Hidden variable models

Comments

- Mini-project sign-up google doc
- No coding on final (just fundamentals)

Summary of last time

- Discussed more about dimensionality reduction
  - and how can use auto-encoders or factorization
- Discussed generalization to losses and priors
  - maximum likelihood and MAP again
- Discussed matrix completion

Matrix completion

- Learned embeddings for users and movies into a common space
  - i.e., h_i and d_j are both k-dimensional representations
  - we can actually look at similarities between users by comparing their embedding vectors (or even similarity between a user and a movie!)
- Why is this useful?
  - allows prediction (completion) of unknown elements in the matrix
  - its a way to learn metrics

Embeddings with co-occurence

- Embed complex items into a shared (Euclidean) space based on their relationships to other complex items
- Examples:
  - word2vec
  - users and movies
  - gene sequences

Consider word features

- Imagine want to predict whether a sentence is positive or negative (say with logistic regression)
- How do we encode words?
- One basic option: a one-hot encoding. If there are 10000 words, the ith word has a 1 in the ith location of a 10000 length vector, and zero everywhere else.
- This is a common way to deal with categorical variables, but with 10000 words this can get big!
- Can we get a more compact representation of a word?

Co-occurence matrix example: word2vec

- X is count of words (rows) and context (columns), where for word i the count is the number of times a context word j is seen within 2 words (say) of word i
- Each word is a one-hot encoding; if there are 10000 words, each row corresponds to 1 word, and X is 10000x10000
  - I like deep learning.
  - I like NLP.
  - I enjoy flying.

counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

How obtain embeddings?

- Words i and s that have similar context counts should be embedded similarly

- Factorize co-occurrence matrix
  - or some measure of how items are related, e.g. rank or probability

- X = H D —> What is H_{i:}, and what is D_{:j}? Which should we use?

- Take complex (non-numerical) data like words, and extract a numerical vector of k features

How might you use an auto-encoder?

- Recall equivalence between PCA and linear auto-encoders
- Can factorize X = HD to get embeddings
- Can learn auto-encoder X B A, where XB is embedding
  - can we get the embedding for the context words?
- What is the input, if want to think of this as samples?
- For a new word, how do I get the embedding?

Does it have to be co-occurrence?

- First layer of a neural network could generate an embedding, even if just learning to predict accurately
  - say want to predict if a blog post is about politics
- Input could be one-hot encoding, XB produces embedding
  - i.e., the representation of the word
- Depending on the prediction problem, supervised approach may not result in interesting embeddings
  - why not?

Thought exercise

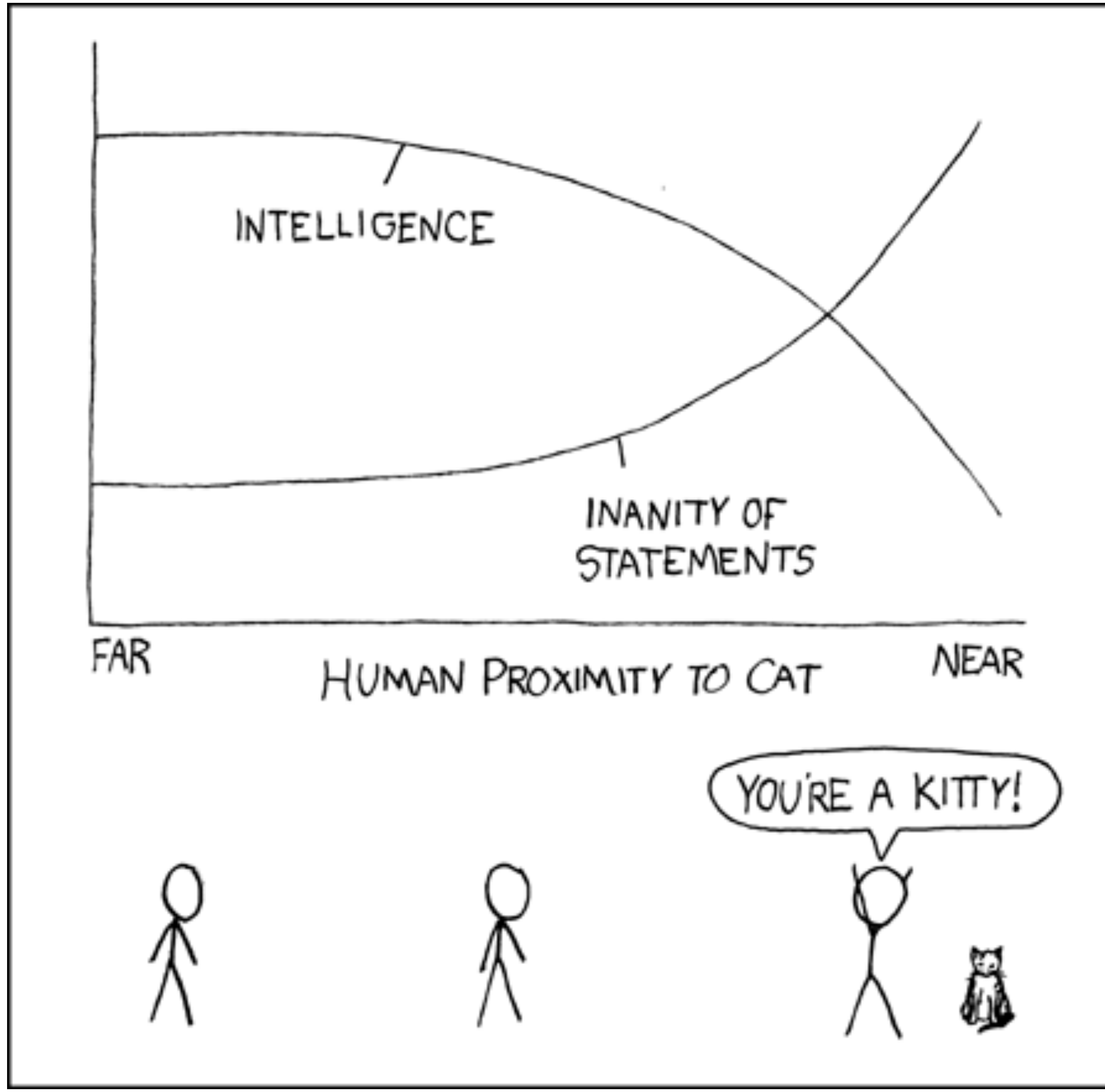
- Assume someone gave you a count matrix for words
- How might you learn a sparse embedding?
- Why might this be useful?

Hidden variables

- Different from missing variables, in the sense that for missing information we *could* have observed the missing data
  - e.g., if the person had just filled in the box on the form
- Hidden variables are never observed; rather they are useful for model description
  - e.g., hidden, latent representation
  - e.g., hidden state that drives dynamics
- Hidden variables make specification of distribution simpler
  - p(x) = \int_h p(x I h) p(h)
  - p(x I h) can be much simpler to specify

Intuitive example

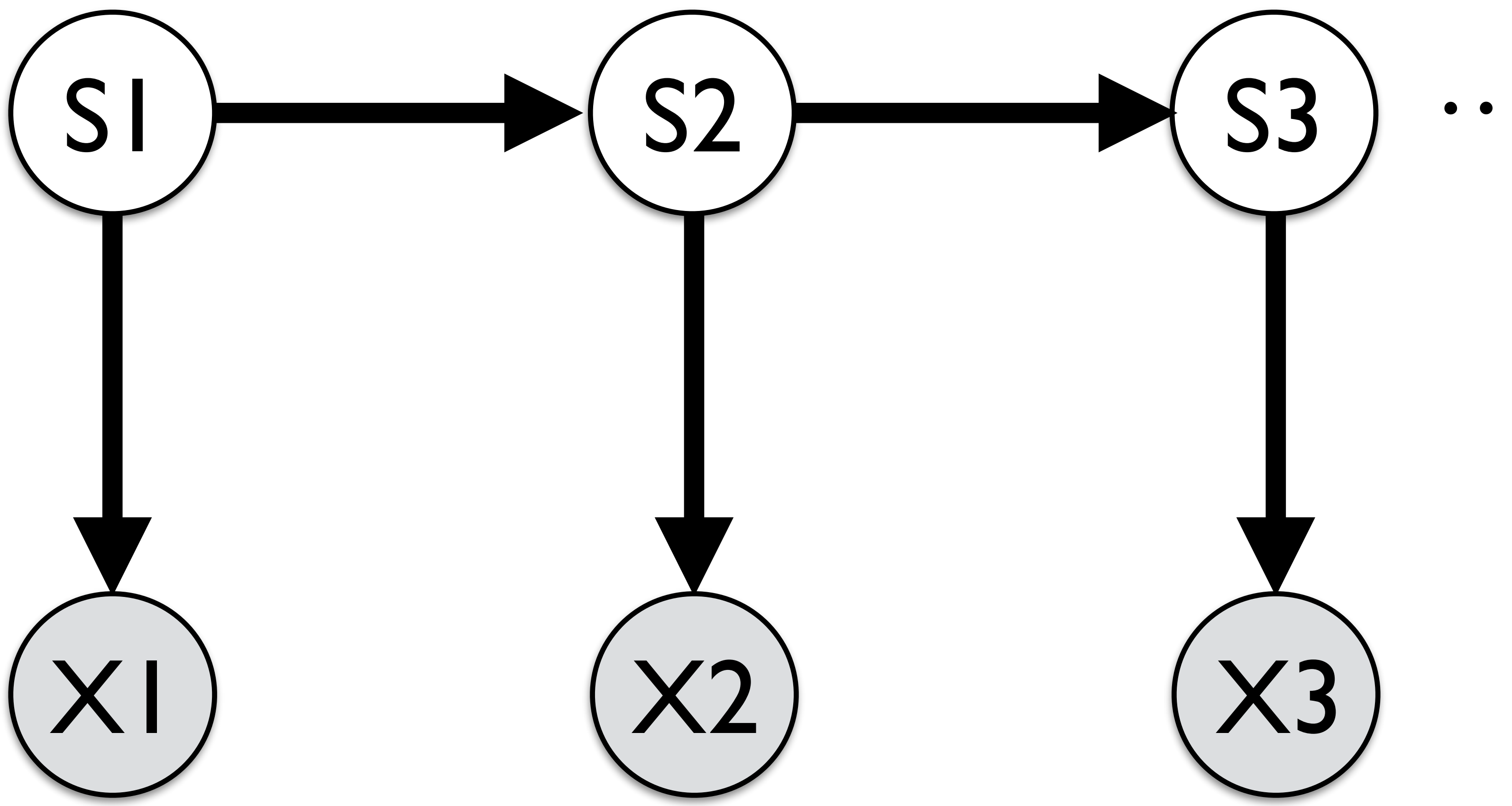
- Underlying “state” influencing what we observe; partial observability makes what we observe difficult to interpret
- Imagine we can never see that a kitten is present; but it clearly helps to explain the data



Hidden variable models

- Probabilistic PCA and factor analysis
  - common in psychology
- Mixture models
- Hidden Markov Models
  - commonly used for NLP and modeling dynamical systems

Hidden Markov Model



The observations are x1, x2, x3
Temporally related
Dynamics driven by hidden state

Gaussian mixture model

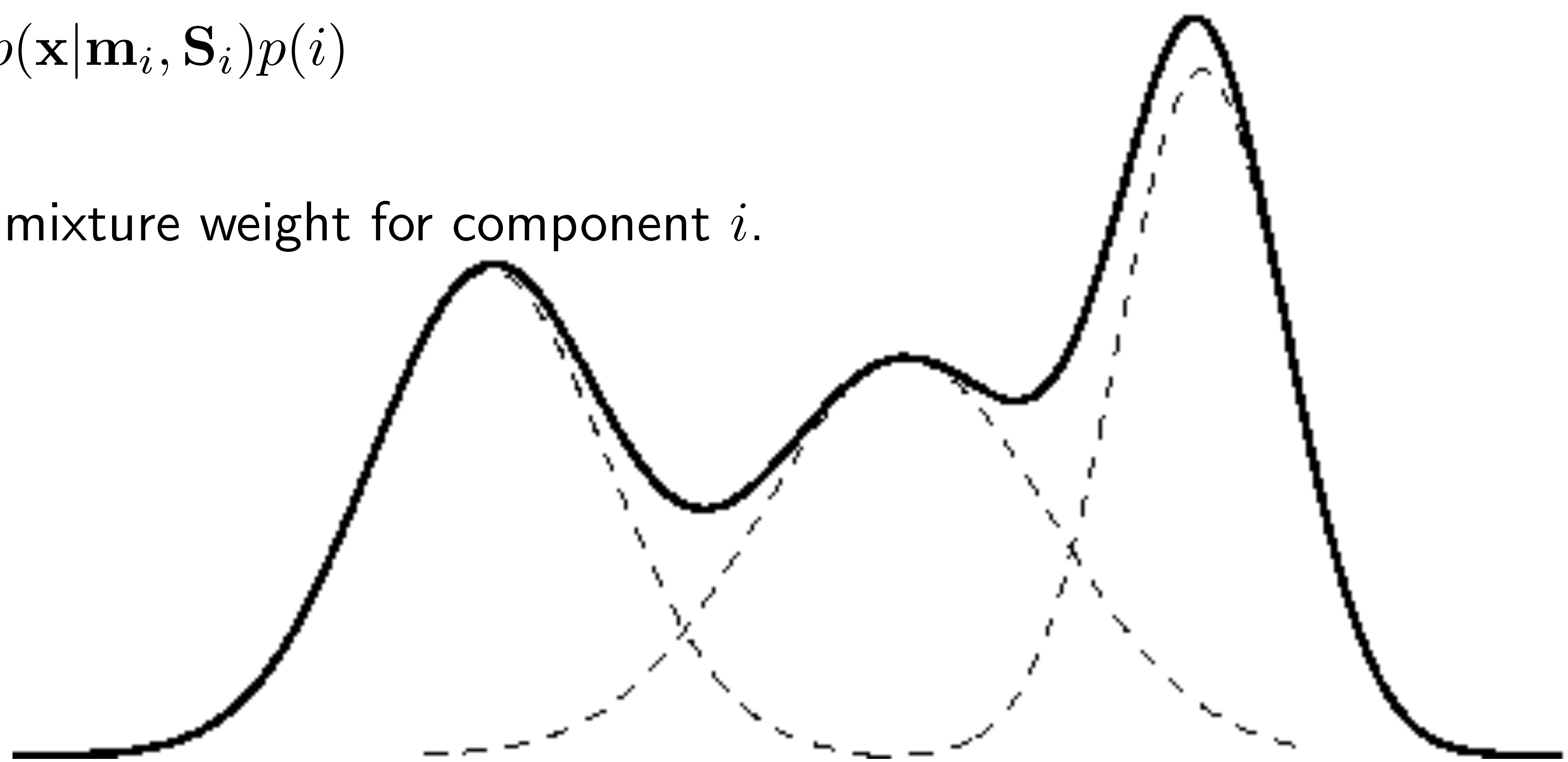
A D dimensional Gaussian distribution for a continuous variable $\mathbf{x}$ is

$$p(\mathbf{x}|\mathbf{m},\mathbf{S}) = \frac{1}{\sqrt{\det\left(2\pi\mathbf{S}\right)}}\exp\left\{-\frac{1}{2}\left(\mathbf{x}-\mathbf{m}\right)^{\mathsf{T}}\mathbf{S}^{-1}\left(\mathbf{x}-\mathbf{m}\right)\right\}$$

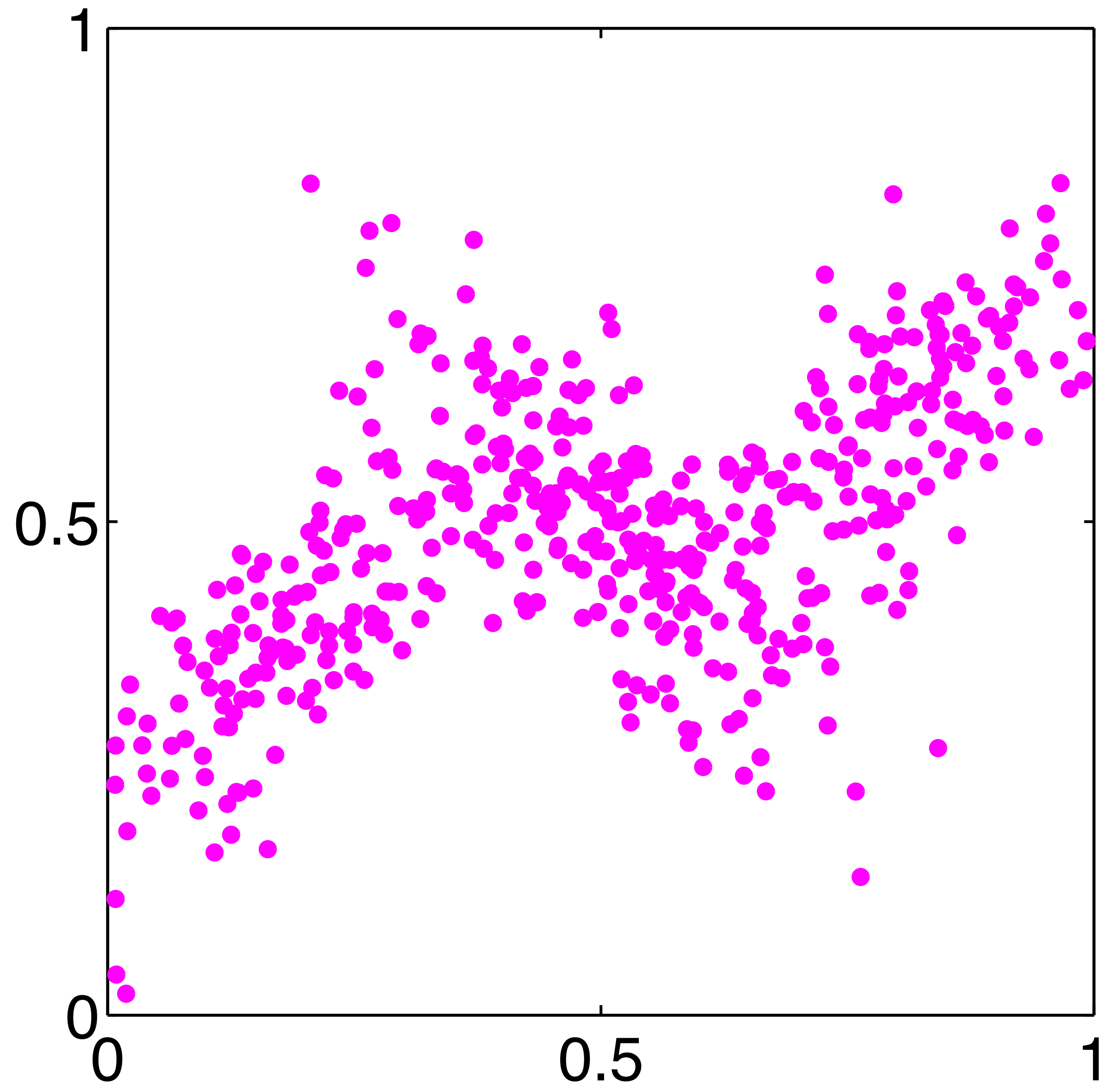
where $\mathbf{m}$ is the mean and $\mathbf{S}$ is the covariance matrix. A mixture of Gaussians is then

$$p\left(\mathbf{x}\right) = \sum_{i=1}^{H} p(\mathbf{x}|\mathbf{m}_i,\mathbf{S}_i)p(i)$$

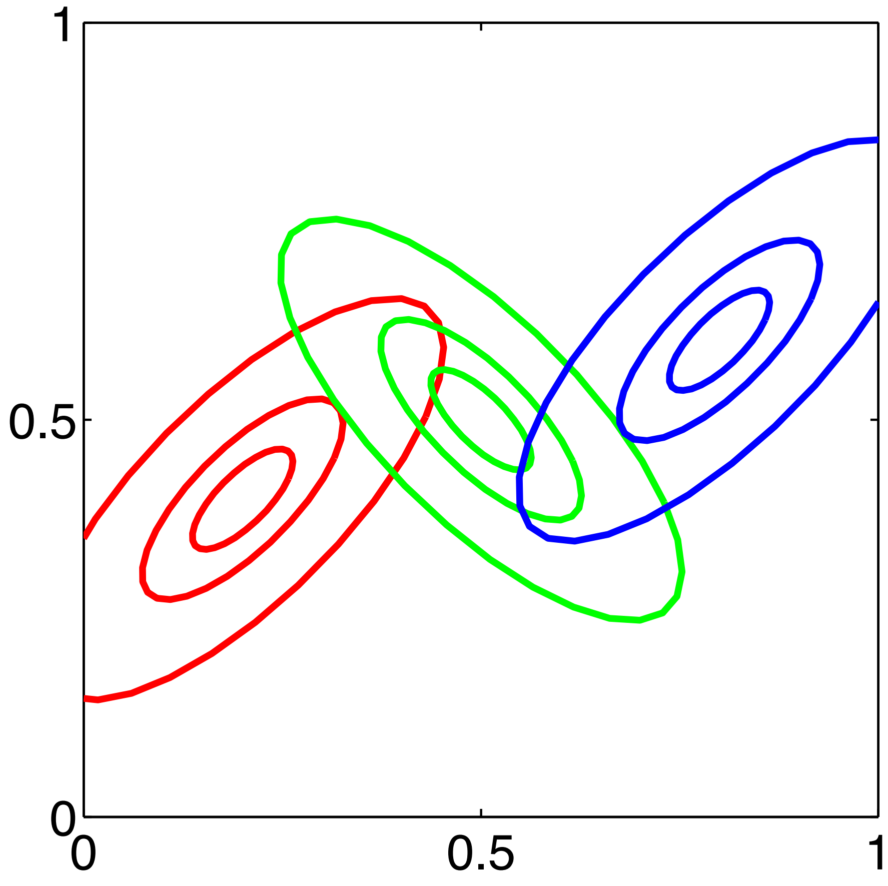
where $p(i)$ is the mixture weight for component i.



Example of 2-d data

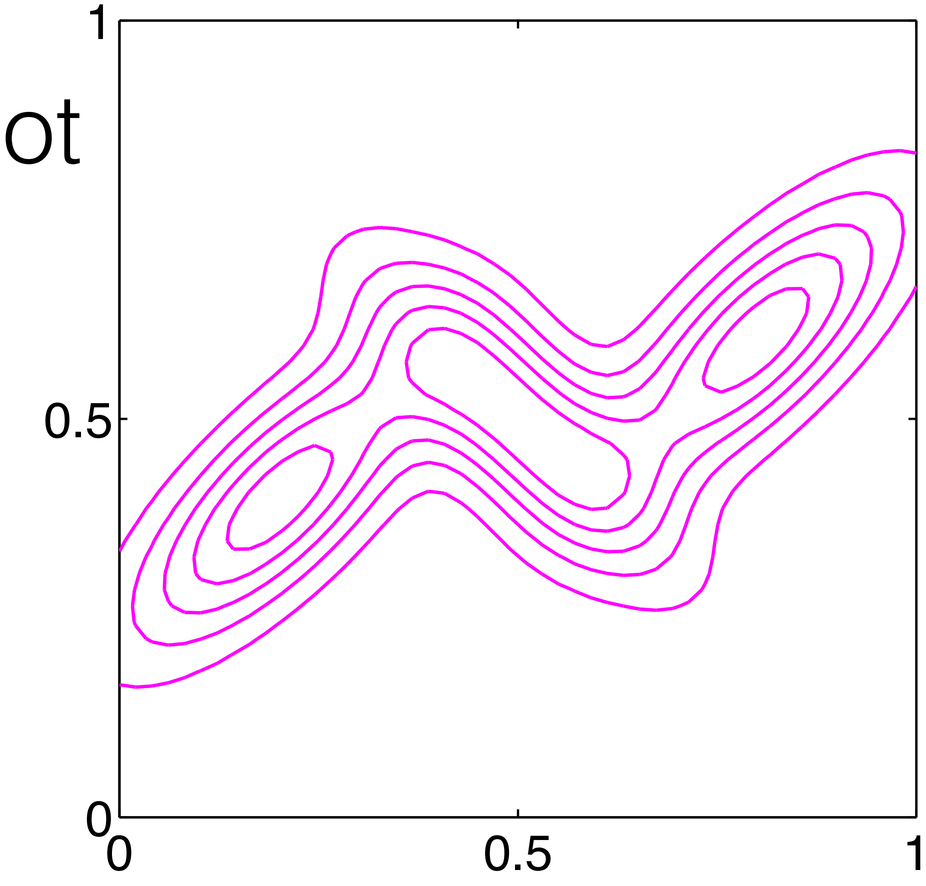


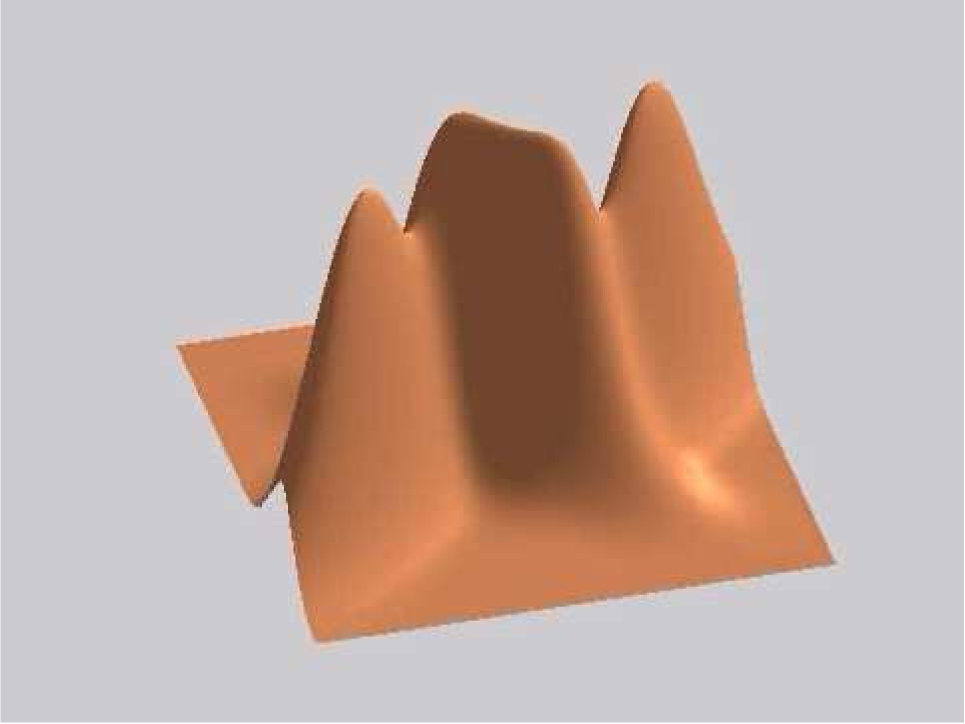
Mixture of 3 Gaussians



3 contours

Contour plot





Surface plot

Probabilistic PCA

- In PCA, we learned p(x l D, h)
  - What were the assumptions on p(x l D, h)?

- For Probabilistic PCA, we learn p(x l D)

- Given some prior p(h), we have

$$p(\mathbf{x}|\mathbf{D}) = \int_{\mathcal{H}} p(\mathbf{x}|\mathbf{D}, \mathbf{h}) p(\mathbf{h}) d\mathbf{h}$$

Hidden variables complicate the solution

- When take log-likelihood, have log of a sum
  - But logs of products is where we get wins
- For a mixture model, log sum_{h=1}^H p(x l h) p(h)
- p(xlh) is Gaussian, but log does not come inside the sum
- Have a complicated objective; could use gradient descent, but more complicated to compute the gradient

Closed-form solutions

- For some hidden variable models, have a closed form solution
  - probabilistic PCA and factor analysis
- For others, no closed form solution, still want to maximize likelihood of the data
  - e.g., mixture models

$$p(\mathbf{x}|\mathbf{m},\mathbf{S}) = \frac{1}{\sqrt{\det(2\pi\mathbf{S})}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mathbf{m})^{\mathsf{T}}\mathbf{S}^{-1}(\mathbf{x}-\mathbf{m})\right\}$$

$$p(\mathbf{x}) = \sum_{i=1}^{H} p(\mathbf{x}|\mathbf{m}_i,\mathbf{S}_i)p(i) \qquad \log p(\mathbf{x})?$$

Closed-form solutions

- For some hidden variable models, have a closed form solution
  - probabilistic PCA
  - factor analysis

- Probabilistic PCA solution:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$$

$$\mathbf{D} = \mathbf{U}_k(\boldsymbol{\Sigma}_k^2 - \sigma^2\mathbf{I}_k)^{1/2}$$

$$\sigma^2 = \frac{1}{d-k}\sum_{i=k+1}^{d}\sigma_i^2$$

Expectation-maximization

- We can use an expectation-maximization approach instead to incrementally compute the solution

- Similar to alternating descent approach taken for factorizations

- Enables logarithm and sum over hidden variables to be swapped, by minimizing instead a lower bound

$$\log p(\theta|\mathbf{x}) \geq \mathbb{E}[\log p(\mathbf{x}, \mathbf{h}|\theta)$$

$$\log p(\mathbf{x}, \mathbf{h}|\theta) = \log p(\mathbf{x}|\mathbf{h}, \theta) + \log p(\mathbf{h}|\theta)$$

If expectation w.r.t. $p(\mathbf{h}|\mathbf{x}, \theta)$ then equal, rather than $\geq$

Expectation-maximization

1. Approximate $p(\mathbf{h}|\mathbf{x},\theta)$

2. Optimize theta for

$$\log p(\mathbf{x},\mathbf{h}|\theta) = \log p(\mathbf{x}|\mathbf{h},\theta) + \log p(\mathbf{h}|\theta)$$

e.g. mixture model

$$p(\mathbf{x}|\mathbf{m},\mathbf{S}) = \frac{1}{\sqrt{\det\left(2\pi\mathbf{S}\right)}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\mathbf{m}\right)^{\mathsf{T}}\mathbf{S}^{-1}\left(\mathbf{x}-\mathbf{m}\right)\right\}$$

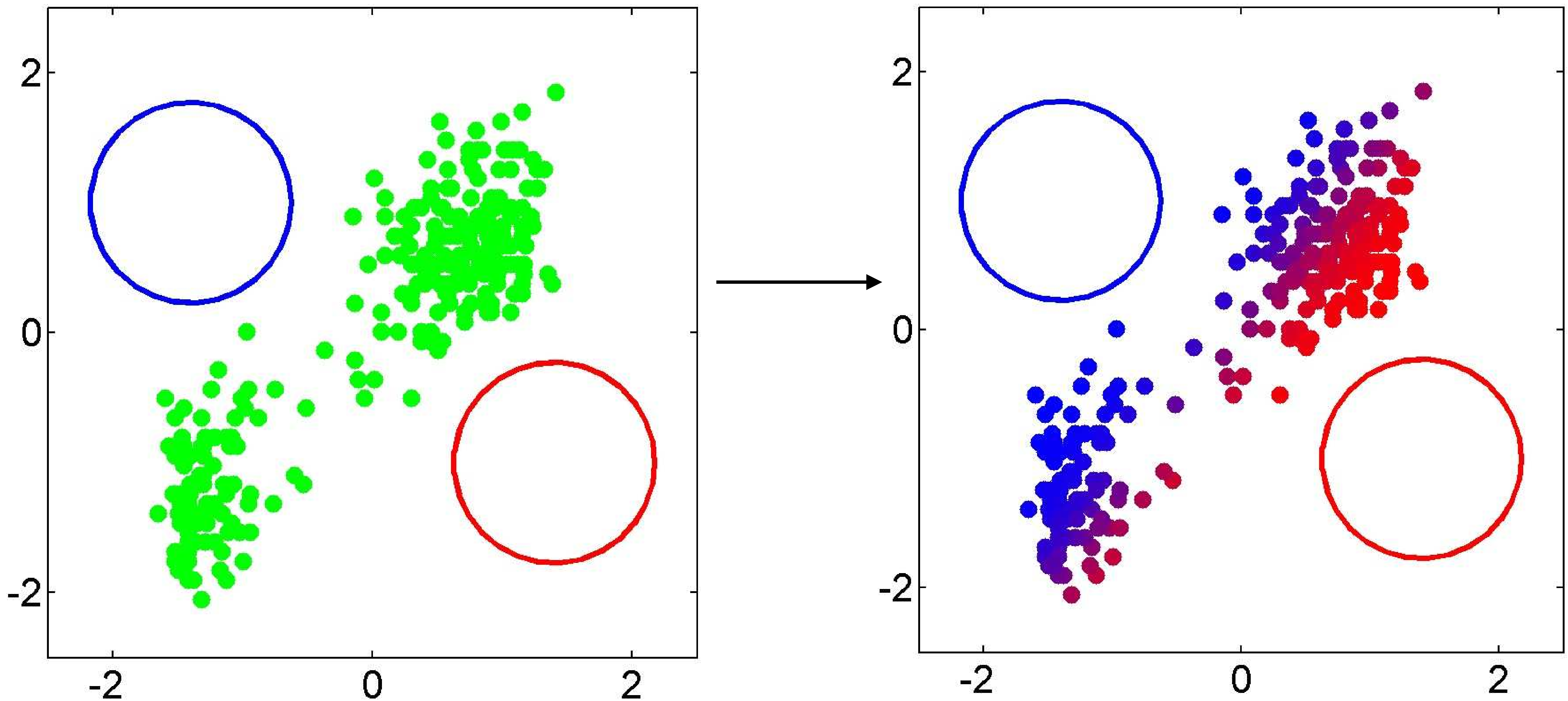
$$p\left(\mathbf{x}\right) = \sum_{i=1}^{H} p(\mathbf{x}|\mathbf{m}_i,\mathbf{S}_i)p(i)$$

$$\log p(\mathbf{x}|h=i) \text{ simple}$$

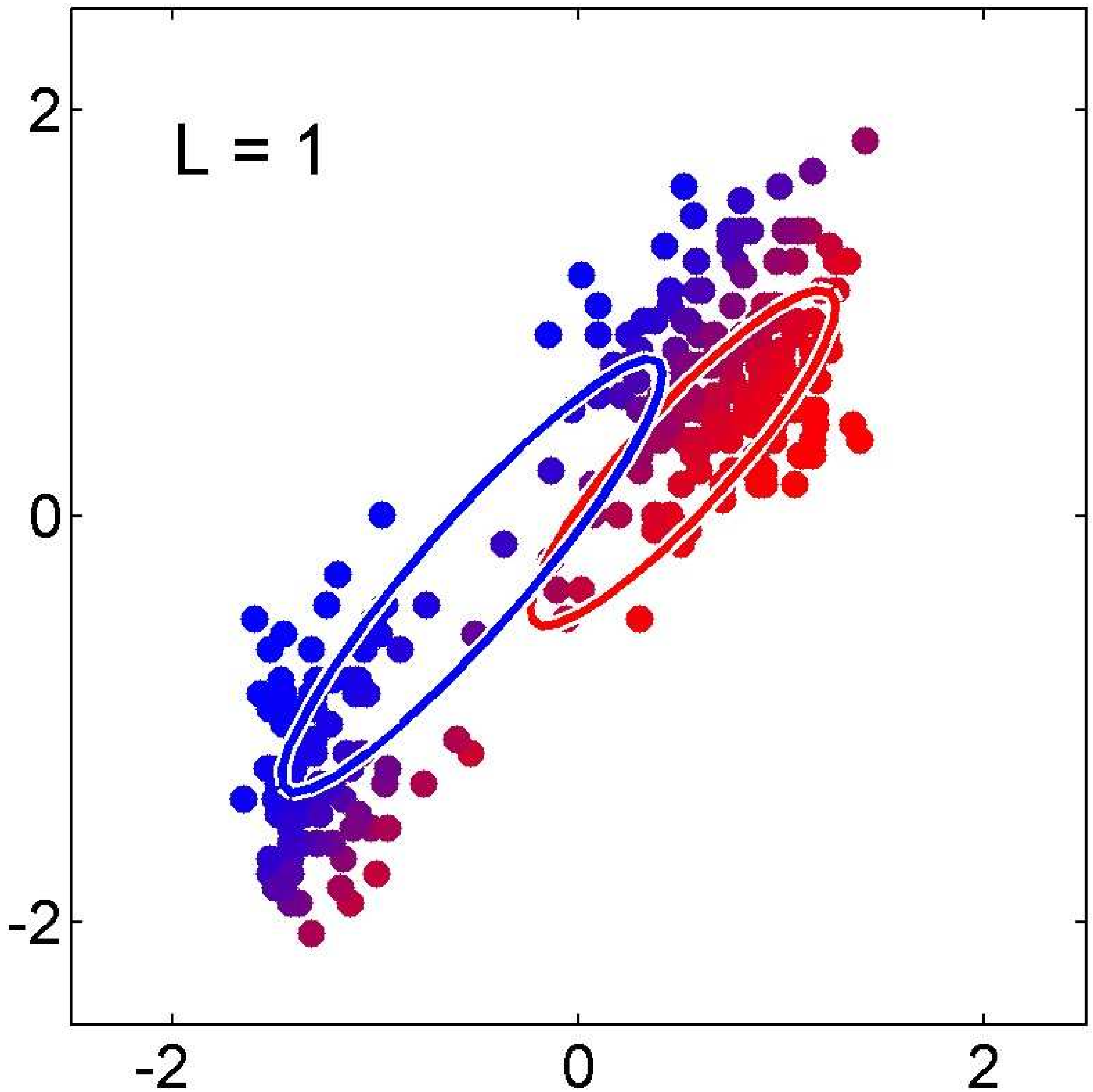
EM algorithm for mixtures

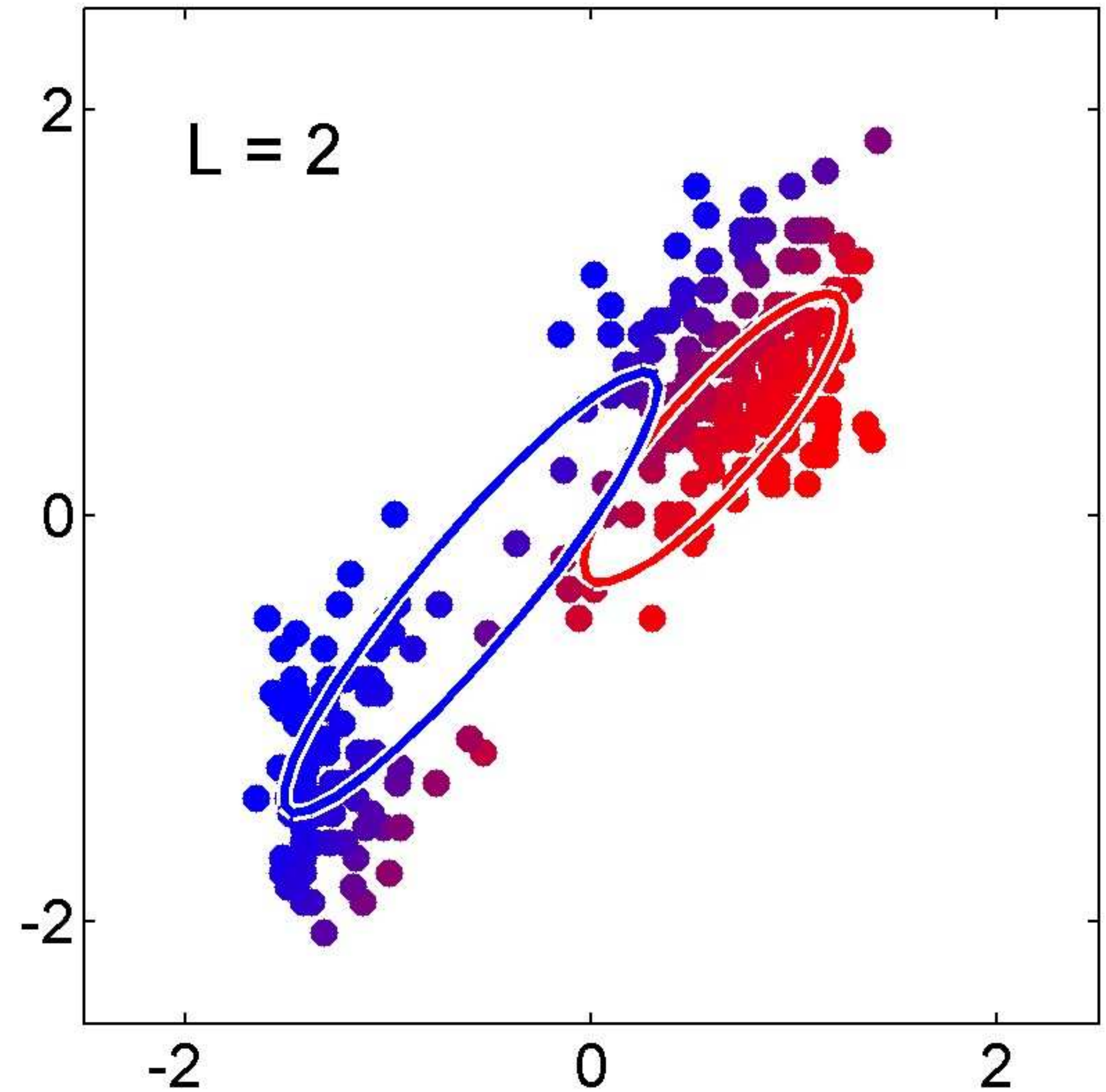
- Procedure: initialize parameters to some initial guess/random
- Alternate between:
  - E-step: fix parameter, approximate p(h l x, theta)
  - M-step: fix p(h l x, theta) obtaining maximum likelihood parameters for means, covariances and weights on each distribution
- Each cycle guaranteed not to decrease likelihood, converge to a local minimum

Simulation of EM for mixtures



Simulation of EM for mixtures





Simulation of EM for mixtures