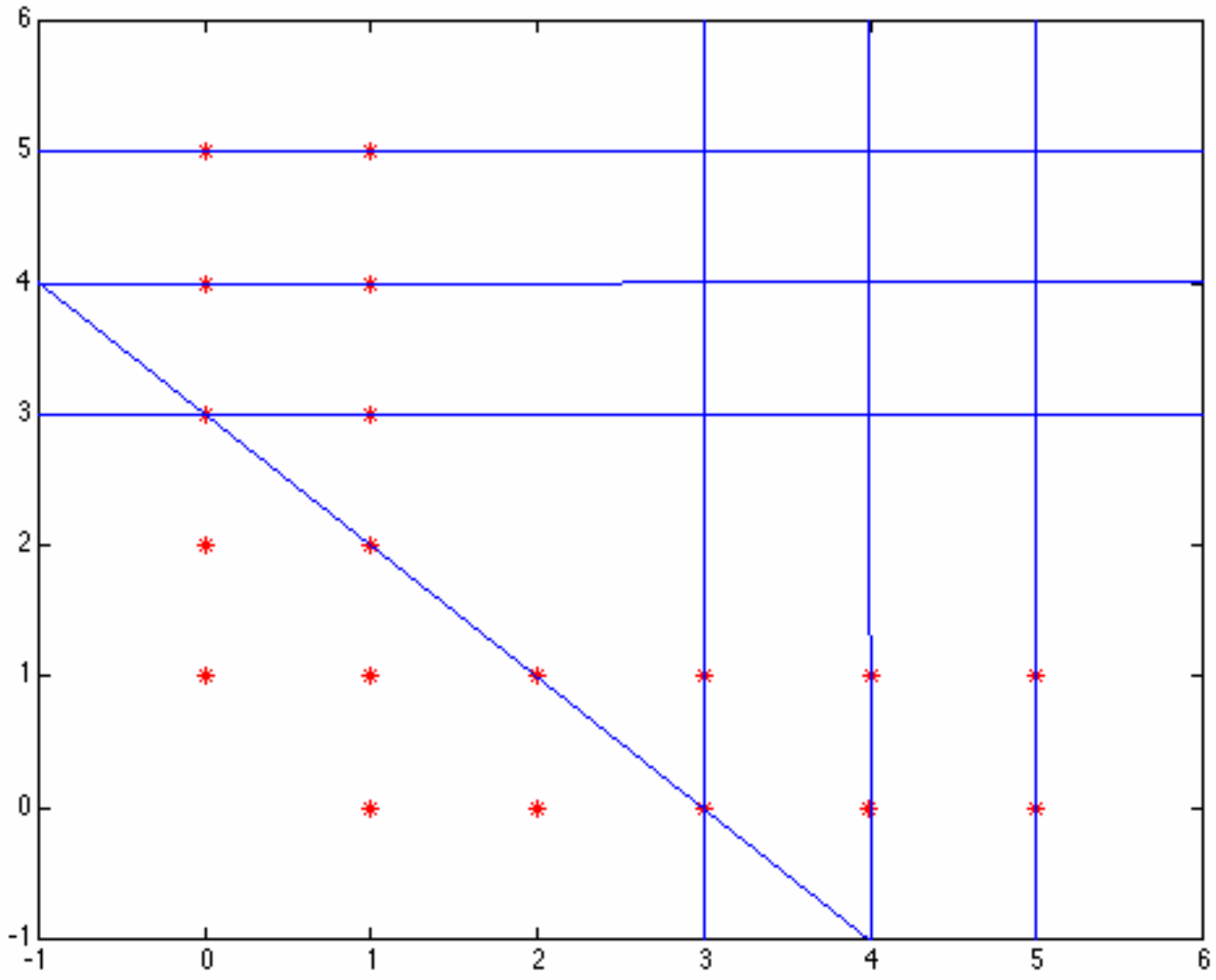
Result of EM fitting, with one line (or at least, one available local maximum).



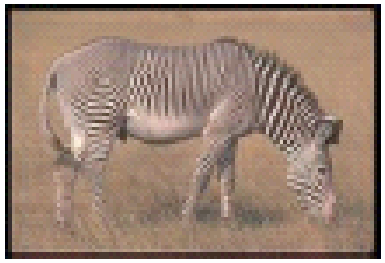
Result of EM fitting, with two lines (or at least, one available local maximum).



Seven lines can produce a rather logical answer



Segmentation with EM



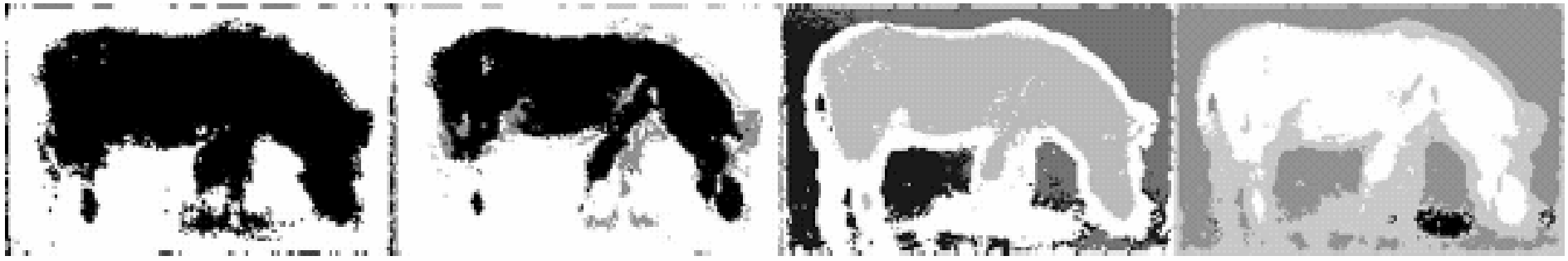


Figure from “Color and Texture Based Image Segmentation Using EM and Its Application to Content Based Image Retrieval”,S.J. Belongie et al., Proc. Int. Conf. Computer Vision, 1998, c1998, IEEE

Motion segmentation with EM

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

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

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

- Likelihood
  - assume

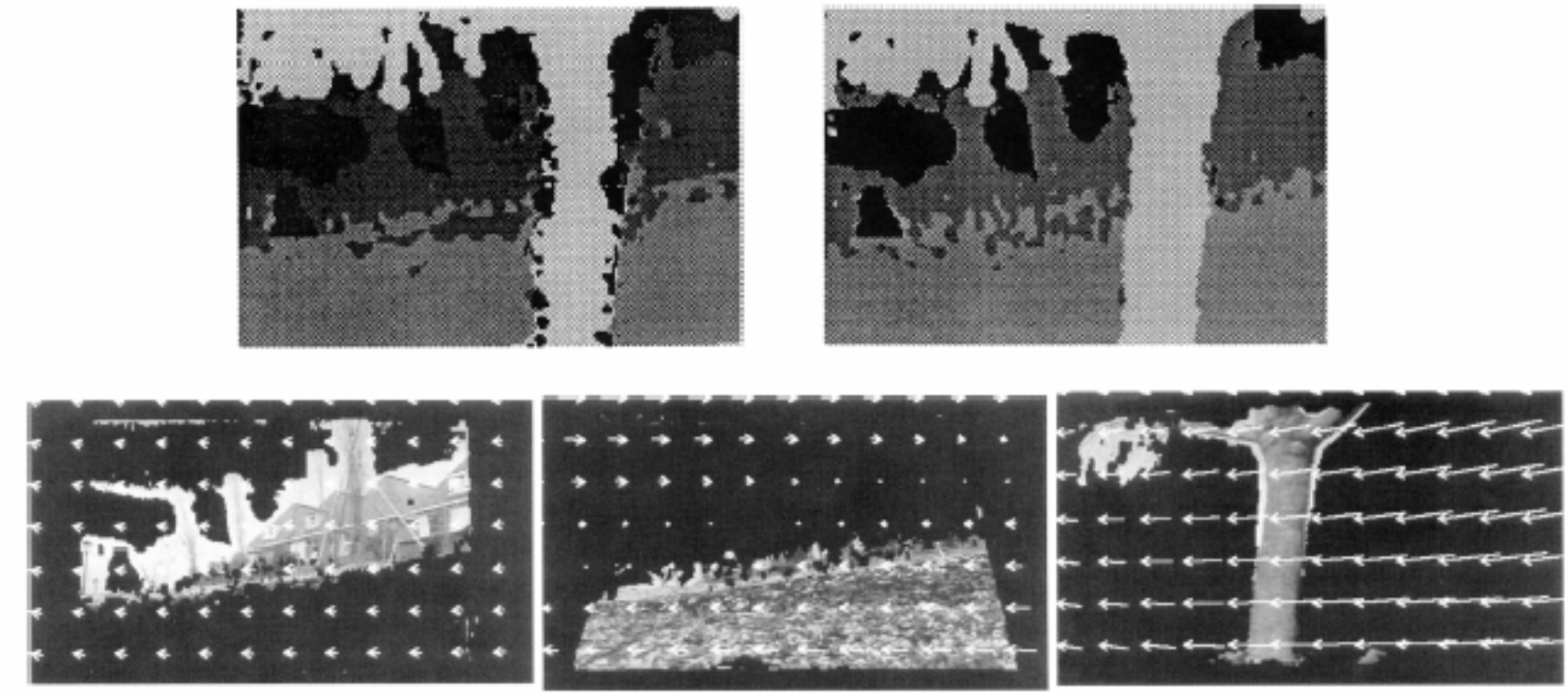
$$I(x, y, t) = I(x + v_x, y + v_y, t + 1) + noise$$

- Straightforward missing variable problem, rest is calculation

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

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

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

Probabilistic Interpretation

- We want P(u | y)
  - (A probability distribution of models given data).
- Or maybe P(u,z | y). Or argmax(u) P(u|y).
- We compute: argmax(u,z) P(y | u,z).
  - Find the model and assignments that make the data as likely to have occurred as possible.
  - This is similar to finding most likely model and assignments given data, ignoring prior on models.