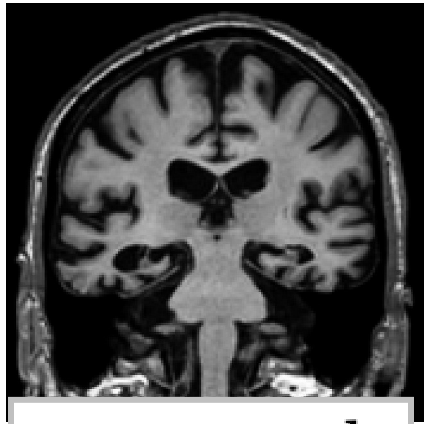
Model-based Segmentation: Part II

Aalto-yliopisto
Aalto-universitetet
Aalto University

Medical Image Analysis
Koen Van Leemput
Fall 2023

The problem to be solved



MRI image $\mathbf{d}$

Label image $\mathbf{l}$

N voxels

$$\mathbf{d} = (d_1, \ldots, d_N)^{\mathrm{T}}$$

d_n: intensity in voxel n

$$\mathbf{l} = (l_1, \ldots, l_N)^{\mathrm{T}}$$

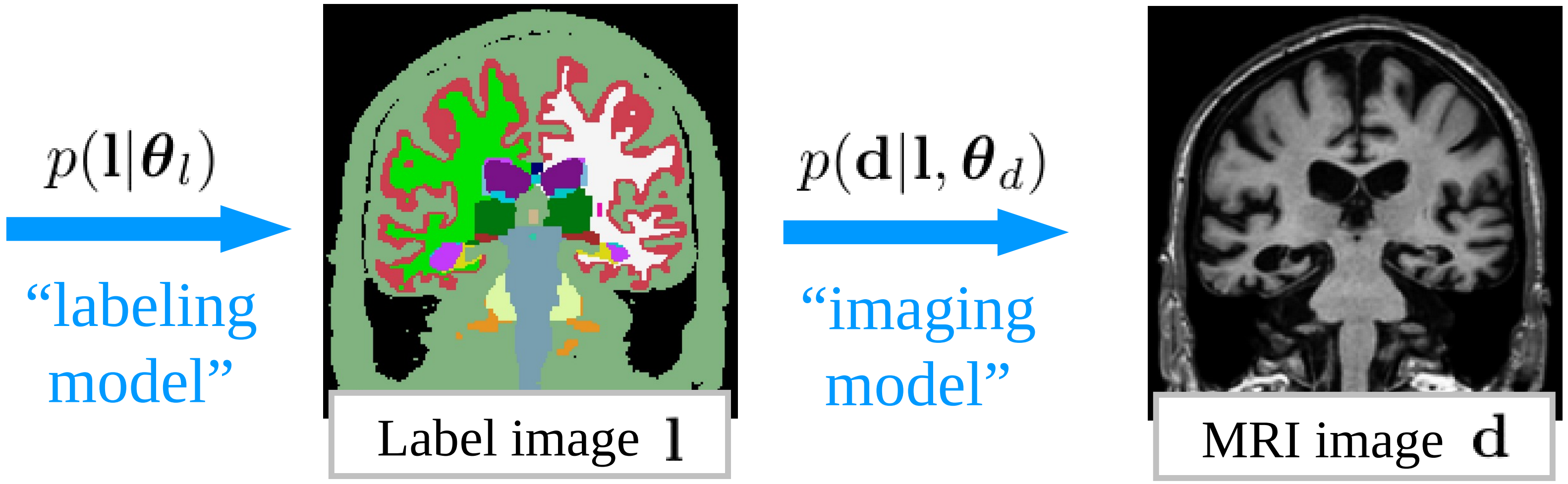
$$l_n \in \{1, \ldots, K\}$$

K: number of classes

Aalto-yliopisto
Aalto-universitetet
Aalto University

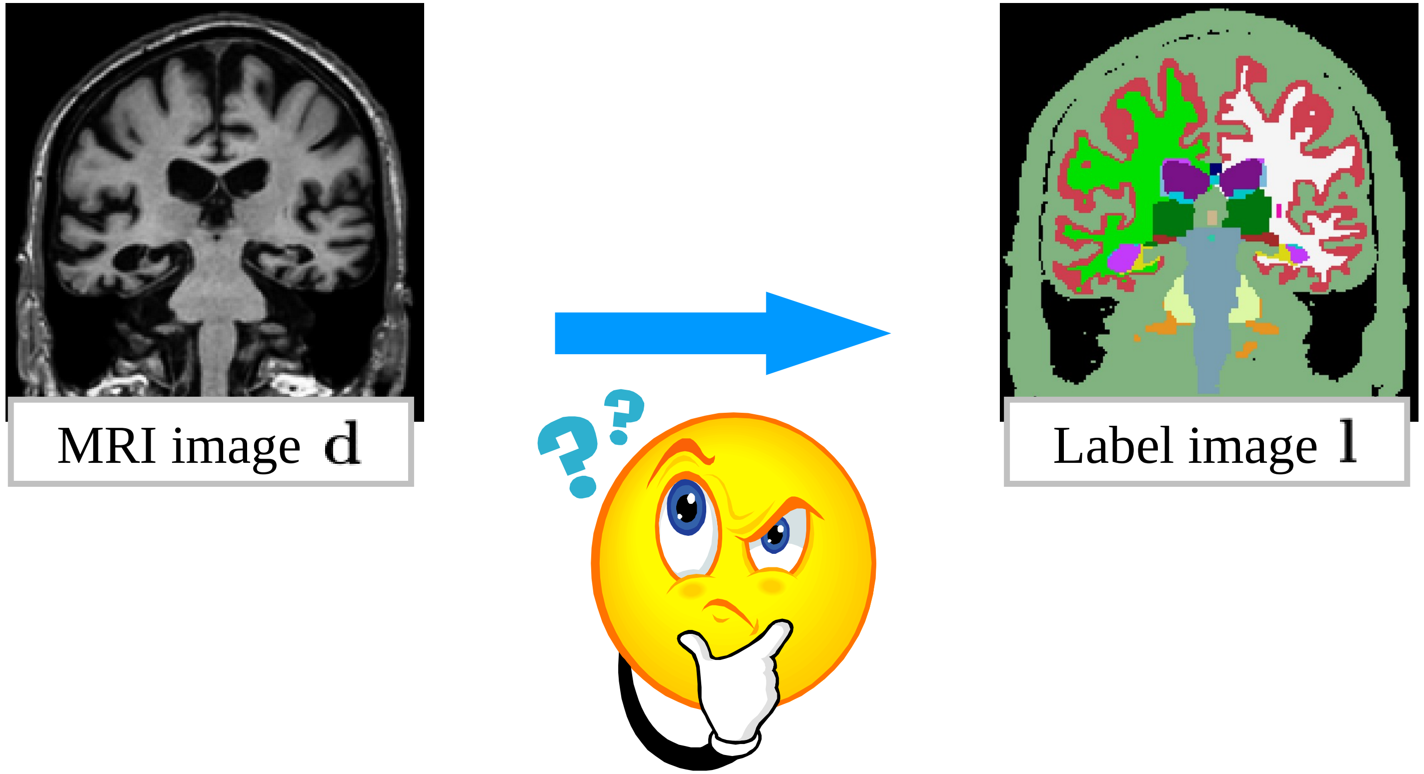
One solution: generative modeling

– Formulate a statistical model of image formation



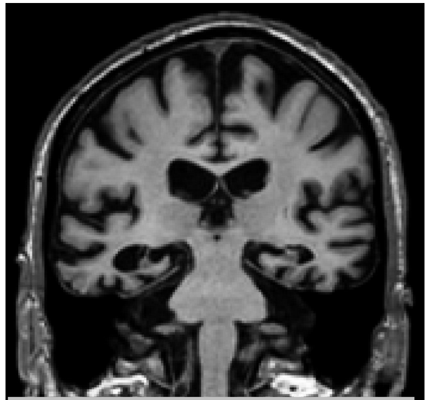
– The model depends on some parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_l^{\mathrm{T}}, \boldsymbol{\theta}_d^{\mathrm{T}})^{\mathrm{T}}$

Segmentation = inverse problem

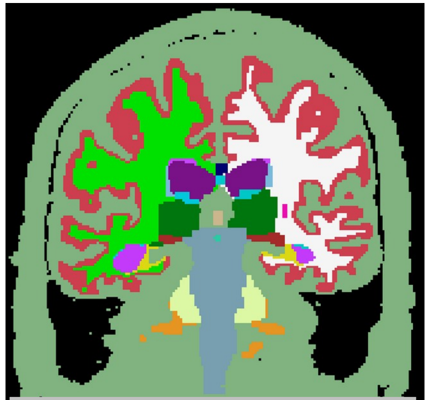


Aalto-yliopisto
Aalto-universitetet
Aalto University

Segmentation = inverse problem



MRI image $\mathbf{d}$



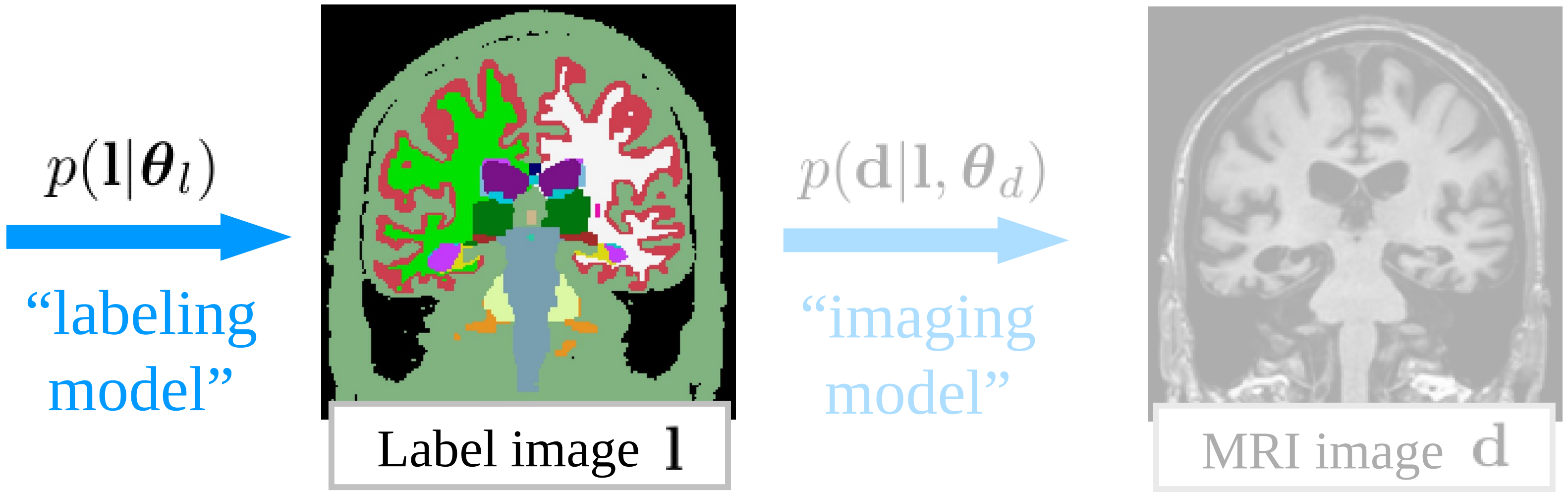
Label image $\mathbf{l}$

$$\hat{\mathbf{l}} = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \boldsymbol{\theta})$$

Bayesian inference

- Play with the mathematical rules of probability

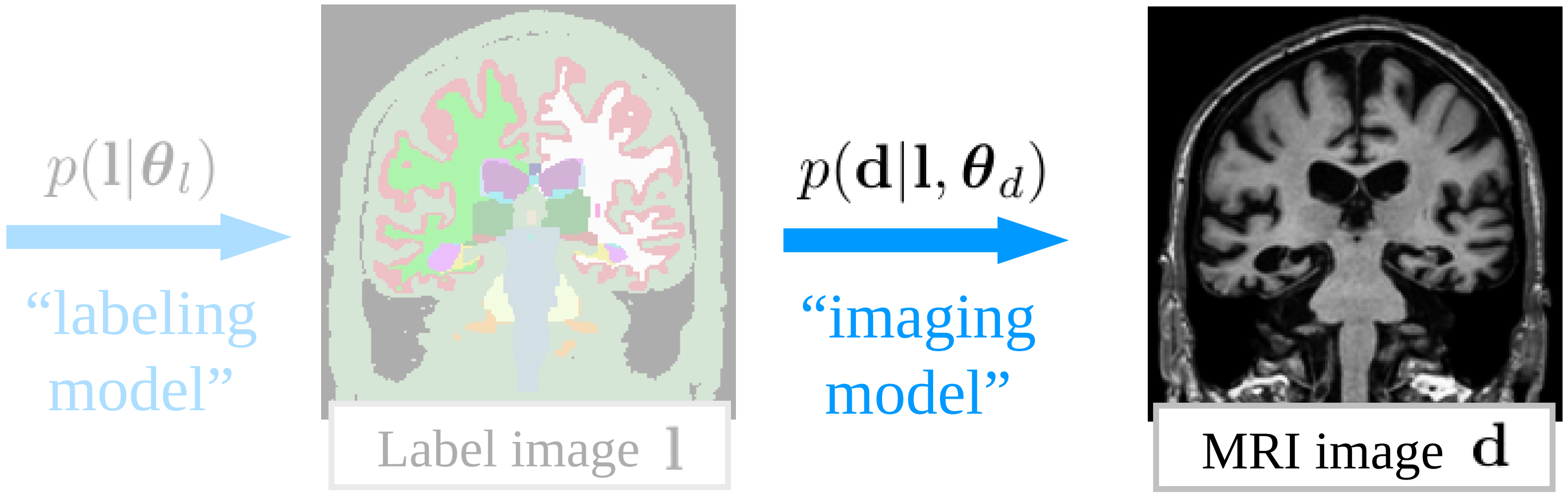
Gaussian mixture model



- Assign a label to each voxel independently
- Probability of assigning label k is π_k

$$p(\mathbf{l}|\boldsymbol{\theta}_l) = \prod_n \pi_{l_n} \;, \qquad \boldsymbol{\theta}_l = (\pi_1, \ldots, \pi_K)^{\mathrm{T}}$$

Gaussian mixture model

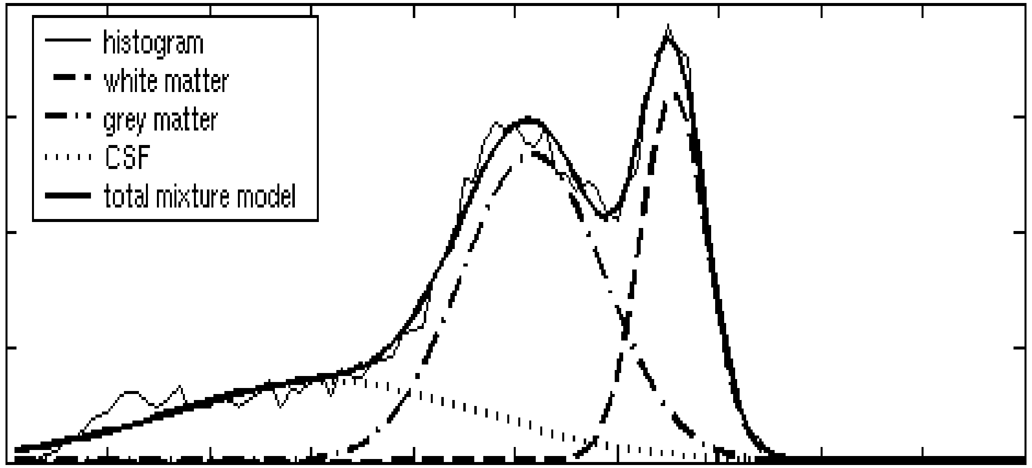


– Draw the intensity in each voxel with label k from a Gaussian distribution with mean μ_k and variance σ_k^2

$$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d) = \prod_n \mathcal{N}(d_n|\mu_{l_n}, \sigma_{l_n}^2), \qquad \boldsymbol{\theta}_d = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots \sigma_K^2)^{\mathrm{T}}$$

$$\mathcal{N}(d|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(d-\mu)^2}{2\sigma^2}\right]$$

Gaussian mixture model

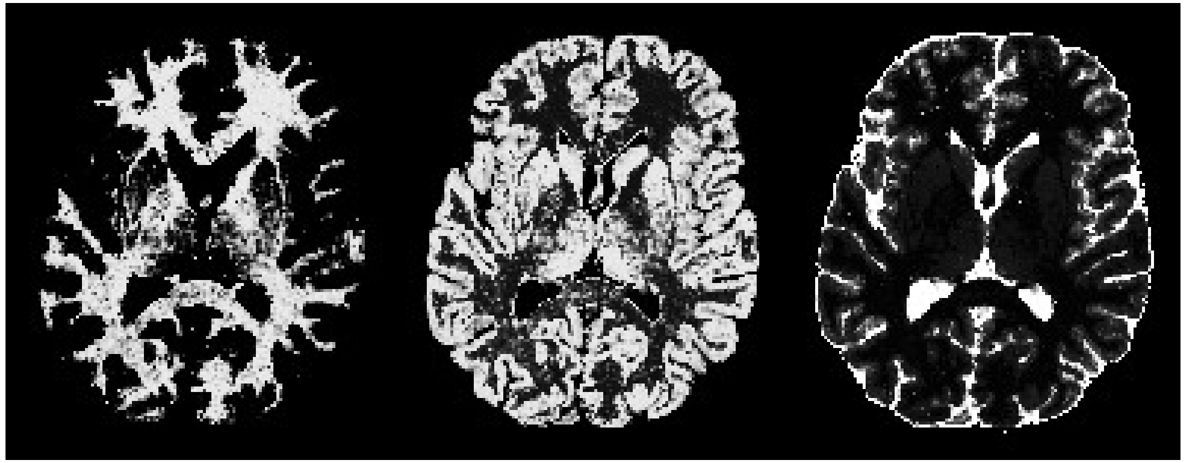


$K = 3$ labels

$$p(\mathbf{d}|\boldsymbol{\theta}) = \prod_n \left(\sum_k \mathcal{N}(d_n|\mu_k, \sigma_k^2)\, \pi_k \right)$$

$$\boldsymbol{\theta} = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots \sigma_K^2, \pi_1, \ldots, \pi_K)^{\mathrm{T}}$$

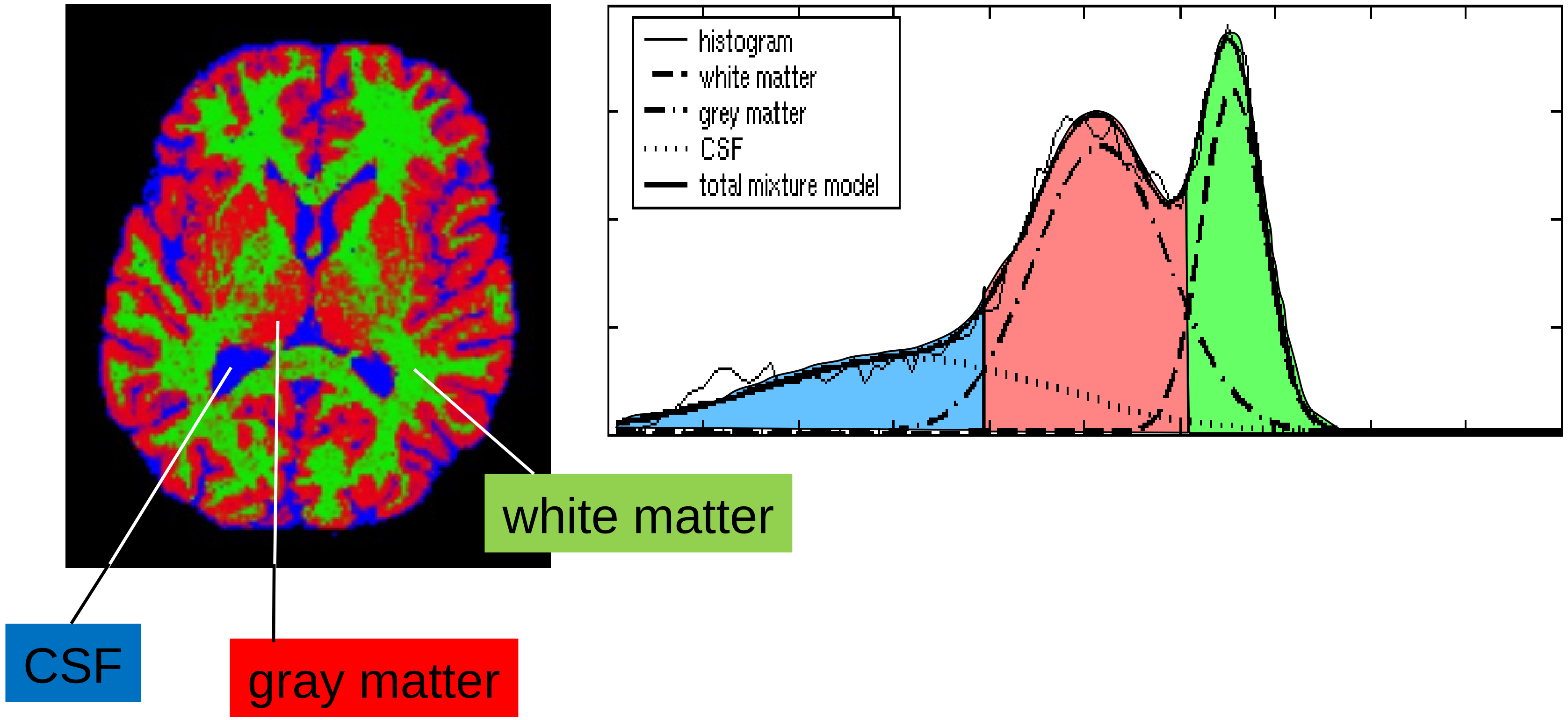
Gaussian mixture model

– Apply Bayes' rule: $p(\mathbf{l}|\mathbf{d}, \boldsymbol{\theta}) = \prod_n p(l_n|d_n, \boldsymbol{\theta})$

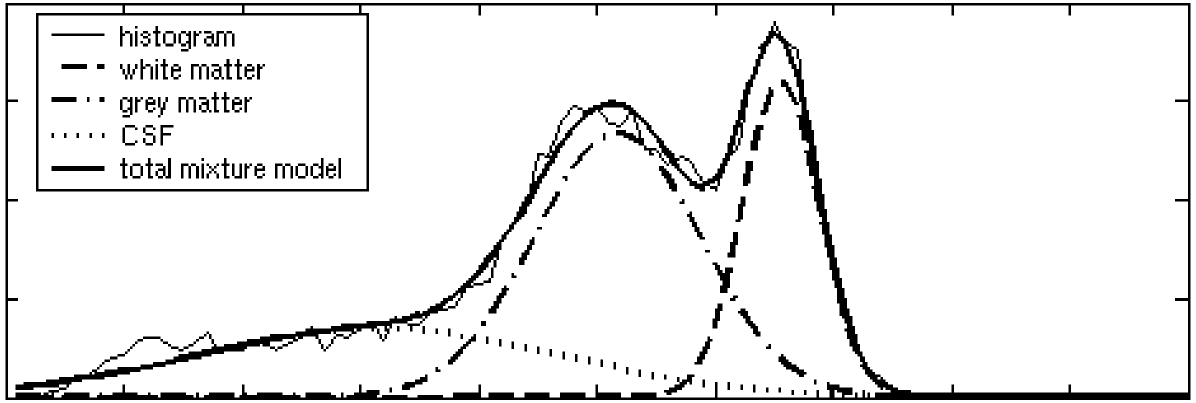
$$p(l_n|d_n, \boldsymbol{\theta}) \;\propto\; \mathcal{N}(d_n|\mu_{l_n}, \sigma^2_{l_n})\,\pi_{l_n}$$

Gaussian mixture model



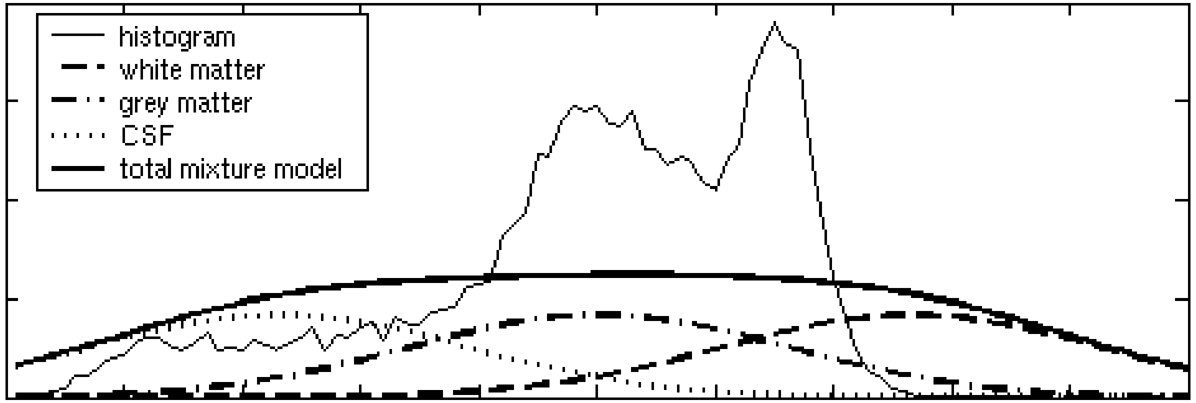
$$\hat{\mathbf{l}} = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d},\boldsymbol{\theta}) = \arg\max_{l_1,\ldots,l_N} p(l_n|d_n,\boldsymbol{\theta})$$

Today's lecture



How to obtain $\boldsymbol{\theta} = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots \sigma_K^2, \pi_1, \ldots, \pi_K)^{\mathrm{T}}$

Today's lecture

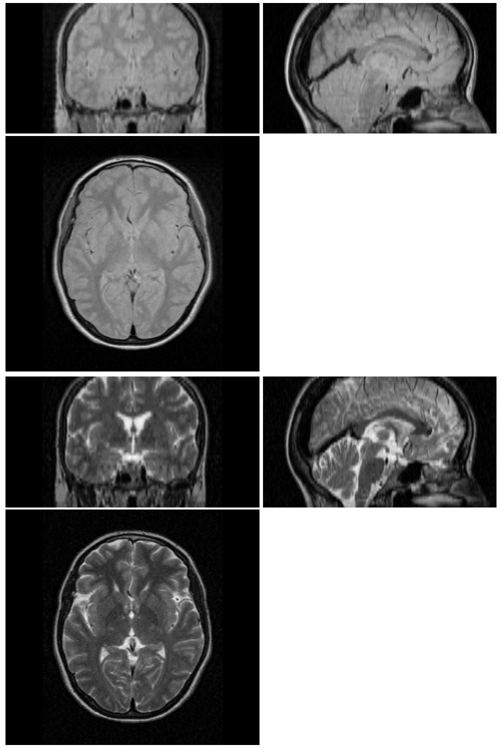


How to obtain $\boldsymbol{\theta} = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots \sigma_K^2, \pi_1, \ldots, \pi_K)^{\mathrm{T}}$

How to obtain model parameters?

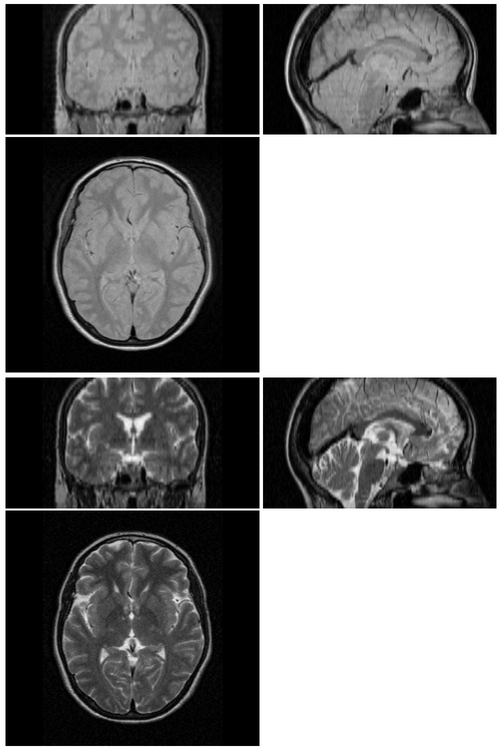
- Click manually on some representative points for each label
- “Train once, apply forever”

- Doesn't work well in MRI:
  - ✓ different imaging protocols
  - ✓ different scanner platforms (make, version)
  - ✓ software/hardware upgrades
  - ✓ ….



How to obtain model parameters?

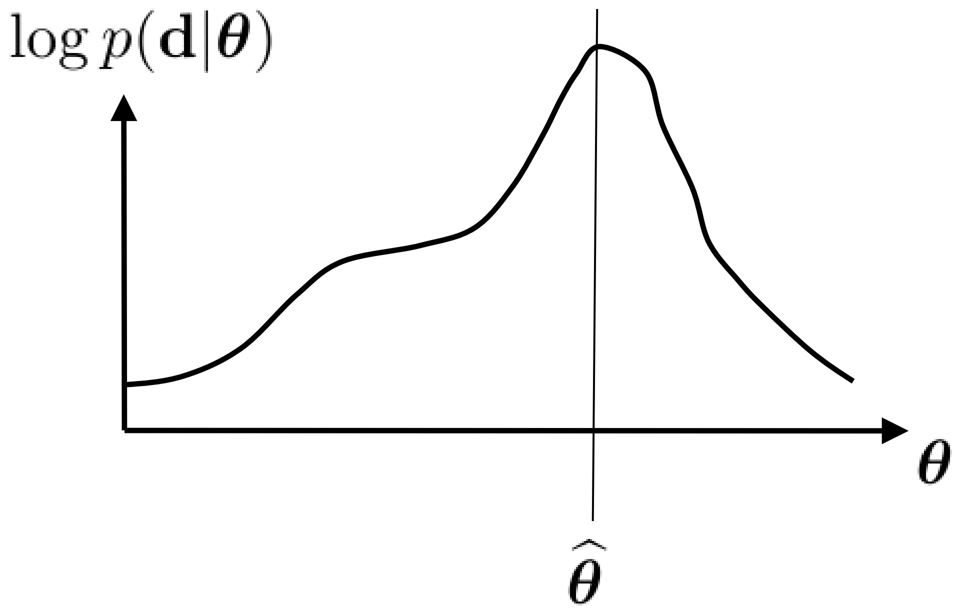
- Click manually on some representative points for each label
- “Train once, apply forever”



- Doesn't work well in MRI:
  - ✓ different imaging protocols
  - ✓ different scanner platforms (make, version)
  - ✓ software/hardware upgrades
  - ✓ ….

Let’s estimate the model parameters automatically from each individual scan

Parameter optimization

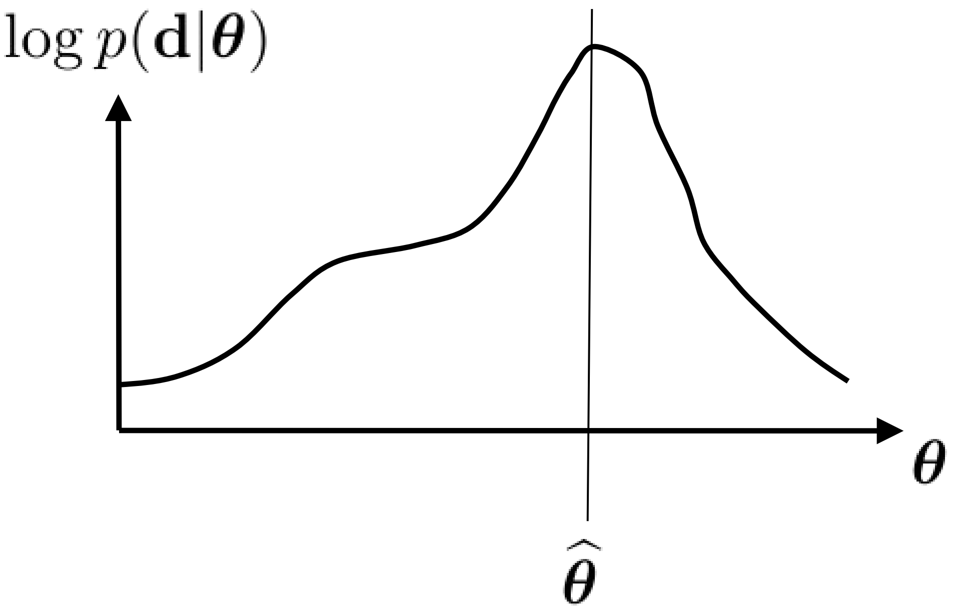


Estimate the *maximum likelihood* parameters:

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{d}|\boldsymbol{\theta})$$

$$= \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{d}|\boldsymbol{\theta})$$

Parameter optimization



Estimate the *maximum likelihood* parameters:

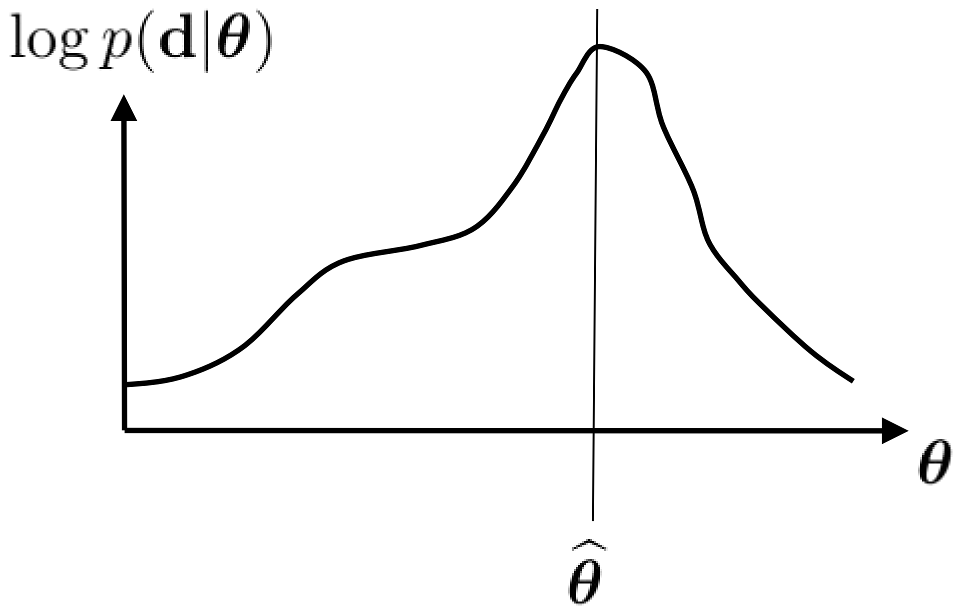
$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{d}|\boldsymbol{\theta})$$

$$= \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{d}|\boldsymbol{\theta})$$

Task:

1. Is this valid? Could I use sine() instead of log()?
2. Benefit? Hint: compute (0.01)^1000 in Matlab/Python

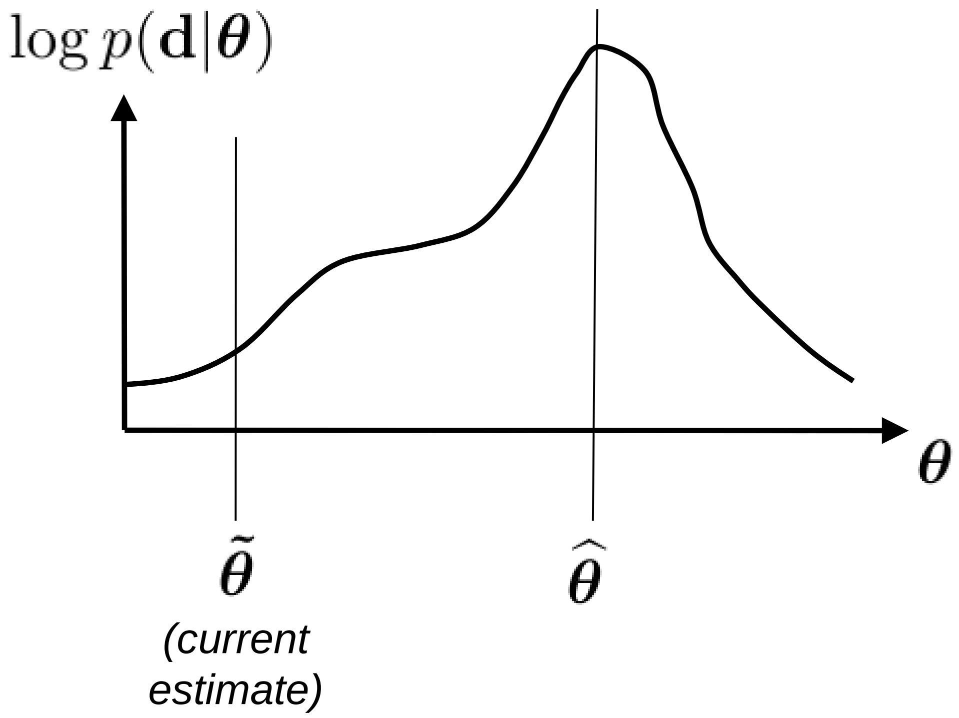
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function

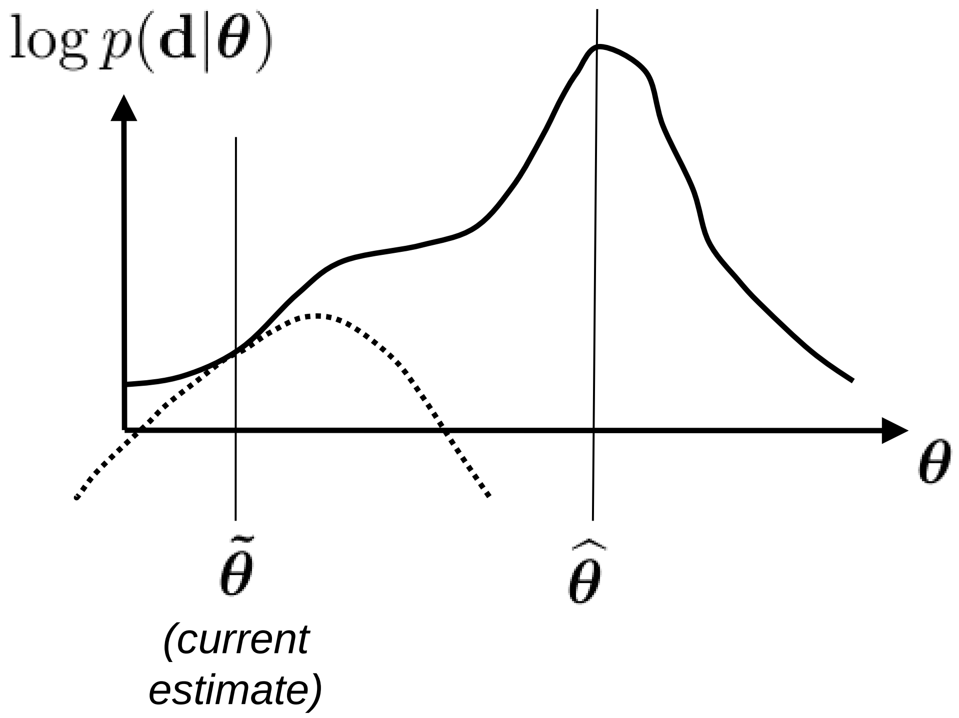
Parameter optimization



Expectation Maximization (EM) algorithm:

– Repeatedly maximize a lower bound to the objective function

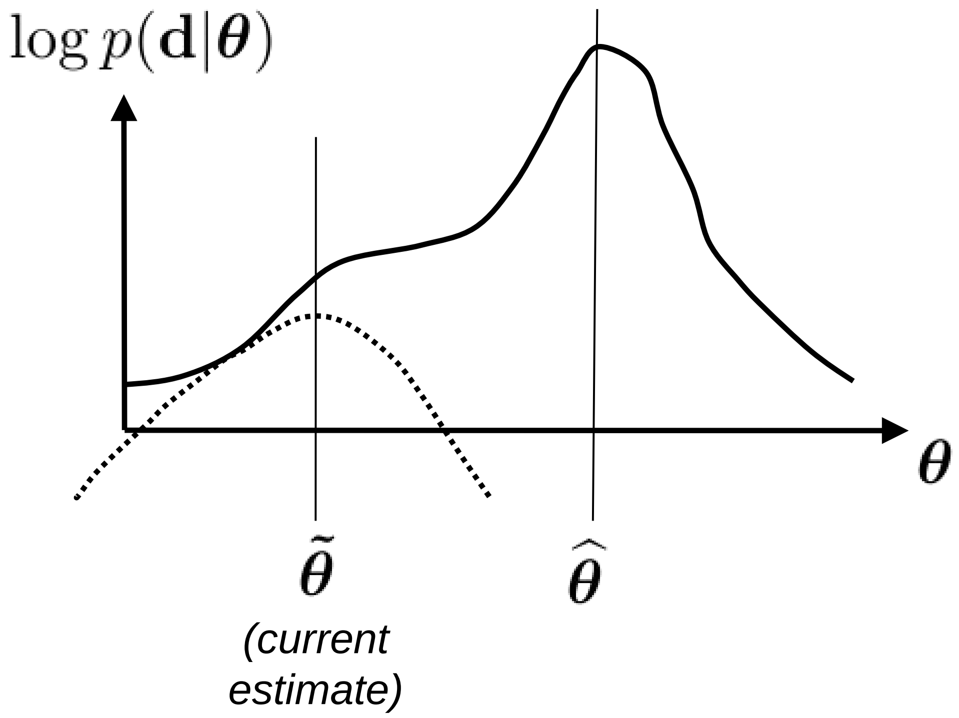
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function

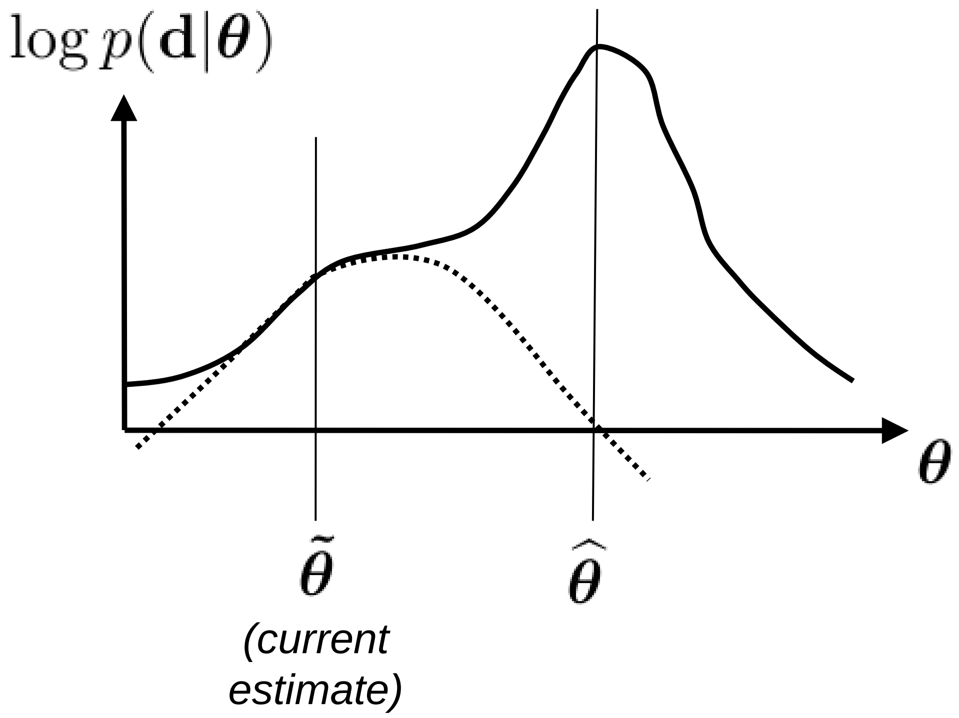
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function

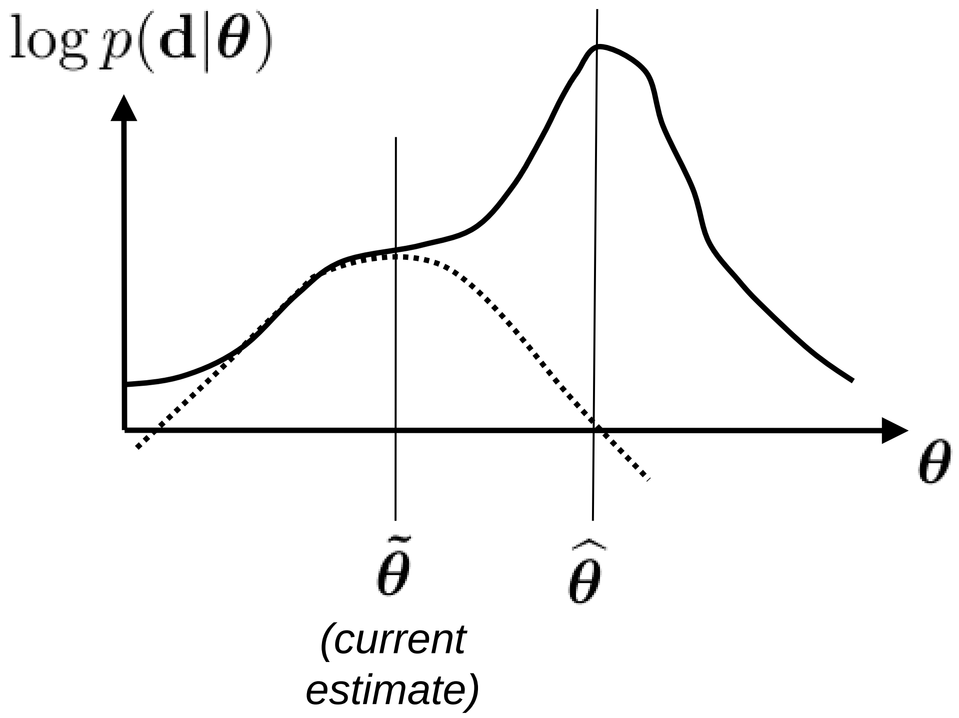
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function

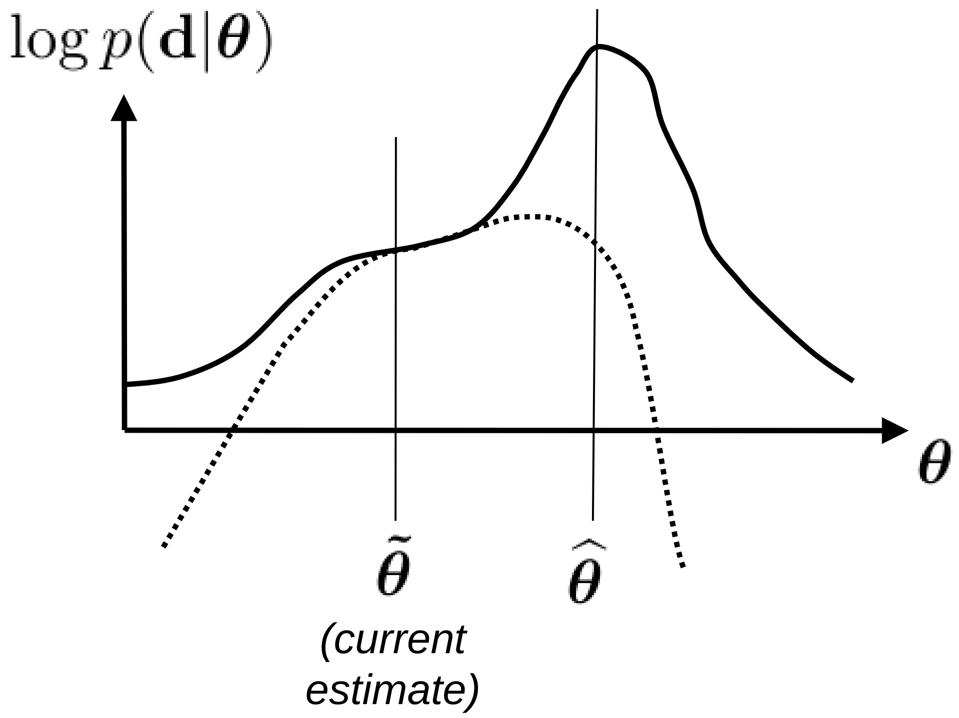
Parameter optimization



Expectation Maximization (EM) algorithm:

– Repeatedly maximize a lower bound to the objective function

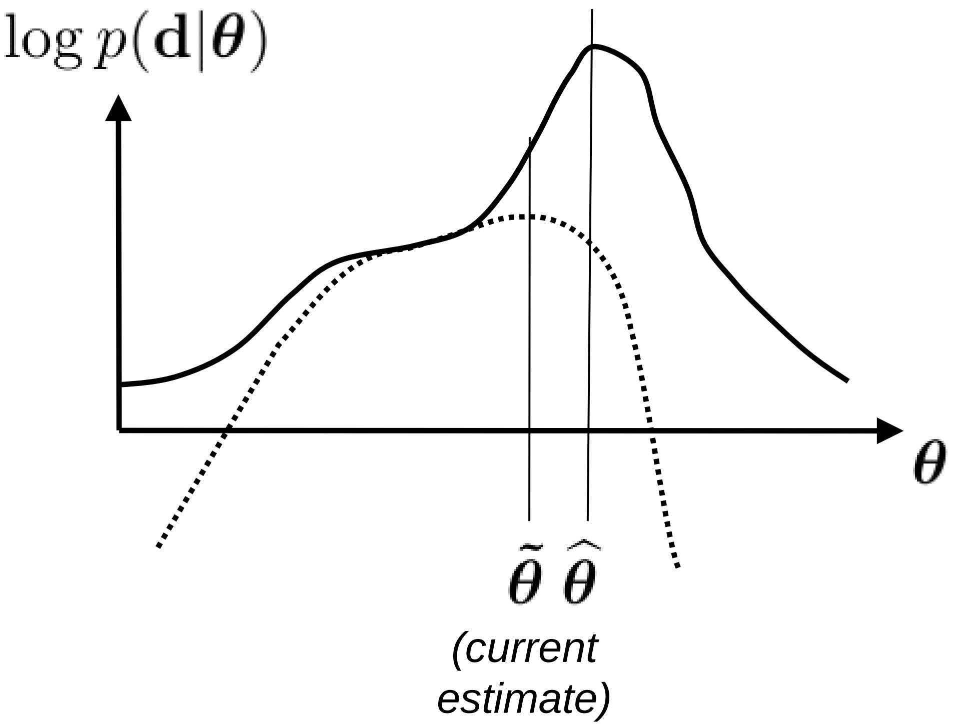
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function

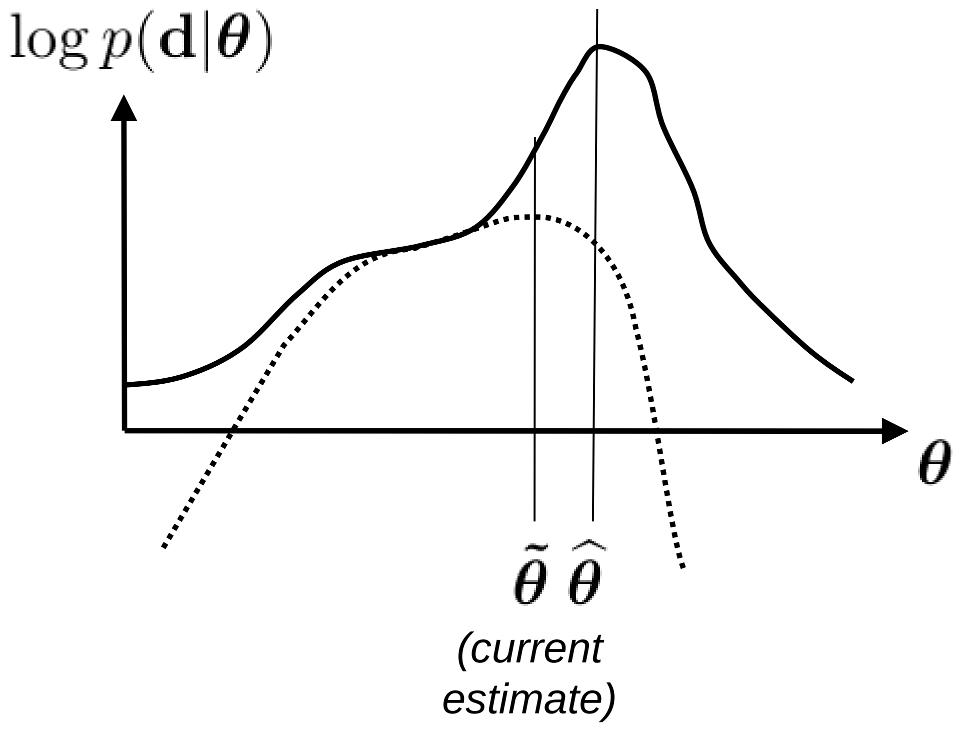
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function

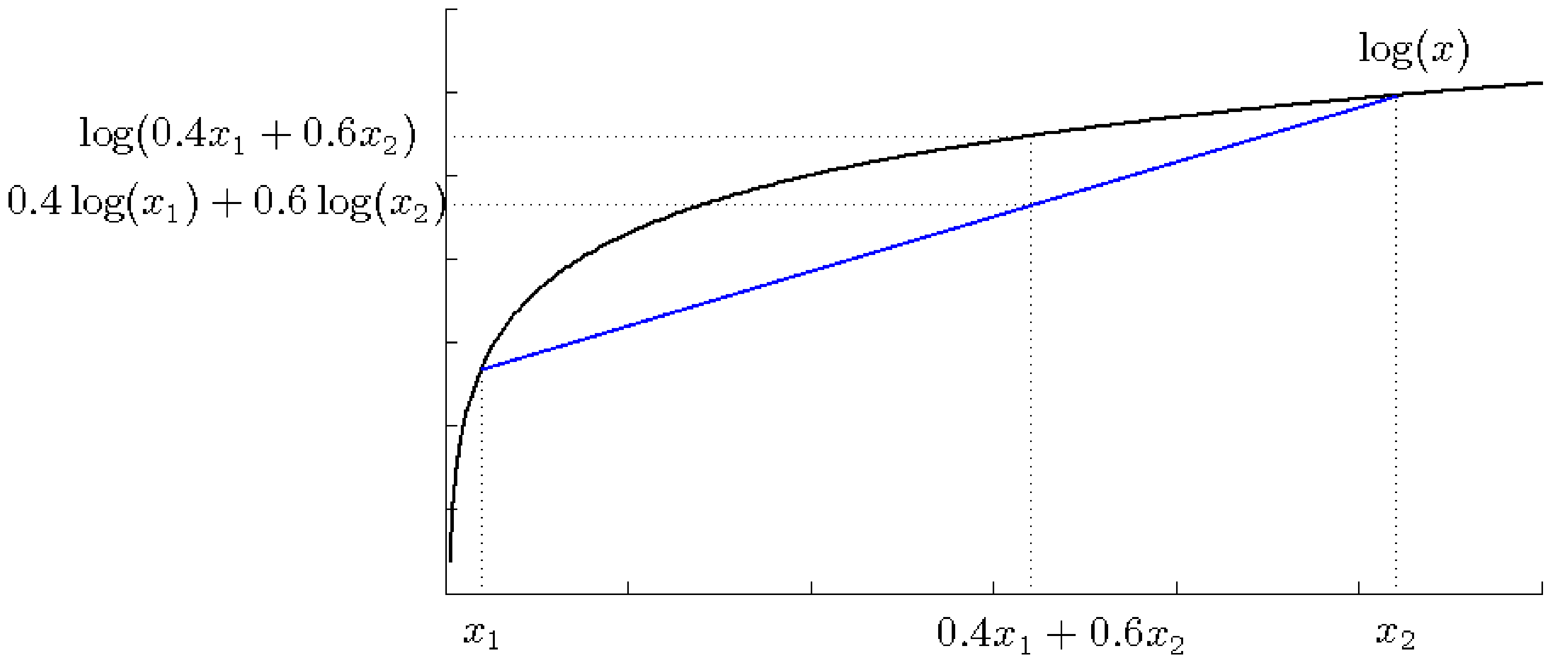
Parameter optimization



Expectation Maximization (EM) algorithm:

- Repeatedly maximize a lower bound to the objective function
- Guaranteed to **never** move in a wrong direction!

Constructing the lower bound



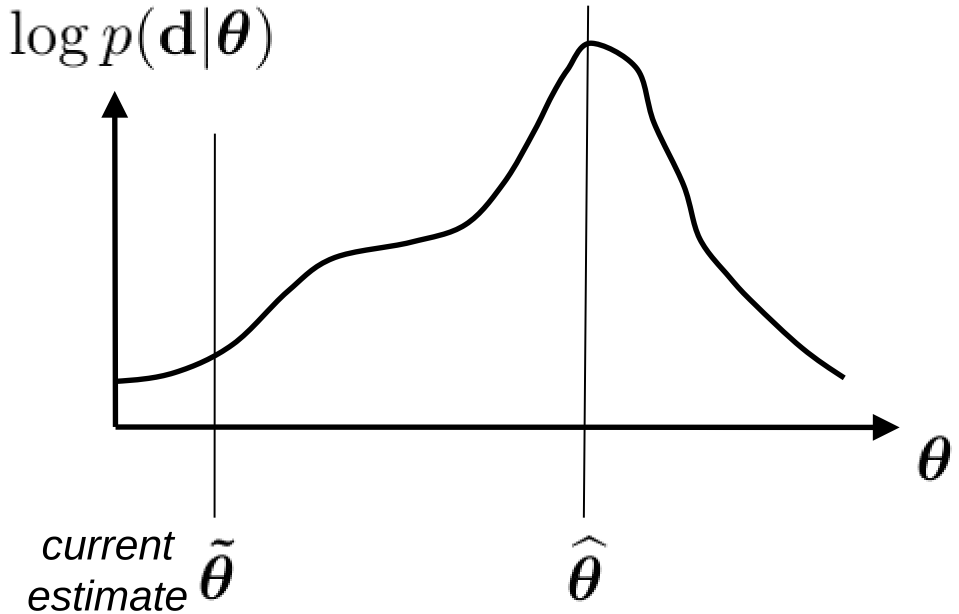
Extends easily to more than two variables (Jensen's inequality):

$$\log\Big(\sum_k w_k x_k\Big) \geq \sum_k w_k \log(x_k)$$

$$w_k \geq 0, \forall k \quad \text{and} \quad \sum_k w_k = 1$$

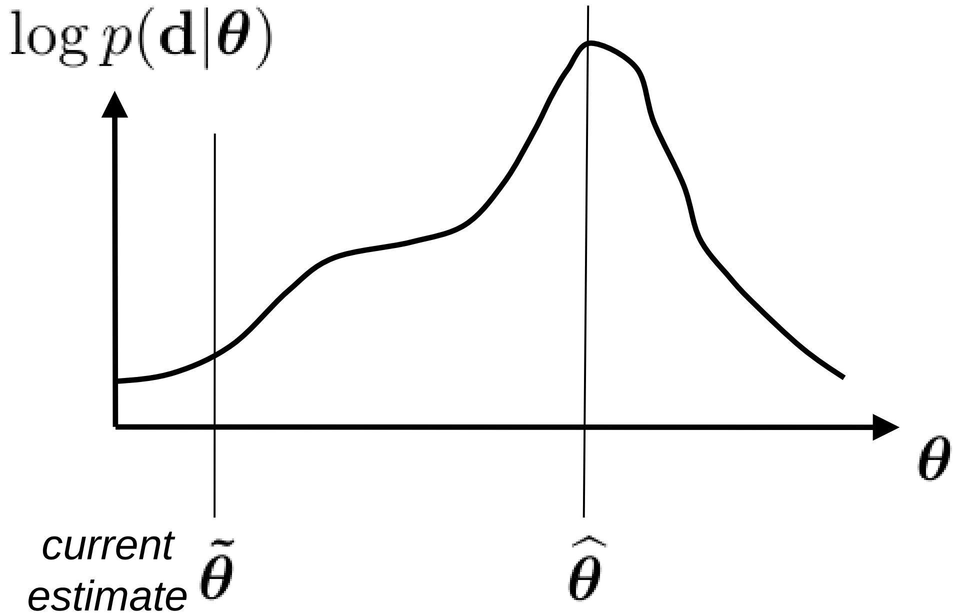
Constructing the lower bound

$$\log p(\mathbf{d}|\boldsymbol{\theta}) = \sum_n \log \left(\sum_k \mathcal{N}(d_n|\mu_k, \sigma_k^2)\, \pi_k \right)$$



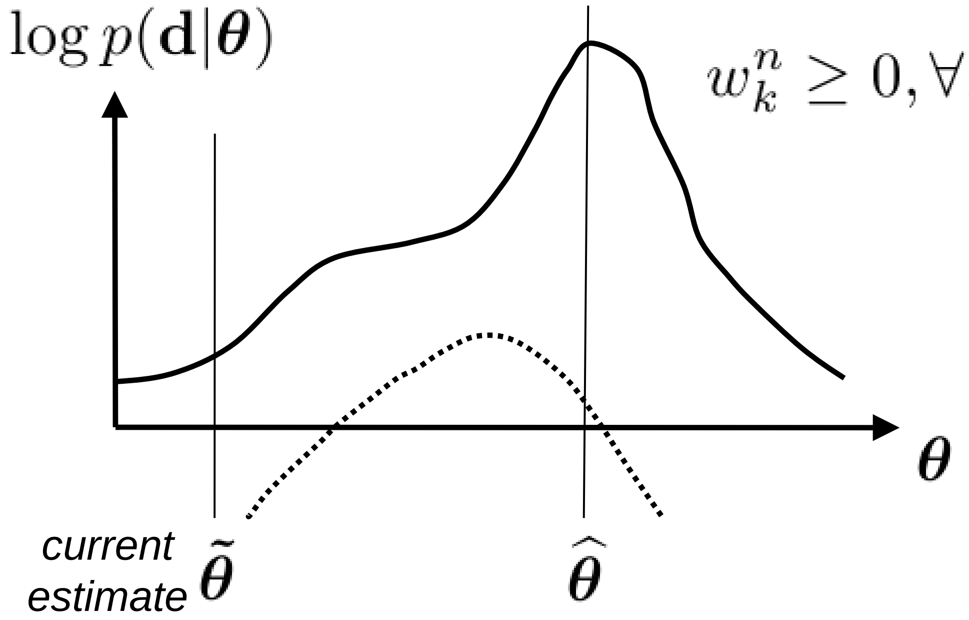
Constructing the lower bound

$$\log p(\mathbf{d}|\boldsymbol{\theta}) = \sum_n \log \left(\sum_k \left[\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\, \pi_k}{w_k^n} \right] w_k^n \right)$$



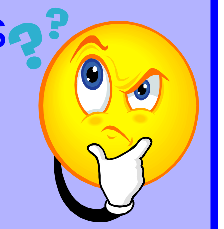
Constructing the lower bound

$$\log p(\mathbf{d}|\boldsymbol{\theta}) = \sum_n \log \left(\sum_k \left[\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\, \pi_k}{w_k^n} \right] w_k^n \right)$$
$$\geq \sum_n \sum_k w_k^n \log \left(\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\, \pi_k}{w_k^n} \right)$$



$$w_k^n \geq 0, \forall k, n \quad \text{and} \quad \sum_k w_k^n = 1, \forall n$$

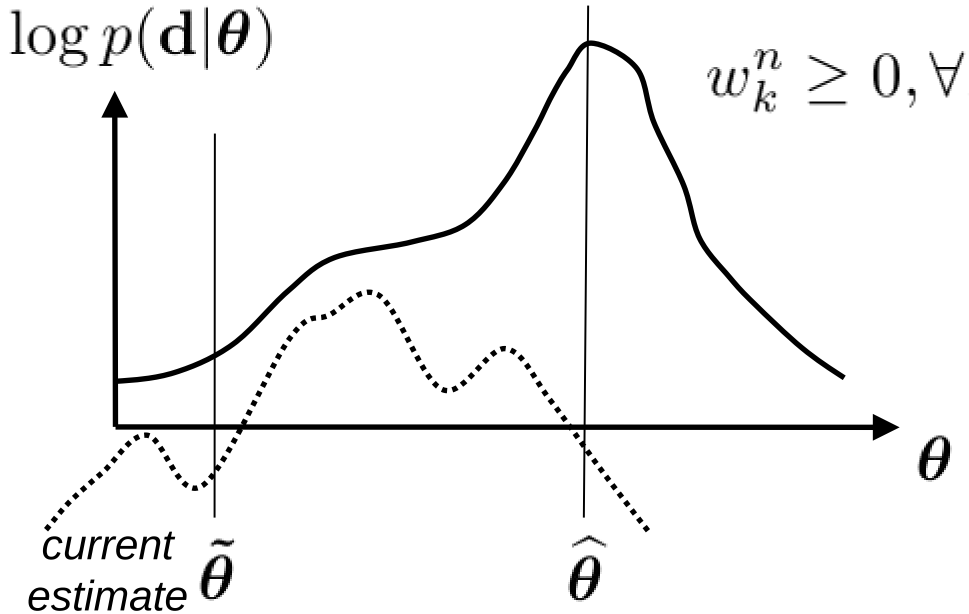
Question: are random weights enough?



Constructing the lower bound

$$\log p(\mathbf{d}|\boldsymbol{\theta}) = \sum_n \log\left(\sum_k \left[\frac{\mathcal{N}(d_n|\mu_k,\sigma_k^2)\,\pi_k}{w_k^n}\right] w_k^n\right)$$

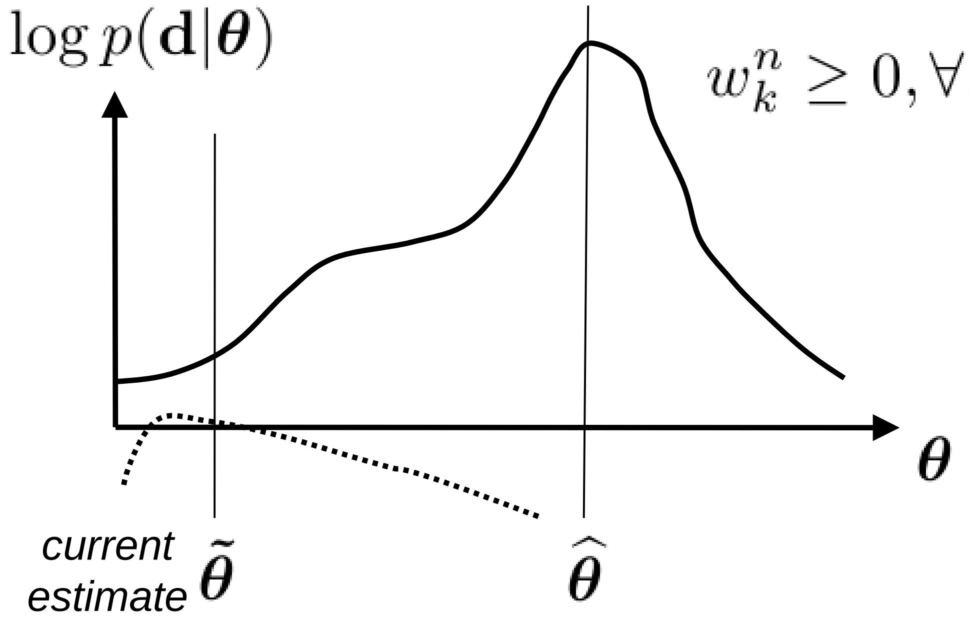
$$\geq \sum_n \sum_k w_k^n \log\left(\frac{\mathcal{N}(d_n|\mu_k,\sigma_k^2)\,\pi_k}{w_k^n}\right)$$



$w_k^n \geq 0, \forall k, n$ and $\sum_k w_k^n = 1, \forall n$

Constructing the lower bound

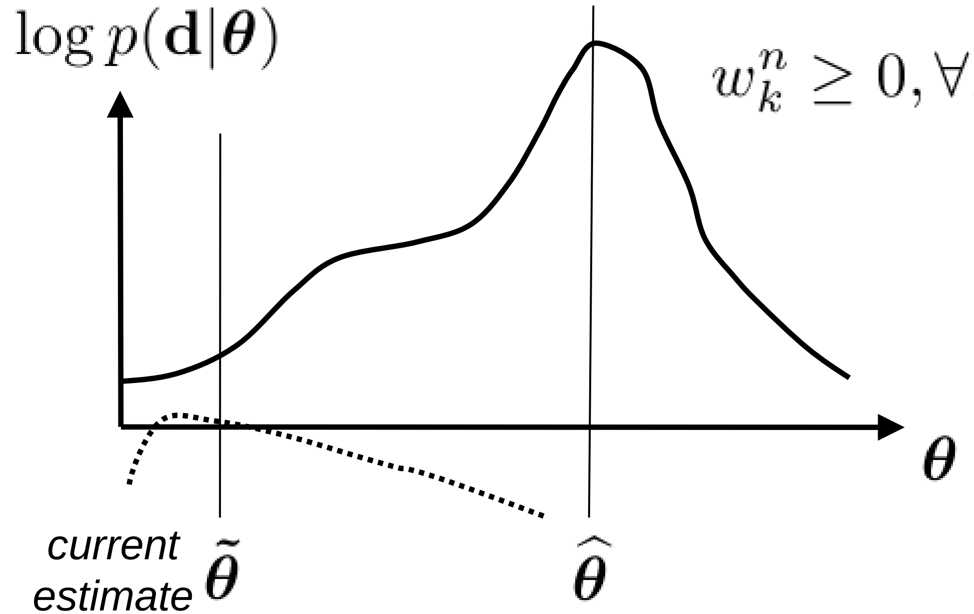
$$\log p(\mathbf{d}|\boldsymbol{\theta}) = \sum_n \log\left(\sum_k \left[\frac{\mathcal{N}(d_n|\mu_k,\sigma_k^2)\,\pi_k}{w_k^n}\right] w_k^n\right)$$
$$\geq \sum_n \sum_k w_k^n \log\left(\frac{\mathcal{N}(d_n|\mu_k,\sigma_k^2)\,\pi_k}{w_k^n}\right)$$



$$w_k^n \geq 0, \forall k, n \quad \text{and} \quad \sum_k w_k^n = 1, \forall n$$

Constructing the lower bound

$$\log p(\mathbf{d}|\boldsymbol{\theta}) = \sum_n \log \left(\sum_k \left[\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\,\pi_k}{w_k^n} \right] w_k^n \right)$$
$$\geq \sum_n \sum_k w_k^n \log \left(\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\,\pi_k}{w_k^n} \right)$$

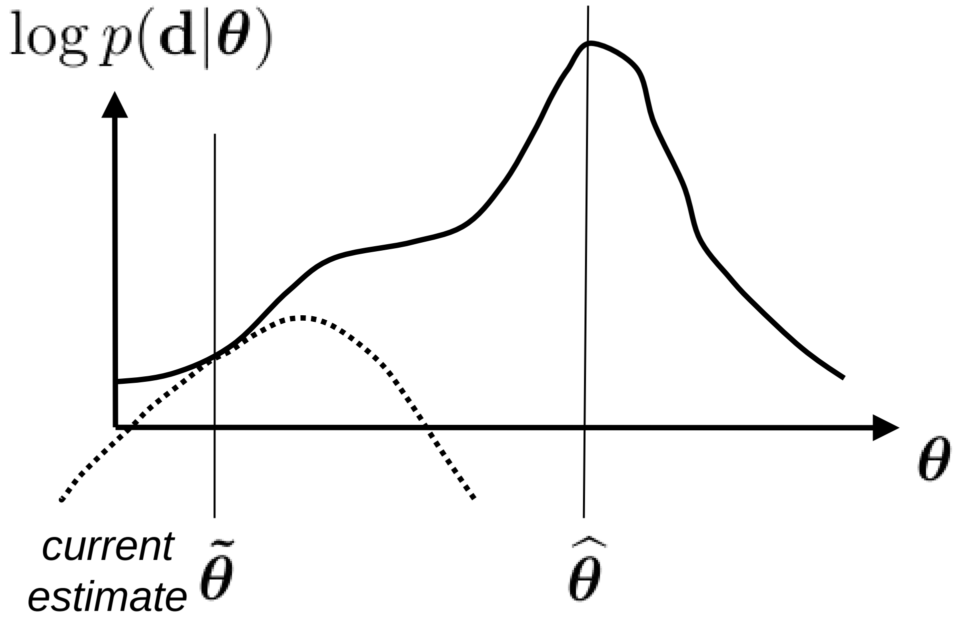


$$w_k^n \geq 0, \forall k, n \quad \text{and} \quad \sum_k w_k^n = 1, \forall n$$

No!

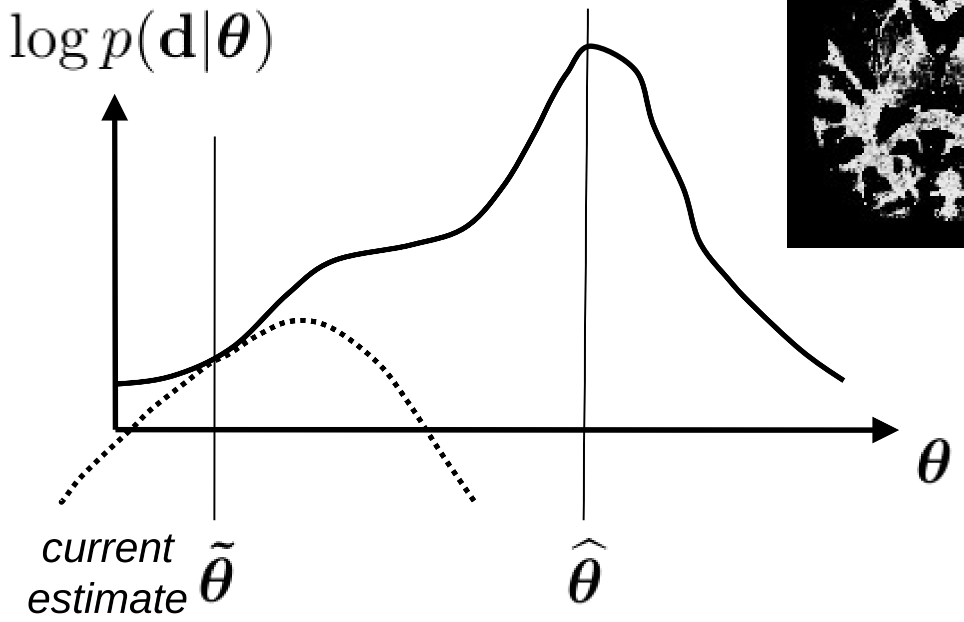
Constructing the lower bound

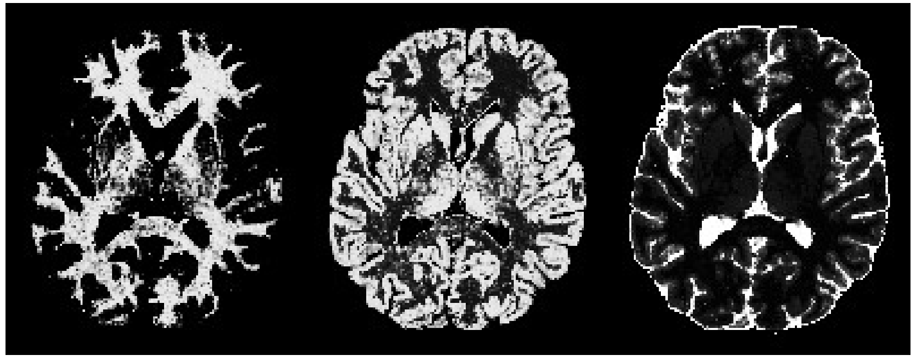
The lower bound touches the objective function at the current parameter estimate if $w_k^n \propto \mathcal{N}(d_n|\tilde{\mu}_k, \tilde{\sigma}_k^2)\,\tilde{\pi}_k$



Constructing the lower bound

The lower bound touches the objective function at the current parameter estimate if $w_k^n \propto \mathcal{N}(d_n | \tilde{\mu}_k, \tilde{\sigma}_k^2)\, \tilde{\pi}_k$



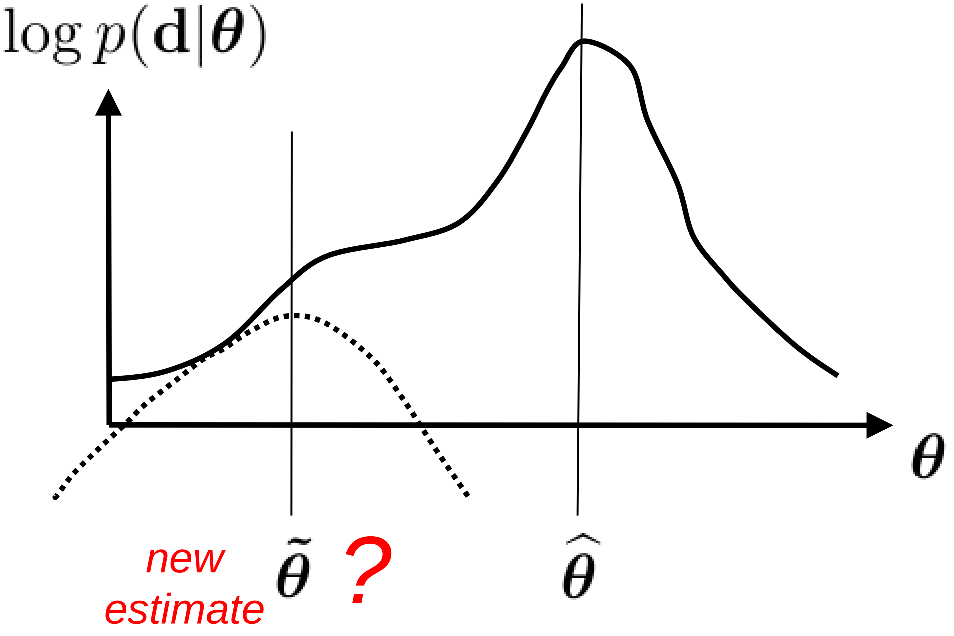


Maximizing the lower bound

Lower bound:

$$\sum_n \left[\sum_k w_k^n \log \left(\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\pi_k}{w_k^n} \right) \right] = -\frac{1}{2} \sum_k \left[\frac{1}{\sigma_k^2} \sum_n w_k^n (d_n - \mu_k)^2 + \left(\sum_n w_k^n \right) \log \sigma_k^2 \right]$$

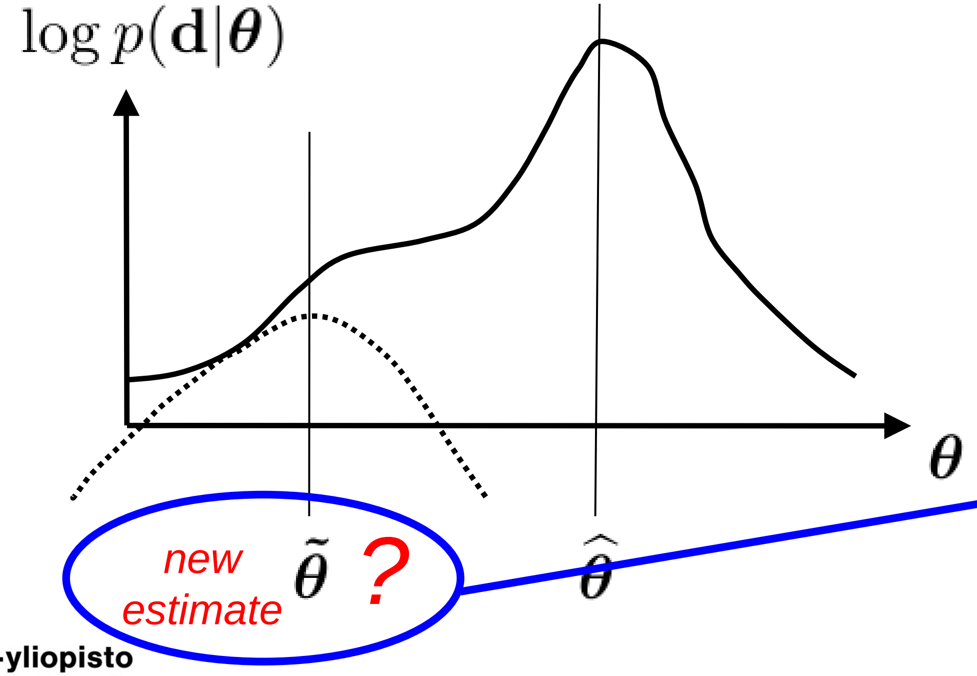
$$+ \sum_k \left[\left(\sum_n w_k^n \right) \log \pi_k \right] + \mathrm{C}$$



Maximizing the lower bound

Lower bound:

$$\sum_n \left[\sum_k w_k^n \log \left(\frac{\mathcal{N}(d_n | \mu_k, \sigma_k^2) \pi_k}{w_k^n} \right) \right] = -\frac{1}{2} \sum_k \left[\frac{1}{\sigma_k^2} \sum_n w_k^n (d_n - \mu_k)^2 + \left(\sum_n w_k^n \right) \log \sigma_k^2 \right] + \sum_k \left[\left(\sum_n w_k^n \right) \log \pi_k \right] + \mathrm{C}$$



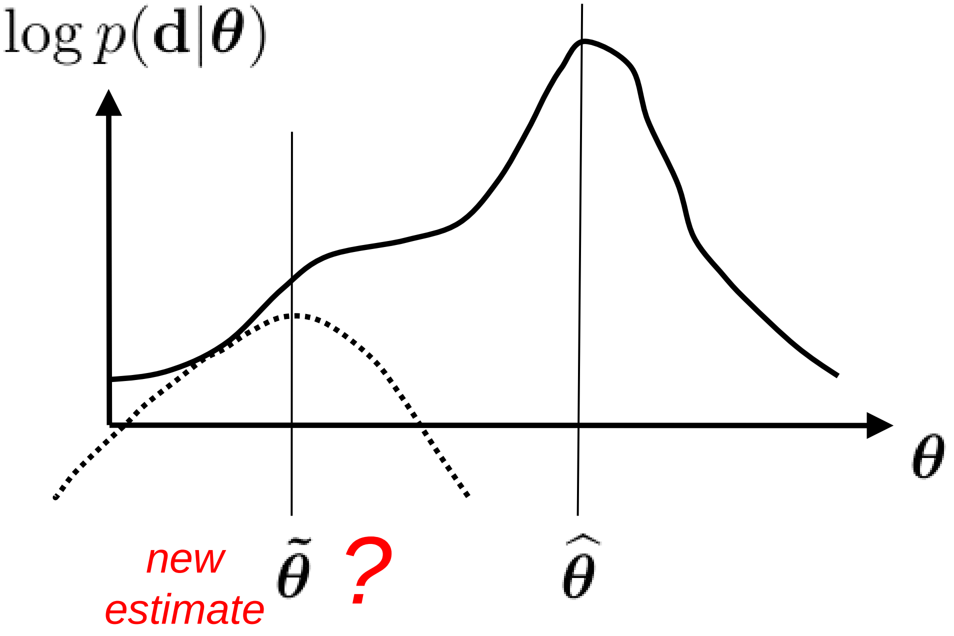
Task:

1. What is $\tilde{\mu}_k$?
2. What is $\tilde{\sigma}_k^2$?
3. What is $\tilde{\pi}_k$?

Maximizing the lower bound

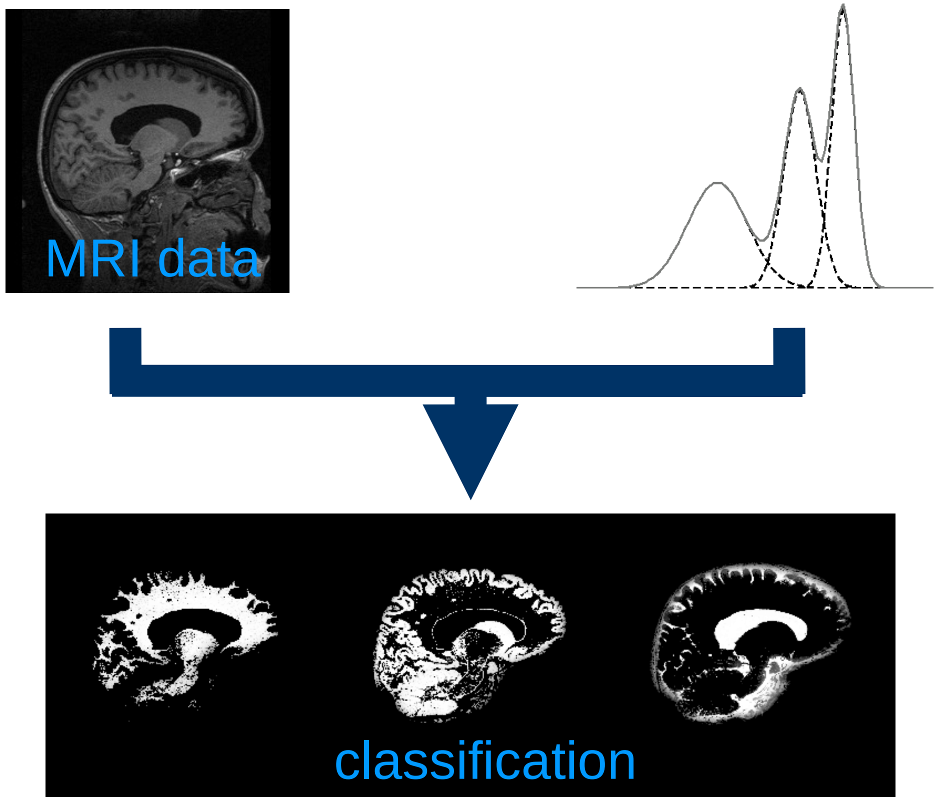
Lower bound:

$$\sum_n \left[\sum_k w_k^n \log \left(\frac{\mathcal{N}(d_n|\mu_k, \sigma_k^2)\pi_k}{w_k^n} \right) \right] = -\frac{1}{2} \sum_k \left[\frac{1}{\sigma_k^2} \sum_n w_k^n (d_n - \mu_k)^2 + \left(\sum_n w_k^n \right) \log \sigma_k^2 \right] + \sum_k \left[\left(\sum_n w_k^n \right) \log \pi_k \right] + \mathrm{C}$$



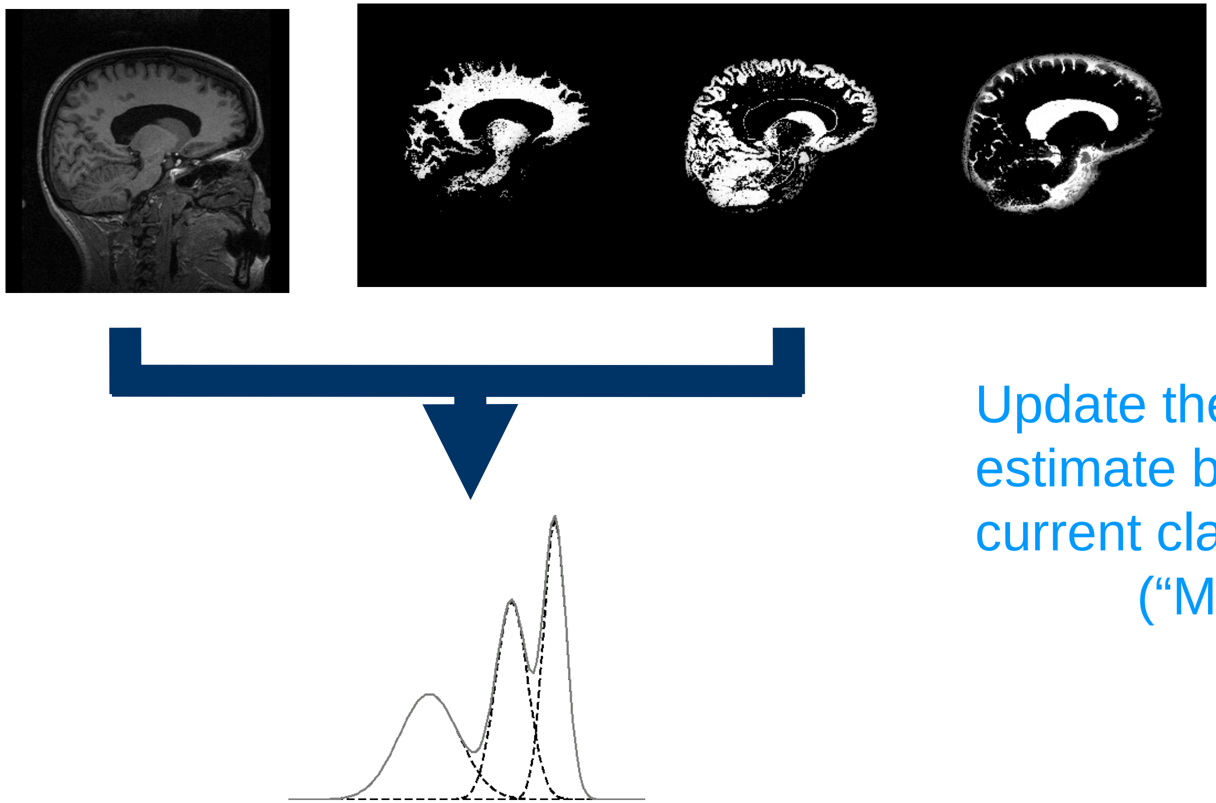
$$\frac{\partial}{\partial \boldsymbol{\theta}} = 0 \quad \Rightarrow \quad \begin{aligned} \tilde{\mu}_k &\leftarrow \frac{\sum_n w_k^n d_n}{\sum_n w_k^n} \\ \tilde{\sigma}_k^2 &\leftarrow \frac{\sum_n w_k^n (d_n - \tilde{\mu}_k)^2}{\sum_n w_k^n} \\ \tilde{\pi}_k &\leftarrow \frac{\sum_n w_k^n}{N} \end{aligned}$$

Parameter optimizer summarized



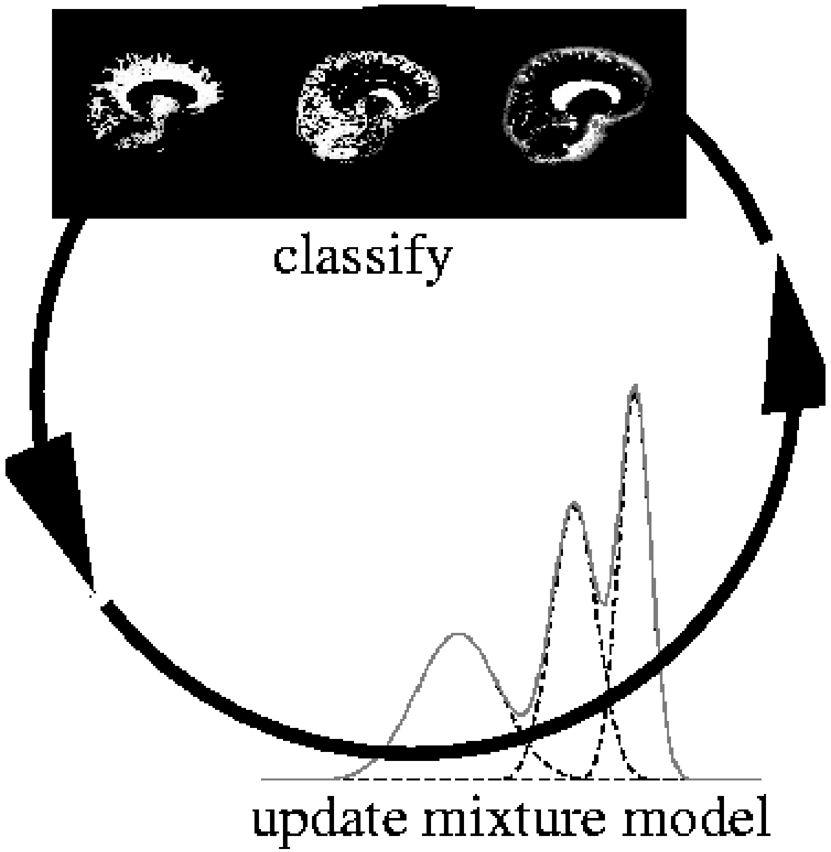
Classify the image voxels according to the current parameter estimate
("E-step")

Parameter optimizer summarized



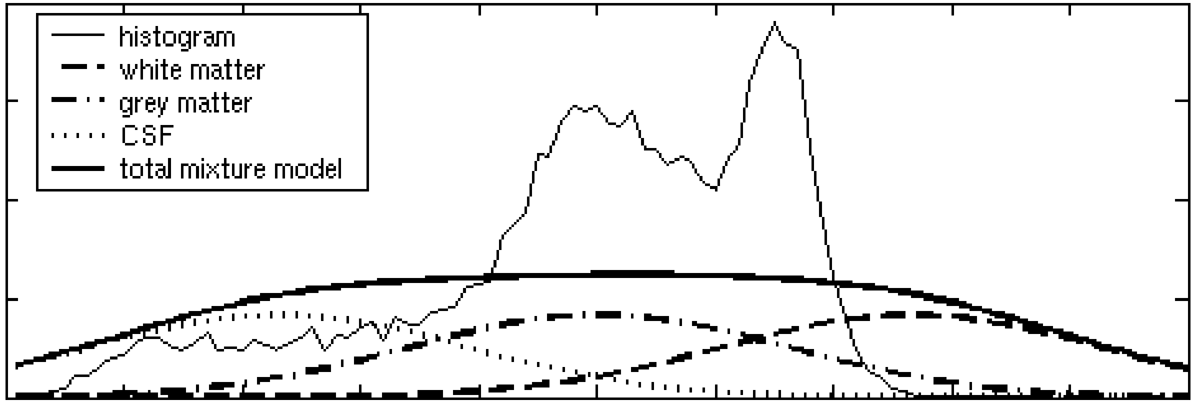
Update the parameter estimate based on the current classification ("M-step")

Parameter optimizer summarized



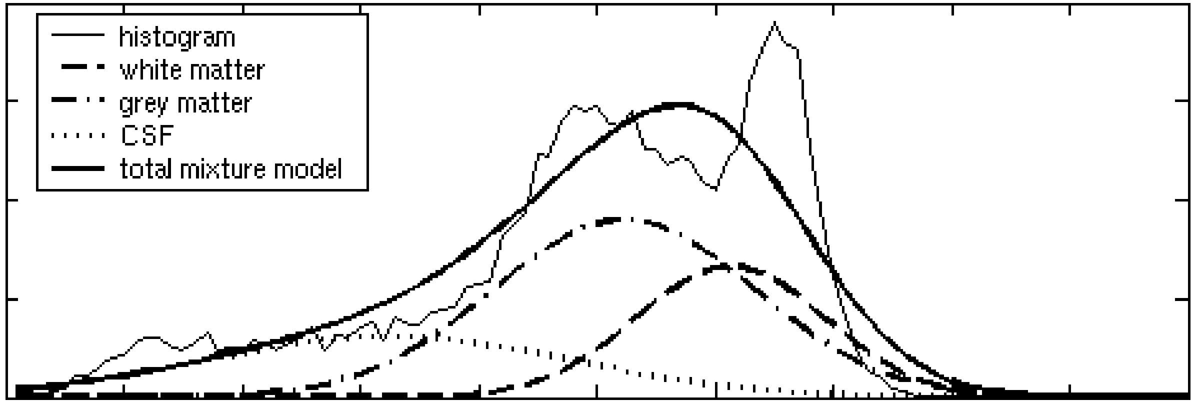
- Repeatedly apply closed-form parameter updates
- Each iteration improves the log likelihood

Example



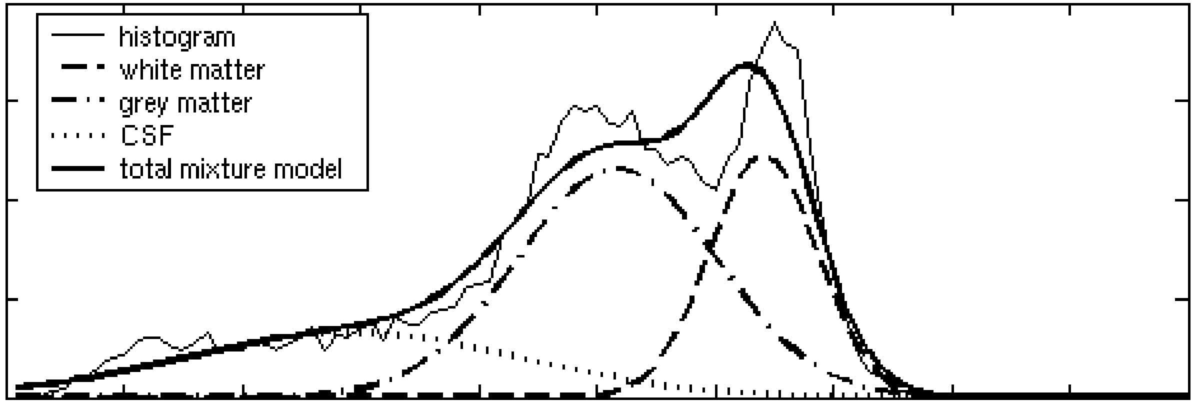
initialization

Example



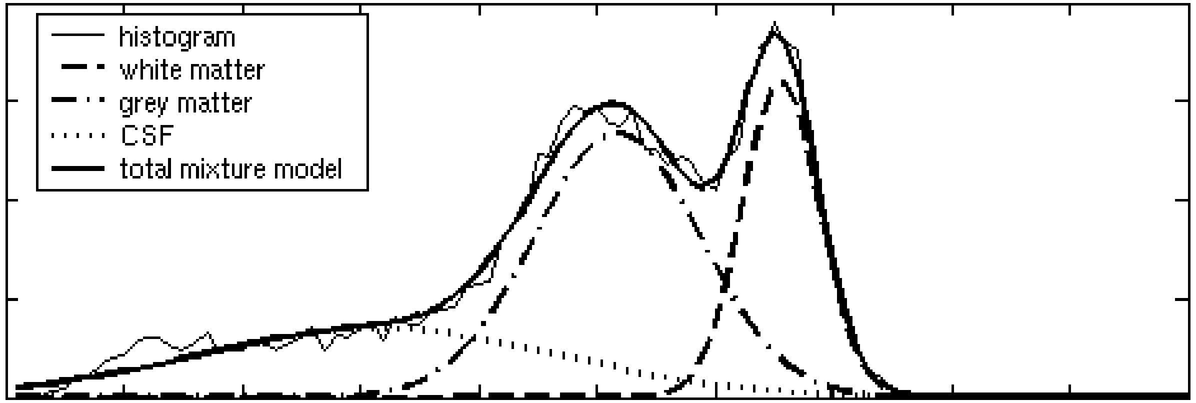
after one iteration

Example



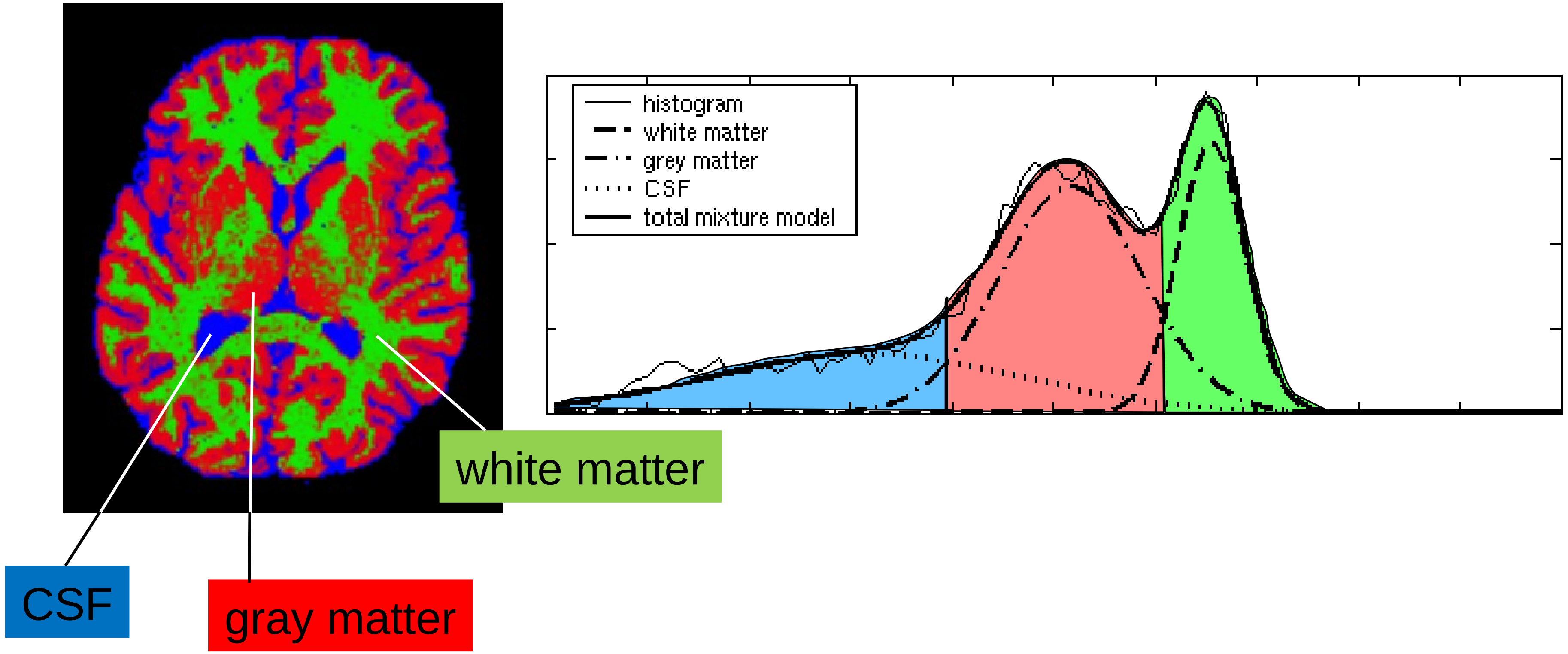
after 10 iterations

Example



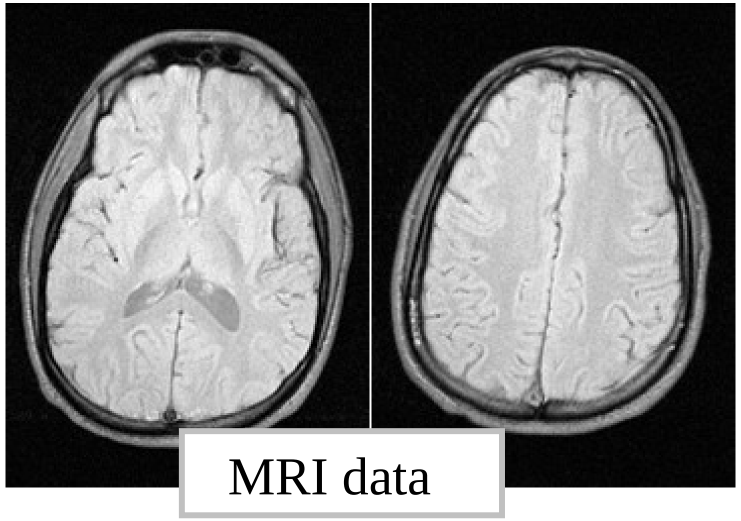
after 30 iterations

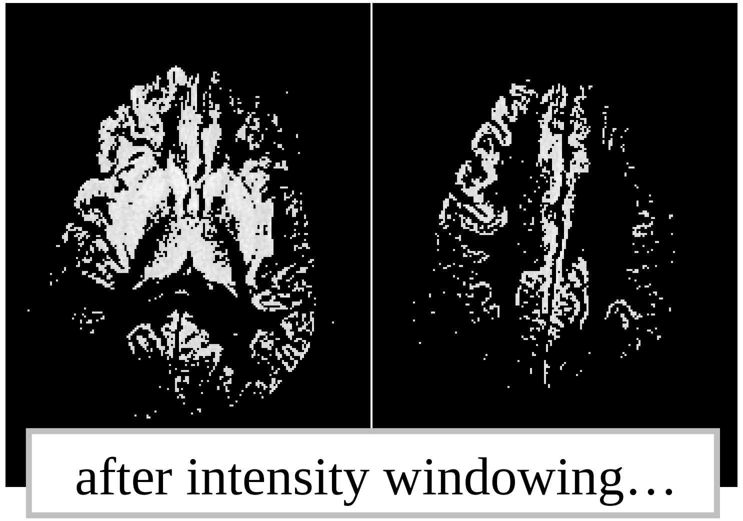
Example


histogram
white matter
grey matter
CSF
total mixture model
white matter
CSF
gray matter

MRI “bias field” artifact

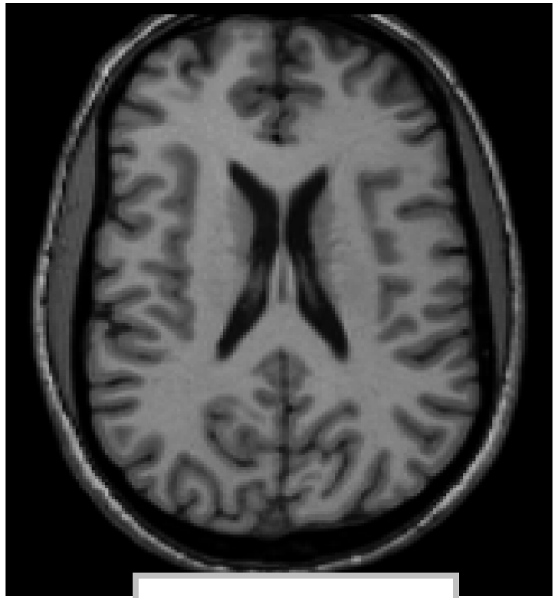
- Imaging artifact in MRI
- Smooth intensity variations across the image area





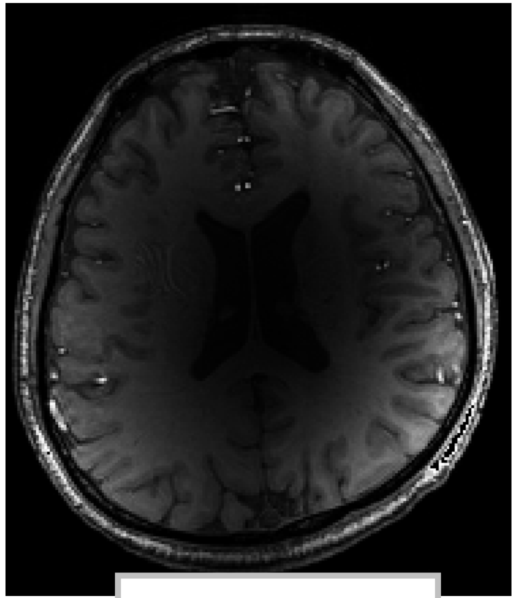
MRI “bias field” artifact

- Depends on the object being scanned
- More pronounced on high-field scanners



1.5 Tesla scanner

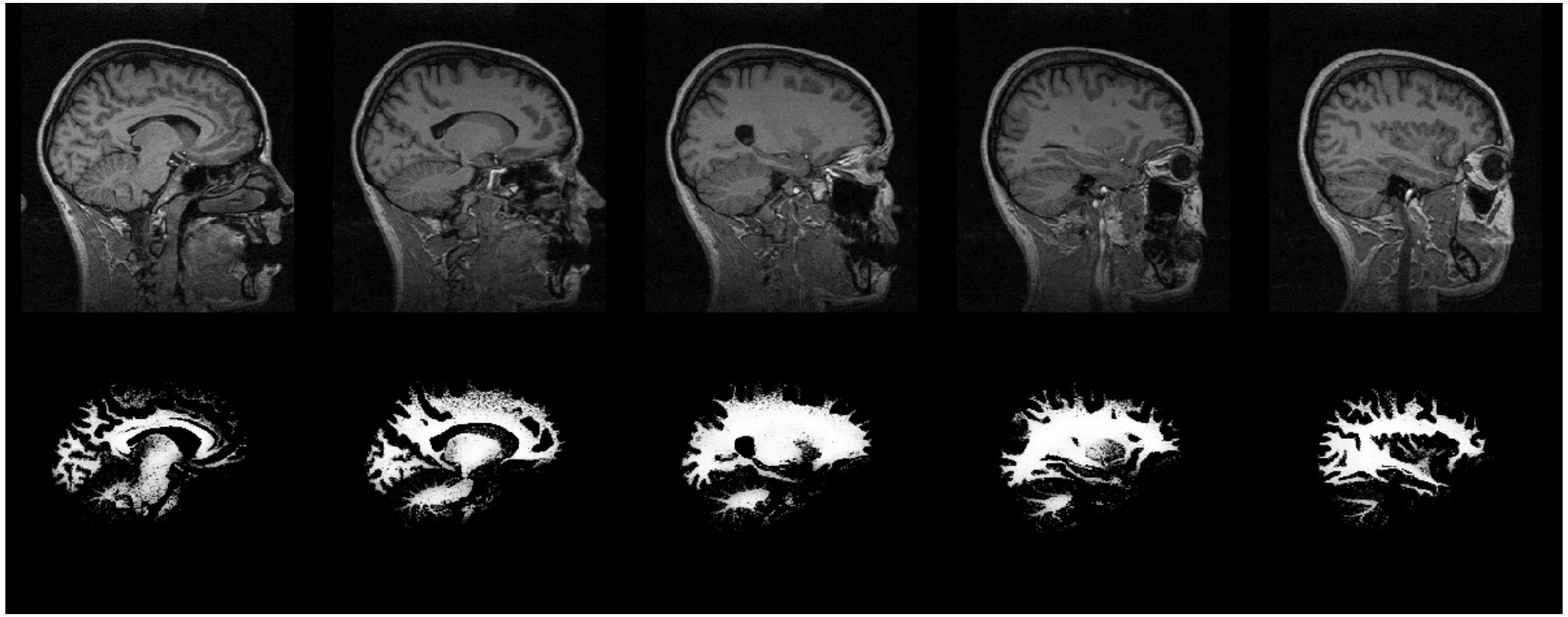
3 Tesla scanner



7 Tesla scanner

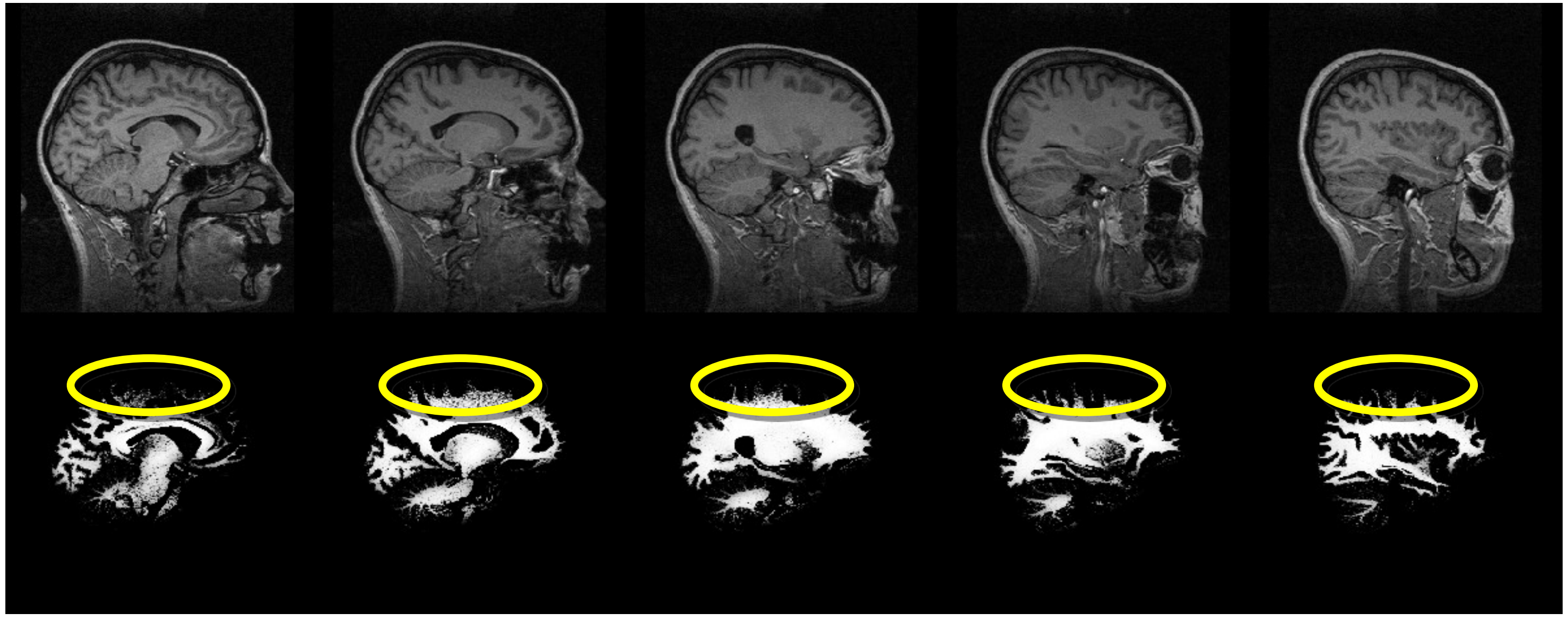
MRI “bias field” artifact

Causes segmentation errors with our segmentation procedure so far…



MRI “bias field” artifact

Causes segmentation errors with our segmentation procedure so far…



Generative model

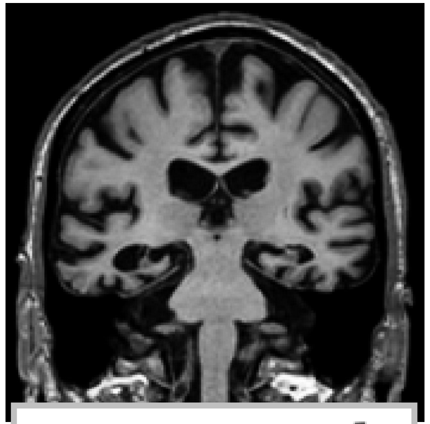
$p(\mathbf{l}|\boldsymbol{\theta}_l)$

“labeling model”

Label image $\mathbf{l}$

$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d)$

“imaging model”

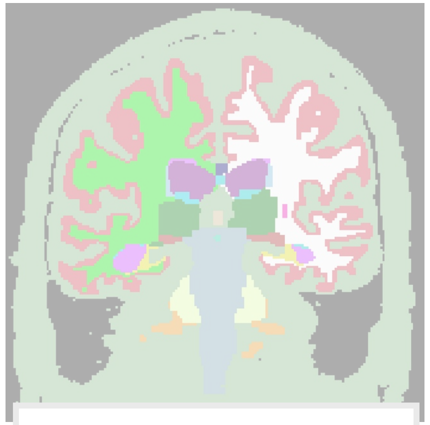


MRI image $\mathbf{d}$

Improved imaging model

$p(\mathbf{l}|\boldsymbol{\theta}_l)$

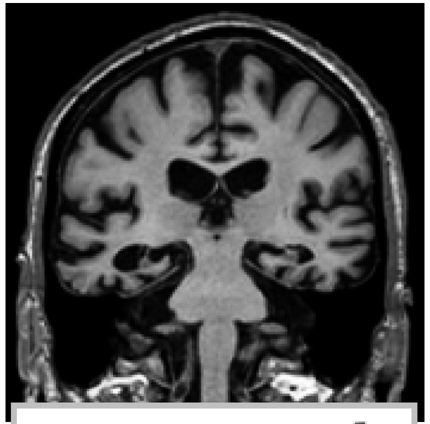
"labeling model"



Label image $\mathbf{l}$

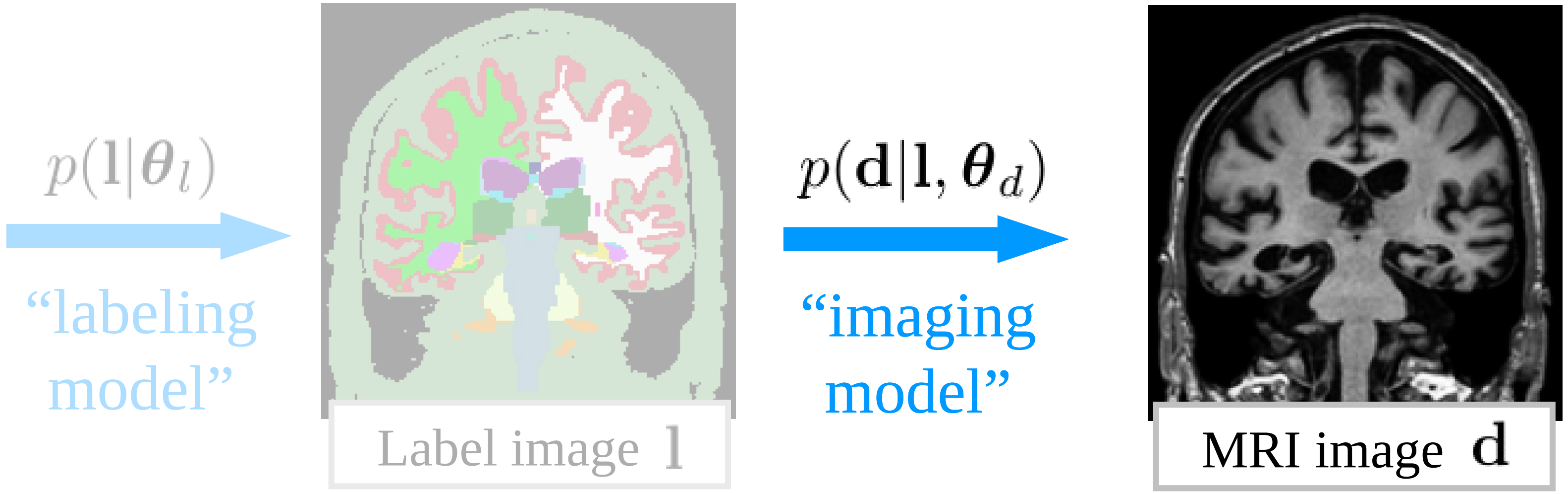
$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d)$

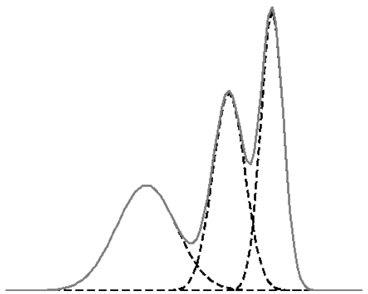
"imaging model"



MRI image $\mathbf{d}$

Improved imaging model



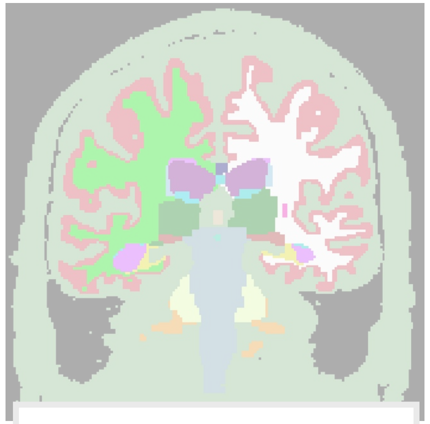


old model

Improved imaging model

$p(\mathbf{l}|\boldsymbol{\theta}_l)$

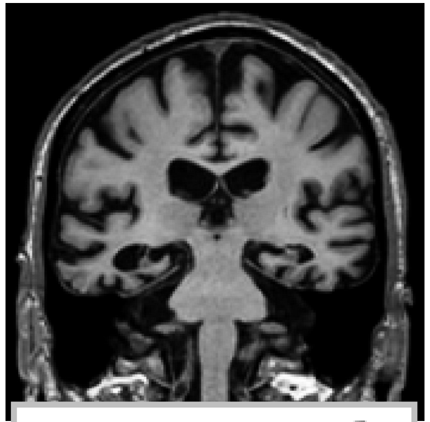
"labeling model"



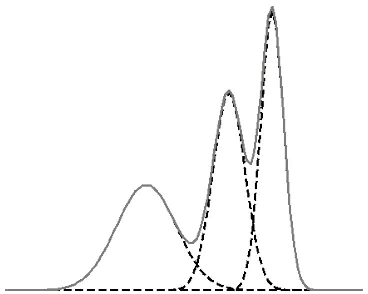
Label image $\mathbf{l}$

$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d)$

"imaging model"

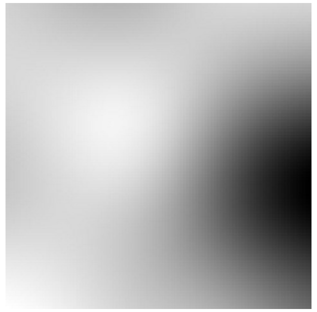


MRI image $\mathbf{d}$



old model

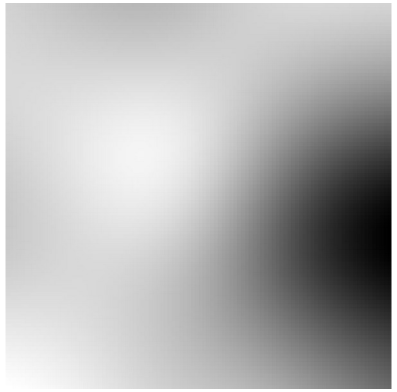
+



parametric bias field model

Bias field model

Linear combination of M smooth basis functions



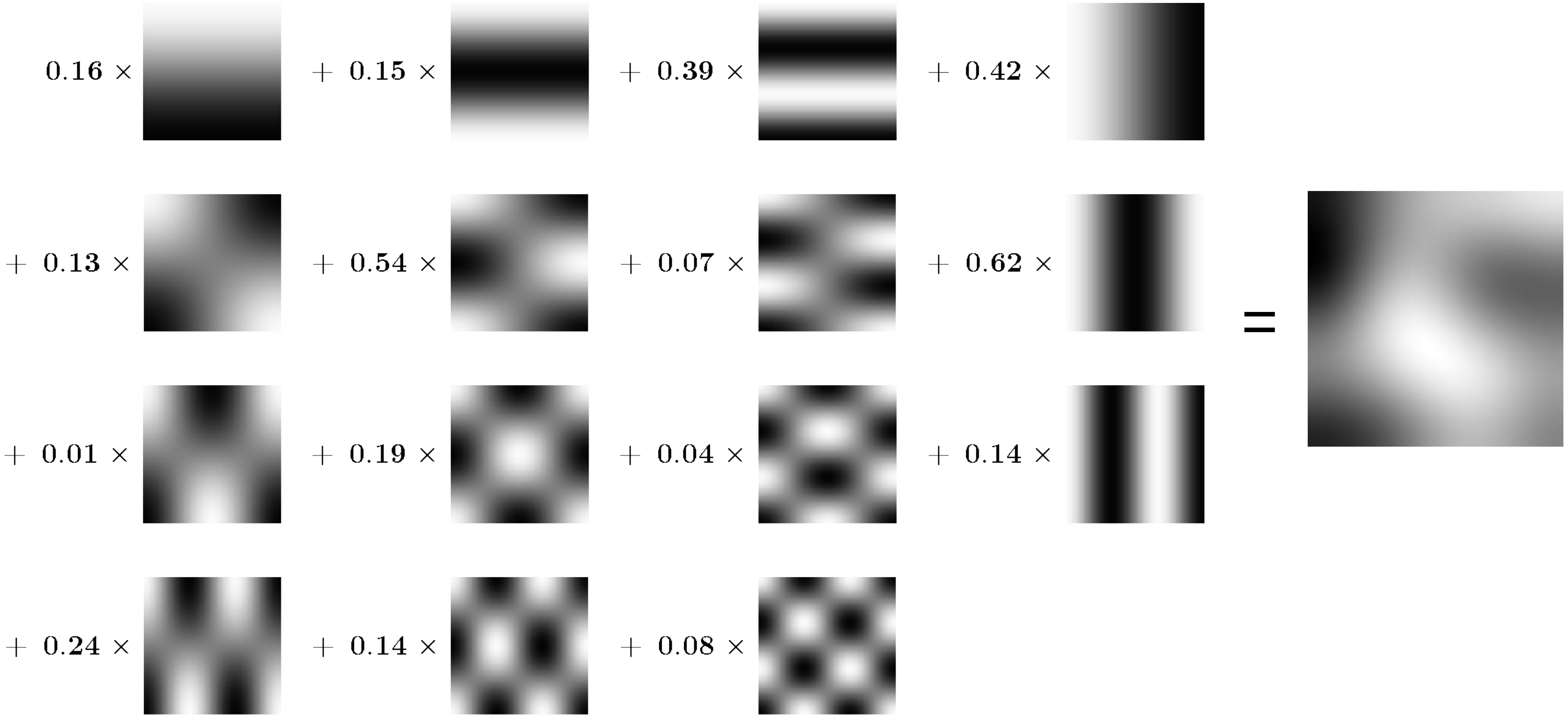
$\mathbf{b} = (b_1, \ldots, b_N)^{\mathrm{T}}$

$$b_n = \sum_{m=1}^{M} c_m \phi_m^n$$

ϕ_m^n : value of the *m*th basis function in voxel n

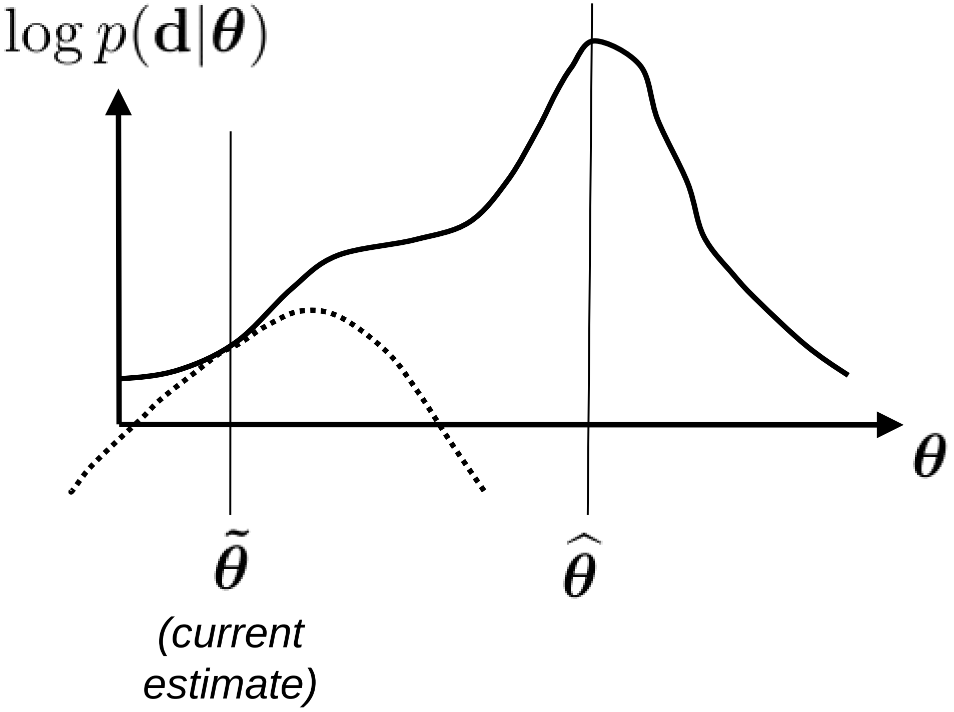
$\mathbf{c} = (c_1, \ldots, c_M)^{\mathrm{T}}$: parameters of the bias field model

Bias field model



Parameter optimization

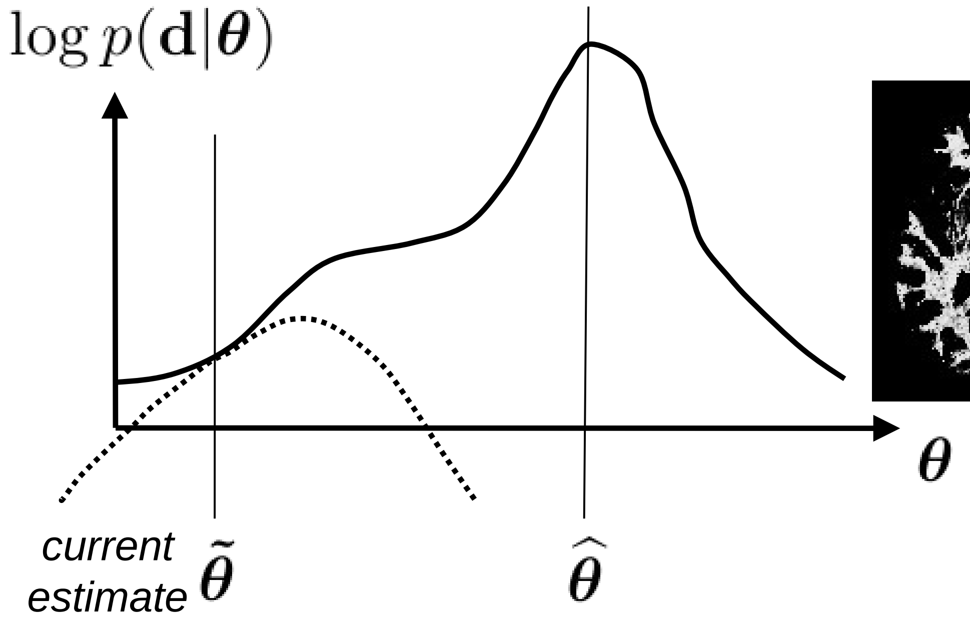
- Bias field parameters are part of the model parameters
- Parameter optimization with a ***Generalized*** Expectation Maximization (GEM) algorithm

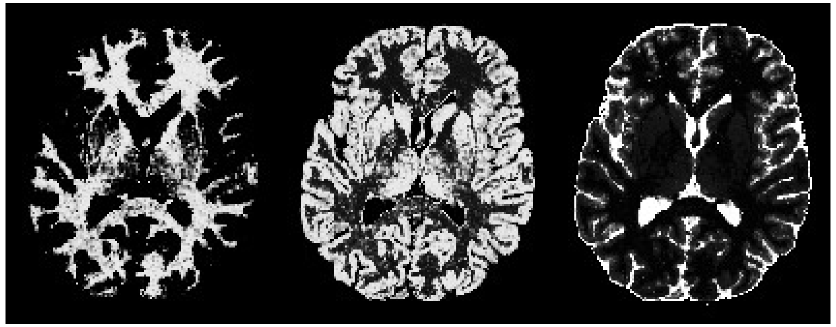


- Repeatedly ***improve*** a lower bound to the log likelihood function
- Still guaranteed to **never** move in a wrong direction!

Constructing the lower bound

- Same derivations as before
- The lower bound touches the objective function at current parameter estimate if $w_k^n \propto \mathcal{N}\left(d_n - \sum_m \tilde{c}_m \phi_m^n \,\middle|\, \tilde{\mu}_k, \tilde{\sigma}_k^2\right) \tilde{\pi}_k$



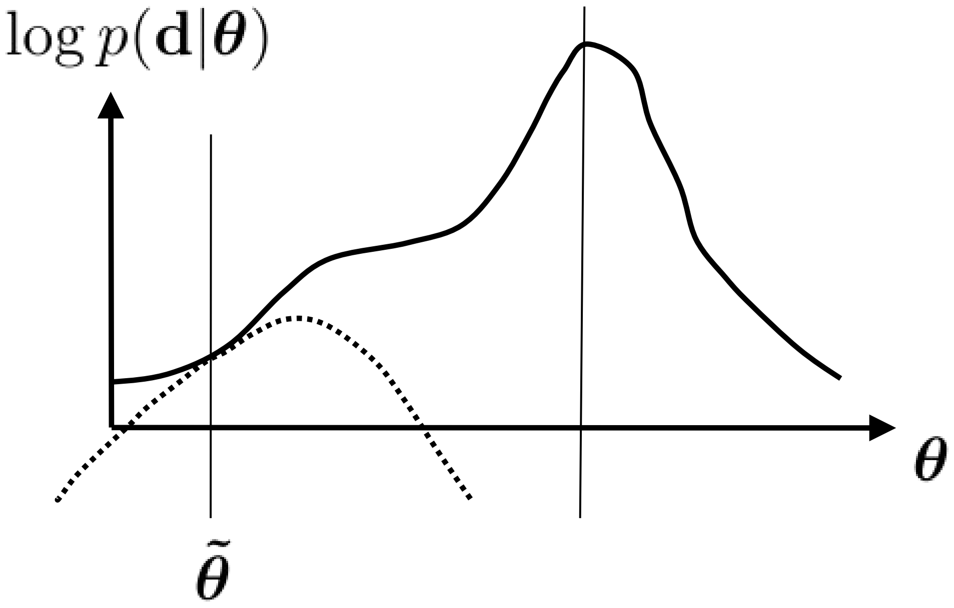


Improving the lower bound

$$\frac{\partial}{\partial \boldsymbol{\theta}} = 0 \qquad \Rightarrow \qquad \tilde{\mu}_k \leftarrow \frac{\sum_n w_k^n (d_n - \sum_m \tilde{c}_m \phi_m^n)}{\sum_n w_k^n}$$

$$\tilde{\sigma}_k^2 \leftarrow \frac{\sum_n w_k^n (d_n - \sum_m \tilde{c}_m \phi_m^n - \tilde{\mu}_k)^2}{\sum_n w_k^n}$$

$$\tilde{\pi}_k \leftarrow \frac{\sum_n w_k^n}{N}$$

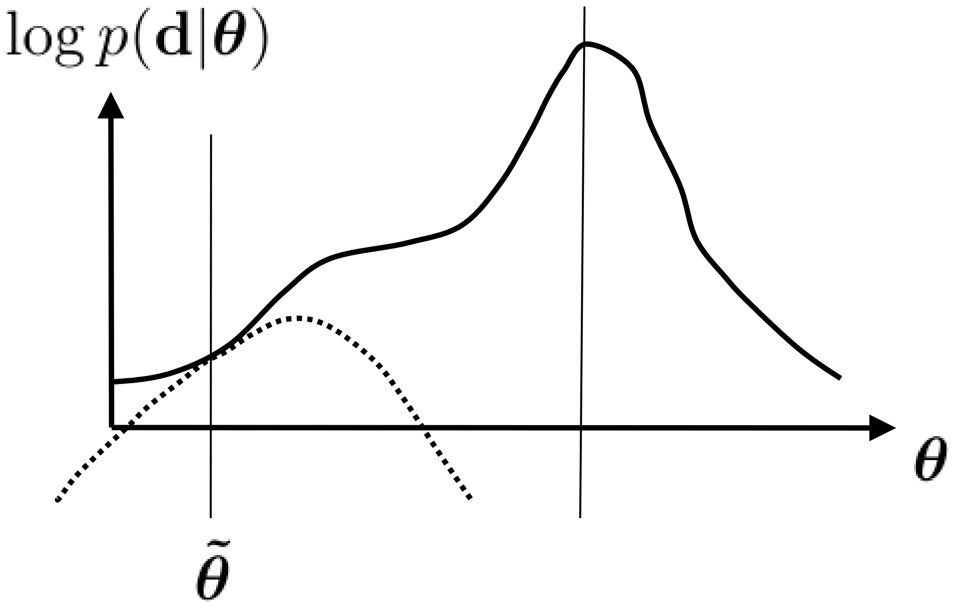


Improving the lower bound

$$\frac{\partial}{\partial \boldsymbol{\theta}} = 0 \qquad \Rightarrow \qquad \tilde{\mu}_k \leftarrow \frac{\sum_n w_k^n (d_n - \sum_m \tilde{c}_m \phi_m^n)}{\sum_n w_k^n}$$

$$\tilde{\sigma}_k^2 \leftarrow \frac{\sum_n w_k^n (d_n - \sum_m \tilde{c}_m \phi_m^n - \tilde{\mu}_k)^2}{\sum_n w_k^n}$$

$$\tilde{\pi}_k \leftarrow \frac{\sum_n w_k^n}{N}$$



Improving the lower bound (cont.)

$$\tilde{\mathbf{c}} \leftarrow (\mathbf{\Phi}^{\mathrm{T}}\mathbf{S}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{S}\mathbf{r}$$

- cf. linear regression
- smoothing operation

$$\mathbf{\Phi} = \begin{pmatrix} \phi_1^1 & \phi_2^1 & \cdots & \phi_M^1 \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^N & \phi_2^N & \cdots & \phi_M^N \end{pmatrix}$$

$$s_k^n = \frac{w_k^n}{\tilde{\sigma}_k^2}, \quad s_n = \sum_k s_k^n, \quad \boldsymbol{S} = \mathrm{diag}(s_n), \quad \tilde{d}_n = \frac{\sum_k s_k^n \tilde{\mu}_k}{\sum_k s_k^n}, \quad \boldsymbol{r} = \begin{pmatrix} d_1 - \tilde{d}_1 \\ \vdots \\ d_N - \tilde{d}_N \end{pmatrix}$$

Aalto-yliopisto
Aalto-universitetet
Aalto University

Improving the lower bound (cont.)

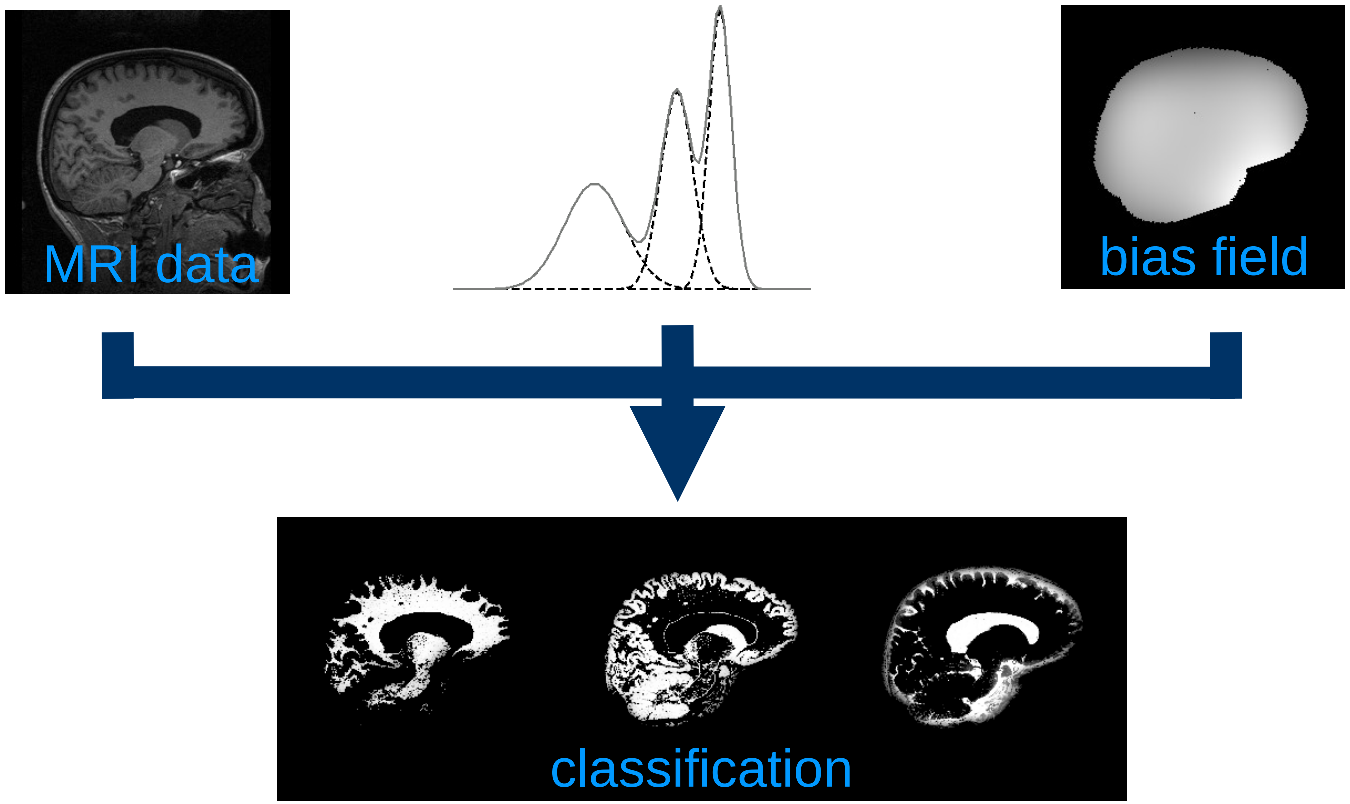
$$\tilde{\mathbf{c}} \leftarrow (\mathbf{\Phi}^{\mathrm{T}}\mathbf{S}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{S}\mathbf{r}$$

- cf. linear regression
- smoothing operation

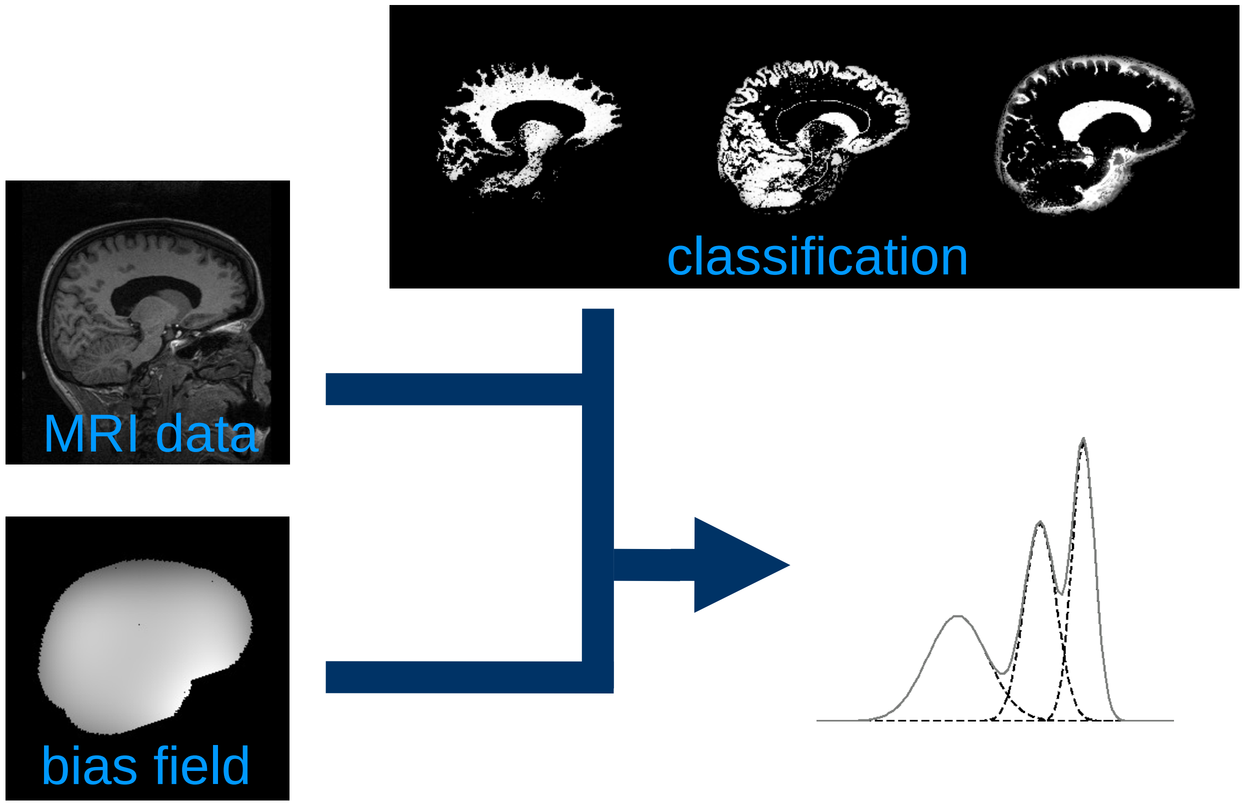
$$\mathbf{\Phi} = \begin{pmatrix} \phi_1^1 & \phi_2^1 & \cdots & \phi_M^1 \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1^N & \phi_2^N & \cdots & \phi_M^N \end{pmatrix}$$

$$s_k^n = \frac{w_k^n}{\tilde{\sigma}_k^2}, \quad s_n = \sum_k s_k^n, \quad \boldsymbol{S} = \mathrm{diag}(s_n), \quad \tilde{d}_n = \frac{\sum_k s_k^n \tilde{\mu}_k}{\sum_k s_k^n}, \quad \boldsymbol{r} = \begin{pmatrix} d_1 - \tilde{d}_1 \\ \vdots \\ d_N - \tilde{d}_N \end{pmatrix}$$

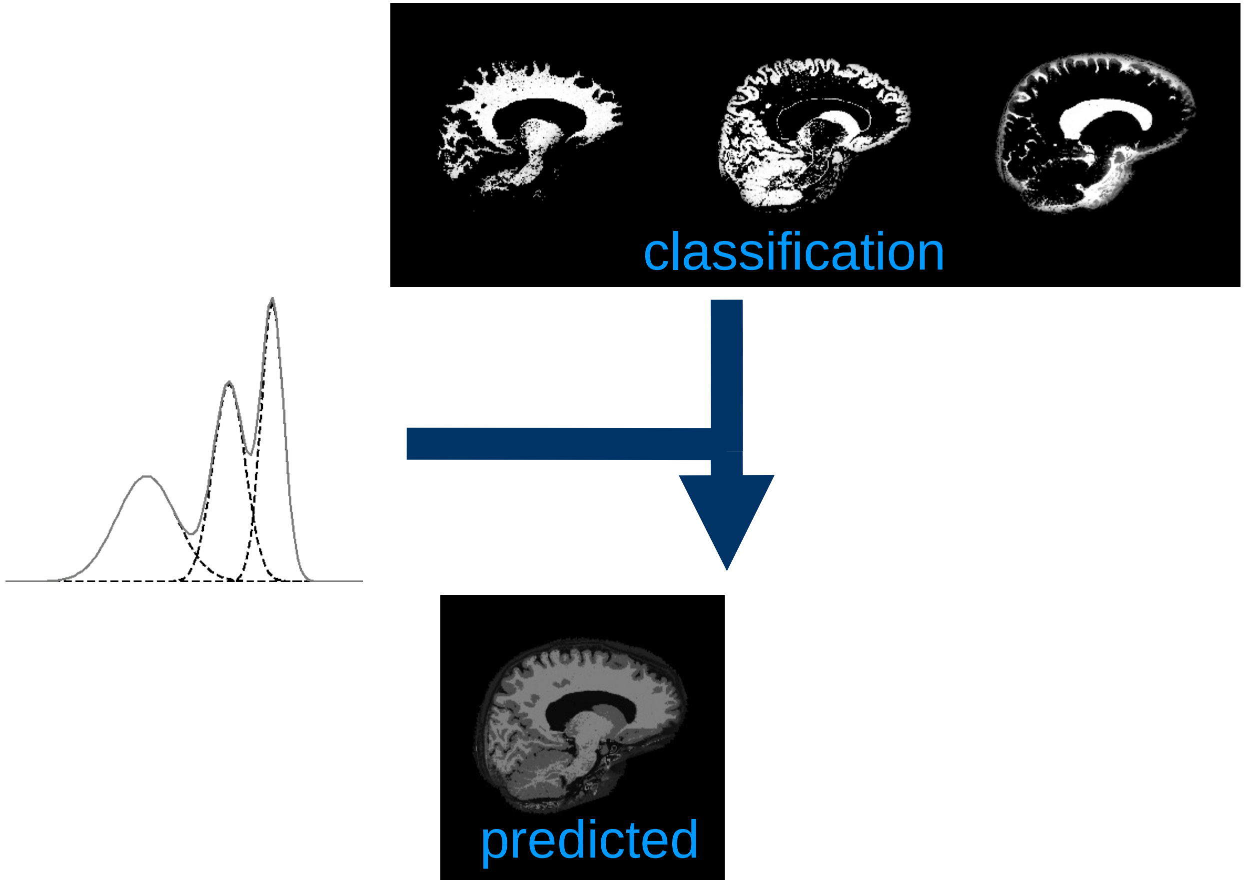
E-step



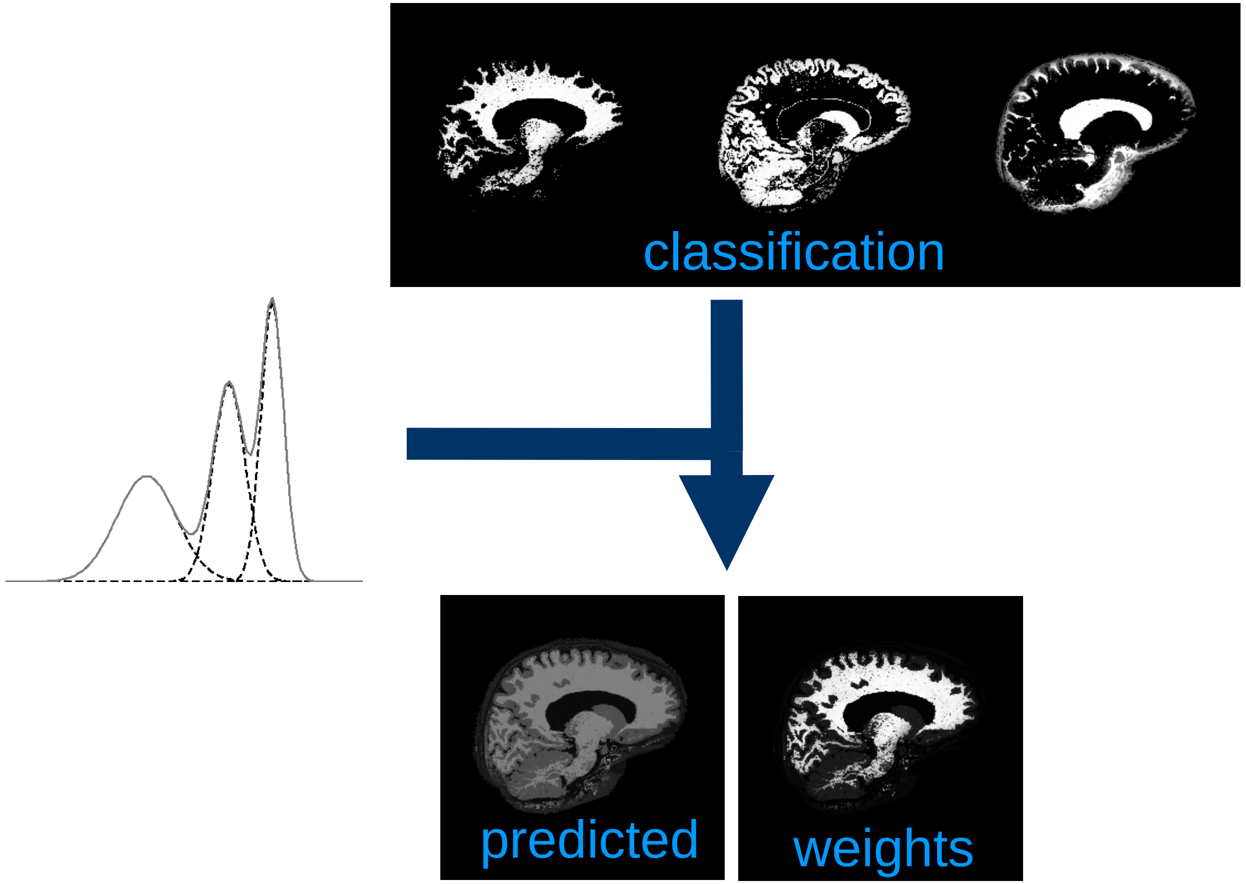
M-step part 1: distribution estimation



M-step part 2: bias field estimation

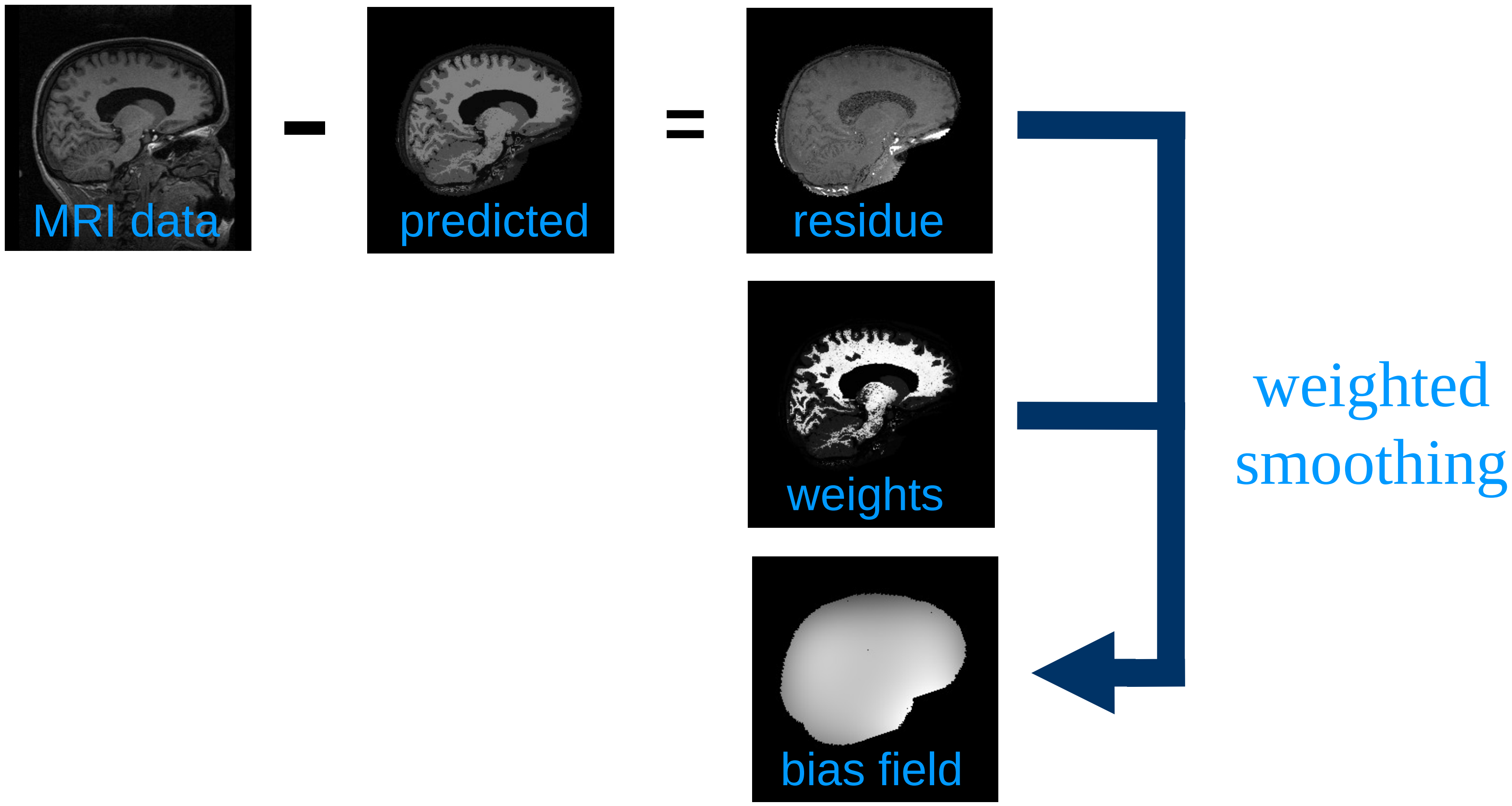


M-step part 2: bias field estimation

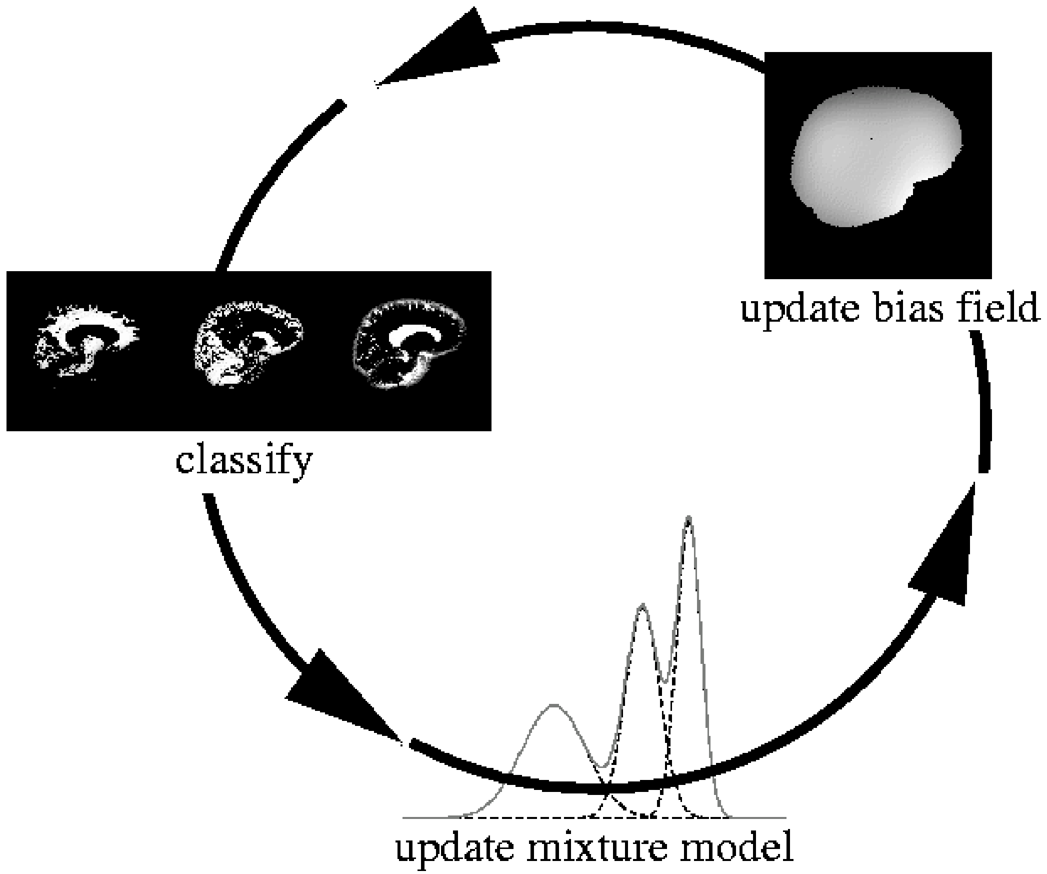


Aalto-yliopisto
Aalto-universitetet
Aalto University

M-step part 2: bias field estimation



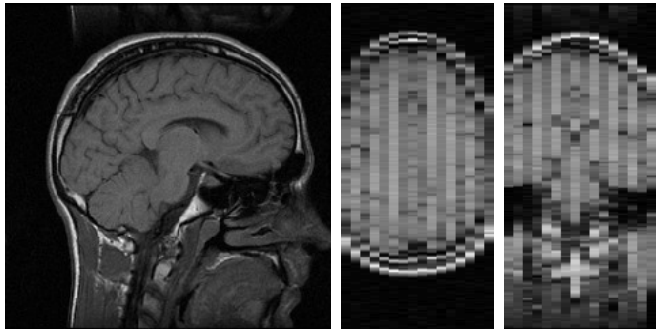
Parameter optimizer summarized



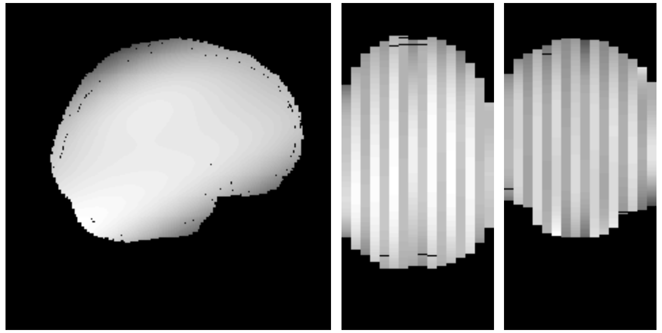
– Repeatedly apply closed-form parameter updates

– Each iteration improves the likelihood

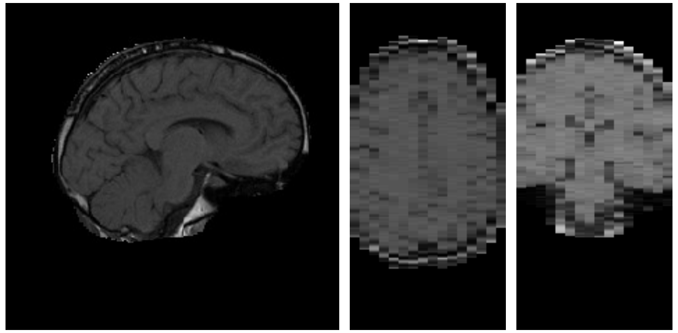
Example



MRI data



Estimated bias field



Bias-corrected MRI data

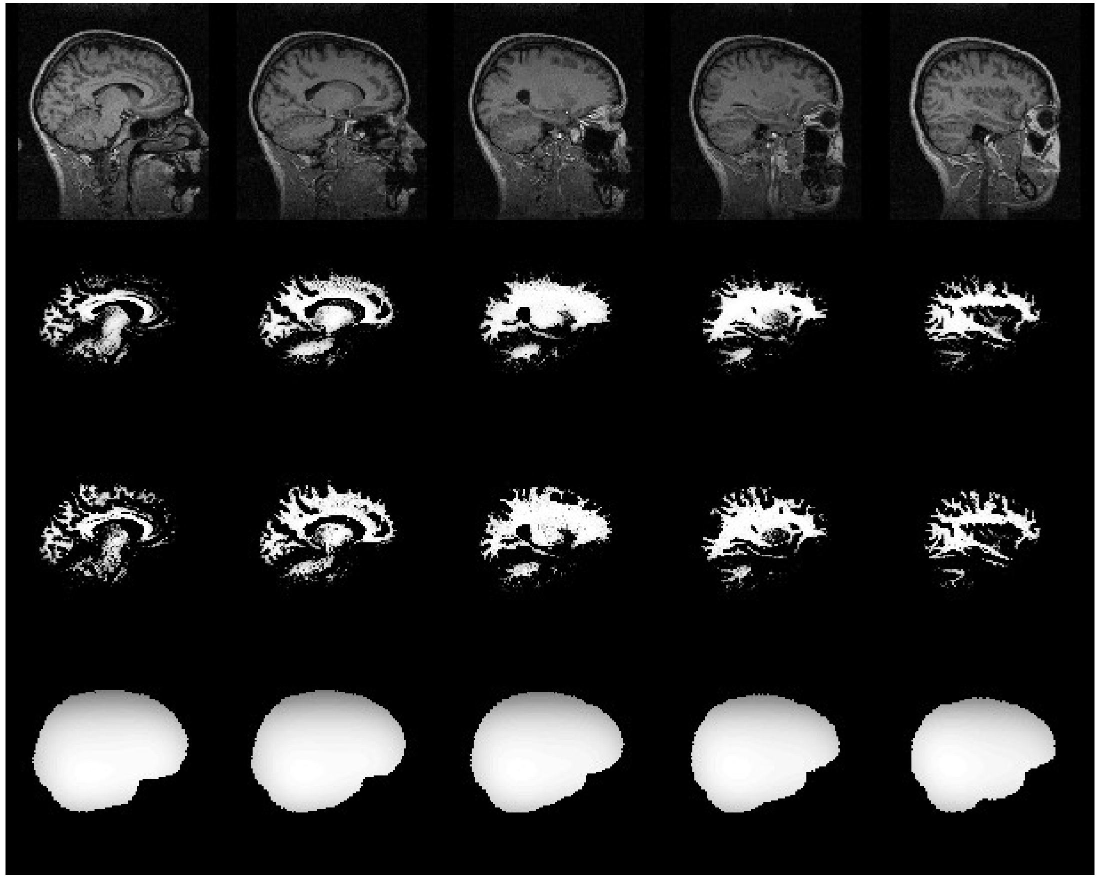
Example

MRI data

White matter without
bias field model

White matter with
bias field model

Estimated bias field



Aalto-yliopisto
Aalto-universitetet
Aalto University

Example

MRI data

White matter without
bias field model

White matter with
bias field model

Estimated bias field