

---

# Probabilistic Modelling in Machine Learning

**Peter Tiño**

School of Computer Science  
University of Birmingham, UK

# Probabilistic modelling and ML...

---

many different flavors

start with data  $\mathcal{D}$

think - Q1: How could the data have been generated?  
source (model)

think - Q2: What interesting aspects in the data you'd like to capture?  
This will also inform the model structure!

# The model...

---

Model (parametrized) -  $p(\mathcal{D}|\mathbf{w})$

Or you may want to be smart and formulate the model in a fully Bayesian framework - "parameter free" ...

... but at some point there will be *some* parameters (of prior), or you may rely solely on hierarchical Bayes...

# Generative probabilistic model - advantages?

---

- principled formulation
- transparent and interpretable model structure
- principled model selection
- consistent coping with missing data
- consistent building of hierarchies
- ...

# Model structure and model fitting

---

Probabilistic modelling involves **two main steps/tasks**:

1. Design the **model structure** by considering Q1 and Q2.

2. **Fit your model to the data.**

– Sometimes the two tasks are interleaved -  
e.g. when model fitting involves both parameters and model structure (e.g. infinite mixtures...)

# Model structure and model fitting

---

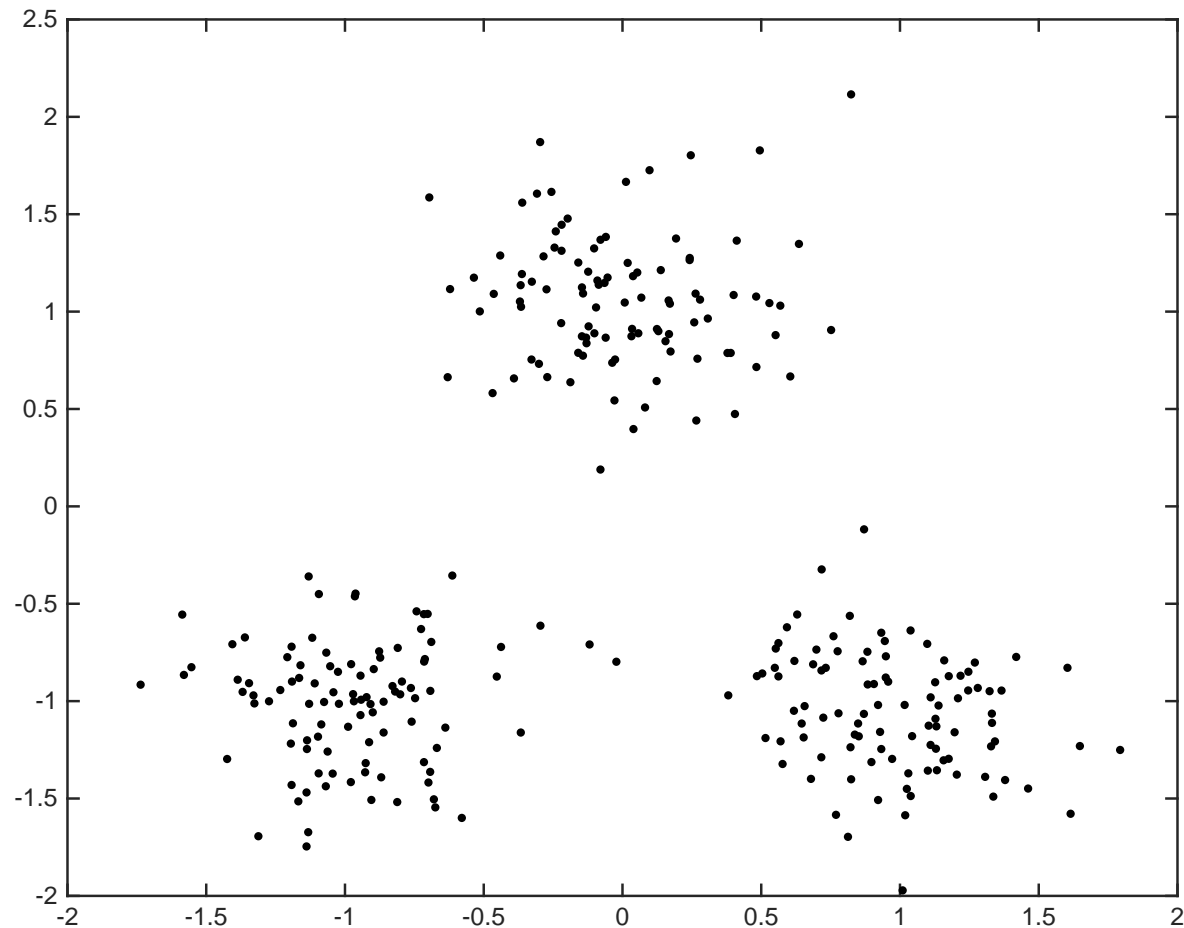
Model fitting: ML, MAP, Type II ML, full Bayesian treatment...

We will put more emphasis on task 1.

Experience the way probabilistic modelers in ML think -  
examples of different data structures and different "questions on the data".

# Start with the data... (simple example)

---



Think: How could have this data been generated?

# Mixture models

---

## Mixture of Gaussians

Given a set of data points  $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$ , coming from what looks like a mixture of 3 Gaussians  $G_1, G_2, G_3$ , fit a 3-component Gaussian mixture model to  $\mathcal{D}$ .

$$p(\mathbf{x}) = \sum_{j=1}^3 P(j, \mathbf{x}) = \sum_{j=1}^3 P(j) \cdot p(\mathbf{x}|j),$$

- $P(j)$  – mixing coefficient (prior probability) of Gaussian  $G_j$
- $p(\mathbf{x}|j)$  – ‘probability mass’ given to the data item  $\mathbf{x}$  by Gaussian  $G_j$ .

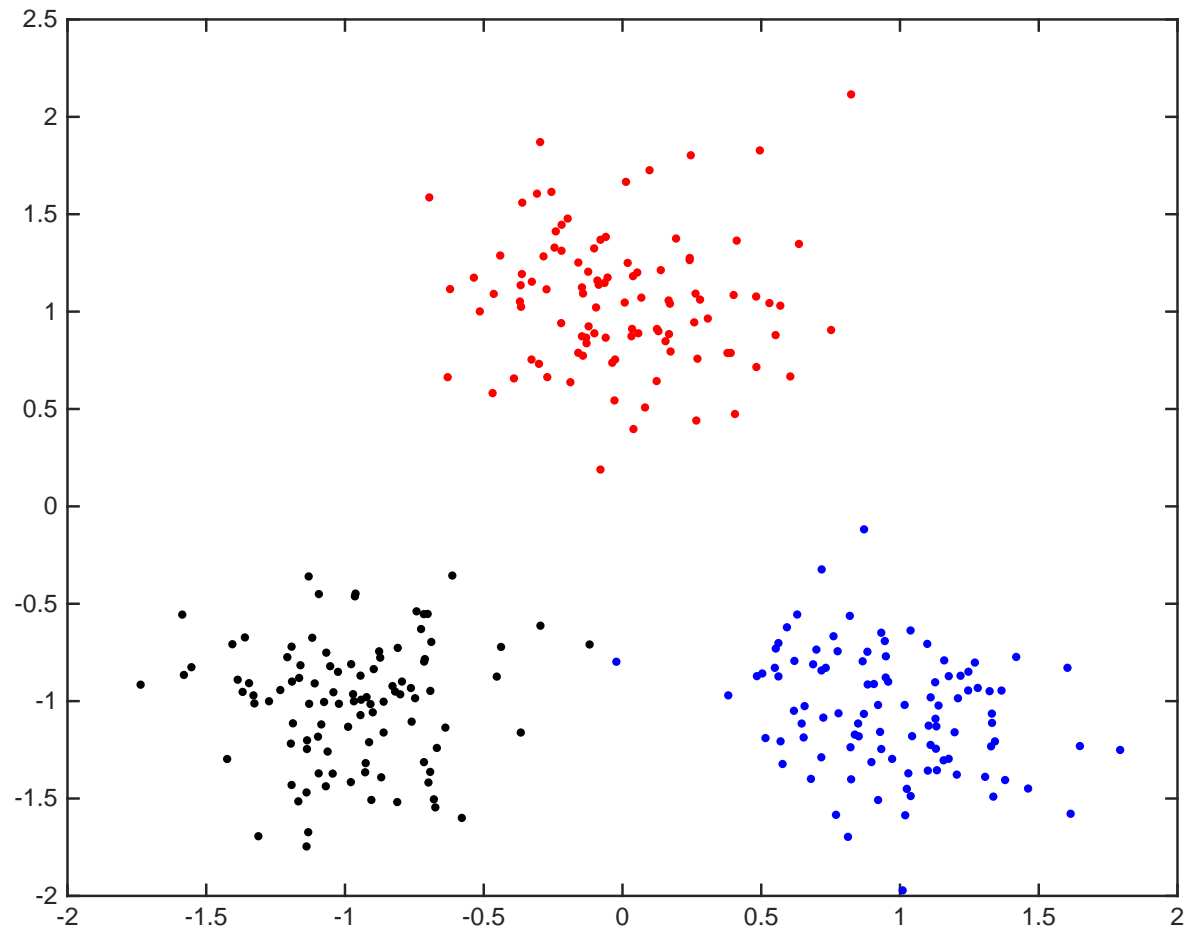
Generative process: Repeat 2 steps

1. generate  $j$  from  $P(j)$
2. generate  $\mathbf{x}$  from  $p(\mathbf{x}|j)$



# Fitting the model is easy!

---



It is fairly "obvious" which point comes from which Gaussian!

# Three Gaussians ...

---

Given that we know which data point comes from which Gaussian, things are easy:

- Collect data points from  $\mathcal{D}$  that are known to be generated from  $G_1$  in  $\mathcal{D}_1$ .
- Estimate free parameters (mean, covariance matrix) of  $G_1$  (e.g. ML):

$$\hat{\mu}_1 = \frac{1}{|\mathcal{D}_1|} \sum_{\mathbf{x} \in \mathcal{D}_1} \mathbf{x},$$

$$\hat{\Sigma}_1 = \frac{1}{|\mathcal{D}_1|} \sum_{\mathbf{x} \in \mathcal{D}_1} (\mathbf{x} - \hat{\mu}_1)(\mathbf{x} - \hat{\mu}_1)^T.$$

Do the same for Gaussians  $G_2$  and  $G_3$ .

- $P(j)$  - proportions of sizes  $|\mathcal{D}_j|$ .

# Formulation through indicator variables

We can represent the knowledge about which data point came from which Gaussian through indicator variables  $z_j^i$ ,  $i = 1, 2, \dots, N$  (data points),  $j = 1, 2, 3$  (mixture components - Gaussians):

$z_j^i = 1$ , iff  $\mathbf{x}^i$  was generated by  $G_j$ ;

$z_j^i = 0$ , otherwise.

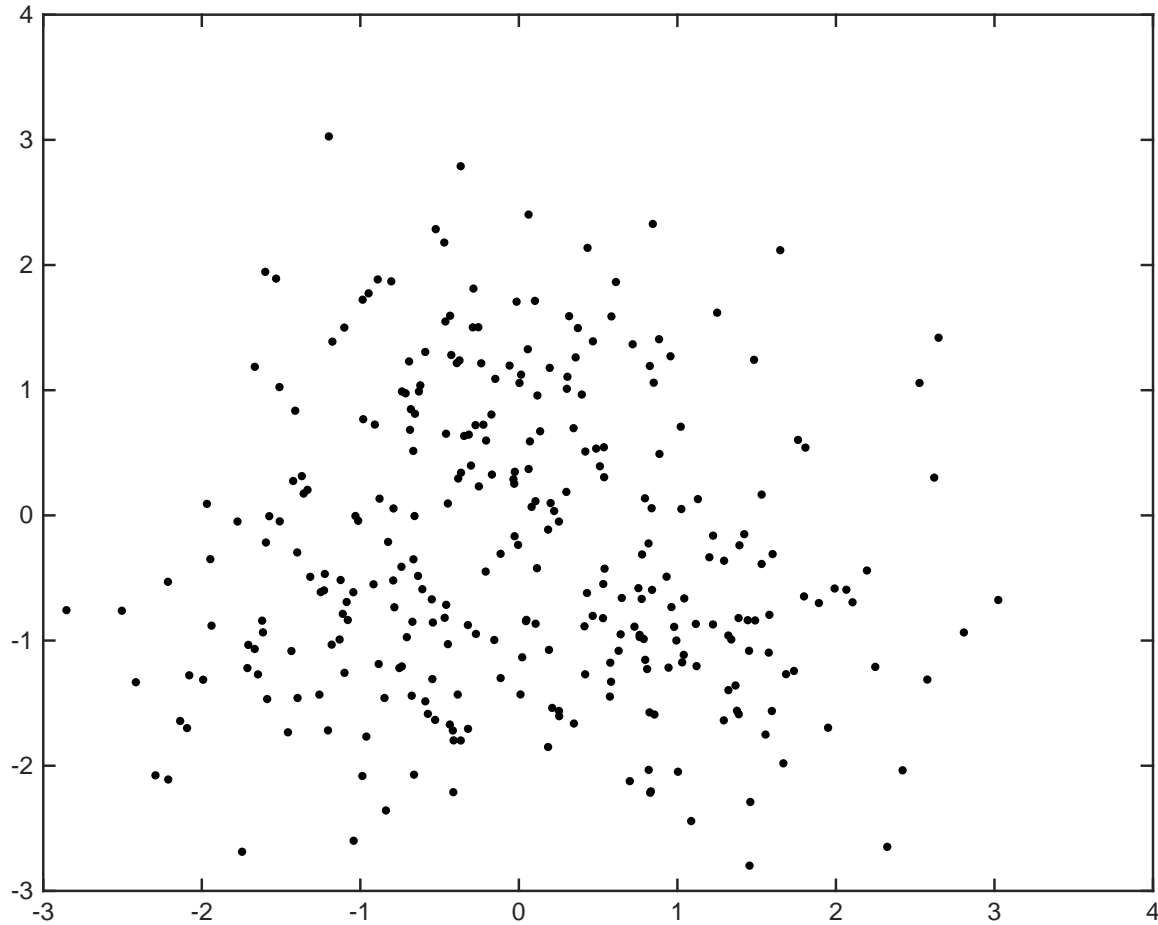
Then,

$$\hat{\mu}_j = \frac{1}{\sum_{i=1}^N z_j^i} \sum_{i=1}^N z_j^i \cdot \mathbf{x}^i,$$

$$\hat{\Sigma}_j = \frac{1}{\sum_{i=1}^N z_j^i} \sum_{i=1}^N z_j^i \cdot (\mathbf{x}^i - \hat{\mu}_j)(\mathbf{x}^i - \hat{\mu}_j)^T.$$

# Now that you are an expert, model this...

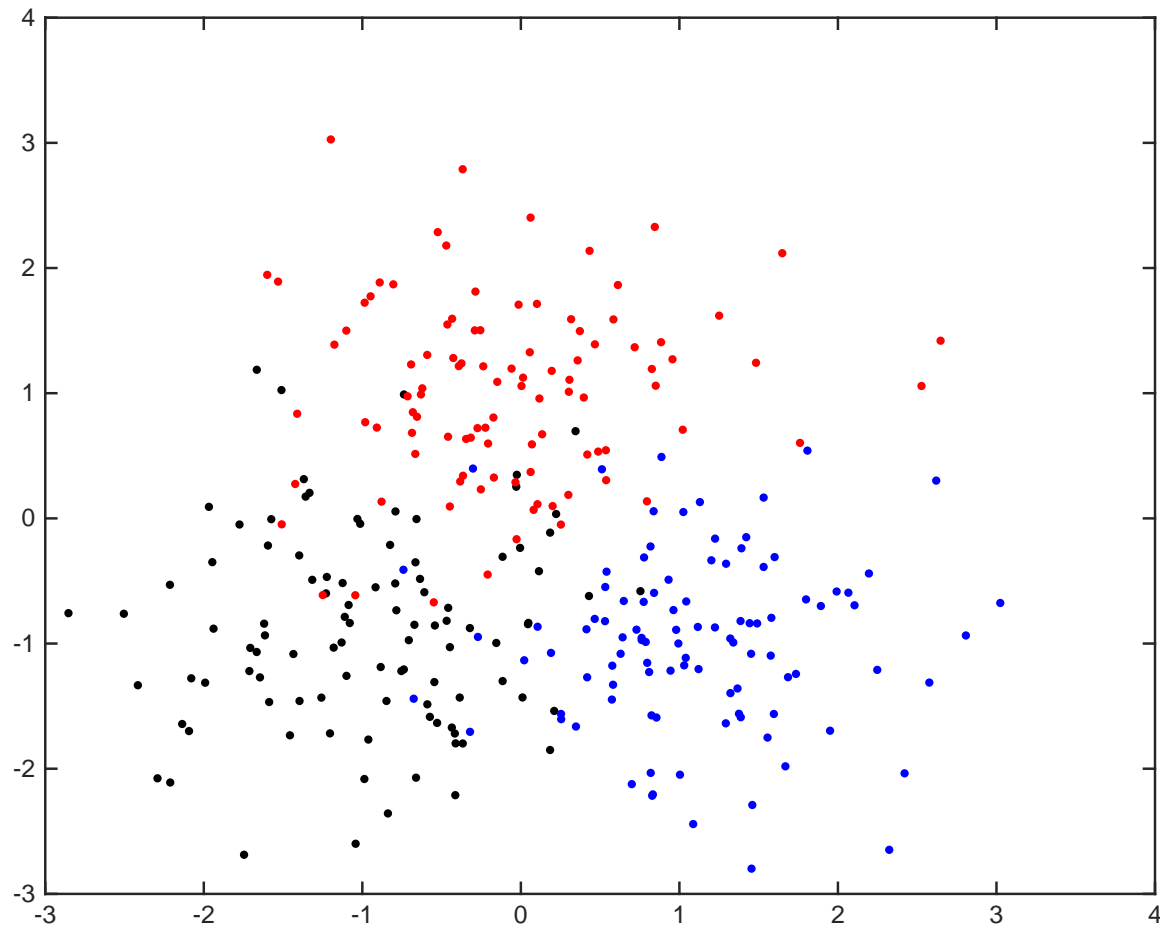
---



ooooops!

# The ground truth

---



But how can I know that? – **You cannot!**

# Don't know values of the indicator variables...

---

The assignments of points to Gaussians  $z_j^i$  are **hidden/latent/non-observable variables!**

Collect all the indicator variables in a **compound** (vector) **latent variable**  
 $Z = \{z_j^i\} \in \{0, 1\}^{N \cdot 3}$ .

Each setting of  $Z$  represents **one particular situation of assigning points  $\mathbf{x}^i$  to Gaussians  $G_1, G_2$  and  $G_3$ .**

This can be anything from trivial settings, such as all points come from  $G_1$ , to very mixed situations.

# Probabilistic way of dealing with uncertain $Z$

---

Assume we already have some guess about where the Gaussians should be ( $\mu_j$ ) and what their shape is ( $\Sigma_j$ ).

Since we don't know  $Z$ , we simply **need to consider all possibilities** (cases) **for assignments  $Z$** .

Obviously, looking at the data, **not all assignments  $Z$**  will be **equally likely**.

This is expressed through **posterior  $P(Z|\mathcal{D}, G_1, G_2, G_3)$**  that evaluates how likely, **given the data and current positions/shapes of Gaussians  $G_1, G_2, G_3$** , is the particular assignment scheme  $Z$ .

# Guessing $Z$ , given the current model and data

---

$z_j^i \in \{0, 1\}$  is unobserved, so calculate its mean value instead -  
"how much  $z_j^i$  wants to be set to 1"

$$\begin{aligned} E_{P(Z|\mathcal{D}, G_1, G_2, G_3)}[z_j^i] &= 0 \cdot P(z_j^i = 0|\mathcal{D}, G_1, G_2, G_3) \\ &\quad + 1 \cdot P(z_j^i = 1|\mathcal{D}, G_1, G_2, G_3) \\ &= P(z_j^i = 1|\mathcal{D}, G_1, G_2, G_3) \\ &= R_j^i \end{aligned}$$

$R_j^i$  is the 'responsibility' of Gaussian  $j$  for the data point  $\mathbf{x}^i$ .

Substitute  $z_j^i$  in the crisp case calculations with 'softer' probabilistic  $R_j^i$ .



# Dealing with uncertainty in $Z$

---

Instead of

$$\hat{\mu}_j = \frac{1}{\sum_{i=1}^N z_j^i} \sum_{i=1}^N z_j^i \cdot \mathbf{x}^i$$

we have

$$\hat{\mu}_j = \frac{1}{\sum_{i=1}^N R_j^i} \sum_{i=1}^N R_j^i \cdot \mathbf{x}^i.$$

Instead of

$$\hat{\Sigma}_j = \frac{1}{\sum_{i=1}^N z_j^i} \sum_{i=1}^N z_j^i \cdot (\mathbf{x}^i - \hat{\mu}_j)(\mathbf{x}^i - \hat{\mu}_j)^T$$

we have

$$\hat{\Sigma}_j = \frac{1}{\sum_{i=1}^N R_j^i} \sum_{i=1}^N R_j^i \cdot (\mathbf{x}^i - \hat{\mu}_j)(\mathbf{x}^i - \hat{\mu}_j)^T.$$

# What next?

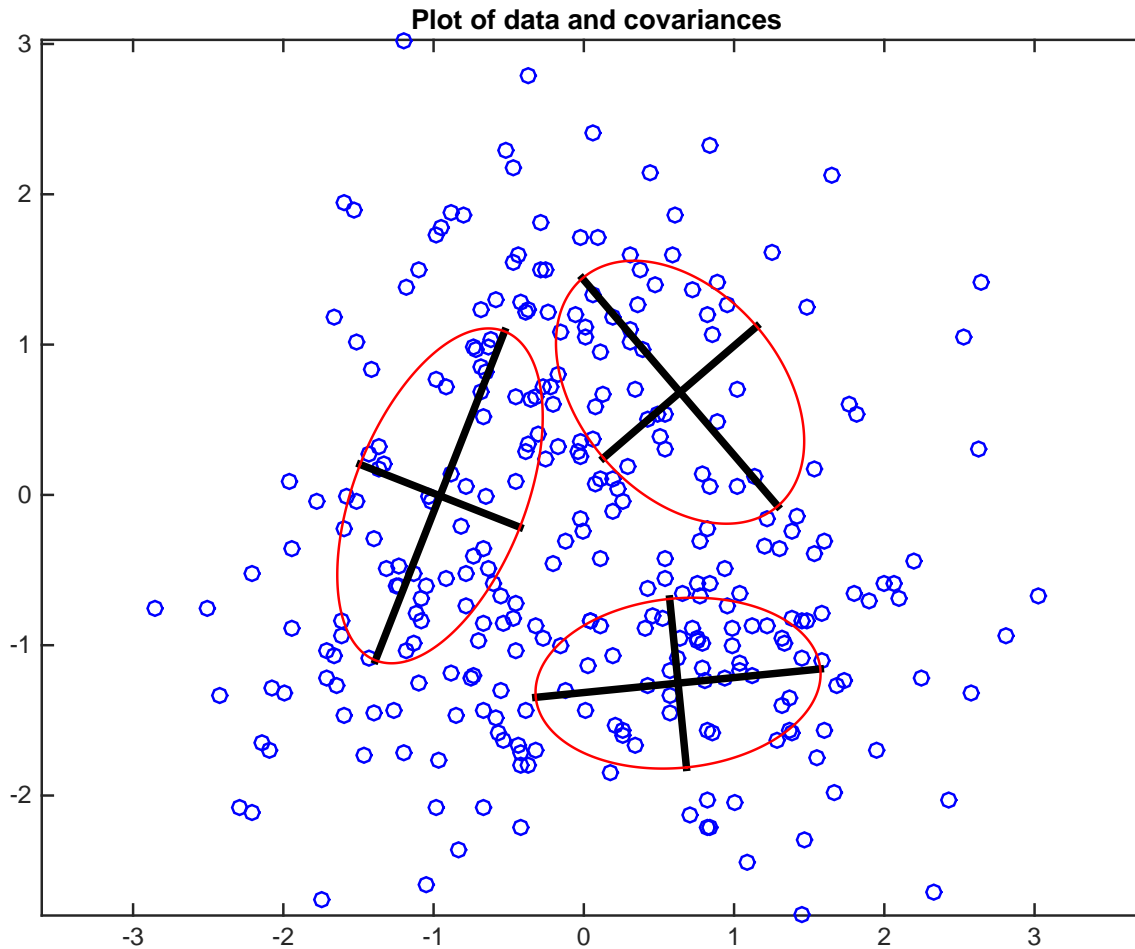
---

Having refined our model (positions and shapes of the Gaussians), **refine our ideas about possible assignments  $Z$  of points  $\mathbf{x}^i$  to Gaussians  $G_1, G_2, G_3$** . Of course, we still cannot be certain about exact values of  $Z$  - so **still a probabilistic formulation!**

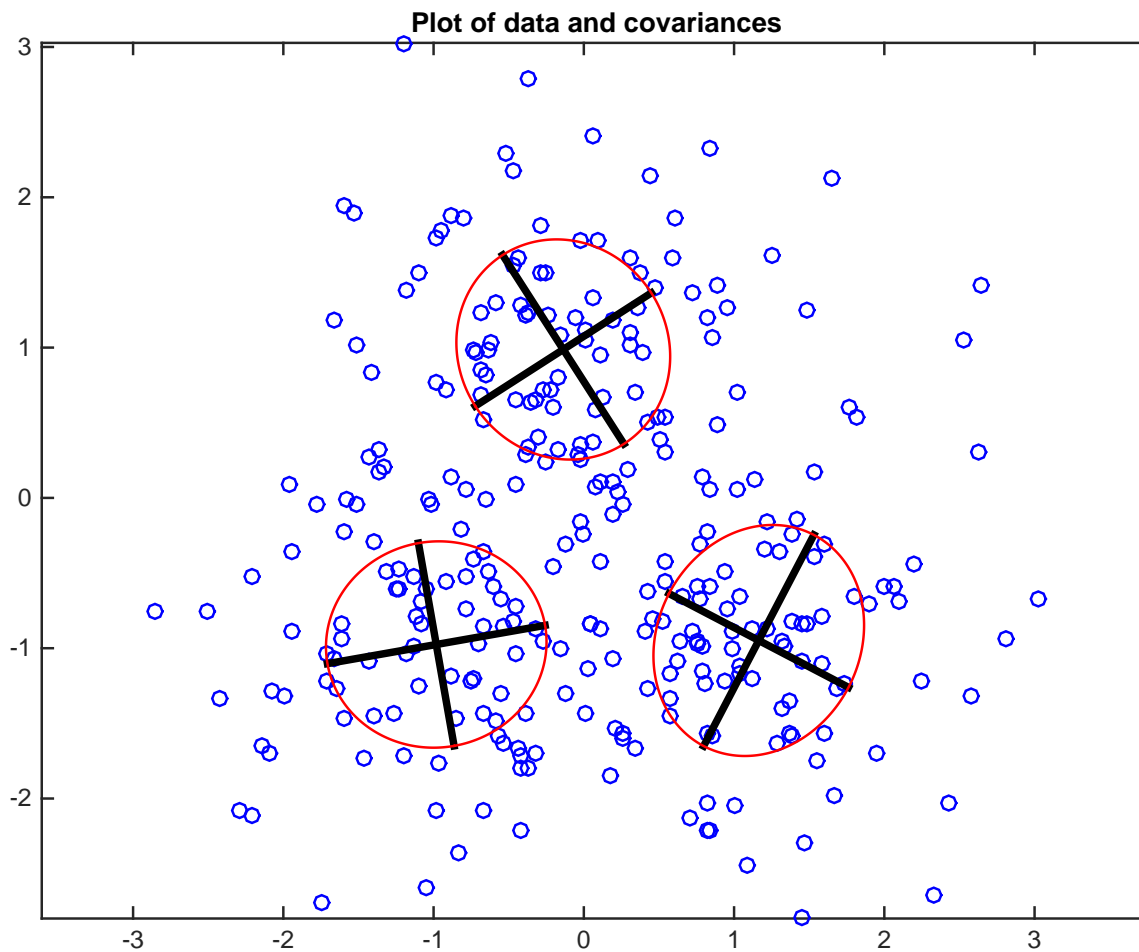
$$R_j^i = P(G_j | \mathbf{x}^i) = \frac{P(\mathbf{x}^i | G_j) \cdot P(j)}{\sum_{q=1}^3 P(\mathbf{x}^i | G_q) \cdot P(q)}$$

**Repeat the parameter estimation and assignment refinement steps until ‘convergence’.**

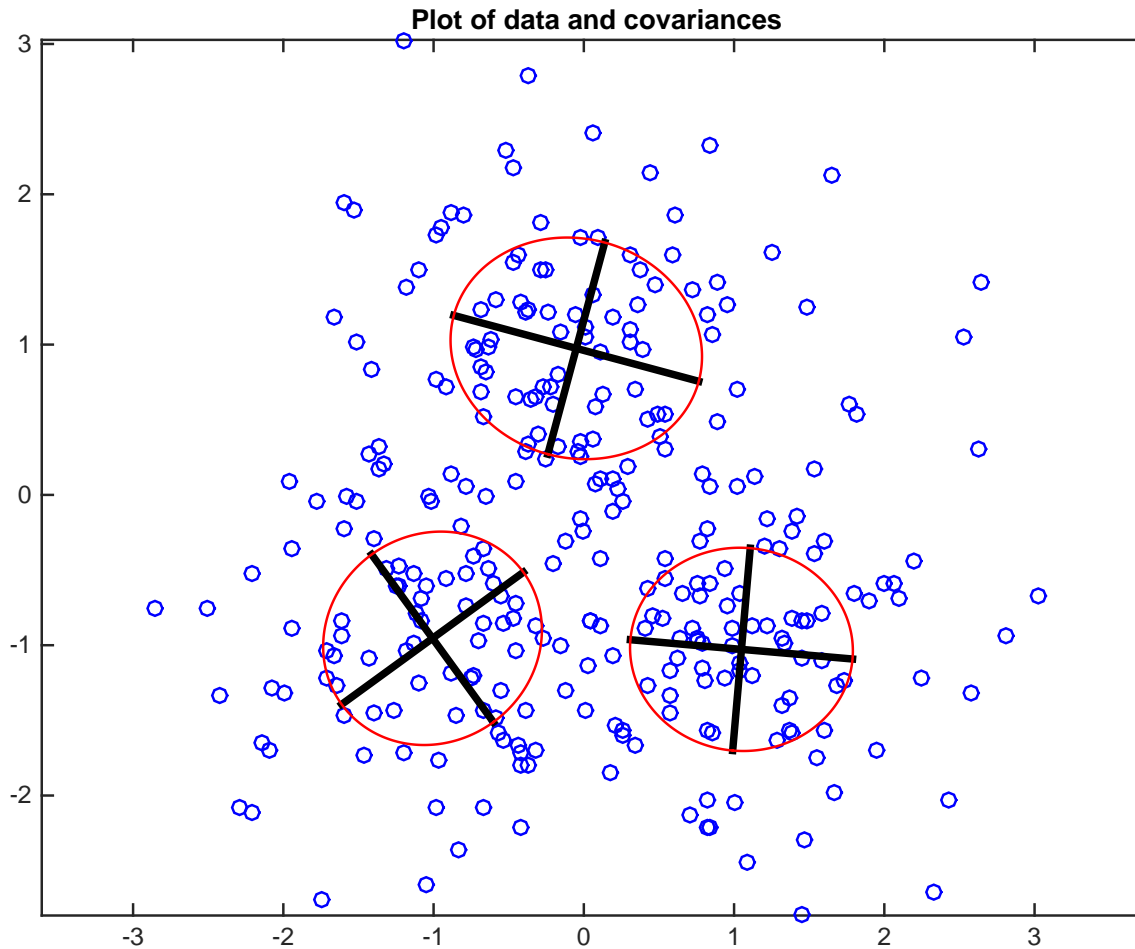
# Apply to our data - iteration 1



# Apply to our data - iteration 2



# Apply to our data - iteration 30

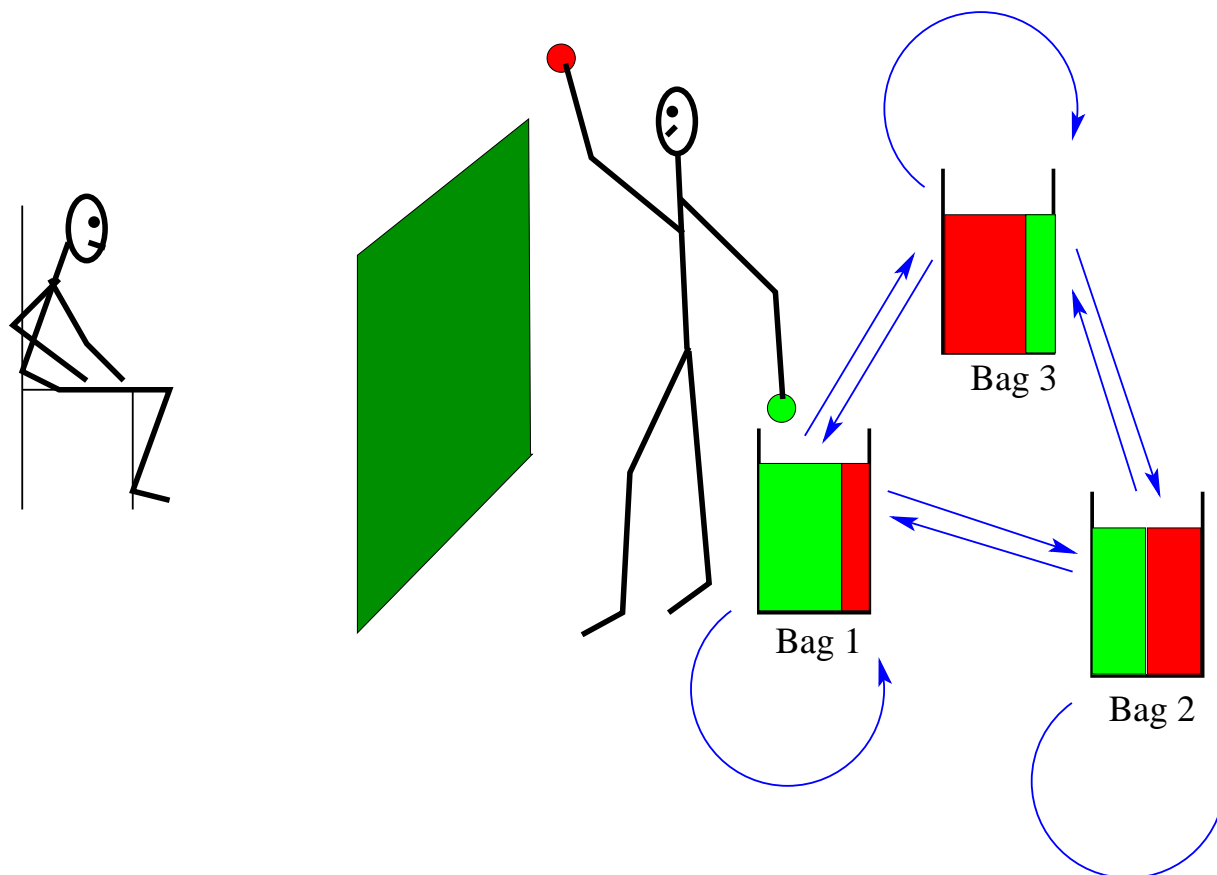


# Another latent variable model

## Hidden Markov Model

Stationary emissions conditional on hidden (unobservable) states.

Hidden states represent basic operating "regimes" of the process.



# Temporal structure - Hidden Markov Model

---

We have  $M$  bags of balls of different colors (Red - R, Green - G).

We are standing behind a curtain and at each point in time we select a bag  $j$ , draw (with replacement) a ball from it and show the ball to an observer. Color of the ball shown at time  $t$  is  $C_t \in \{R, G\}$ . We do this for  $T$  time steps.

The observer can only see the balls, it has no access to the information about how we select the bags.

Assume: we select bag at time  $t$  based only on our selection at the previous time step  $t - 1$  (1st-order Markov assumption).

# If only we knew ...

---

If we knew which bags were used at which time steps, things would be very easy! ... just counting

Hidden variables  $z_t^j$ :

$z_t^j = 1$ , iff bag  $j$  was used at time  $t$ ;

$z_t^j = 0$ , otherwise.

$$P(\text{bag}_j \rightarrow \text{bag}_k) = \frac{\sum_{t=1}^{T-1} z_t^j \cdot z_{t+1}^k}{\sum_{q=1}^M \sum_{t=1}^{T-1} z_t^j \cdot z_{t+1}^q} \quad [\text{state transitions}]$$

$$P(\text{color} = c \mid \text{bag}_j) = \frac{\sum_{t=1}^T z_t^j \cdot \delta(c = C_t)}{\sum_{g \in \{R, G\}} \sum_{t=1}^T z_t^j \cdot \delta(g = C_t)} \quad [\text{emissions}]$$



# But we don't ...

---

We need to estimate probabilities for hidden events such as:

- $z_t^j \cdot z_{t+1}^k = 1$   
at time  $t$  - bag  $j$ , at the next time step - bag  $k$
- $z_t^j \cdot \delta(c = C_t) = 1$   
at time  $t$  - bag  $j$ , ball of color  $c$

Again, the probability estimates need to be based on observed data  $\mathcal{D}$  and our current model of state transition and emission probabilities.

# Estimating values of the hidden variables

---

$$P(z_t^j \cdot z_{t+1}^k = 1 \mid \mathcal{D}, \text{current model}) = R_t^{j \rightarrow k}$$

$$P(z_t^j \cdot \delta(c = C_t) = 1 \mid \mathcal{D}, \text{current model}) = R_t^{j,c}$$

I will not deal with the crucial question of how to compute those posteriors over hidden variables, given the observed data and current model parameters.

This can be done efficiently - [Forward-Backward algorithm](#).

# Re-estimate the model

$$P(\text{bag}_j \rightarrow \text{bag}_k) = \frac{\sum_{t=1}^{T-1} z_t^j \cdot z_{t+1}^k}{\sum_{q=1}^M \sum_{t=1}^{T-1} z_t^j \cdot z_{t+1}^q} \rightarrow$$

$$P(\text{bag}_j \rightarrow \text{bag}_k) = \frac{\sum_{t=1}^{T-1} R_t^{j \rightarrow k}}{\sum_{q=1}^M \sum_{t=1}^{T-1} R_t^{j \rightarrow q}} \quad [\text{state transitions}]$$

$$P(\text{color} = c \mid \text{bag}_j) = \frac{\sum_{t=1}^T z_t^j \cdot \delta(c = C(t))}{\sum_{g \in \{R, G\}} \sum_{t=1}^T z_t^j \cdot \delta(g = C(t))} \rightarrow$$

$$P(\text{color} = c \mid \text{bag}_j) = \frac{\sum_{t=1}^T R_t^{j, c}}{\sum_{g \in \{R, G\}} \sum_{t=1}^T R_t^{j, g}} \quad [\text{emissions}]$$

# Learning - can we do better than hand-waving?

---

Observed data:  $\mathcal{D}$

Parameterized model of data items  $\mathbf{x}$ :  $p(\mathbf{x}|\mathbf{w})$

Log-likelihood of  $\mathbf{w}$ :  $\log p(\mathcal{D}|\mathbf{w})$

Train via **Maximum Likelihood**:

$$\mathbf{w}_{ML} = \underset{\mathbf{w}}{\operatorname{argmax}} \log p(\mathcal{D}|\mathbf{w}).$$

# Complete data

---

Observed data:  $\mathcal{D}$

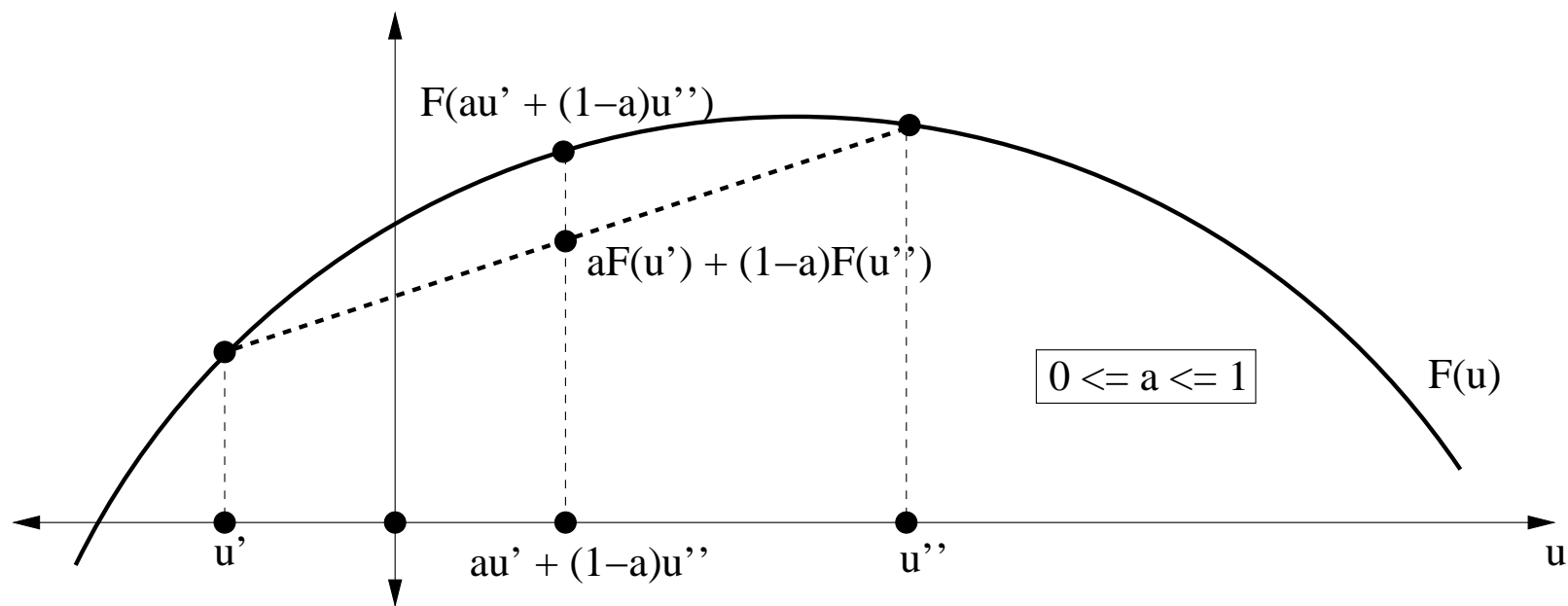
Unobserved data:  $\mathcal{Z}$  (realization of compound hidden variable  $Z$ )

Complete data:  $(\mathcal{D}, \mathcal{Z})$

By marginalization ("integrate out the uncertainty in  $Z$ ):

$$p(\mathcal{D}|\mathbf{w}) = \sum_{\mathcal{Z}} p(\mathcal{D}, \mathcal{Z}|\mathbf{w})$$

# Concave function



For any concave function  $F : \mathcal{R} \rightarrow \mathcal{R}$  and any  $u', u'' \in \mathcal{R}$ ,  $a \in [0, 1]$ :

$$F(au' + (1 - a)u'') \geq aF(u') + (1 - a)F(u'').$$

$$F\left(\sum_i a_i u_i\right) \geq \sum_i a_i F(u_i), \quad a_i \geq 0, \quad \sum_i a_i = 1$$

# A lower bound on log-likelihood

---

Pick ‘any’ distribution  $Q$  for hidden variable  $Z$ .

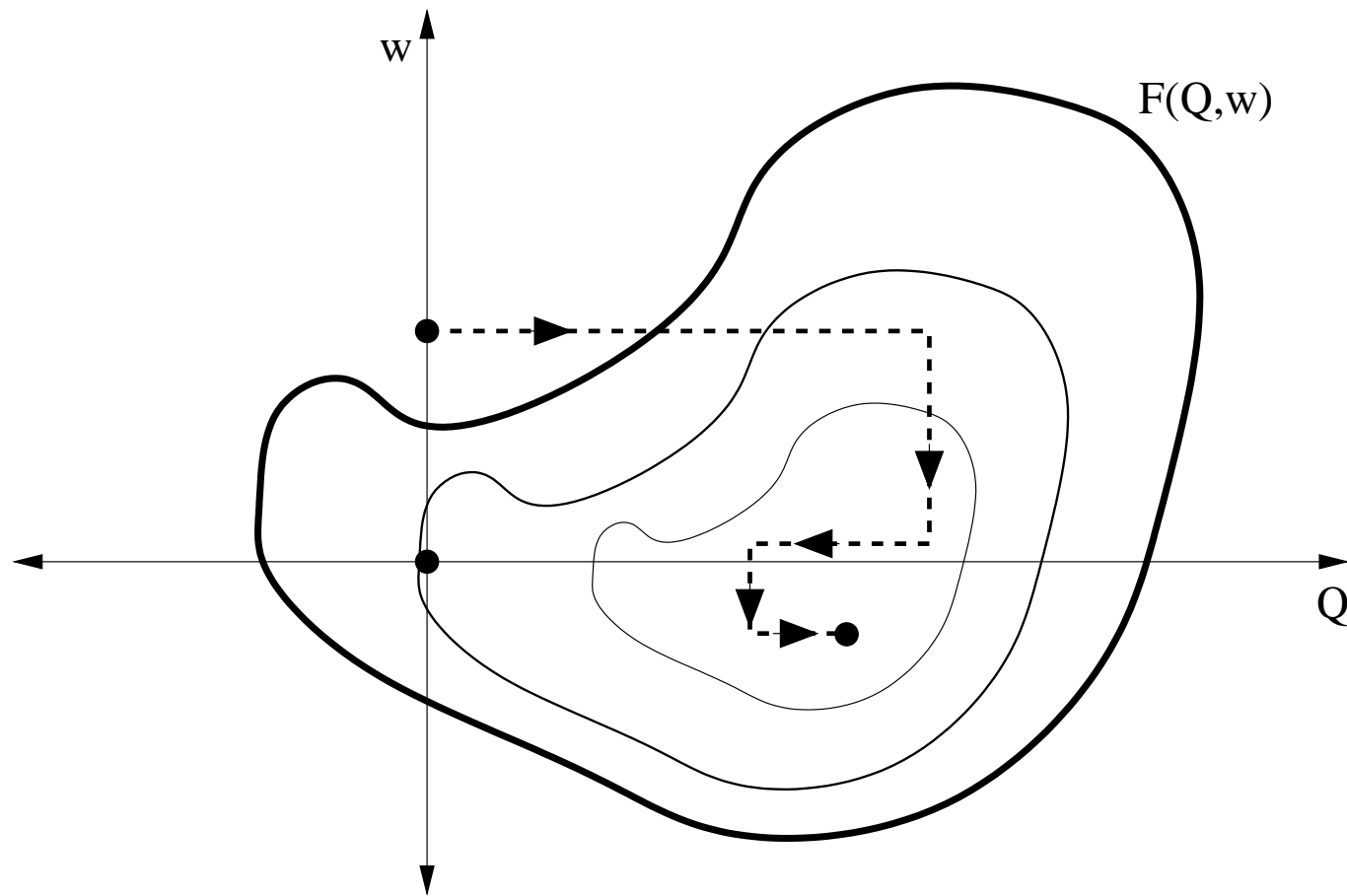
$$\sum_{\mathcal{Z}} Q(\mathcal{Z}) = 1, Q(\mathcal{Z}) > 0.$$

$\log(\cdot)$  is a concave function

$$\begin{aligned} \log p(\mathcal{D}|\mathbf{w}) &= \log \left( \sum_{\mathcal{Z}} p(\mathcal{D}, \mathcal{Z}|\mathbf{w}) \right) \\ &= \log \left( \sum_{\mathcal{Z}} Q(\mathcal{Z}) \frac{p(\mathcal{D}, \mathcal{Z}|\mathbf{w})}{Q(\mathcal{Z})} \right) \\ &\geq \sum_{\mathcal{Z}} Q(\mathcal{Z}) \log \left( \frac{p(\mathcal{D}, \mathcal{Z}|\mathbf{w})}{Q(\mathcal{Z})} \right) = \mathcal{F}(Q, \mathbf{w}) \end{aligned}$$

# Max $\mathcal{F}(Q, \mathbf{w})$ - lower bound on log-likelihood

Do 'coordinate-wise' ascent on  $\mathcal{F}(Q, \mathbf{w})$ .





# Maximize $\mathcal{F}(Q, \mathbf{w})$ w.r.t. $Q$ ( $\mathbf{w}$ fixed)

---

$$\begin{aligned}\mathcal{F}(Q, \mathbf{w}) &= \sum_{\mathcal{Z}} Q(\mathcal{Z}) \log \left( \frac{p(\mathcal{D}, \mathcal{Z} | \mathbf{w})}{Q(\mathcal{Z})} \right) \\ &= \sum_{\mathcal{Z}} Q(\mathcal{Z}) \log \left( \frac{p(\mathcal{Z} | \mathcal{D}, \mathbf{w}) \cdot p(\mathcal{D} | \mathbf{w})}{Q(\mathcal{Z})} \right) \\ &= \sum_{\mathcal{Z}} Q(\mathcal{Z}) \log \frac{p(\mathcal{Z} | \mathcal{D}, \mathbf{w})}{Q(\mathcal{Z})} + \sum_{\mathcal{Z}} Q(\mathcal{Z}) \cdot \log p(\mathcal{D} | \mathbf{w}) \\ &= - \sum_{\mathcal{Z}} Q(\mathcal{Z}) \log \frac{Q(\mathcal{Z})}{p(\mathcal{Z} | \mathcal{D}, \mathbf{w})} + \log p(\mathcal{D} | \mathbf{w}) \sum_{\mathcal{Z}} Q(\mathcal{Z}) \\ &= -D_{KL}[Q(\mathcal{Z}) || P(\mathcal{Z} | \mathcal{D}, \mathbf{w})] + \log p(\mathcal{D} | \mathbf{w}).\end{aligned}$$

# Maximize $\mathcal{F}(Q, \mathbf{w})$ w.r.t. $Q$ ( $\mathbf{w}$ fixed)

---

Since  $\log p(\mathcal{D}|\mathbf{w})$  is constant w.r.t.  $Q$ , we only need to **maximize**  $-D_{KL}[Q(Z)||P(Z|\mathcal{D}, \mathbf{w})]$ , which is equivalent to **minimizing**

$$D_{KL}[Q(Z)||P(Z|\mathcal{D}, \mathbf{w})] \geq 0.$$

This is achieved when  $D_{KL}[Q(Z)||P(Z|\mathcal{D}, \mathbf{w})] = 0$ , i.e. when

$$Q_*(Z) = P(Z|\mathcal{D}, \mathbf{w})$$

Note:  $\mathcal{F}(Q, \mathbf{w}) = \log p(\mathcal{D}|\mathbf{w})$  - no longer a loose lower bound!

# E-step

---

The optimal way for guessing the values of hidden variables  $Z$  is to set the distribution of  $Z$  to the posterior over  $\mathcal{Z}$ , given the observed data  $\mathcal{D}$  and current parameter settings  $\mathbf{w}$ .

# Maximize $\mathcal{F}(Q, \mathbf{w})$ w.r.t. $\mathbf{w}$ ( $Q$ is fixed to $Q_*$ )

---

$$\begin{aligned}\mathcal{F}(Q_*, \mathbf{w}) &= \sum_{\mathcal{Z}} Q_*(\mathcal{Z}) \log \left( \frac{p(\mathcal{D}, \mathcal{Z} | \mathbf{w})}{Q_*(\mathcal{Z})} \right) \\ &= \sum_{\mathcal{Z}} Q_*(\mathcal{Z}) \log p(\mathcal{D}, \mathcal{Z} | \mathbf{w}) - \sum_{\mathcal{Z}} Q_*(\mathcal{Z}) \log Q_*(\mathcal{Z}) \\ &= E_{Q_*(Z)}[\log p(\mathcal{D}, \mathcal{Z} | \mathbf{w})] + H(Q_*).\end{aligned}$$

Since the entropy of  $Q_*$ ,  $H(Q_*)$ , is constant ( $Q_*$  is fixed), we only need to maximize  $E_{Q_*(Z)}[\log p(\mathcal{D}, Z | \mathbf{w})]$ :

$$\mathbf{w}_* = \underset{\mathbf{w}}{\operatorname{argmax}} E_{Q_*(Z)}[\log p(\mathcal{D}, Z | \mathbf{w})].$$

# M-step

---

The **optimal way** for estimating the parameters  $\mathbf{w}$  is to select the parameter values  $\mathbf{w}_*$  that **maximize the expected value of the complete data log-likelihood**  $p(\mathcal{D}, \mathcal{Z}|\mathbf{w})$ , where the **expectation** is taken **w.r.t. the posterior distribution over the hidden data  $\mathcal{Z}$ ,  $P(\mathcal{Z}|\mathcal{D}, \mathbf{w})$**  (our best guess).

Find a single parameter vector  $\mathbf{w}_*$  for all hidden variable settings  $\mathcal{Z}$  (since we don't know the true values of  $Z$ ), but while doing this, **weight the importance of each particular setting  $\mathcal{Z}$  by the posterior probability  $P(\mathcal{Z}|\mathcal{D}, \mathbf{w})$** .

# E-M Algorithm

---

Given the current parameter setting  $\mathbf{w}^{old}$  do:

■ E-step:

Estimate  $P(Z|\mathcal{D}, \mathbf{w}^{old})$ , the posterior distribution over  $\mathcal{Z}$ , given the observed data  $\mathcal{D}$  and current parameter settings  $\mathbf{w}^{old}$ .

■ M-step:

Obtain new parameter values  $\mathbf{w}^{new}$  by **maximizing**

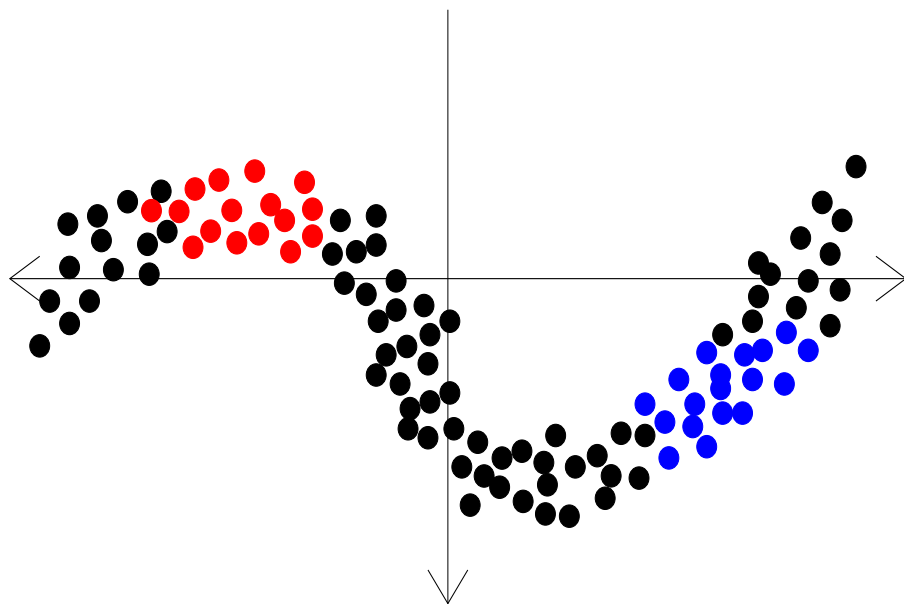
$$E_{P(Z|\mathcal{D}, \mathbf{w}^{old})} [\log p(\mathcal{D}, Z|\mathbf{w})].$$

■ Set  $\mathbf{w}^{old} := \mathbf{w}^{new}$  and go to E-step.

# Manifold learning (for ants)

---

Imagine you are an ant and you are faced with this 2-dim data set:



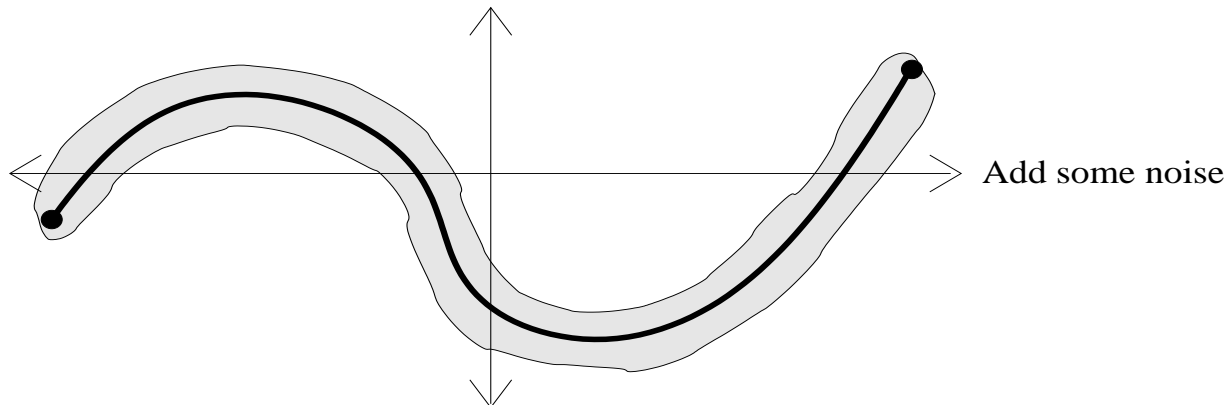
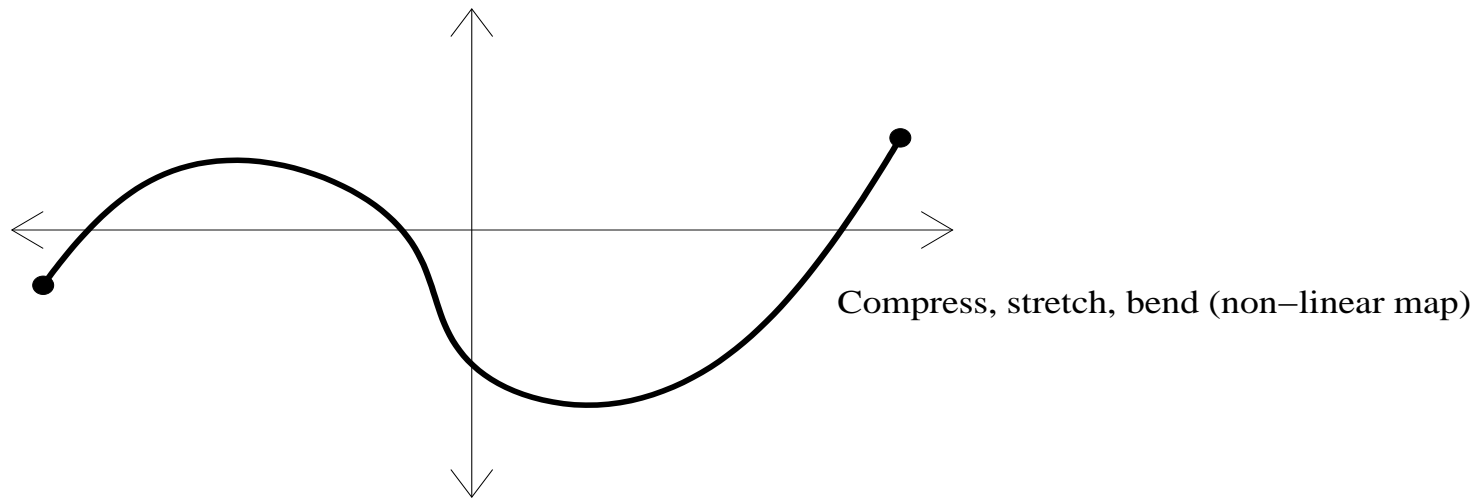
but you can comprehend only 1-dim patterns

We should be able to understand the data - it corresponds to a ‘noisy’  
1-dim manifold

# Build a latent space model of data distribution

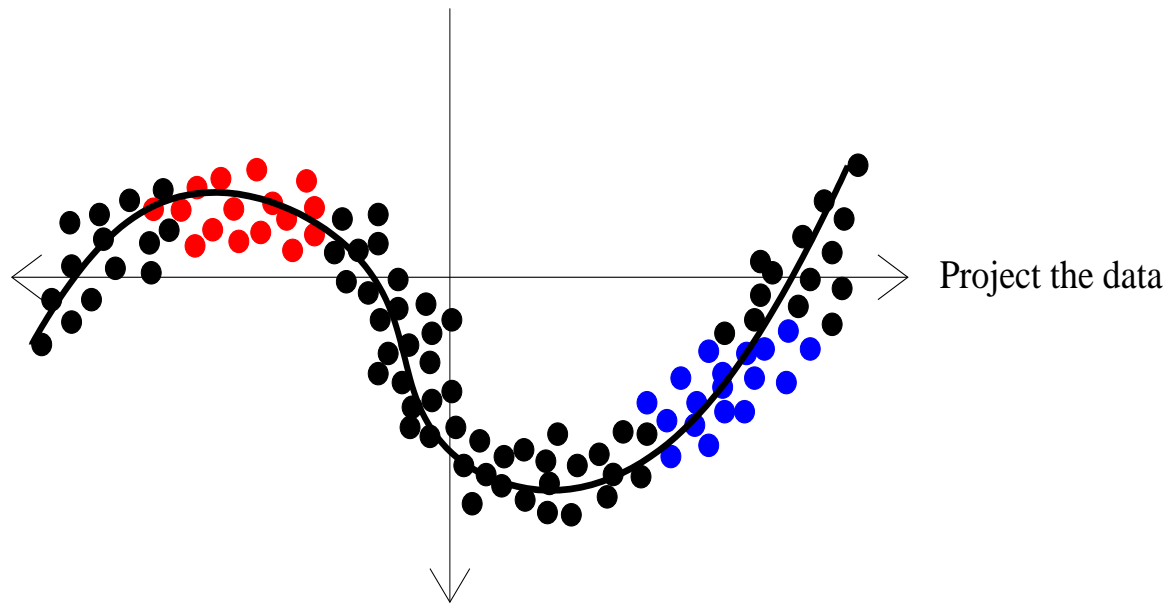
I think I know how the data might have been generated

Everything can be "explained" using a line segment (Latent Space = computer screen)



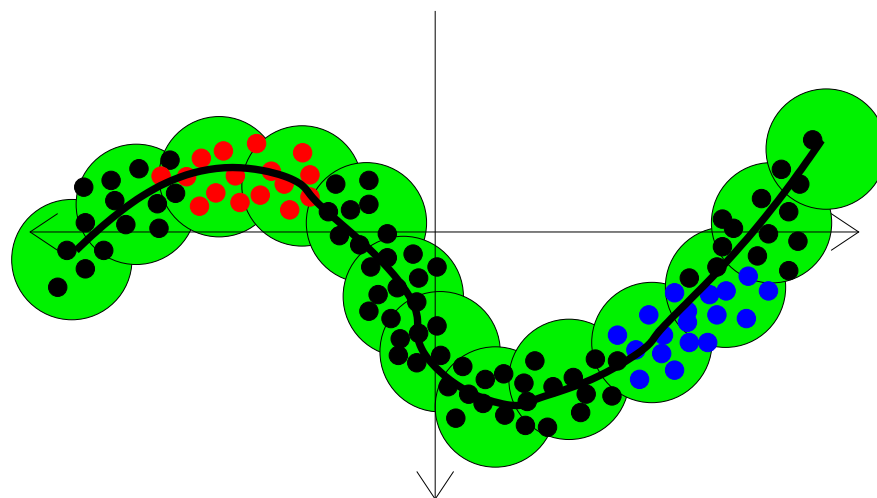


# Project the data



Stretch back to a straight line  
(computer screen)

# Constrained mixture models

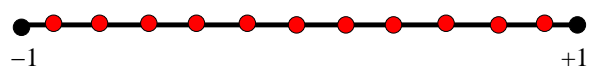


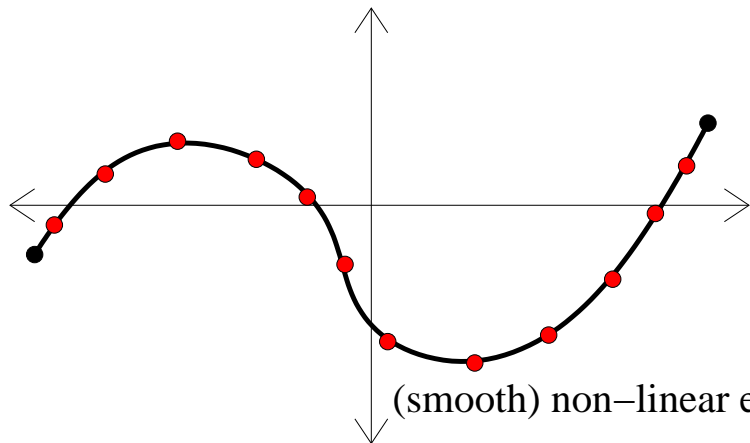
A chain of noise models (Gaussians) along a 1-dim "mean data manifold"

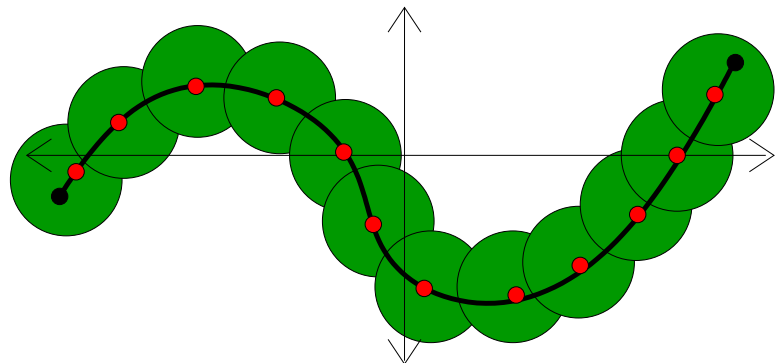
Constrained mixture of noise models - spherical Gaussians.

Still  $p(\mathbf{t}) = \sum_{j=1}^M P(j) \cdot p(\mathbf{t}|j)$ , but now the Gaussians are forced to have their means organized along a smooth 1-dim manifold.

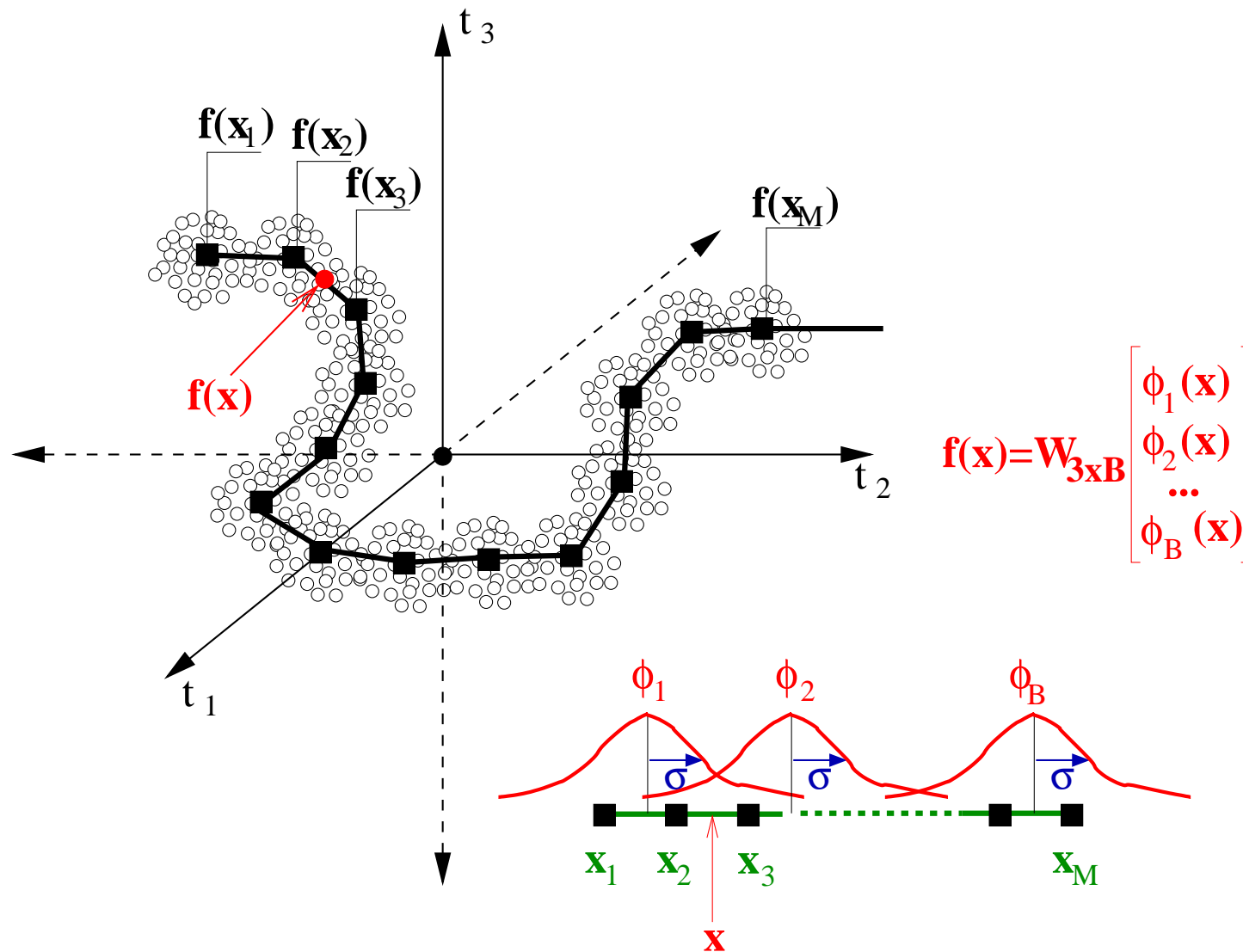
# Smooth embedding of continuous latent space

 low-dim latent space (continuous)

 (smooth) non-linear embedding in high-dim model space

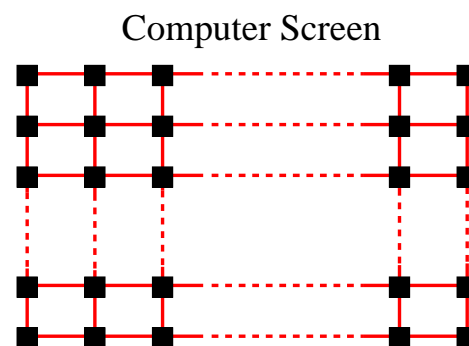
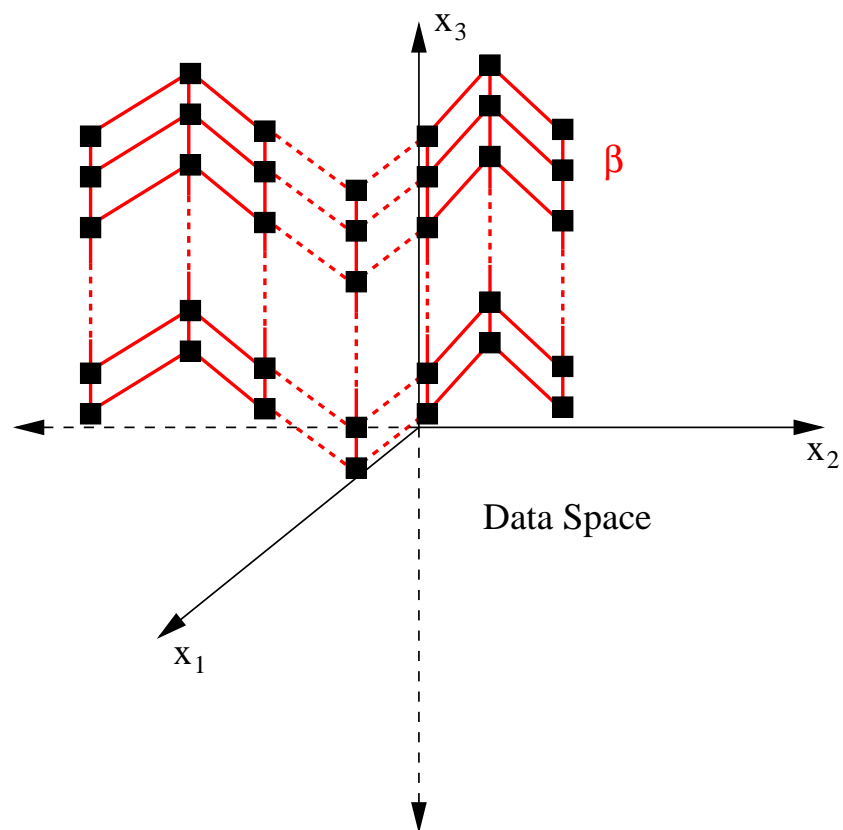
 constrained mixture

# Smooth embedding of continuous latent space

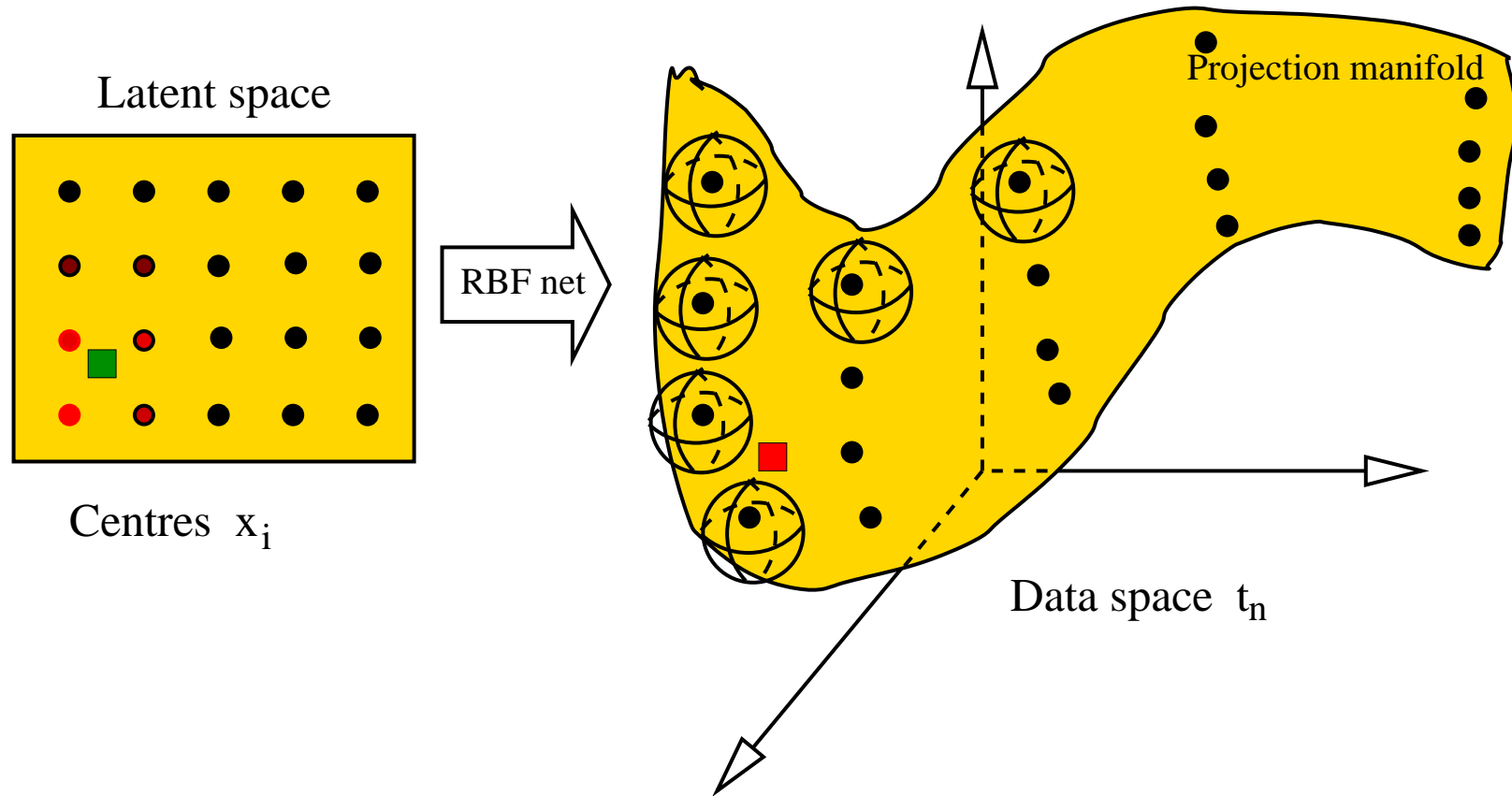


# Data (roughly) along a 2-D manifold

Generalize the notion of ‘bicycle chain’ of codebook vectors: Take advantage of two-dimensional structure of the computer screen. Cover it with a 2-dimensional grid of nodes.



# Generative Topographic Mapping



# GTM - model formulation

---

- Models probability distribution in the (observable) high-dim data space  $\mathfrak{R}^D$  by means of low-dim latent variables. Data is visualized in the latent space  $\mathcal{H} \subset \mathfrak{R}^L$  (e.g.  $[-1, 1]^2$ ).
- Latent space  $\mathcal{H}$  is covered with an array of  $C$  **latent space centers**  $\mathbf{x}_c \in \mathcal{H}$ ,  $c = 1, 2, \dots, C$ .
- **Non-linear GTM map**  $f : \mathcal{H} \rightarrow \mathcal{D}$  is defined using a kernel regression –  $B$  fixed basis functions  $\phi_j : \mathcal{H} \rightarrow \mathfrak{R}$ , collected in  $\phi$ , (Gaussians of the same width  $\sigma$ ),  $D \times B$  matrix of weights  $\mathbf{W}$ :

$$f(\mathbf{x}) = \mathbf{W}_{D \times B} \phi(\mathbf{x})$$

# GTM - model formulation - Cont'd

---

- GTM creates a generative probabilistic model in the data space by placing radially-symmetric Gaussians  $P(\mathbf{t} | \mathbf{x}_c, \mathbf{W}, \beta)$  (zero mean, inverse variance  $\beta$ ) around  $f(\mathbf{x}_c)$ ,  $c = 1, 2, \dots, C$ .
- Defining a uniform prior over  $\mathbf{x}_c$ , the GTM density model is

$$P(\mathbf{t} | \mathbf{W}, \beta) = \frac{1}{C} \sum_{c=1}^C P(\mathbf{t} | \mathbf{x}_c, \mathbf{W}, \beta)$$

- The data is modelled as a constrained mixture of Gaussians. GTM can be trained using an EM algorithm.
- Mixture of Gaussians where we sneaked in a non-linear model (low-dim manifold) where the Gaussian centers can lie.



# GTM - Data Visualization

---

- Posterior probability that the  $c$ -th Gaussian generated  $\mathbf{t}_n$ ,

$$r_{c,n} = \frac{P(\mathbf{t}_n | \mathbf{x}_c, \mathbf{W}, \beta)}{\sum_{j=1}^C P(\mathbf{t}_n | \mathbf{x}_j, \mathbf{W}, \beta)}.$$

- The latent space representation of the point  $\mathbf{t}_n$ , i.e. the projection of  $\mathbf{t}_n$ , is taken to be the mean

$$\sum_{c=1}^C r_{cn} \mathbf{x}_c$$

of the posterior distribution on  $\mathcal{H}$ .

# Differential geometry on projection manifold

---

Unlike many other manifold learning methods, GTM provides explicit parametrized model for the data manifold!

Latent space - global co-ordinate chart.

## Magnification Factors:

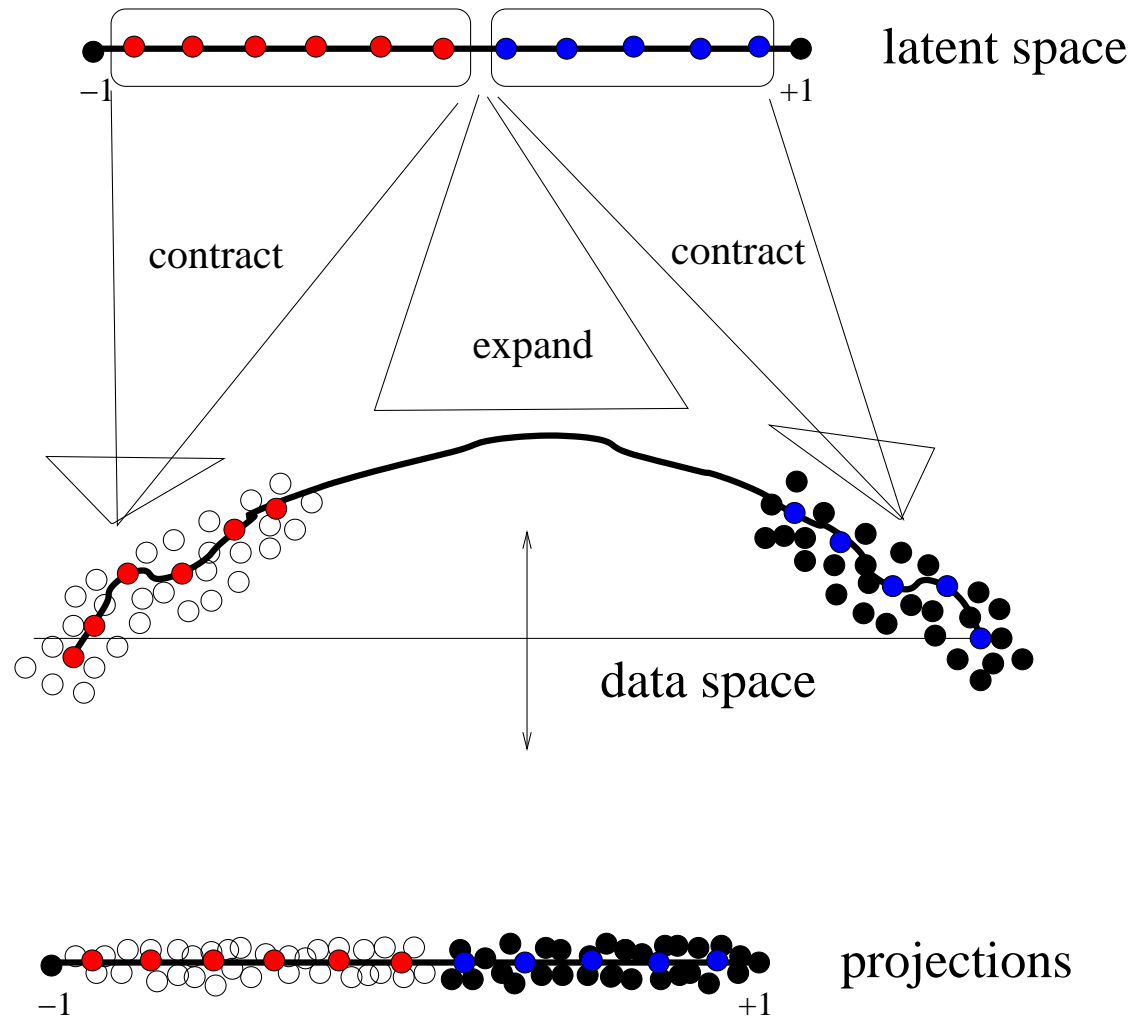
We can measure stretch in the sheet. This can be used to detect the gaps between data clusters.

## Directional Curvatures:

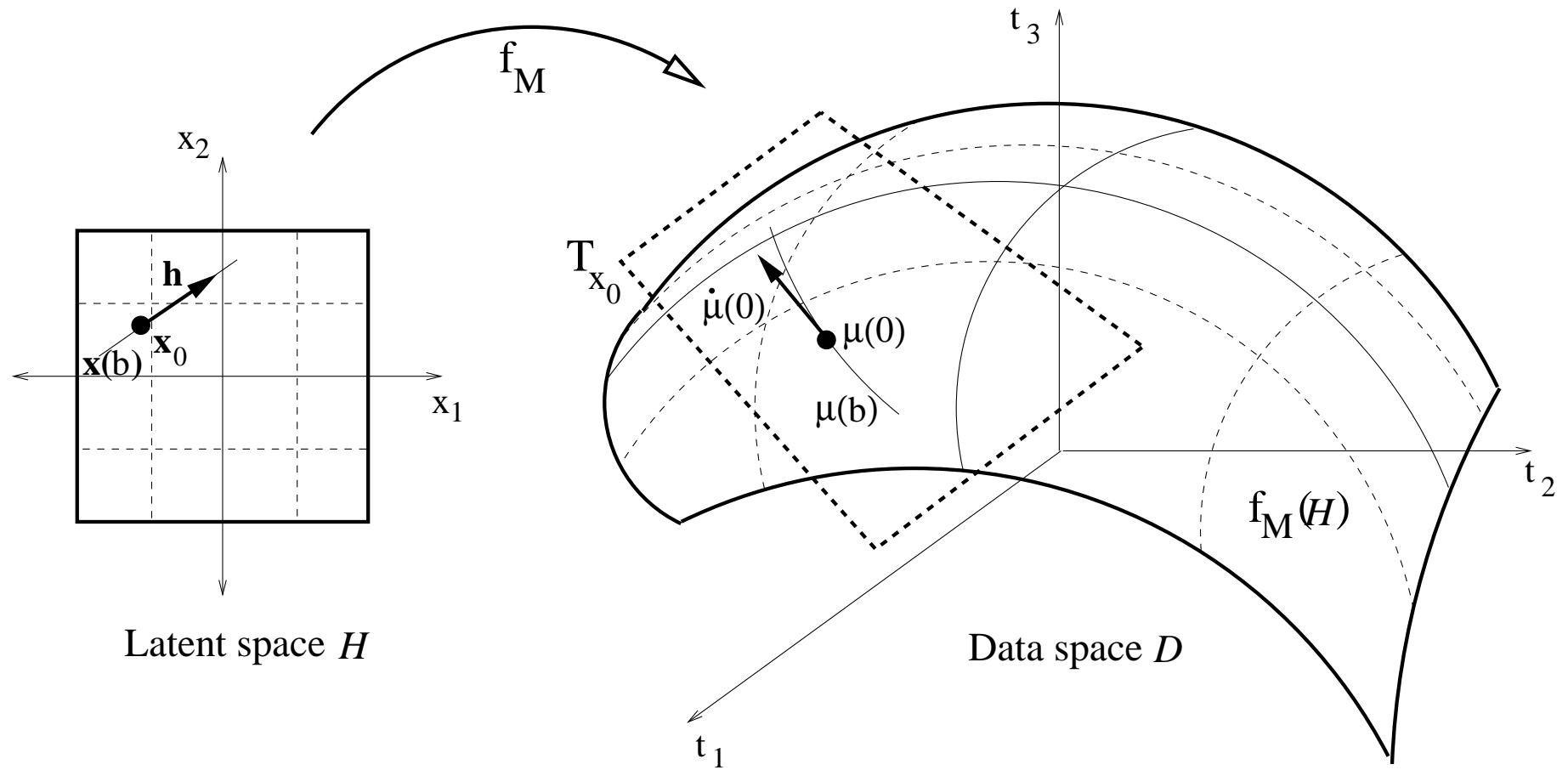
We can also measure the directional curvature of the 2-D sheet embedded in the high-dim data space.

Visualize the magnitude and direction of the local largest curvatures to see where and how the manifold is most folded.

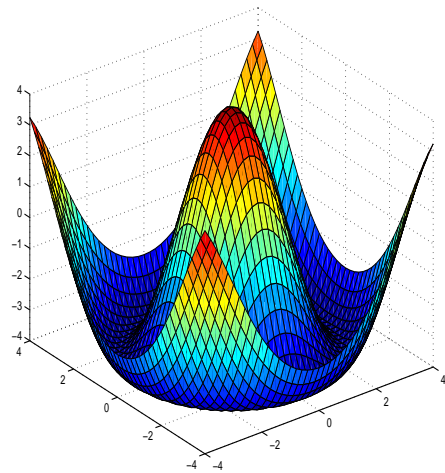
# Magnification Factors (detect clusters)



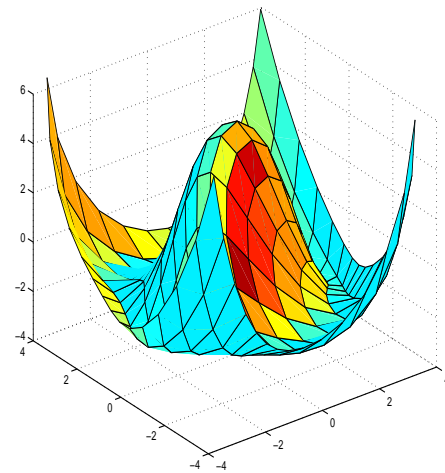
# Directional Curvatures (detect foldings)



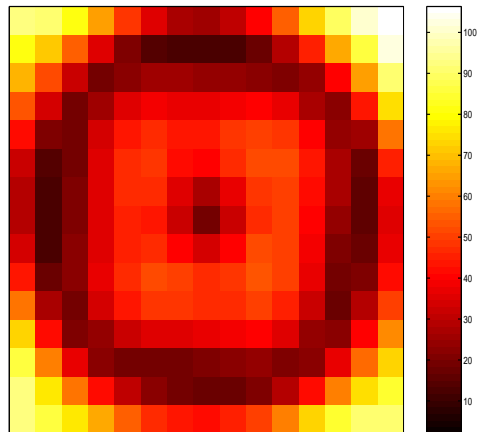
# Toy Example



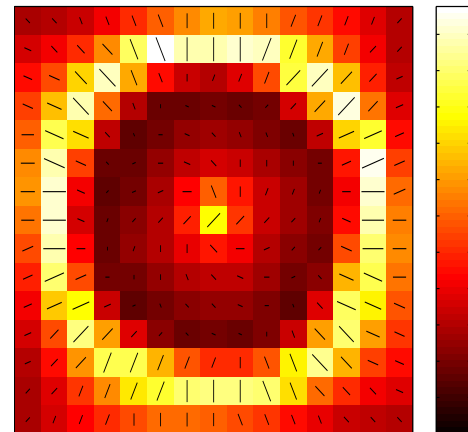
Data manifold



Projection manifold



Magn. factors



Dir. curvatures

# Directional Curvature

---

- The tangent vector  $\dot{\mu}(0)$  to  $\mu$  at  $\mu(0)$  lies in  $\mathbf{T}_{\mathbf{x}_0}$  (dashed rectangle), the tangent plane of the manifold  $f(\mathcal{H})$  at  $\mu(0)$ .
- Let  $\mathbf{\Gamma}_r^{(1)}$  be a (column) vector of partial derivatives of the function

$$f = (f^1, f^2, \dots, f^D)^T,$$

with respect to the  $r$ -th latent space variable at  $\mathbf{x}_0 \in \mathcal{H}$ ,

- Let  $\mathbf{\Gamma}^{(1)}$  be the  $D \times L$  matrix

$$\mathbf{\Gamma}^{(1)} = [\mathbf{\Gamma}_1^{(1)}, \mathbf{\Gamma}_2^{(1)}, \dots, \mathbf{\Gamma}_L^{(1)}].$$

- The range of the matrix  $\mathbf{\Gamma}^{(1)}$  is the tangent plane  $\mathbf{T}_{\mathbf{x}_0}$  of the projection manifold  $\Omega$  at  $f(\mathbf{x}_0) = \mu(0)$ .

# Directional Curvature

---

- Orthogonal projection onto  $\mathbf{T}_{\mathbf{x}_0}$  is a linear operator described by the projection matrix

$$\mathbf{\Pi} = \mathbf{\Gamma}^{(1)} \left( \mathbf{\Gamma}^{(1)} \right)^+,$$

- Decompose the second directional derivative  $\ddot{\mu}(0)$  of  $f$  into two orthogonal components, one lying in the tangent space  $\mathbf{T}_{\mathbf{x}_0}$ , the other lying in its orthogonal complement  $\mathbf{T}_{\mathbf{x}_0}^\perp$ ,

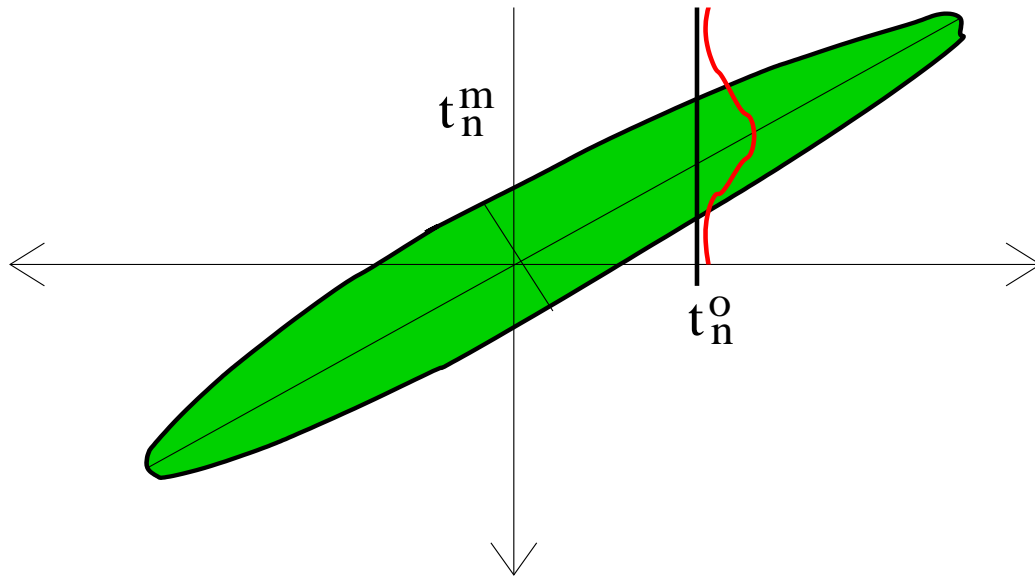
$$\ddot{\mu}(0) = \ddot{\mu}^\parallel(0) + \ddot{\mu}^\perp(0), \quad \ddot{\mu}^\parallel(0) \in \mathbf{T}_{\mathbf{x}_0}, \quad \ddot{\mu}^\perp(0) \in \mathbf{T}_{\mathbf{x}_0}^\perp.$$

- $\ddot{\mu}^\parallel(0)$  - changes in the first-order derivatives due to “varying speed of parameterization”  
 $\ddot{\mu}^\perp(0)$  - changes in the first-order derivatives that are responsible for curving of the projection manifold  $\Omega$

# Missing Data

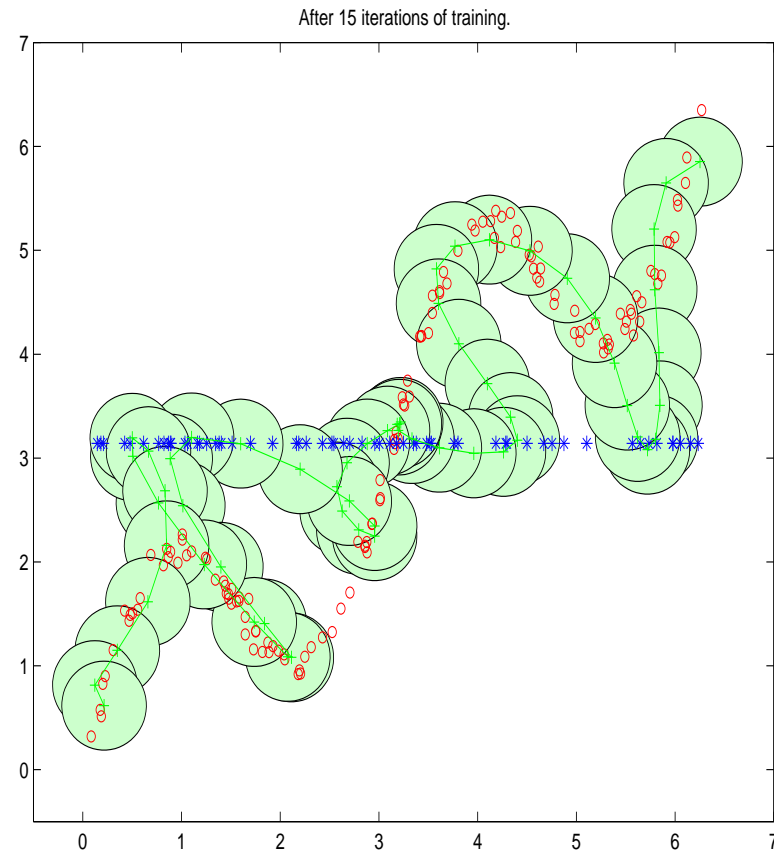
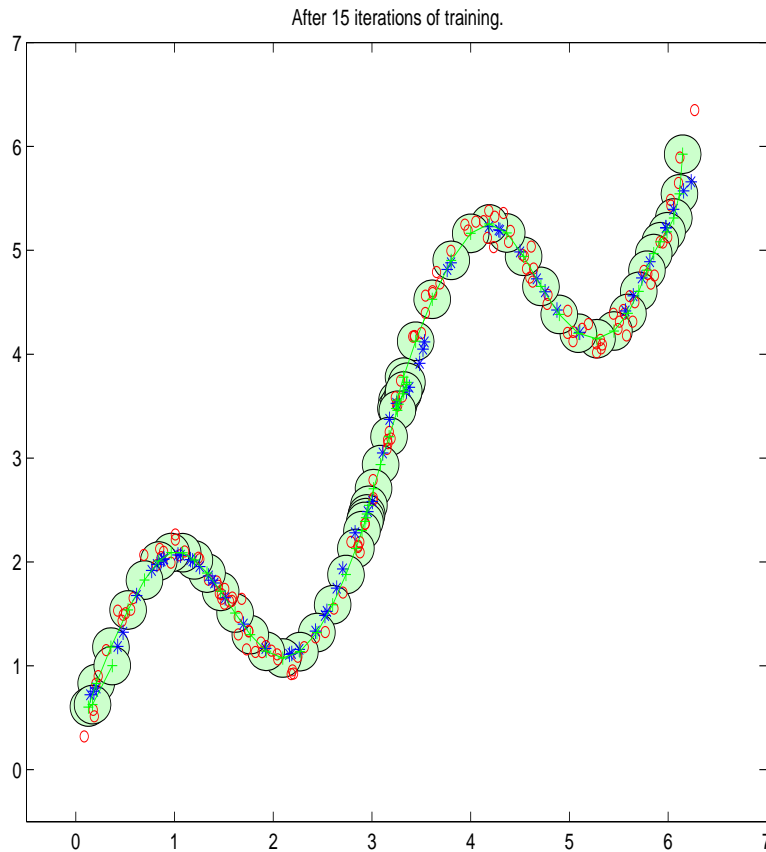
Data point  $\mathbf{t}_n$  divided into an **observed component**  $\mathbf{t}_n^o$  and a **missing component**  $\mathbf{t}_n^m$ .

Having a **generative probabilistic model of the data** can help us to **deal with missing values in a principled manner - treat them as latent variables!**



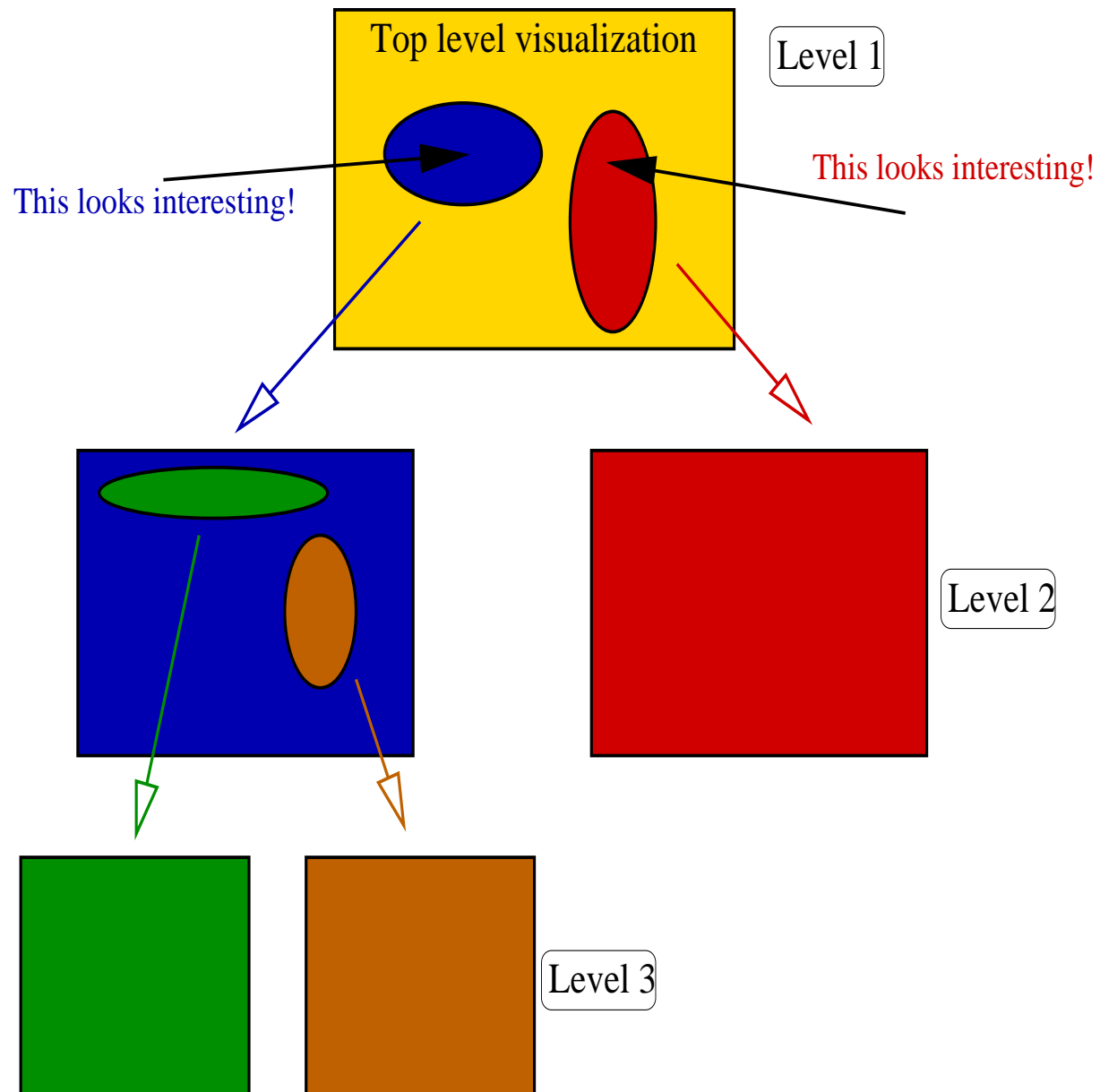


# Toy Example – Missing Data



- o – complete data points
- + – centers of the Gaussian mixture components
- \* – Filled in missing values
- discs – 2 standard deviations of the noise model.

# Hierarchy of GTMs

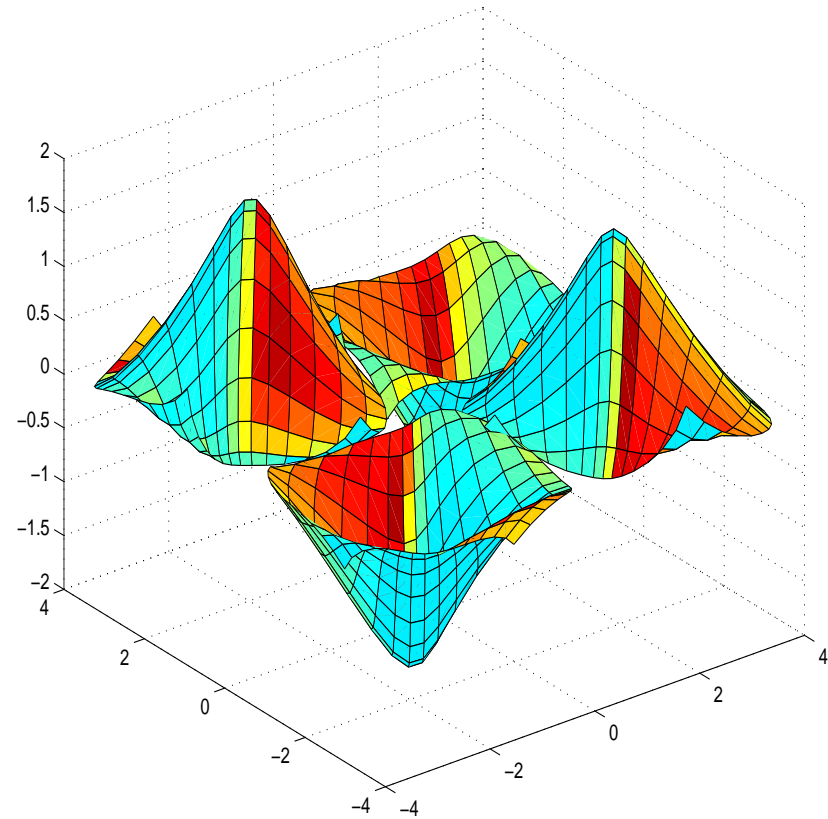
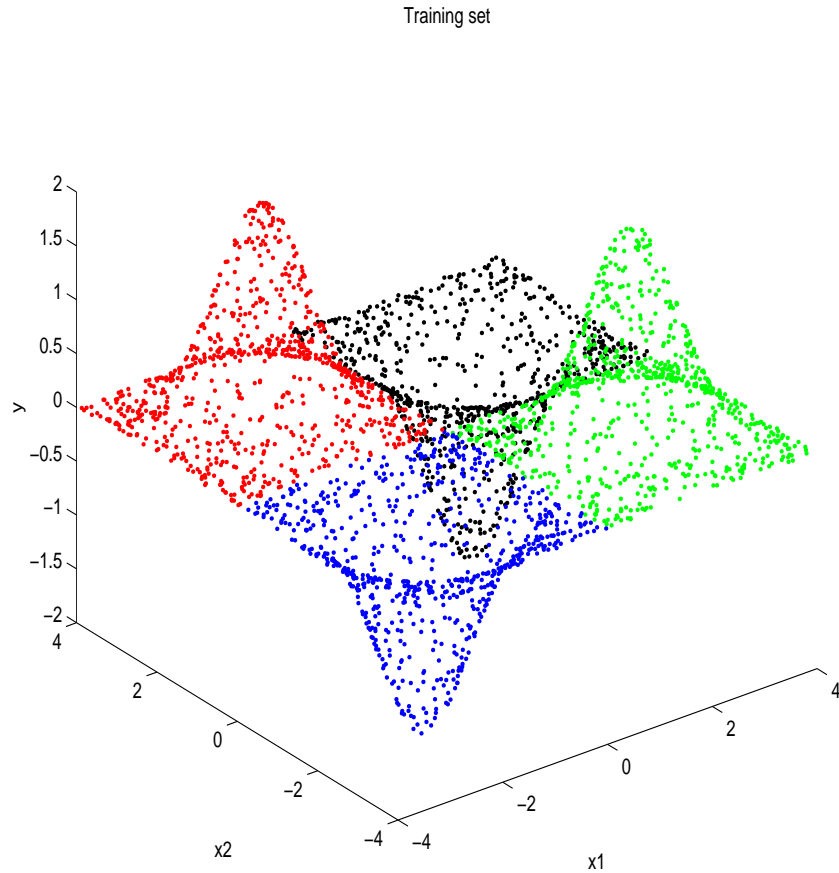


# 3 types of hidden variables!

---

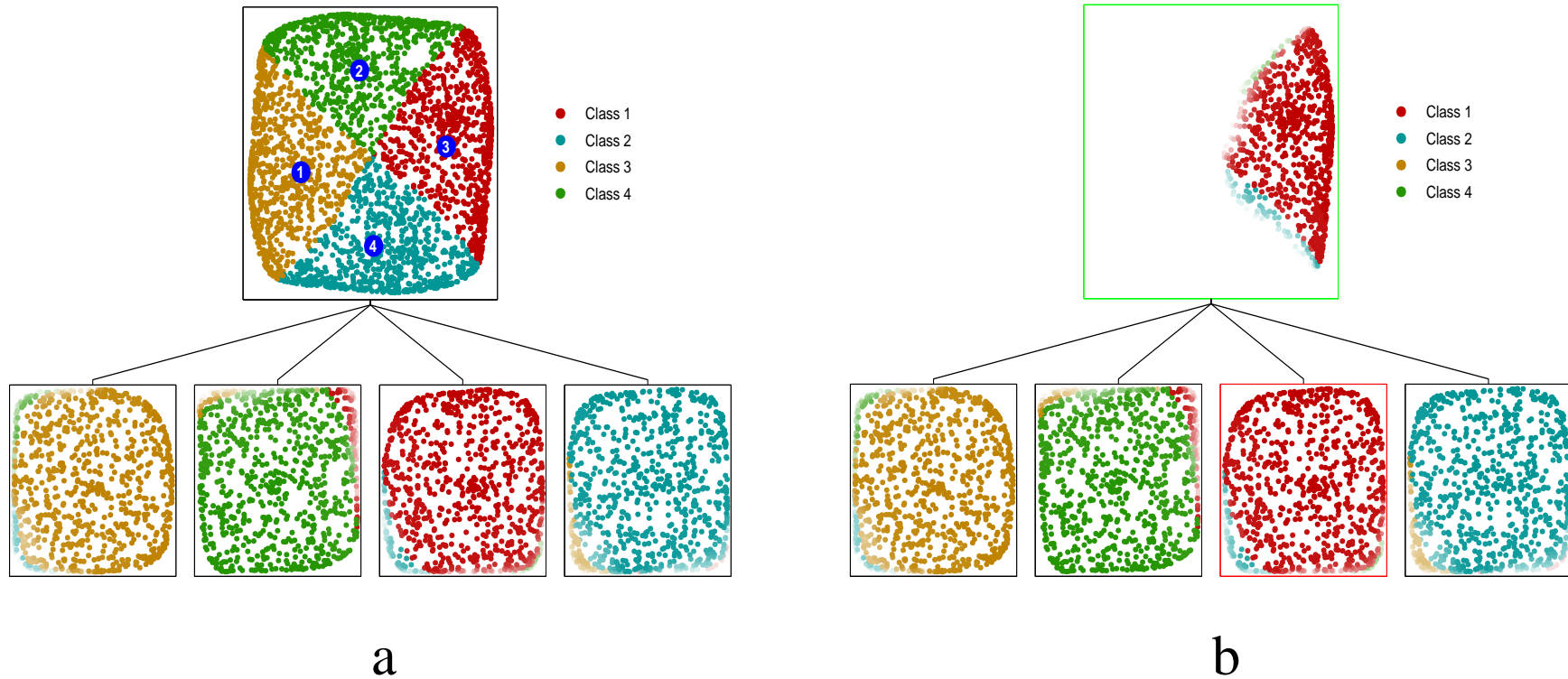
1. We do not know the exact assignments of points  $\mathbf{t}_n$  to GTMs  $\mathcal{N}$  at level  $\ell$ , but we do have model responsibilities  $P(\mathcal{N} | \mathbf{t}_n)$  from the previous step.  
 $P(\mathcal{N} | \mathbf{t}_n)$  is the posterior probability that GTM  $\mathcal{N}$  generated  $\mathbf{t}_n$ .
2. Assuming that GTM  $\mathcal{N}$  at level  $\ell$  generated  $\mathbf{t}_n$ , we do not know the exact assignments of its children  $\mathcal{M}$  at level  $\ell + 1$  to  $\mathbf{t}_n$ . We can calculate parent-conditional responsibilities  $P(\mathcal{M} | \mathcal{N}, \mathbf{t}_n)$
3. Assuming that GTM  $\mathcal{M}$  at level  $\ell + 1$  generated  $\mathbf{t}_n$ , we do not know the exact assignments of its latent space centers  $\mathbf{x}_i^{\mathcal{M}}$  to  $\mathbf{t}_n$ . We can evaluate  $R_{i,n}^{\mathcal{M}}$

# Toy Example – HGTM



Data + child projection manifolds

# Toy Example – HGTM plots



Projections + Child modulated parent plot

# Oil data

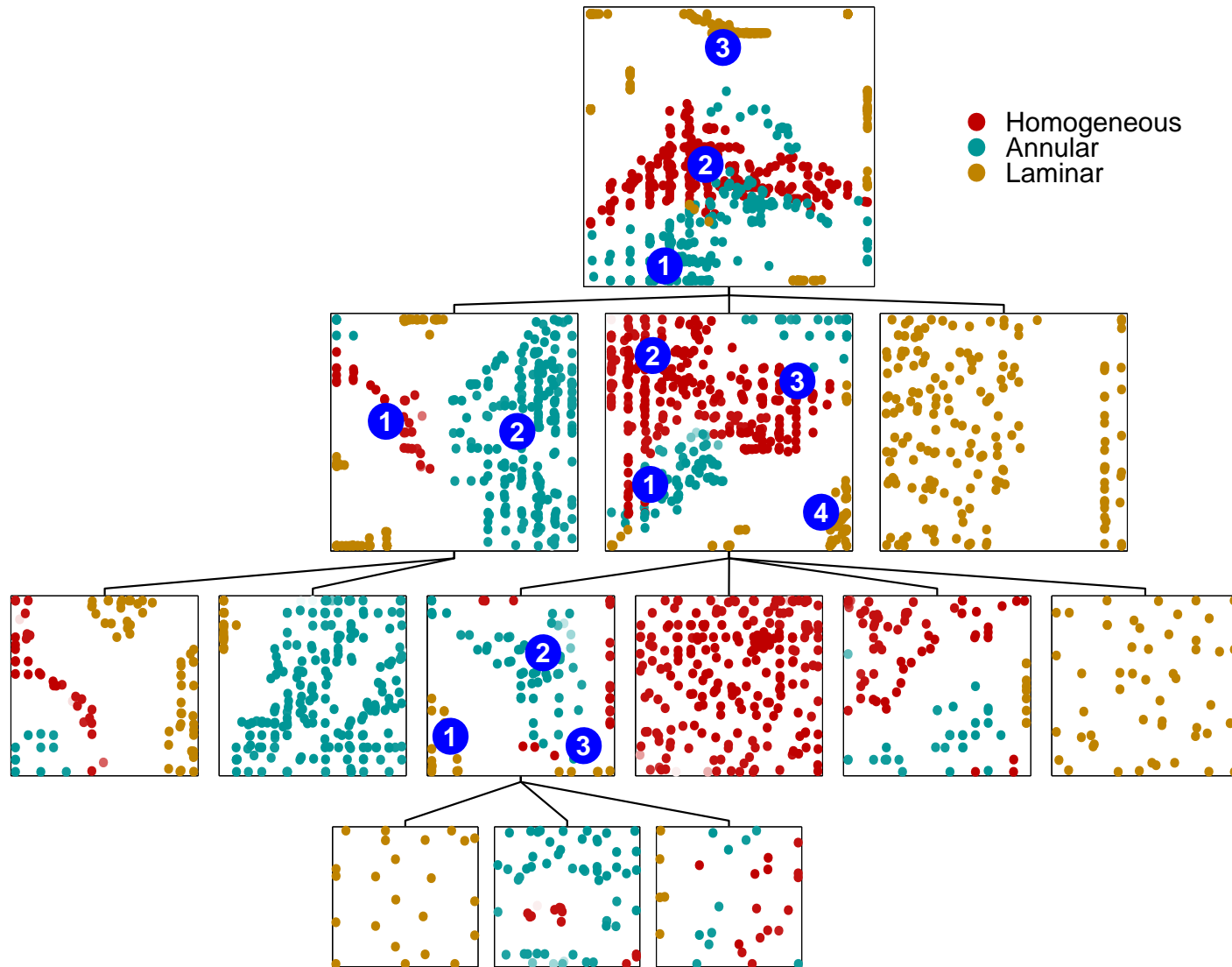
---

Data set arises from a physics-based simulation of non-invasive monitoring system, used to determine the quantity of oil in a multi-phase pipeline containing a mixture of oil, water and gas

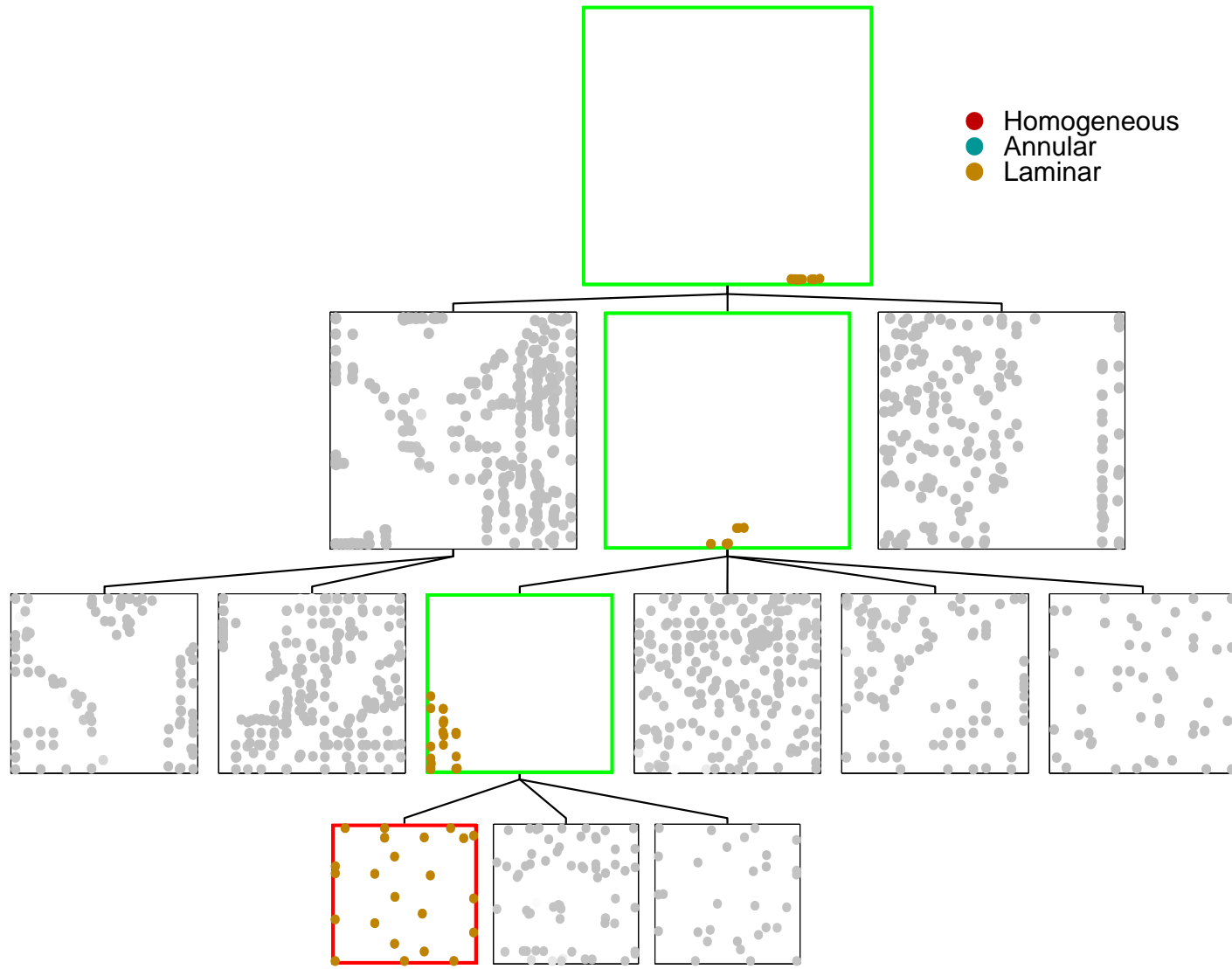
1000 points in 12-dim space

Points in the data set are classified into 3 different multi-phase flow configurations – *homogeneous*, *annular* and *laminar*

# Hierarchical GTM - Oil data



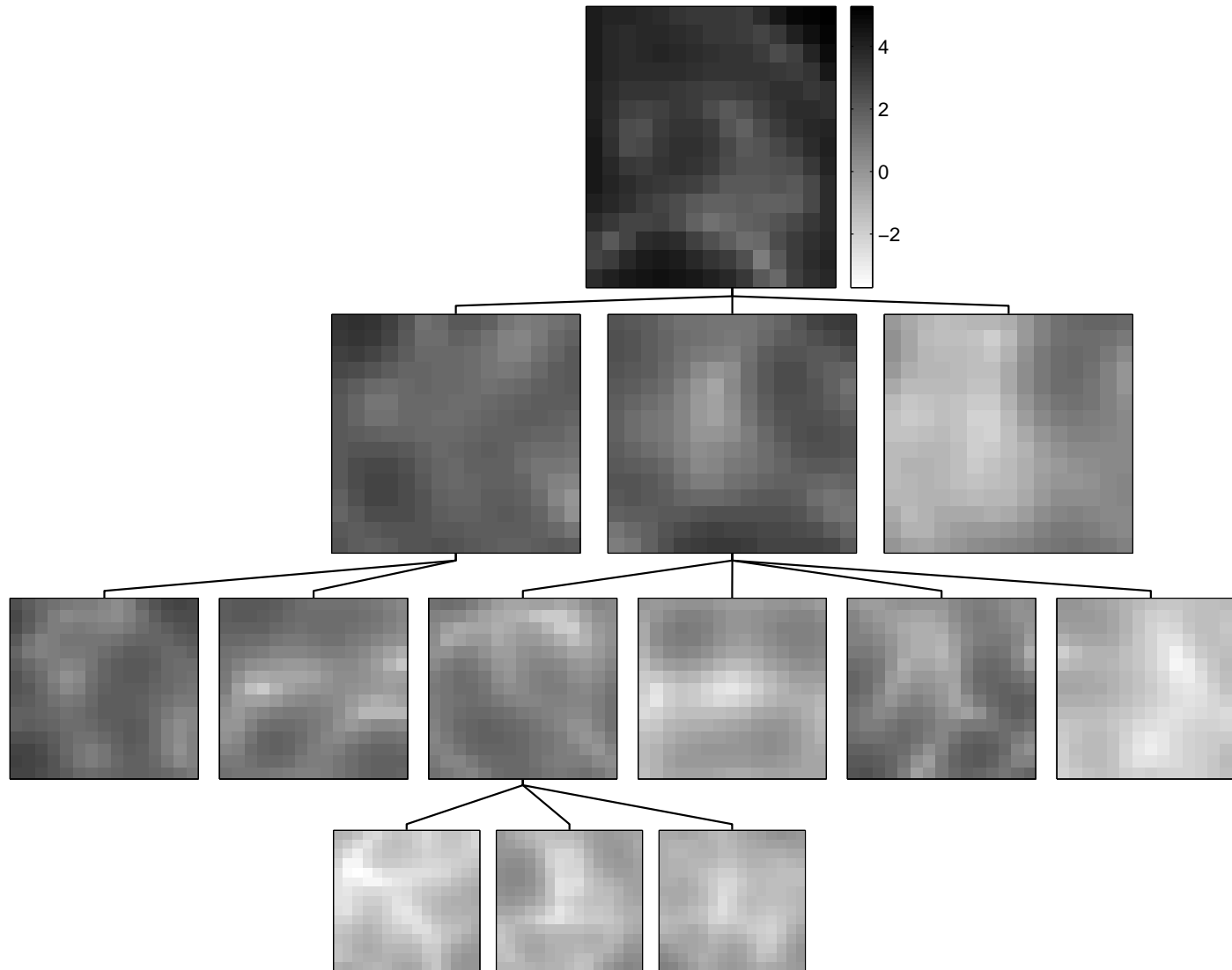
# Understanding the plot



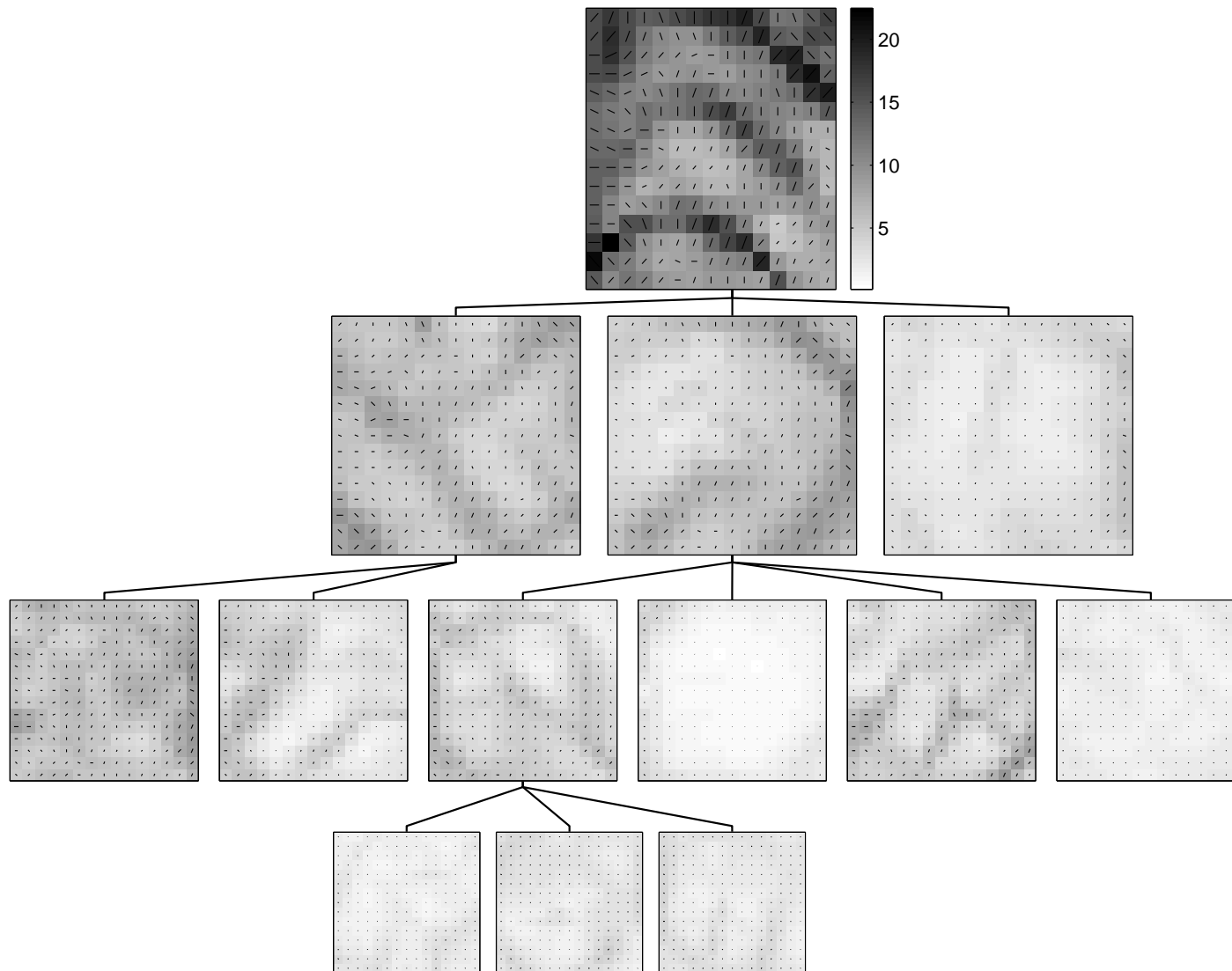


# Magnification factors

---

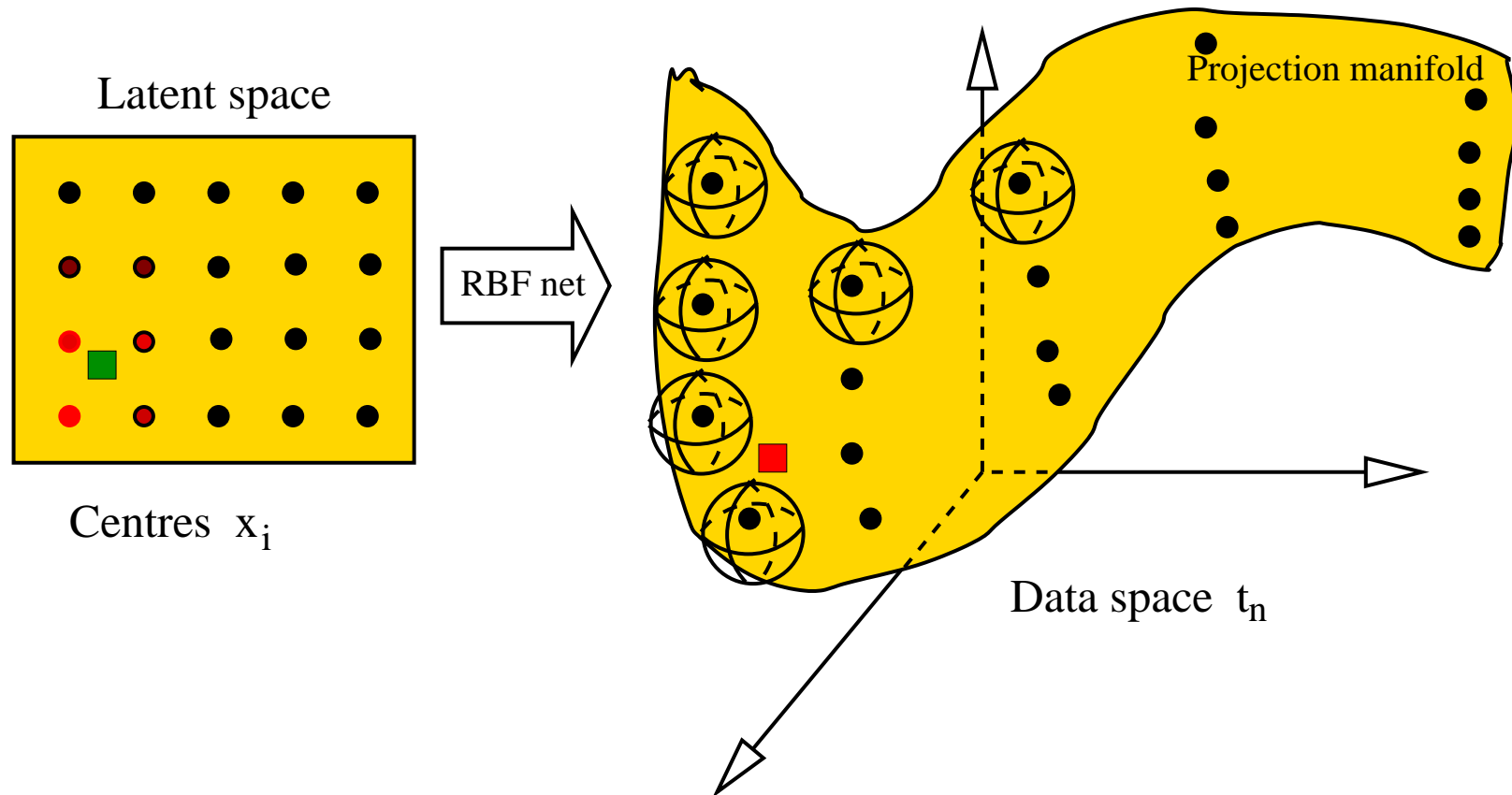


# Directional Curvatures



# Let's get crazy - GTM in the model space!

Other data types  $\implies$  other noise models



# Other data types $\implies$ other noise models

---

- Probabilistic framework is convenient - we can deal with arbitrary data types as long as an appropriate noise model can be formulated.
- The principle will be demonstrated on sequential data, but extensions to other data types (graphs) are possible.
- For sequential data
  - need noise models that take into account temporal correlations within sequences, e.g. Markov chains, HMMs, etc.

# Latent Trait HMM (LTHMM)

---

GTM with HMM as the noise model!

For each HMM (latent center) we need to parameterize several multinomials

- initial state probabilities
- transition probabilities
- emission probabilities (discrete observations)

Multinomials are parameterized through natural parameters.

# LTHMM

---

- alphabet of  $S$  symbols,  $\mathcal{S} = \{1, 2, \dots, S\}$ .
- Consider a set of symbolic sequences,  $\mathbf{s}^{(n)} = (s_t^{(n)})_{t=1:T_n}$ ,  $n = 1, 2, \dots, N$
- **With each latent point  $\mathbf{x} \in \mathcal{H}$ , we associate a generative distribution (HMM with  $K$  hidden states) over sequences  $p(\mathbf{s}|\mathbf{x})$ .**

$$p(\mathbf{s}|\mathbf{x}) = \sum_{\mathbf{h} \in K^{T_n}} p(h_1|\mathbf{x}) \prod_{t=2}^{T_n} p(h_t|h_{t-1}, \mathbf{x}) \prod_{t=1}^{T_n} p(s_t|h_t, \mathbf{x})$$

- Assuming independently generated sequences, the likelihood is

$$\mathcal{L} = \prod_{n=1}^N p(\mathbf{s}^{(n)}) = \prod_{n=1}^N \frac{1}{C} \sum_{c=1}^C p(\mathbf{s}^{(n)}|\mathbf{x}_c).$$

# Smooth manifold in the k-state HMM space

---

LTHMM parameters are obtained through a parameterized *smooth non-linear mapping from the latent space (global coordinate chart) into the HMM natural parameter space (another global coordinate chart)*.

$g(\cdot)$  is the softmax function (the canonical inverse link function of multinomial distributions)

$$g_k \left( (a_1, a_2, \dots, a_\ell)^T \right) = \frac{\exp\{a_k\}}{\sum_{i=1}^{\ell} \exp\{a_i\}}, \quad k = 1, 2, \dots, \ell,$$

Free parameters:  $\mathbf{A}^{(\boldsymbol{\pi})} \in \mathbb{R}^{K \times B}$ ,  $\mathbf{A}^{(\mathbf{T}_l)} \in \mathbb{R}^{K \times B}$  and  $\mathbf{A}^{(\mathbf{B}_k)} \in \mathbb{R}^{S \times B}$

# From latent points to (local) noise models

---

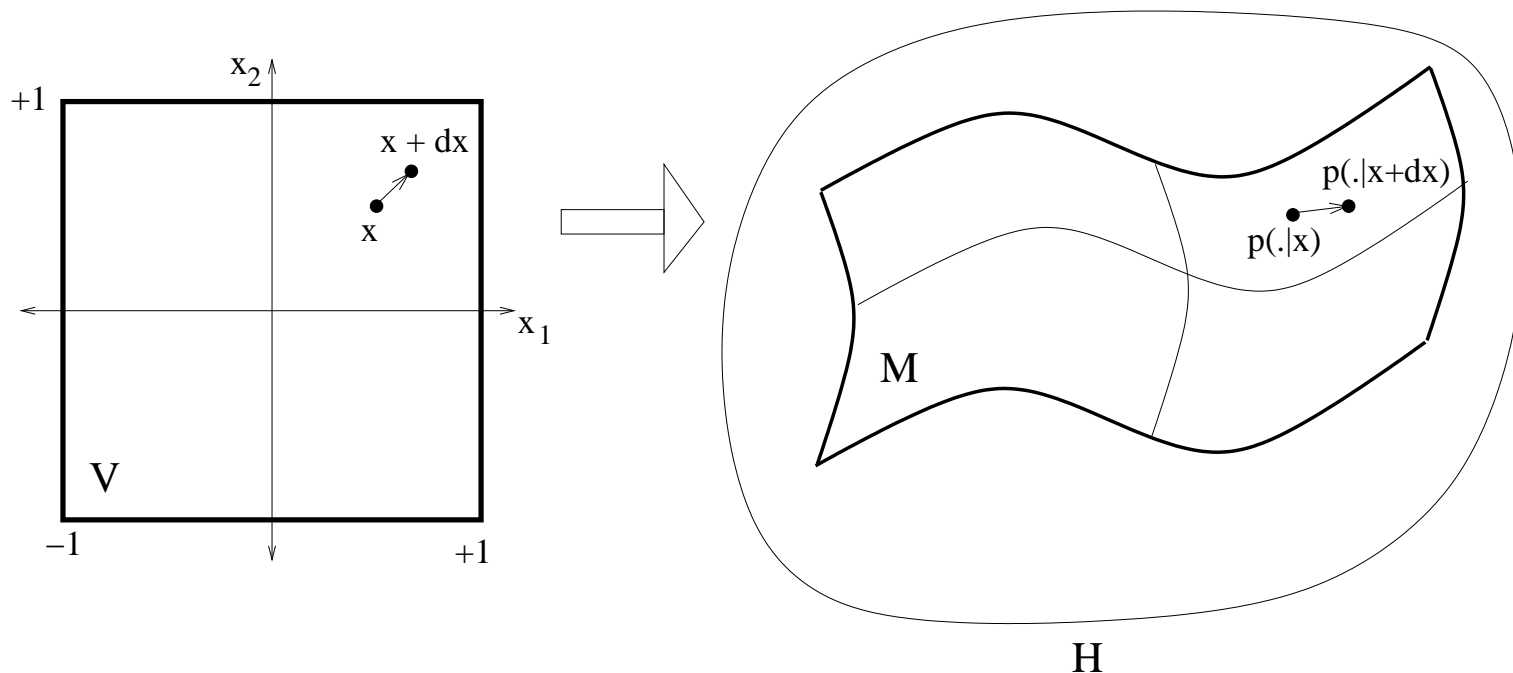
$$\begin{aligned}\boldsymbol{\pi}(\boldsymbol{x}) &= \{p(h_1 = k | \boldsymbol{x})\}_{k=1:K} \\ &= \{g_k(\mathbf{A}^{(\boldsymbol{\pi})} \boldsymbol{\phi}(\boldsymbol{x}))\}_{k=1:K}\end{aligned}$$

$$\begin{aligned}\mathbf{T}(\boldsymbol{x}) &= \{p(h_t = k | h_{t-1} = l, \boldsymbol{x})\}_{k,l=1:K} \\ &= \{g_k(\mathbf{A}^{(\mathbf{T}_l)} \boldsymbol{\phi}(\boldsymbol{x}))\}_{k,l=1:K}\end{aligned}$$

$$\begin{aligned}\mathbf{B}(\boldsymbol{x}) &= \{p(s_t^{(n)} = s | h_t = k, \boldsymbol{x})\}_{s=1:S, k=1:K} \\ &= \{g_s(\mathbf{A}^{(\mathbf{B}_k)} \boldsymbol{\phi}(\boldsymbol{x}))\}_{s=1:S, k=1:K}\end{aligned}$$



# Riemannian manifold of HMM



2-dim manifold  $\mathcal{M}$  of local noise models (HMMs)  $p(\cdot|x)$  parameterized by the latent space through a smooth non-linear mapping.

$\mathcal{M}$  is embedded in manifold  $\mathcal{H}$  of all noise models of the same form.

# Riemannian metric

---

Latent coordinates  $\boldsymbol{x}$  are displaced to  $\boldsymbol{x} + d\boldsymbol{x}$ .

How different are the corresponding noise models (HMMs)?

Need to answer this in a parameterization-free manner...

Local Kullback-Leibler divergence can be estimated by

$$D[p(\boldsymbol{s}|\boldsymbol{x})||p(\boldsymbol{s}|\boldsymbol{x} + d\boldsymbol{x})] \approx d\boldsymbol{x}^T J(\boldsymbol{x})d\boldsymbol{x},$$

where  $J(\boldsymbol{x})$  is the Fisher Information Matrix

$$J_{i,j}(\boldsymbol{x}) = -E_{p(\boldsymbol{s}|\boldsymbol{x})} \left[ \frac{\partial^2 \log p(\boldsymbol{s}|\boldsymbol{x})}{\partial x_i \partial x_j} \right]$$

that acts like a metric tensor on the Riemannian manifold  $\mathcal{M}$

# LTHMM - Fisher Information Matrix

---

HMM is itself a latent variable model.  
 $J(\boldsymbol{x})$  cannot be analytically determined.

There are several approximation schemes and an efficient algorithm for calculating the **observed** Fisher Information Matrix.

# Induced metric in data space

---

Structured data types - careful with the notion of a metric in the data space.

LTHMM naturally induces a metric in the structured data space.

Two data items (sequences) are considered to be close (or similar) if both of them are well-explained by the same underlying noise model (e.g. HMM) from the 2-dimensional manifold of noise models.

Distance between structured data items is implicitly defined by the local noise models that drive topographic map formation.

If the noise model changes, the perception of what kind of data items are considered similar changes as well.

# LTHMM - training

---

Constrained mixture of HMMs is fitted by **Maximum likelihood** using an E-M algorithm

Two types of hidden variables:

- **which HMM generated which sequence**  
(responsibility calculations is in mixture models)
- **within a HMM**, what is the **state sequence** responsible for generating the observed sequence  
(forward-backward-like calculations)

# Two illustrative examples

---

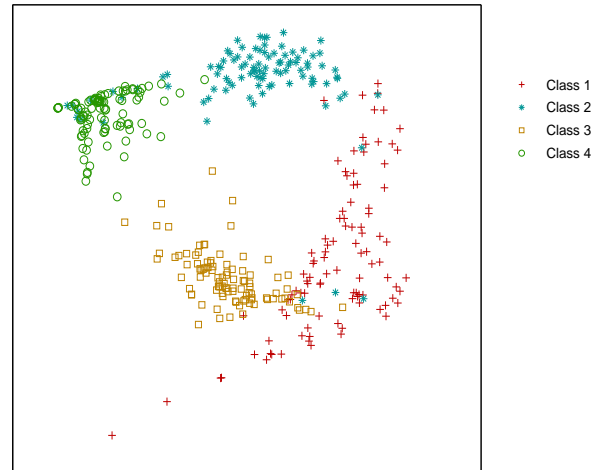
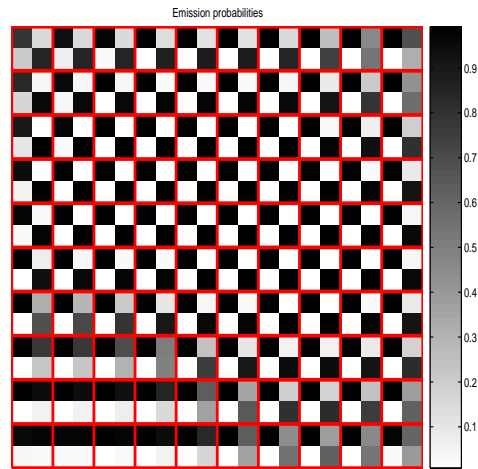
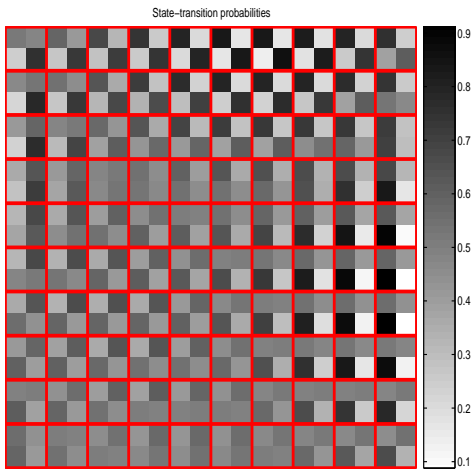
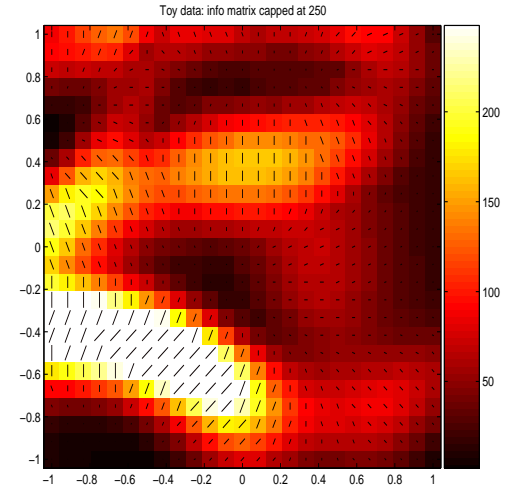
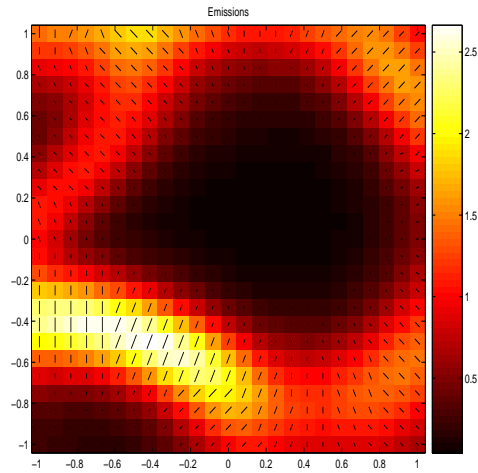
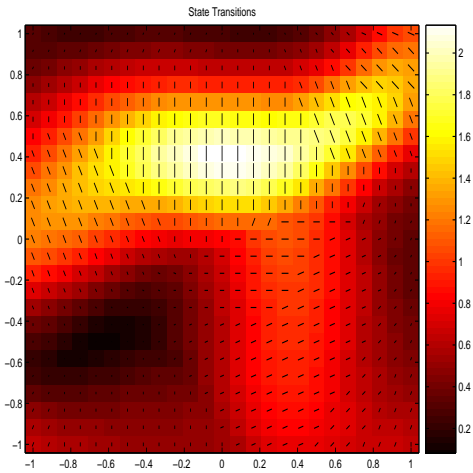
## ■ Toy Data

400 binary sequences of length 40 generated from 4 HMMs (2 hidden states) with identical emission structure (the HMMs differed only in transition probabilities). Each of the 4 HMMs generated 100 sequences.

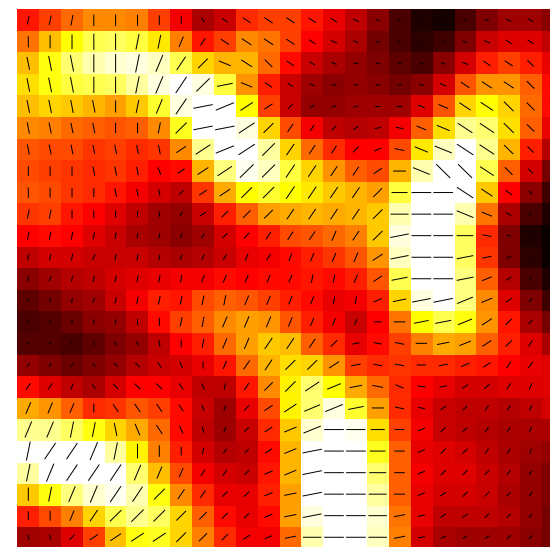
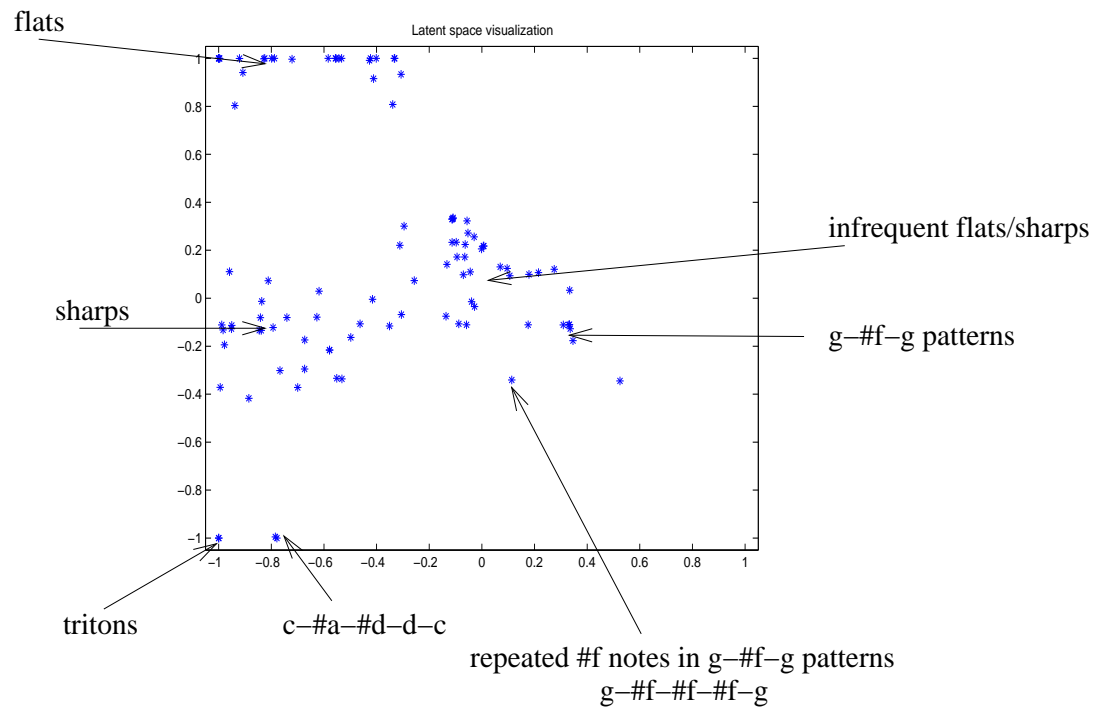
## ■ Melodic Lines of Chorals by J.S. Bach

100 chorales. Pitches are represented in the space of one octave, i.e. the observation symbol space consists of 12 different pitch values.

# Toy data

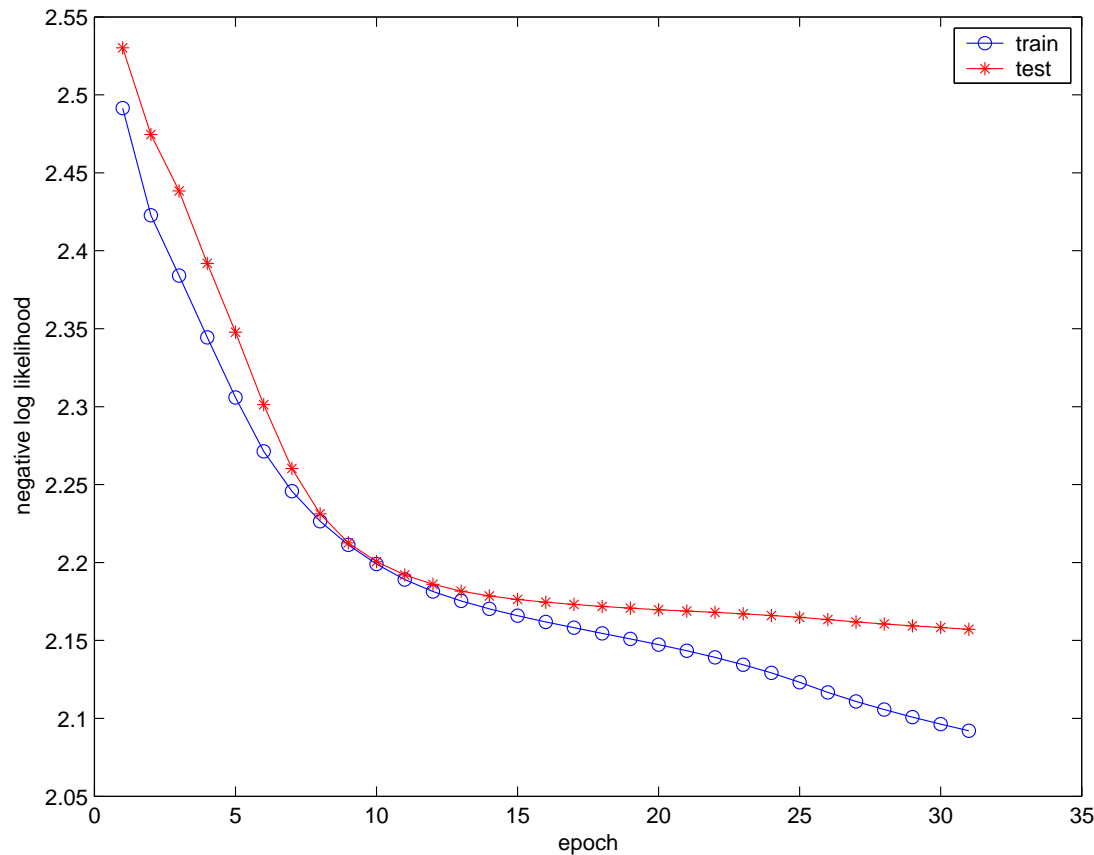


# Bach chorals





# Topographic formulation regularizes the model



Evolution of negative log-likelihood per symbol measured on the training (o) and test (\*) sets (Bach chorals).

# Topographic organization of eclipsing binaries

---

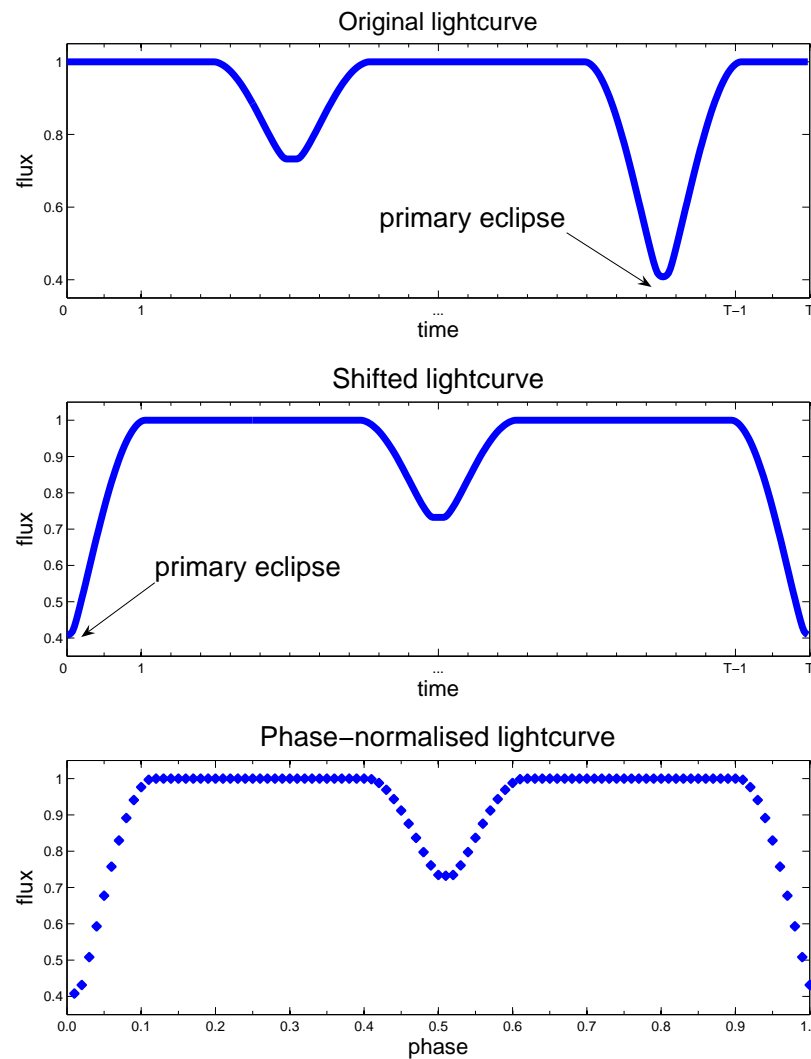
Line of sight of the observer is aligned with orbit plane of a two star system to such a degree that the component stars undergo mutual eclipses.

Even though the light of the component stars does not vary, eclipsing binaries are variable stars - this is because of the eclipses.

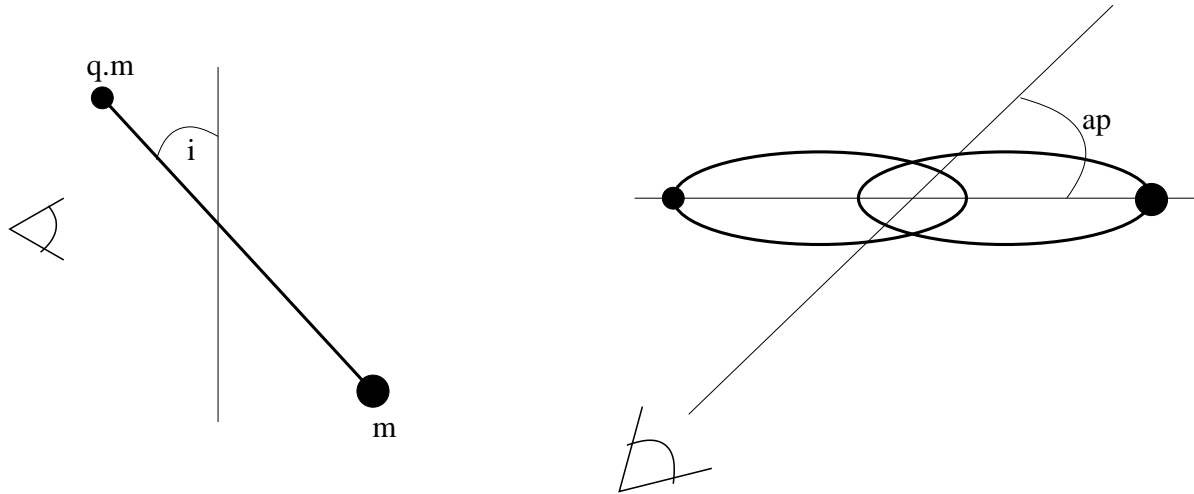
The light curve is characterized by periods of constant light with periodic drops in intensity.

If one of the stars is larger than the other (primary star), one will be obscured by a total eclipse while the other will be obscured by an annular eclipse.

# Eclipsing Binary Star - normalized flux



# Eclipsing Binary Star - the model



## Parameters:

Primary mass:  $m$  (1-100 solar mass)

mass ratio:  $q$  (0-1)

eccentricity:  $e$  (0-1)

inclination:  $i$  ( $0^\circ - 90^\circ$ )

argument of periastron:  $ap$  ( $0^\circ - 180^\circ$ )

log period:  $\pi$  (2-300 days)

# Empirical priors on parameters

---

$$p(m, q, e, i, ap, \pi) = p(m)p(q)p(\pi)p(e|\pi)p(i)p(ap)$$

Primary mass density:

$$p(m) = a \times m^b$$

$$a = \begin{cases} 0.6865, & \text{if } 0.5 \times M_{sun} \leq m \leq 1.0 \times M_{sun} \\ 0.6865, & \text{if } 1.0 \times M_{sun} < m \leq 10.0 \times M_{sun} \\ 3.9, & \text{if } 10.0 \times M_{sun} < m \leq 100.0 \times M_{sun} \end{cases}$$

$$b = \begin{cases} -1.4, & \text{if } 0.5 \times M_{sun} \leq m \leq 1.0 \times M_{sun} \\ -2.5, & \text{if } 1.0 \times M_{sun} < m \leq 10.0 \times M_{sun} \\ -3.3, & \text{if } 10.0 \times M_{sun} < m \leq 100.0 \times M_{sun} \end{cases}$$

# Empirical priors on parameters

---

## Mass ratio density

$$p(q) = p_1(q) + p_2(q) + p_3(q)$$

where

$$p_i(q) = A_i \times \exp\left(-0.5 \frac{(q - q_i)^2}{s_i^2}\right)$$

with

$$A_1 = 1.30, A_2 = 1.40, A_3 = 2.35$$

$$q_1 = 0.30, q_2 = 0.65, q_3 = 1.00$$

$$s_1 = 0.18, s_2 = 0.05, s_3 = 0.10$$

## log-period density

$$p(\pi) = \begin{cases} 1.93337\pi^3 + 5.7420\pi^2 - 1.33152\pi + 2.5205, & \text{if } \pi \leq \log_{10}18 \\ 19.0372\pi - 5.6276, & \text{if } \log_{10}18 < \pi \leq \log_{10}300 \end{cases}$$

etc.

# Flux GTM

---

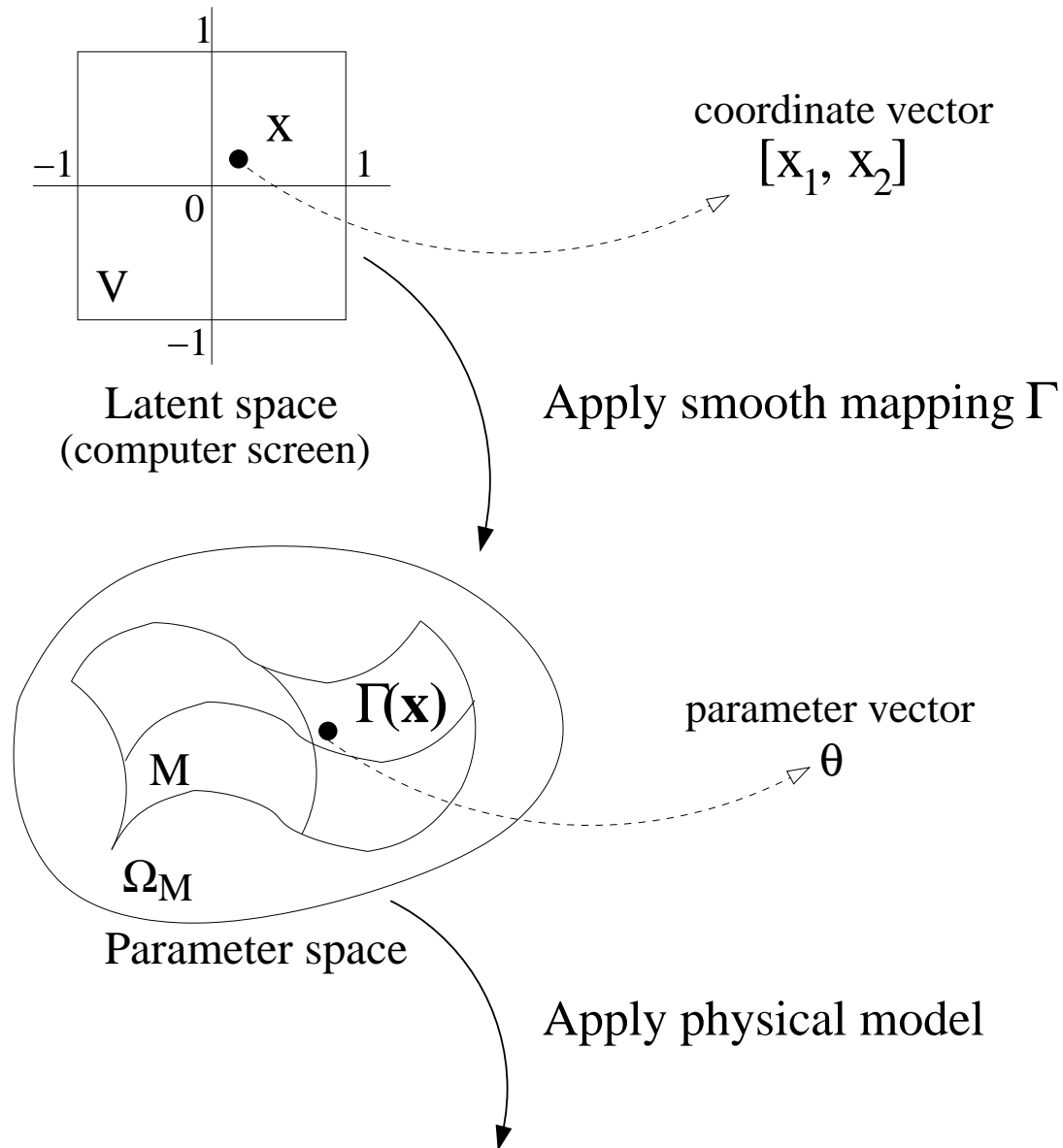
Smooth parameterized mapping  $F$  from 2-dim latent space into the space where 6 parameters of the eclipsing binary star model live.

Model light curves are contaminated by an additive observational noise (Gaussian). This gives a local noise model in the (time,flux)-space.

Each point on the computer screen corresponds to a local noise model and "represents" observed eclipsing binary star light curves that are well explained by the local model.

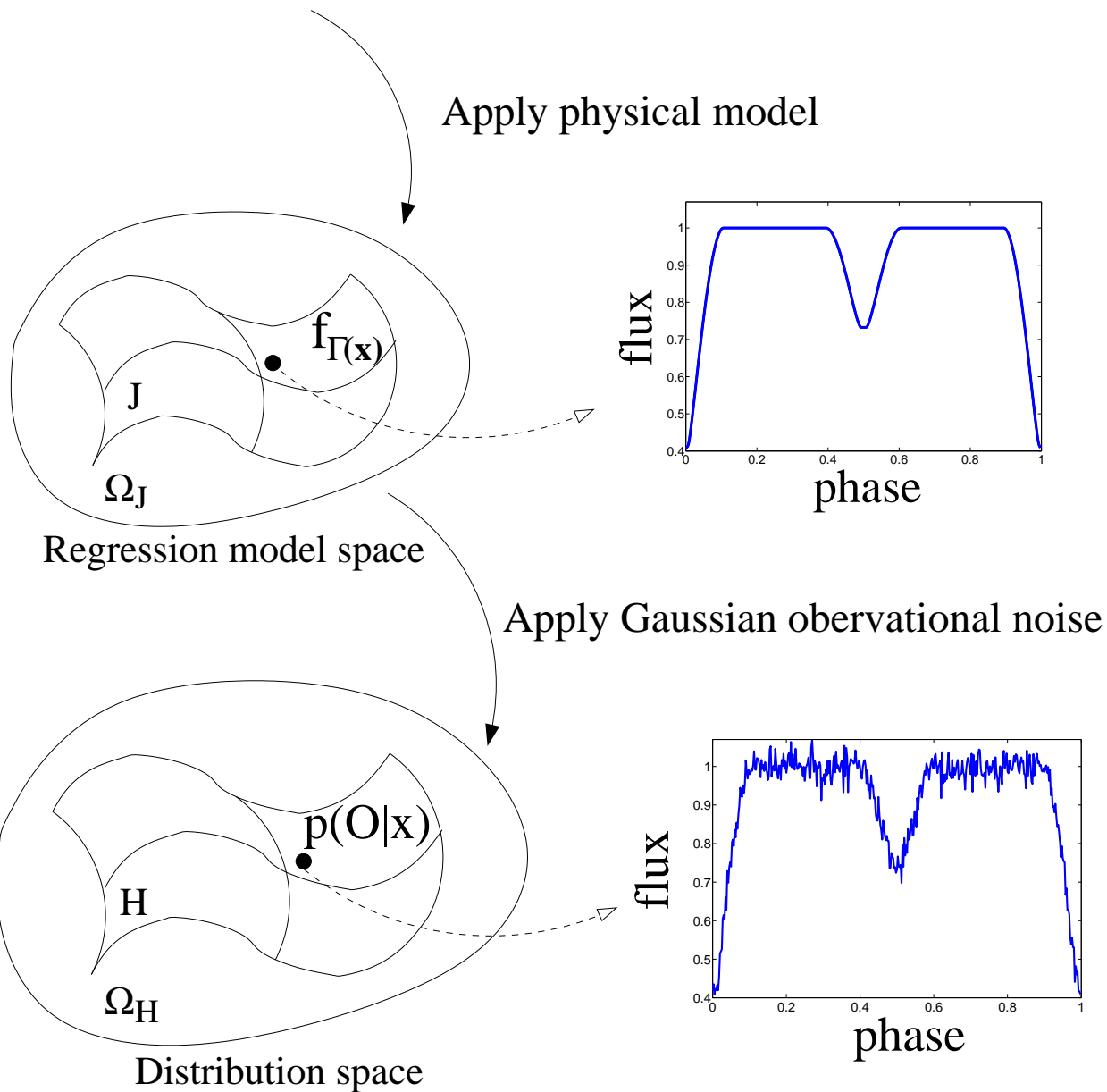
MAP estimation of the mapping  $F$  via E-M.

# Outline of the model (1)



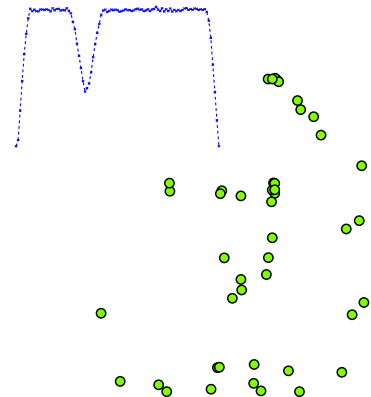
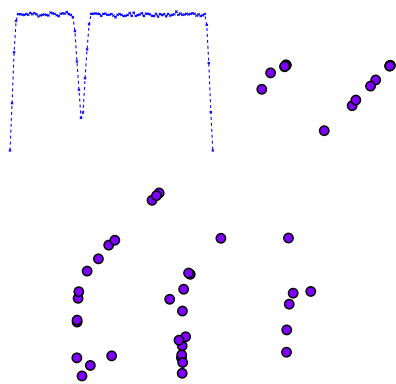
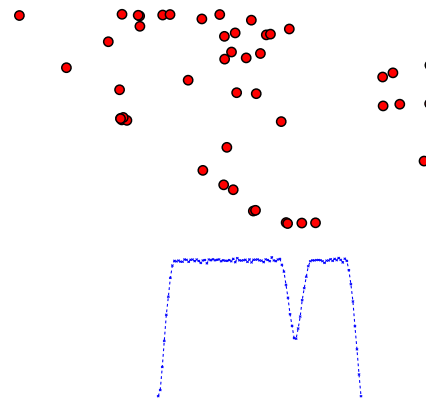
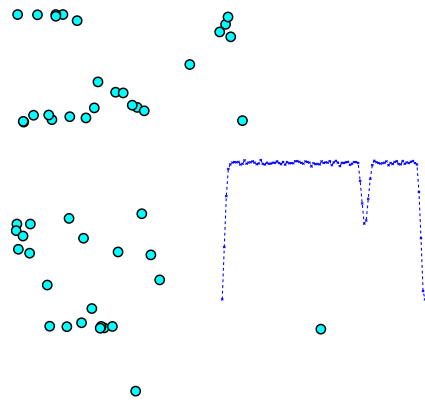


# Outline of the model (2)



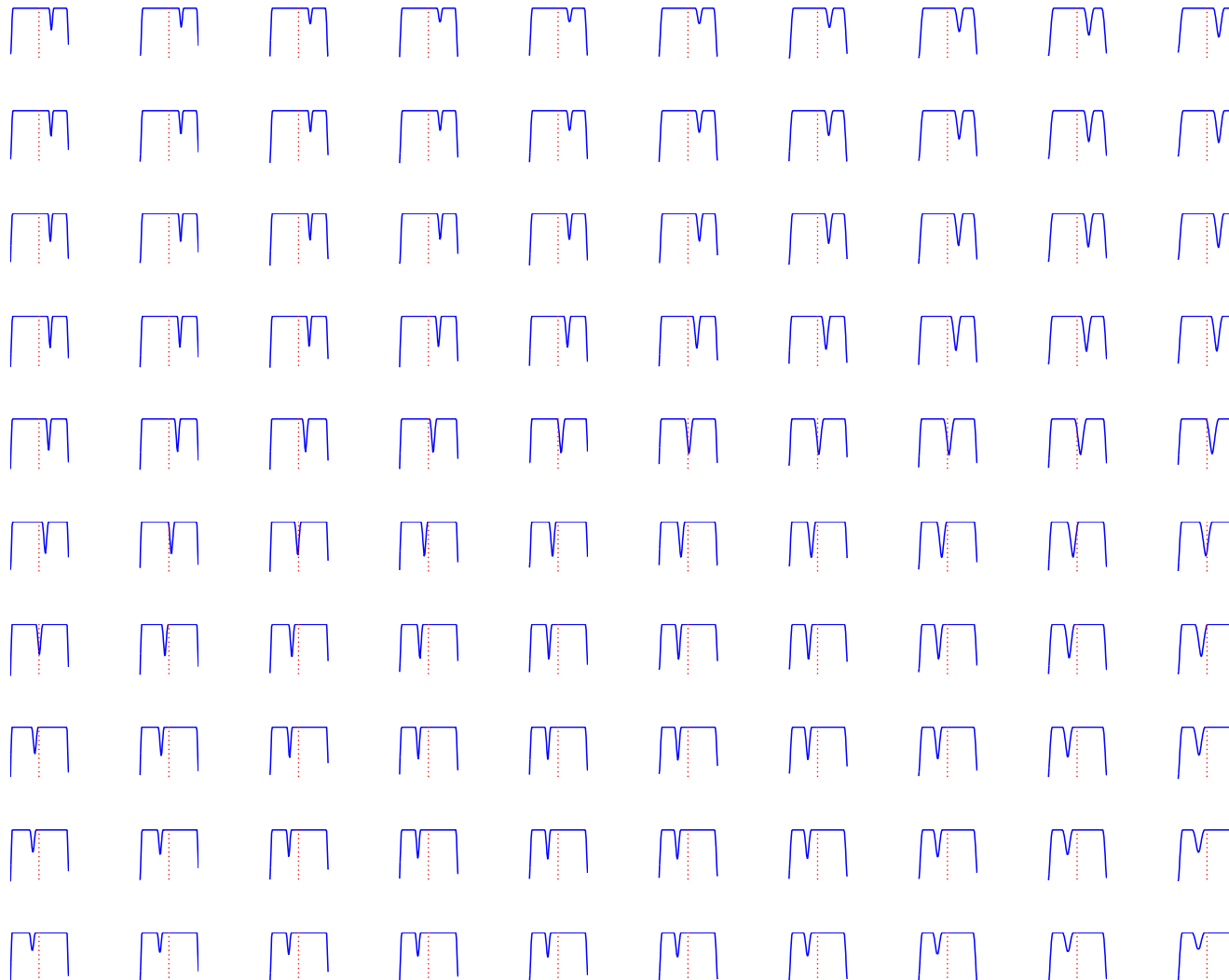
# Artificial fluxes - projections

---

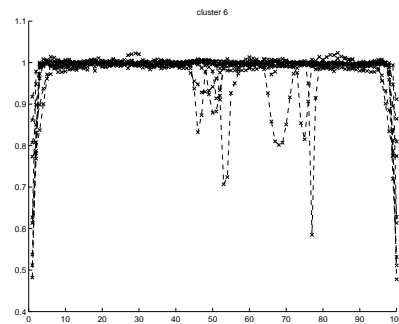
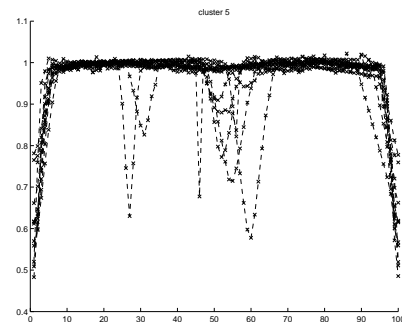
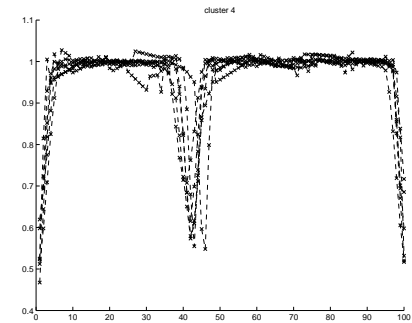
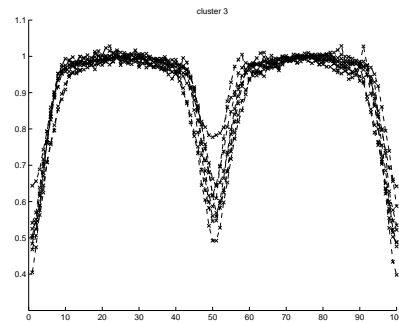
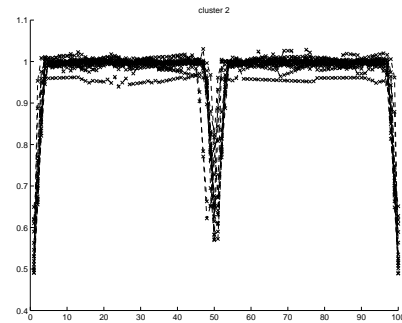
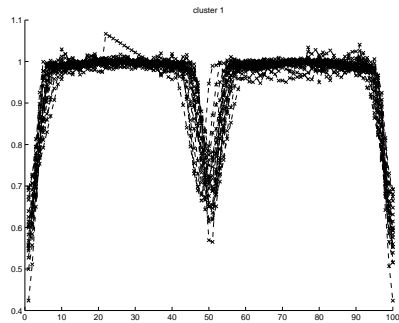


# Artificial fluxes - model

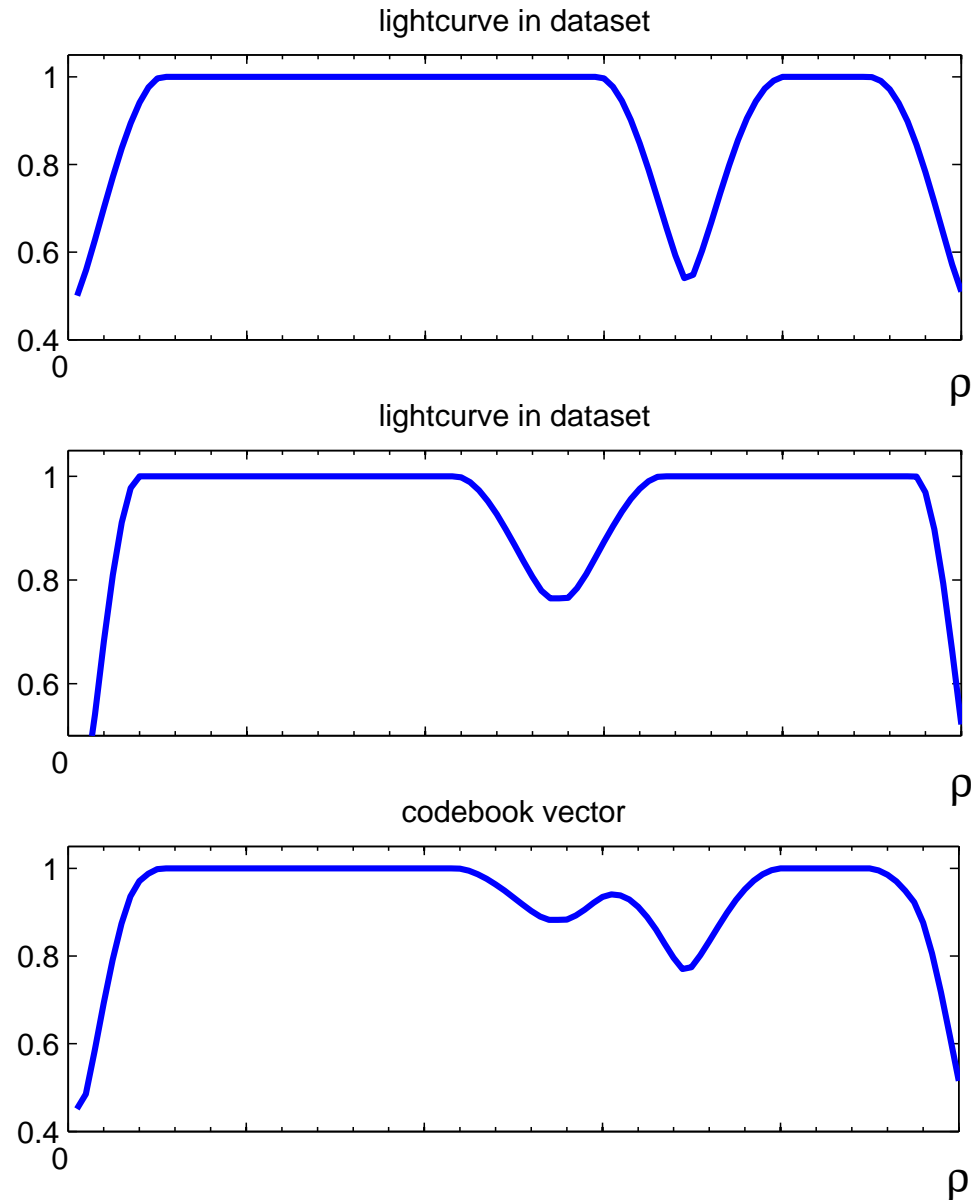
---



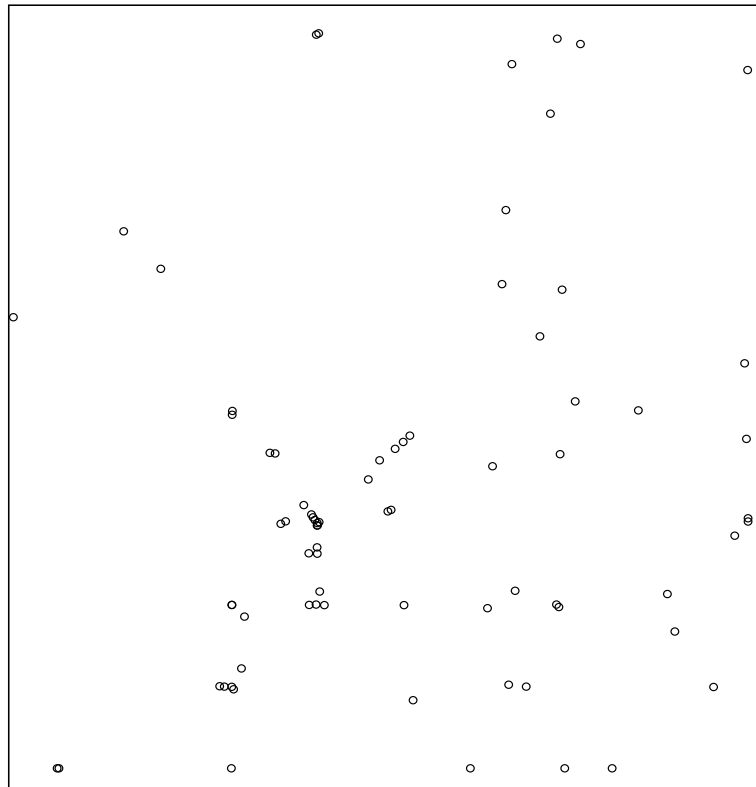
# Real fluxes - clustering mode



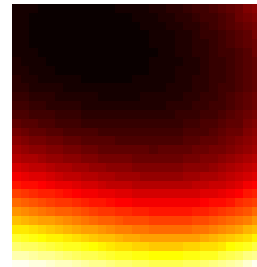
# Off-the-shelf methods may produce nonsense!



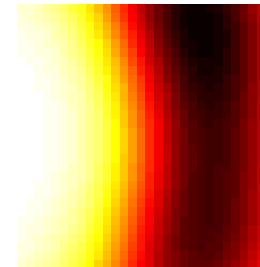
# Real fluxes - projections + model



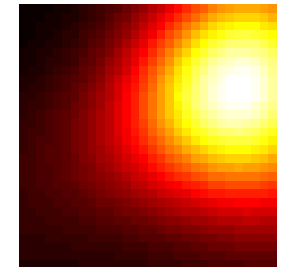
Primary mass



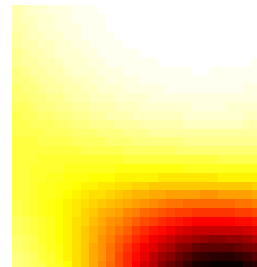
Mass ratio



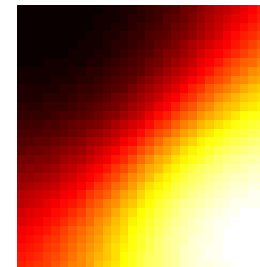
Eccentricity



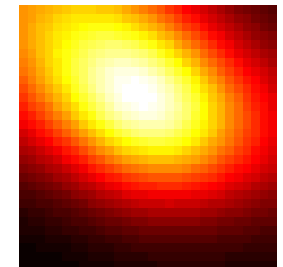
Inclination



Argument



Period



# 3-color cDNA microarrays

---

- Traditional dual-color cDNA microarrays – 2 different fluorescence dyes corresponding to two samples (e.g. “normal” and “disease”).
- 3-color cDNA microarrays – 3rd dye associated with yet another sample hybridized to a single microarray.
- Assess effects of a drug – assay hybridizing three samples: **normal** (dyed red), **disease** (dyed green), **drug-treated** (dyed blue).
- Intensities **R**, **G** and **B** reflect expression levels of the genes in the normal (healthy), disease and drug-treated samples, respectively.

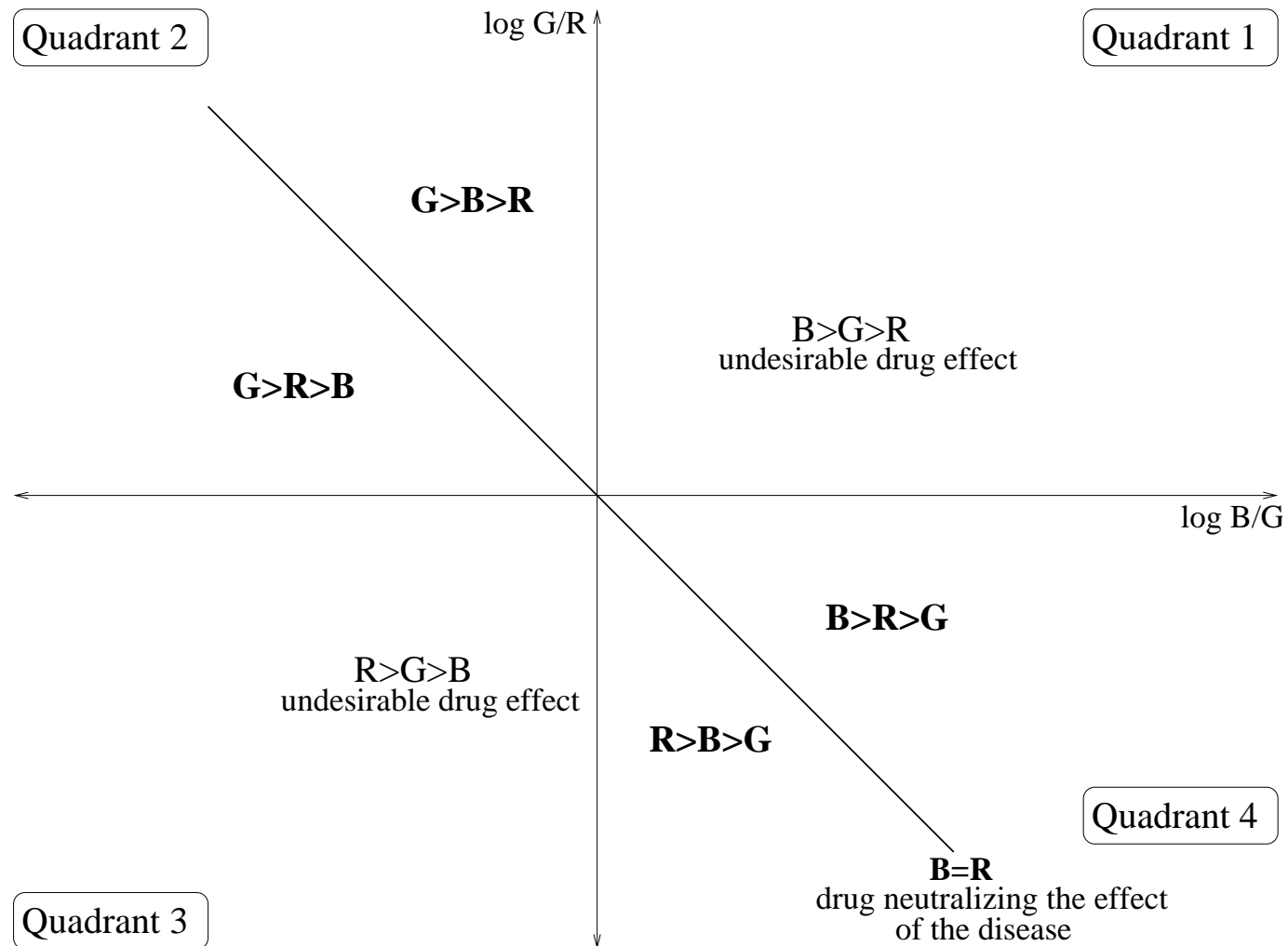
# hexaMplot [Zhao06]

---

- 2-dim representation of R, G and B intensities suited for assessing the drug effect on essayed genes
- hexaMplot coordinates: log ratios of intensity pairs  
 $x_1 = \log_2 B/G$  and  $x_2 = \log_2 G/R$ .
- Genes in the upper and lower half-plane are up- and down-regulated, respectively, by the disease.
- Genes in the left and right half-plane are up- and down-regulated, respectively, by the drug treatment, compared with the disease sample.
- Slant axis  $x_2 = -x_1 \Rightarrow \log_2 B/R = 0$ . Expression levels of genes in the normal and drug-treated samples are the same.



# hexaMplot



# Assessing drug effects through hexaMplot

---

- Drug neutralizes the effect of the disease on the essayed genes – gene representations cluster around the slant axis.
- Deviations form the slant axis within the 4th and 2nd quadrants ( $x_1 > 0, x_2 < 0$  and  $x_1 < 0, x_2 > 0$ , respectively) – still represent drug effects in the right direction.
- Genes in 1st and 3rd quadrants ( $x_1, x_2 > 0$  and  $x_1, x_2 < 0$ , respectively) – undesirable effect of the drug: enhancing the up-regulation, or suppressing the down-regulation of the gene by the disease.

# Past work

---

[Zhao06 ] – correlation coefficient of hexaMplot gene representations calculated and assessed for statistical significance.

[Zhao07 ] – more involved analysis.

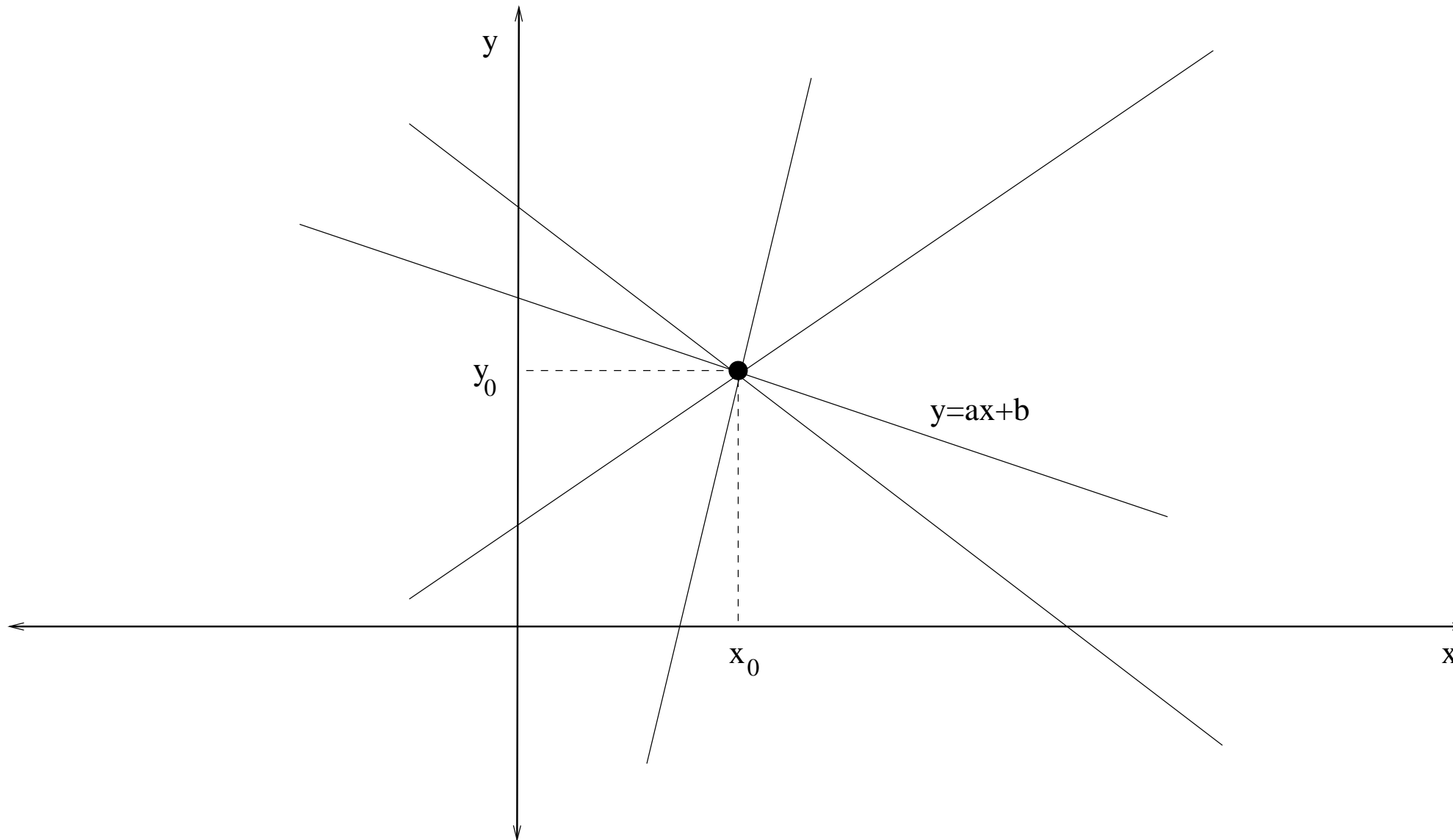
Detect groups of genes with similar expression patterns relative to the disease and the drug.

- Each such group is aligned along a **line ray** starting in the hexaMplot origin.
- **Direction** of the ray signifies whether the drug has positive or negative effect.
- **Angle** measures the drug effect level

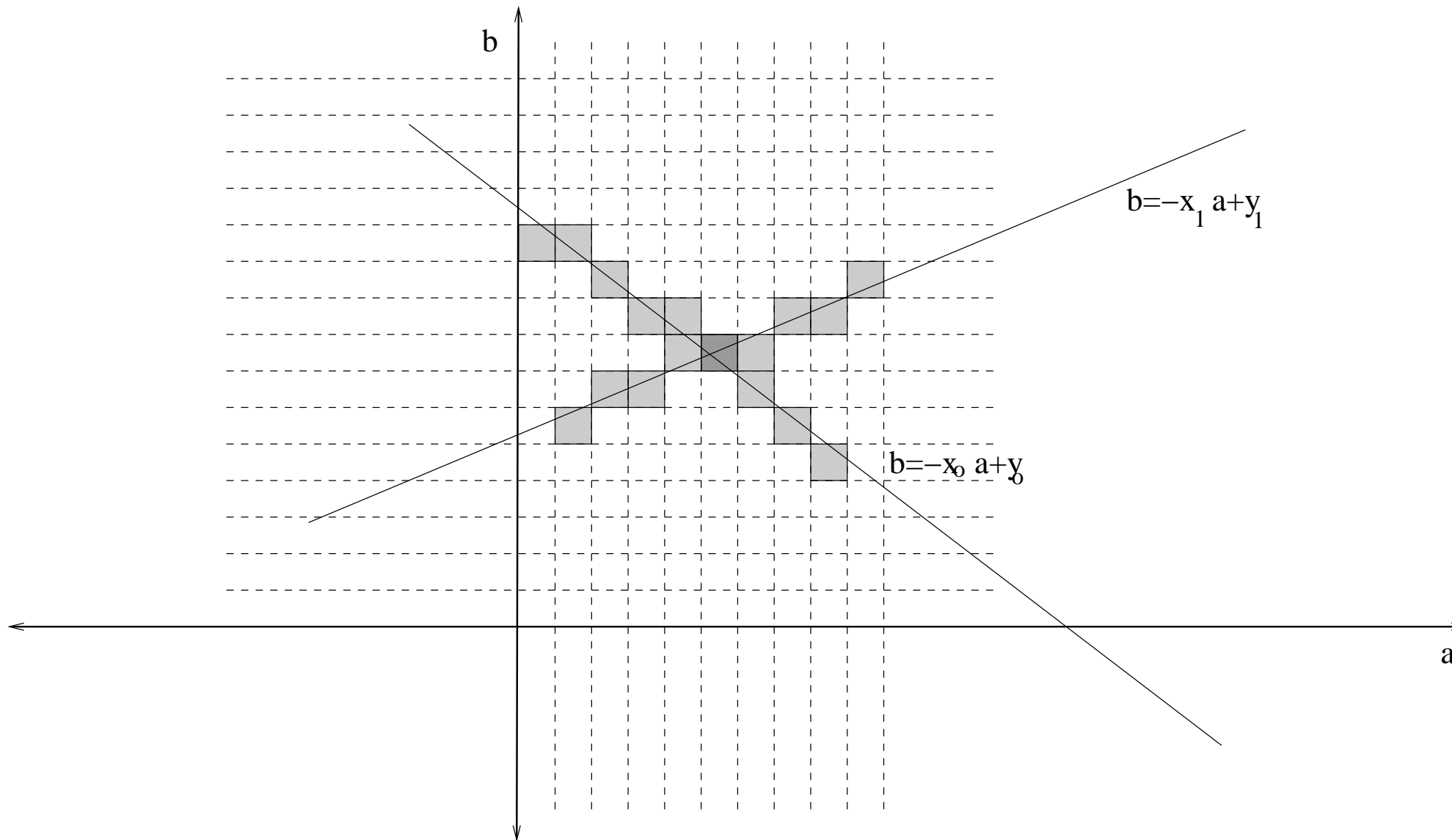
- The lines were detected through Hough Transform

# H Transform

---



# H Transform



# Problem 1

---

- HT applied to the differentially expressed genes only. Their detection done through fitting a global bi-variate Gaussian on hexaMplot gene representations and then applying a probability density threshold.
  - “Hard” separation of genes into equally vs. differentially expressed genes is not optimal – typically there will be a high density of genes around the separating confidence ellipse.
  - Results can be sensitive to the particular choice of the confidence value defining what is differentially expressed and what is not.

# Problem 2

---

- HT implicitly imposes a noise model that does not fit the nature of hexaMplot representations well.
  - Induced noise model depends on the line parametrization used.
  - $(x_1, x_2)$  hexaMplot representations are negatively correlated and there is no direct way of representing this fact in the standard HT.

# Problem 3

---

- Determination of the quantization level in the Hough space should reflect the amount of “measurement” noise in the hexaMplot features.

The quantization level determines the amount of smoothing in the Hough accumulator, which in turn has an effect on the number of distinct peaks (detected lines) in the Hough space.



# Problem 4

---

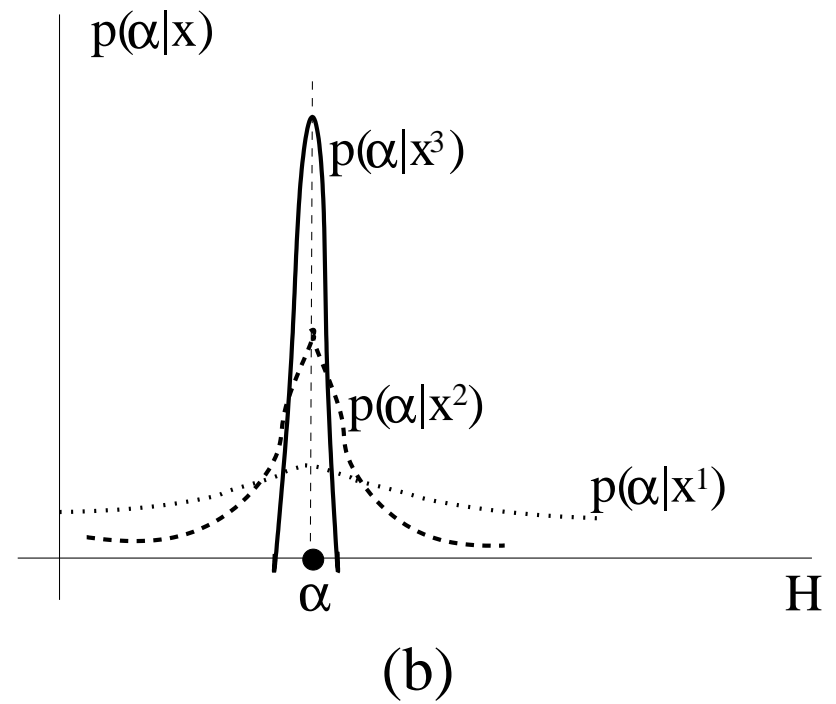
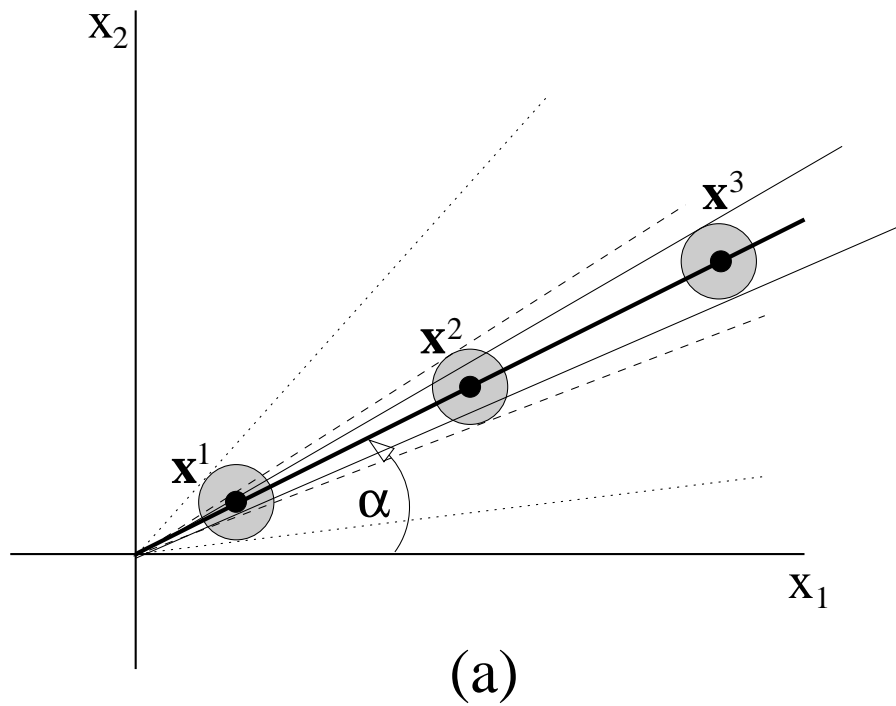
- Given a detected line, there is no principled way of quantifying the strength of association of the points with that line.

# Our proposal – probabilistic HT

---

- Address these shortcomings in the framework of a **principled probabilistic model based formulation**.
- **All essayed genes are considered**. The weaker and stronger contribution of equally and differentially expressed genes is obtained naturally in a “soft” manner from the **probabilistic formulation of the model behind the hexaMplot**.
- The model **explicitly takes into account the size and the negatively correlated nature of the noise** associated with hexaMplot gene representations.
- Both the **strength of association of individual genes with a particular group** (line ray in hexaMplot) and the **support for the group by the selected genes** can be **quantified in a principled manner** through posterior probabilities over the line angles, given the observations.

# Probabilistic HT



# The model

---

- A line ray in  $\mathbb{R}^2$  (hexaMplot space) starting in the origin at an angle  $\alpha \in [-\pi/4, 7\pi/4)$ .
- Bi-variate zero-mean Gaussian measurement noise with covariance matrix  $\Sigma_X$ . The density of possible measurements  $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$  corresponding to the point  $(r \cos \alpha, r \sin \alpha)$  on the line is given by

$$p(\mathbf{x}|\alpha, r) = \frac{1}{2\pi|\Sigma_X|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}^T - (r \cos \alpha, r \sin \alpha)) \Sigma_X^{-1} (\mathbf{x} - (r \cos \alpha, r \sin \alpha)^T) \right\},$$

where  $r > 0$  is the (Euclidean) distance of the point on the line from the origin.

# The model - contd'

---

- Prior knowledge about the parameter values  $(\alpha, r) \in [-\pi/4, 7\pi/4) \times [0, \infty)$ , summarized in the form of a prior distribution  $p(\alpha, r)$ .
- Given an observation  $\mathbf{x}$ , the induced uncertainty in the parameter space is given by the posterior

$$p(\alpha, r|\mathbf{x}) = \frac{p(\mathbf{x}|\alpha, r) p(\alpha, r)}{\int_{[-\pi/4, 7\pi/4) \times [0, \infty)} p(\mathbf{x}|\alpha', r') p(\alpha', r') d\alpha' dr'}.$$

- To obtain the amount of support for the angle parameter  $\alpha$  given the observation  $\mathbf{x}$ , we integrate  $r$  from the posterior:

$$p(\alpha|\mathbf{x}) = \int_{[0, \infty)} p(\alpha, r|\mathbf{x}) dr.$$

## The model - contd'

---

- Given a set of observations  $\mathcal{D} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$ ,  $\mathbf{x}^i \in \mathbb{R}^2$ ,  $i = 1, 2, \dots, N$ , accumulate the evidence contributions in the Hough space  $\mathcal{H}$

$$H(\alpha; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N p(\alpha | \mathbf{x}^i).$$

- Given that a line candidate with inclination angle  $\alpha$  has been detected by inspecting the peaks of the Hough accumulator  $H(\alpha; \mathcal{D})$ , one can ask which points from  $\mathcal{D}$  are strongly associated with it. Consult the posteriors  $p(\alpha | \mathbf{x}^i)$ ,  $i = 1, 2, \dots, N$ , and select points above some threshold value  $\theta$ .
- To enhance the threshold interpretability, we discretized the angle space  $\mathcal{H}$  into a regular grid  $G = \{\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_M\}$  and turned the densities  $p(\alpha | \mathbf{x})$  into probabilities  $P(\tilde{\alpha}_j | \mathbf{x})$  over the  $G$ .

## The model - contd'

---

- Calculate the probability threshold  $\theta \in (0, 1)$  as  $\theta = \kappa/M$ ,  $\kappa \in (0, M)$ , meaning that only observations with posteriors at least  $\kappa$  times greater than the uninformative distribution  $1/M$  will be considered.
- Given a probability threshold  $\theta$  and a (discretized) angle  $\tilde{\alpha}$ , the set of selected points that support the line ray  $\tilde{\alpha}$  reads:

$$S_{\theta}(\tilde{\alpha}) = \{\mathbf{x} \mid \mathbf{x} \in \mathcal{D}, P(\tilde{\alpha}|\mathbf{x}) \geq \theta\}.$$

- Check how much the set as a whole supports that line ray through the posterior

$$P(\tilde{\alpha}|S_{\theta}(\tilde{\alpha})) = \frac{p(S_{\theta}(\tilde{\alpha})|\tilde{\alpha}) P(\tilde{\alpha})}{\sum_{\tilde{\alpha}' \in G} p(S_{\theta}(\tilde{\alpha})|\tilde{\alpha}') P(\tilde{\alpha}')},$$

where  $P(\tilde{\alpha}')$  is the prior distribution over the grid  $G$ .

# The model - contd'

---

- Assuming independence of observations

$$\begin{aligned} p(S_\theta(\tilde{\alpha})|\tilde{\alpha}') &= \prod_{\mathbf{x} \in S_\theta(\tilde{\alpha})} p(\mathbf{x}|\tilde{\alpha}') \\ &= \prod_{\mathbf{x} \in S_\theta(\tilde{\alpha})} \int_0^\infty p(\mathbf{x}|r, \tilde{\alpha}') p(r|\tilde{\alpha}') dr. \end{aligned}$$

Here,  $p(\mathbf{x}|r, \tilde{\alpha}')$  is the noise model and  $p(r|\tilde{\alpha}')$  is the conditional prior on  $r$ .



# Noise model

---

- It is usual to assume that the log intensities are normally distributed.

$$\mathbf{x} = (x_1, x_2)^T = \left( \log \frac{B}{G}, \log \frac{G}{R} \right)^T .$$

- Consider 3 random variables (log intensities)  $Y_1$ ,  $Y_2$  and  $Y_3$  representing  $\log B$ ,  $\log G$  and  $\log R$ , respectively. **hexaMplot** representations  $(x_1, x_2)$  correspond to two random variables  $X_1 = Y_1 - Y_2$  and  $X_2 = Y_2 - Y_3$  coupled through  $Y_2$ .
- Even if we assume that the individual measurement errors of the three log intensities  $Y_1$ ,  $Y_2$  and  $Y_3$  are independent, the implied noise in the hexaMplot coordinates  $X_1$ ,  $X_2$  will be negatively correlated. This simply follows from that fact that while  $Y_2$  contributes negatively to  $X_1$ , its contribution to  $X_2$  is positive.

# Noise model

---

- Assuming that the measurement noise of the log intensity  $Y_i$  is a zero mean Gaussian with variance  $\sigma_i^2$ ,  $i = 1, 2, 3$ ,  $(X_1, X_2)$  will be Gaussian distributed with covariance matrix

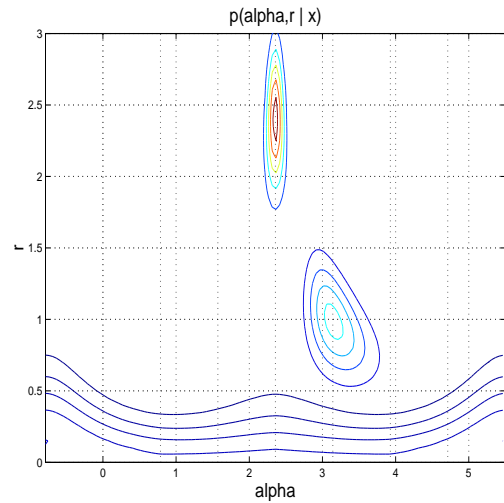
$$\Sigma_X = \begin{bmatrix} \sigma_1^2 + \sigma_2^2 & -\sigma_2^2 \\ -\sigma_2^2 & \sigma_2^2 + \sigma_3^2 \end{bmatrix}. \quad (1)$$

- We assume equal levels of measurement noise across the three colors,  $\sigma^2 = \sigma_1^2 = \sigma_2^2 = \sigma_3^2$ ,

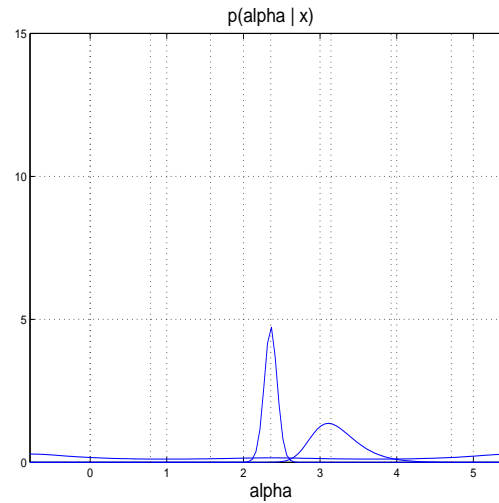
$$\Sigma_X = 2\sigma^2 \begin{bmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{bmatrix}. \quad (2)$$

# Toy example

$$\mathbf{x}^1 = (0.1, -0.1)^T, \mathbf{x}^2 = (-1, 0)^T \text{ and } \mathbf{x}^3 = (-1.75, 1.75)^T$$

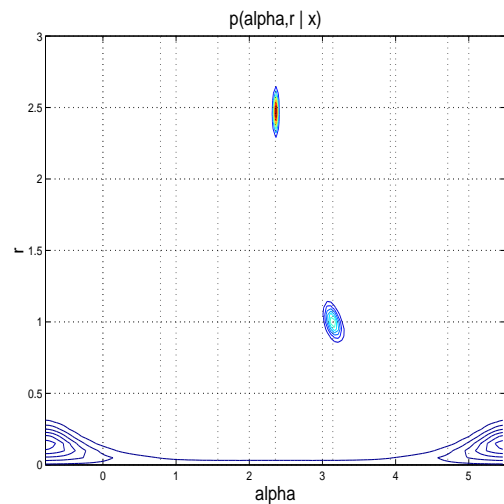


(a)

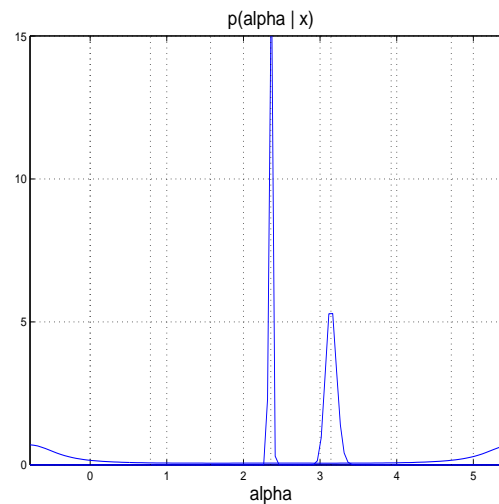


(b)

$\sigma=0.2$



(c)



(d)

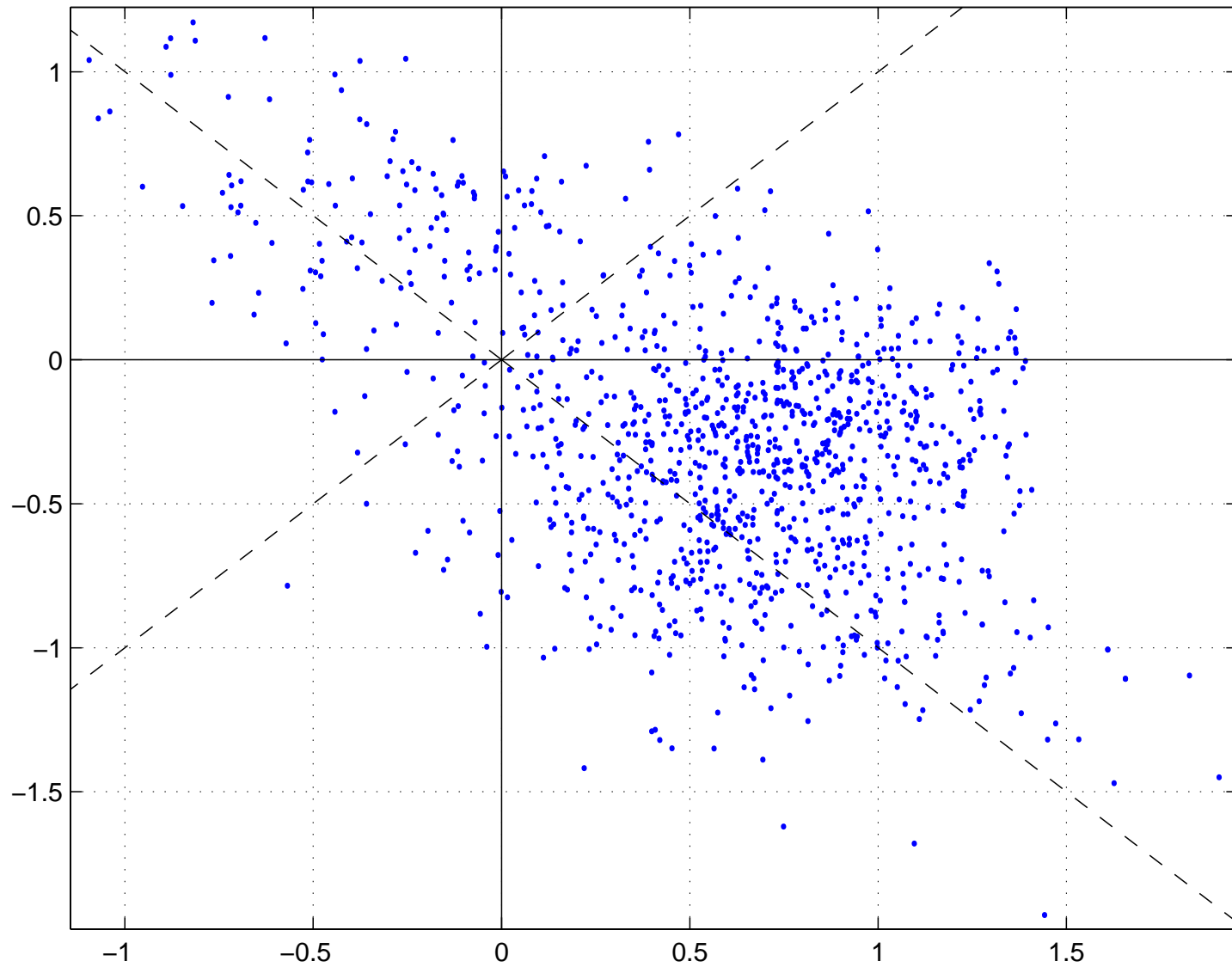
$\sigma=0.05$

# Real data

---

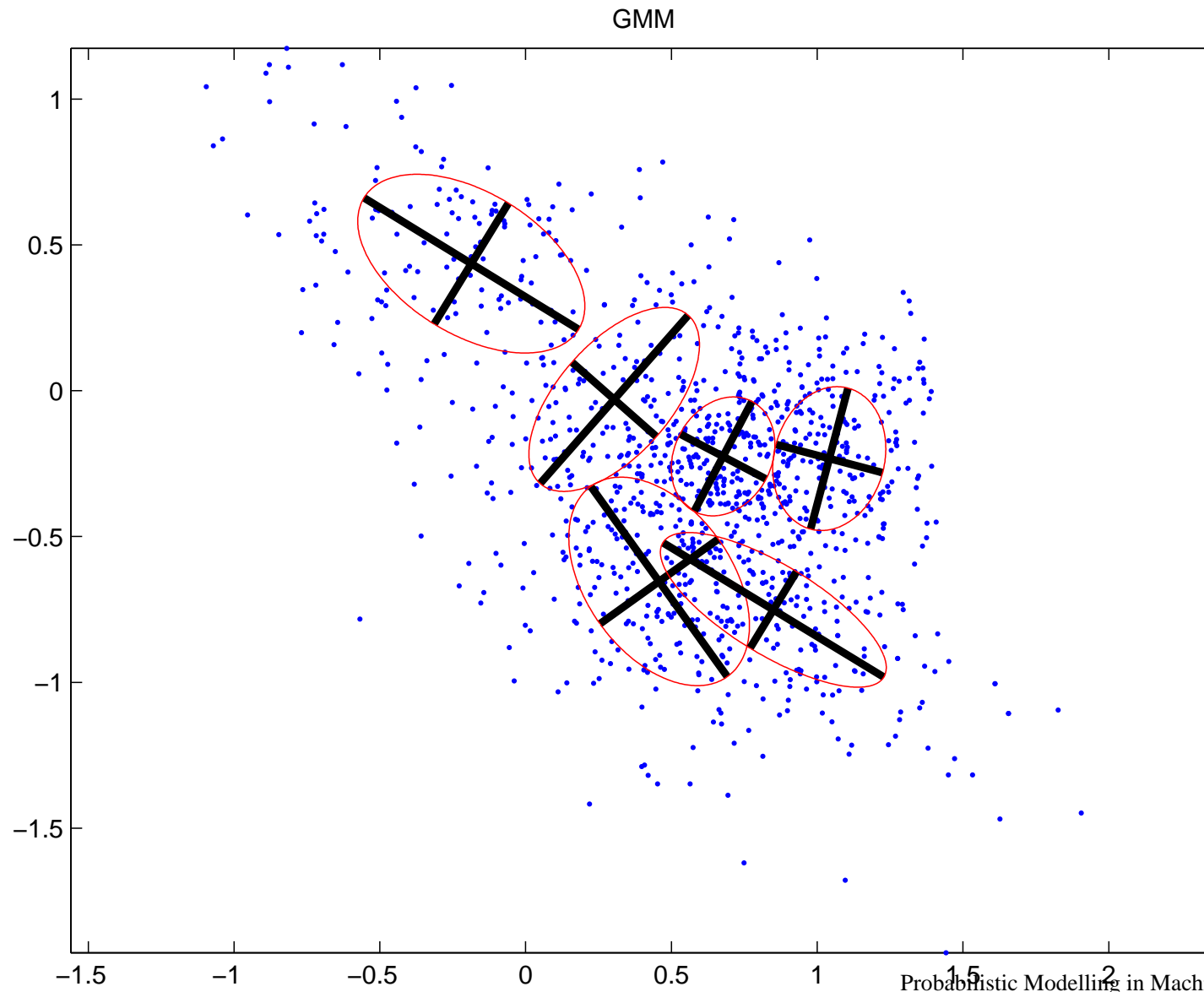
- Analyse action of the drug Rg1 (dominant compound of the extract of ginsenosides in ginseng) on homocysteine-treated human umbilical vein endothelial cells (HUVEC).
- 1128 genes assayed in four microarrays (4 repeats under the same experimental conditions).
- Usual data normalization.

# hexaMplot



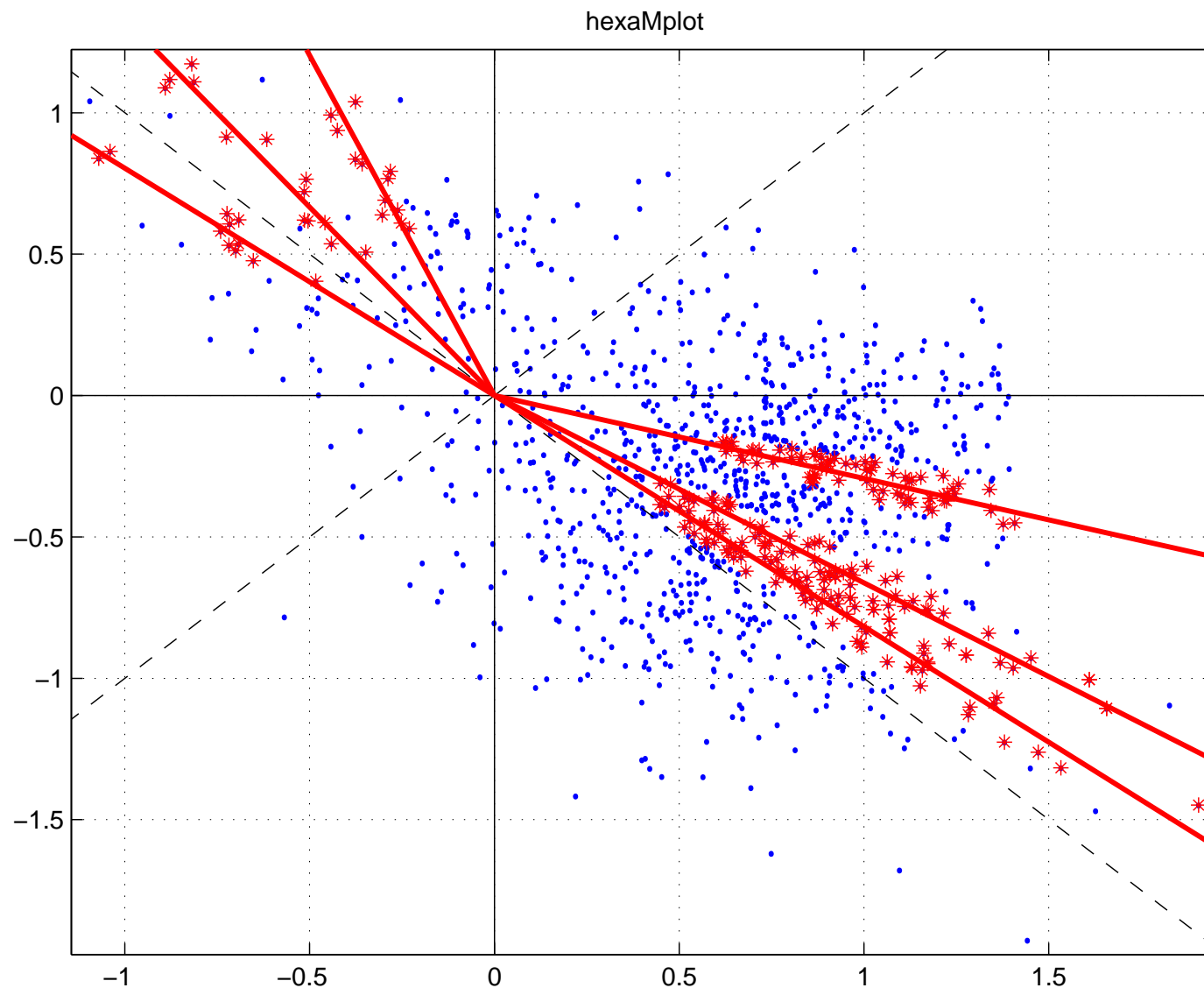
# Need a dedicated grouping mechanism..

GMM model + model evidence

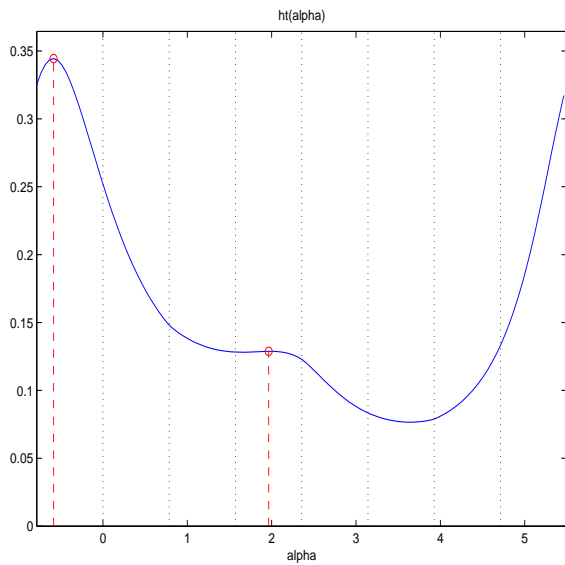


# Probabilistic HT - detected line rays

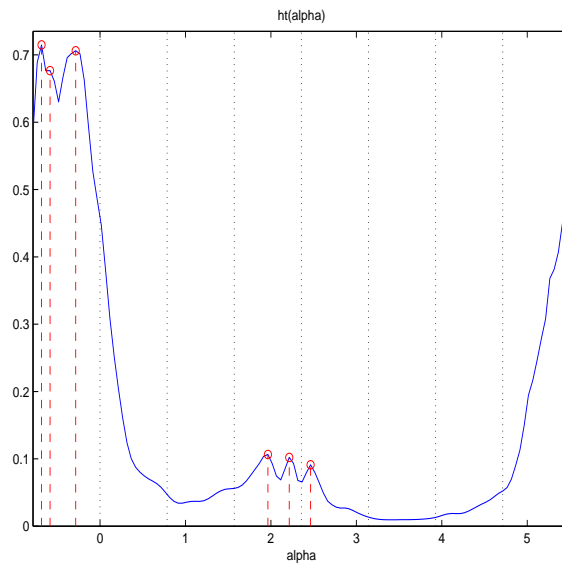
$\sigma = 0.05$ ,  $\mathcal{S}_\theta(\alpha)$  chosen using  $M = 126$  and  $\kappa = 25$



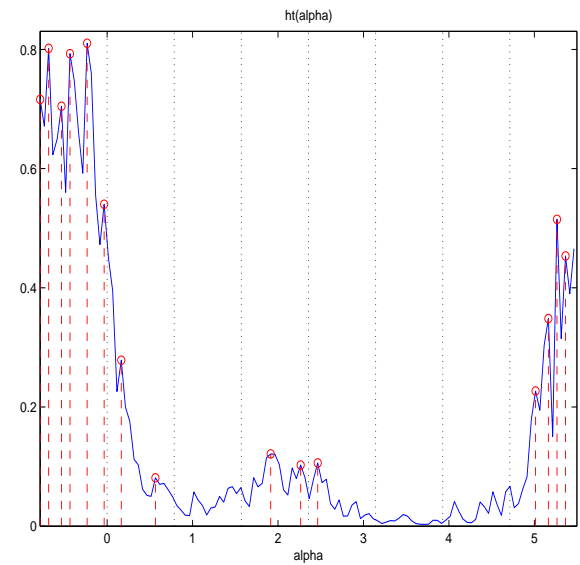
# Probabilistic HT - accumulator



(a)



(b)



(c)



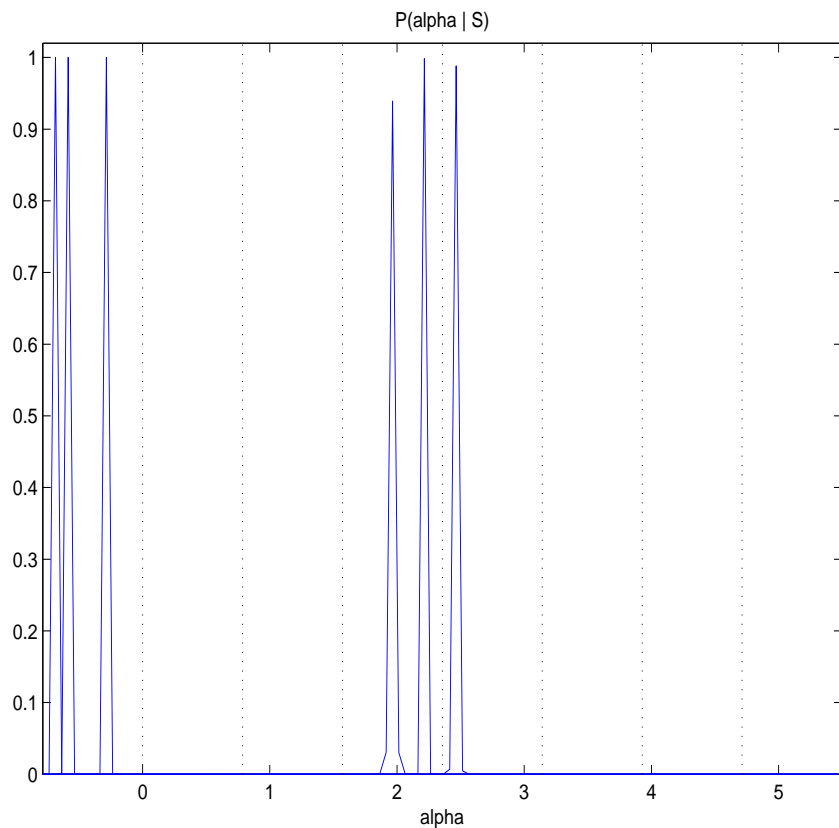
# Detected line rays

The shortest intervals  $(\tilde{\alpha}_-(\sigma), \tilde{\alpha}_+(\sigma))$  containing the estimated line angles and 95% of the posterior mass  $P(\cdot | S_\theta(\tilde{\alpha}))$  around them. The intervals are shown for three levels of observational noise:  $\sigma = 0.05$ ,  $\sigma = 0.3$  and  $\sigma = 1.0$ .

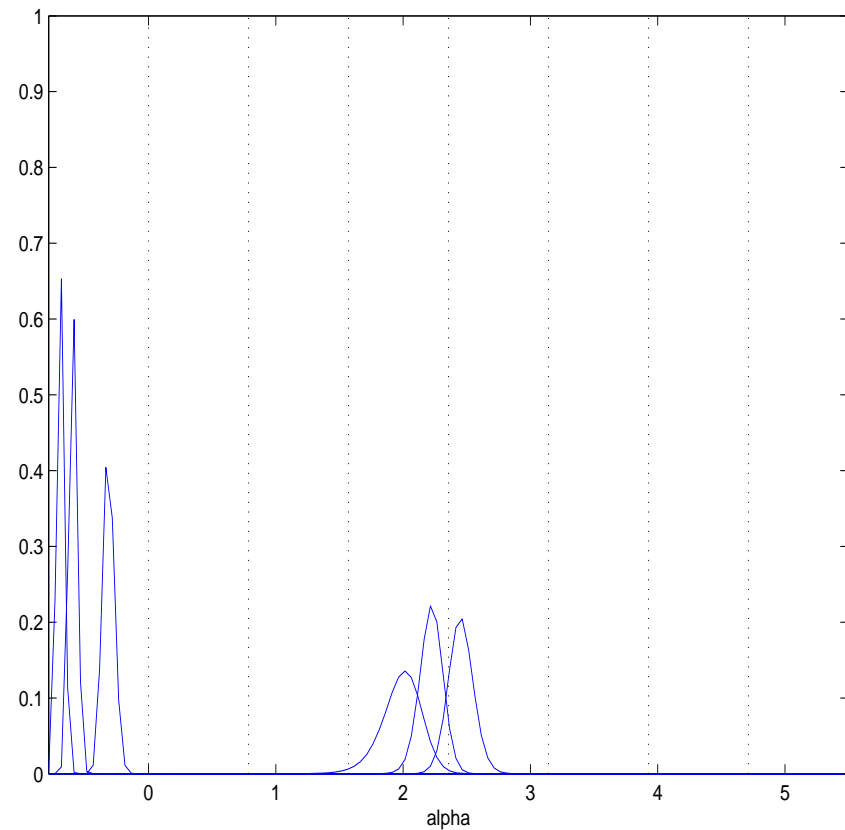
line	$\tilde{\alpha}$	$(\tilde{\alpha}_-(0.05), \tilde{\alpha}_+(0.05))$	$(\tilde{\alpha}_-(0.3), \tilde{\alpha}_+(0.3))$	$(\tilde{\alpha}_-(1.0), \tilde{\alpha}_+(1.0))$
1	-0.685	(-0.735,-0.684)	(-0.736, -0.683)	(-0.885,-0.535)
2	-0.585	( -0.635,-0.584)	(-0.635, -0.535)	(-0.785,-0.435)
3	-0.285	(-0.335,-0.284)	(-0.385, -0.235)	(-0.685,-0.235)
4	1.965	(1.914,1.966)	(1.665, 2.265)	(-0.035, 3.015)
5	2.215	(2.165,2.216)	(2.015, 2.365)	(1.565, 2.765)
6	2.465	(2.415,2.466)	(2.265, 2.615)	(1.865, 3.015)

# Support for the detected groups

Posteriors  $P(\alpha|S_\theta(\alpha))$  of the six detected lines (gene groups) for two levels of observational noise:  $\sigma = 0.05$  (a) and  $\sigma = 0.3$  (b).



(a)



(b)

# GO analysis

line	$ \mathcal{S}_\theta(\tilde{\alpha}) $	GO term ID	# genes	# genes from $\mathcal{S}_\theta(\tilde{\alpha})$	p-value
1	80	GO:0002675	87	37	0.00173
		GO:0002525	85	36	0.00242
		GO:0006953	85	36	0.00242
		GO:0002527	86	36	0.00322
		GO:0002543	86	36	0.00322
		GO:0002539	60	27	0.00441
		GO:0002540	60	27	0.00441
2	64	GO:0044424	176	41	0.00000
		GO:0044444	175	41	0.00000
		GO:0044446	168	39	0.00294
		GO:0030117	165	38	0.00437
		GO:0045265	162	37	0.00536
3	71	GO:0005488	178	51	0.00000
		GO:0050794	163	55	0.00295

# GO analysis

---

- The first three lines with angles in  $(-\pi/4, 0)$  represent genes with  $(R, G, B)$  intensities satisfying  $G < R < B$ .
- The disease decreases expression of a gene, compared with its normal expression level  $R$ , i.e.  $G < R$ . The drug eliminates this effect by overexpressing the genes,  $B > R$ .
- Genes in the group corresponding to the 1st line are related to acute inflammatory response (GO:0002675, GO:0002525) increasing for example the concentration of non-antibody proteins in the plasma (GO:0006953), or increasing the intra- or extra-cellular levels of prostaglandin (GO:0002539) and leukotriene (GO:0002540).

# GO analysis

---

- The 3rd line groups genes that are related to binding mechanisms (GO:0005488) and breakdown of neutral lipids (GO:0046461), membrane lipids (GO:0046466) and glycerolipids (GO:0046503).
- The disease also down-regulates genes related to pathways of the complement cascade which allow for the direct killing of microbes as well as regulation of other immune processes (GO:0001867, GO:0006957). The drug Rg1 corrects this situation by stimulating the pathways.

# GO analysis

---

- The 4th and 5th lines with angles in  $(\pi/2, 3\pi/2)$  represent genes with  $(R, G, B)$  intensities satisfying  $R < B < G$ .
- Compared with its normal expression level, the expression of a gene is increased by the disease ( $G > R$ ). The drug partially eliminates this effect by reducing the expression level to  $B$ , leaving  $B$  still above the normal expression  $R$ .
- The 6th line with  $\alpha \in (3\pi/2, \pi)$  groups genes with  $(R, G, B)$  intensities satisfying  $B < R < G$ .
- The disease causes increased expression of a gene ( $G > R$ ) and the drug compensates for this effect by driving the gene expression below the normal level ( $B < R$ ).
- While genes grouped together by the 4th line are associated with immune and chronic inflammatory response, the genes corresponding to the 5th and 6th lines are again related to cellular components and mechanisms effected by the disease.