

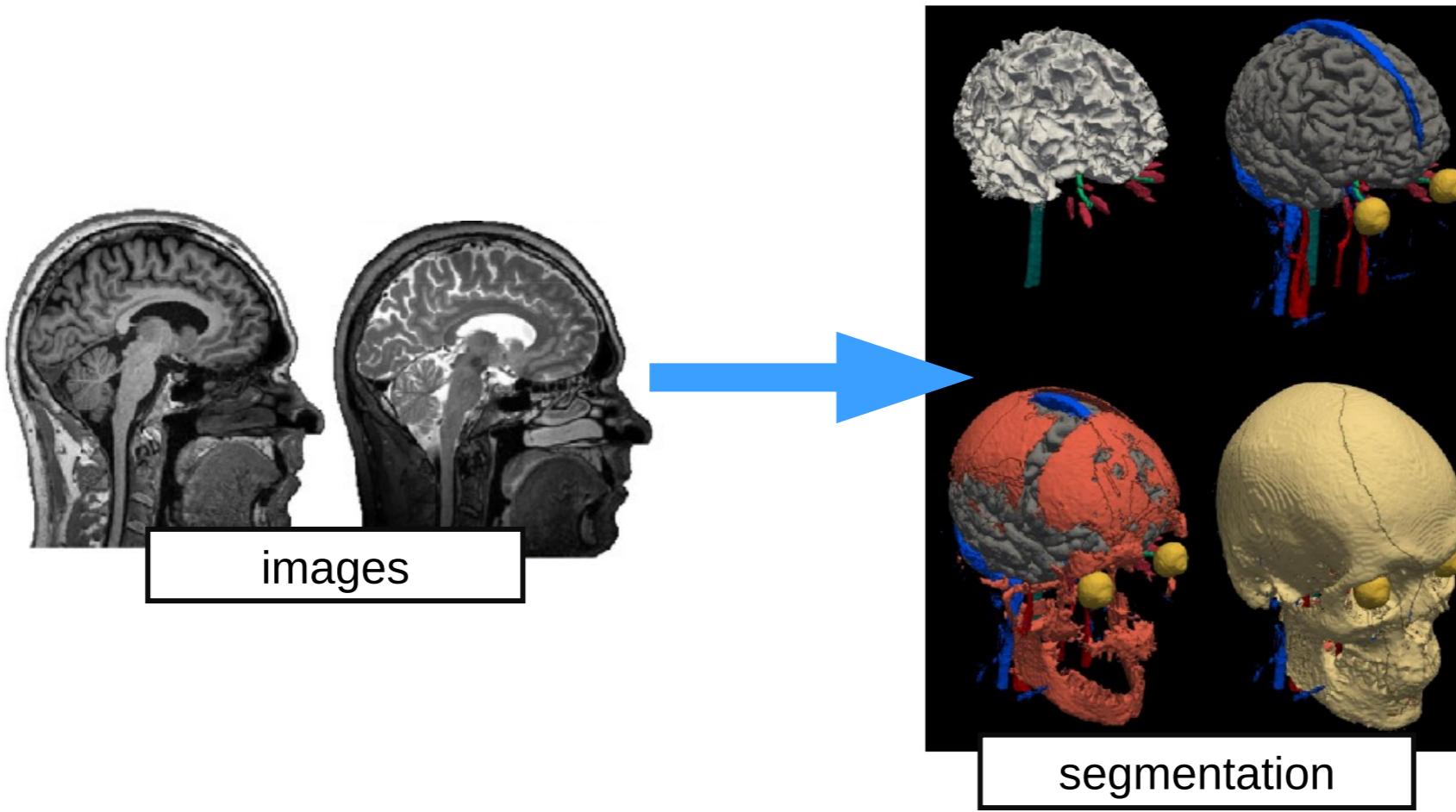
Model-based Segmentation: Part I

A''

Aalto-yliopisto
Aalto-universitetet
Aalto University

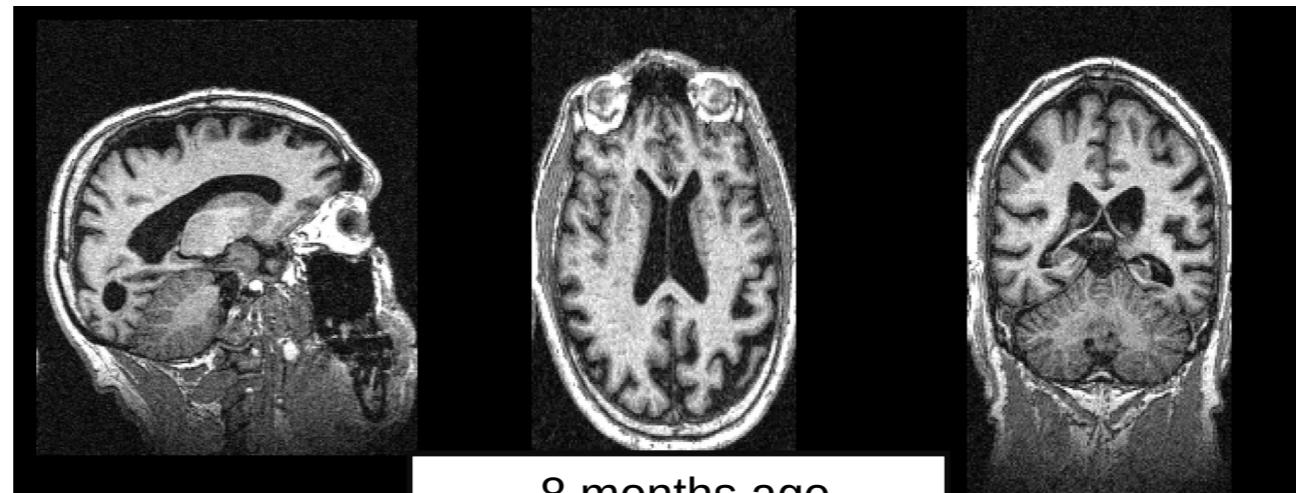
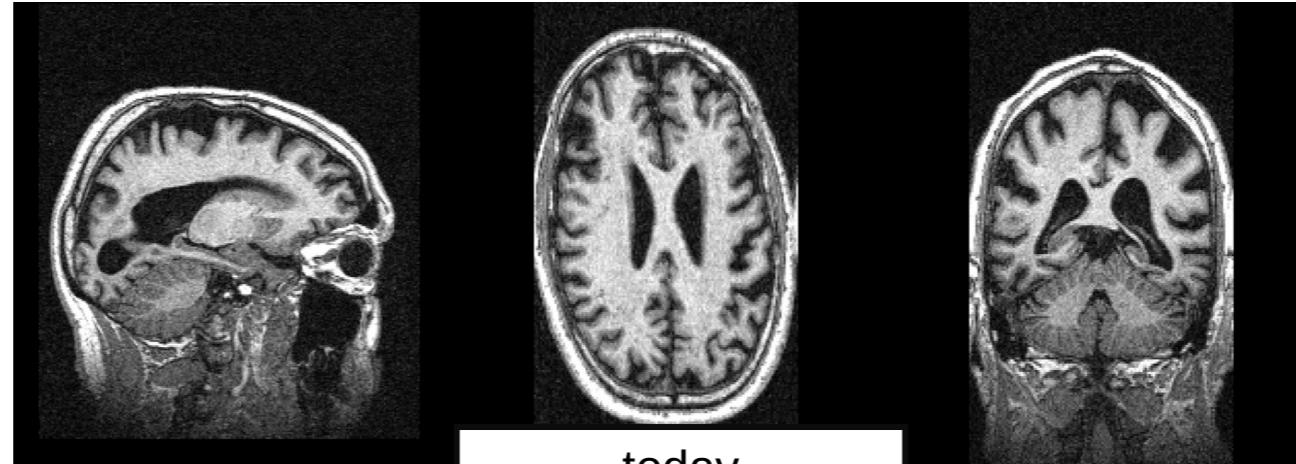
Medical Image Analysis
Koen Van Leemput
Fall 2023

Segmentation



- individual patients (diagnosis, treatment planning, follow-up)
- group studies (drug trials, elucidating disease mechanisms)

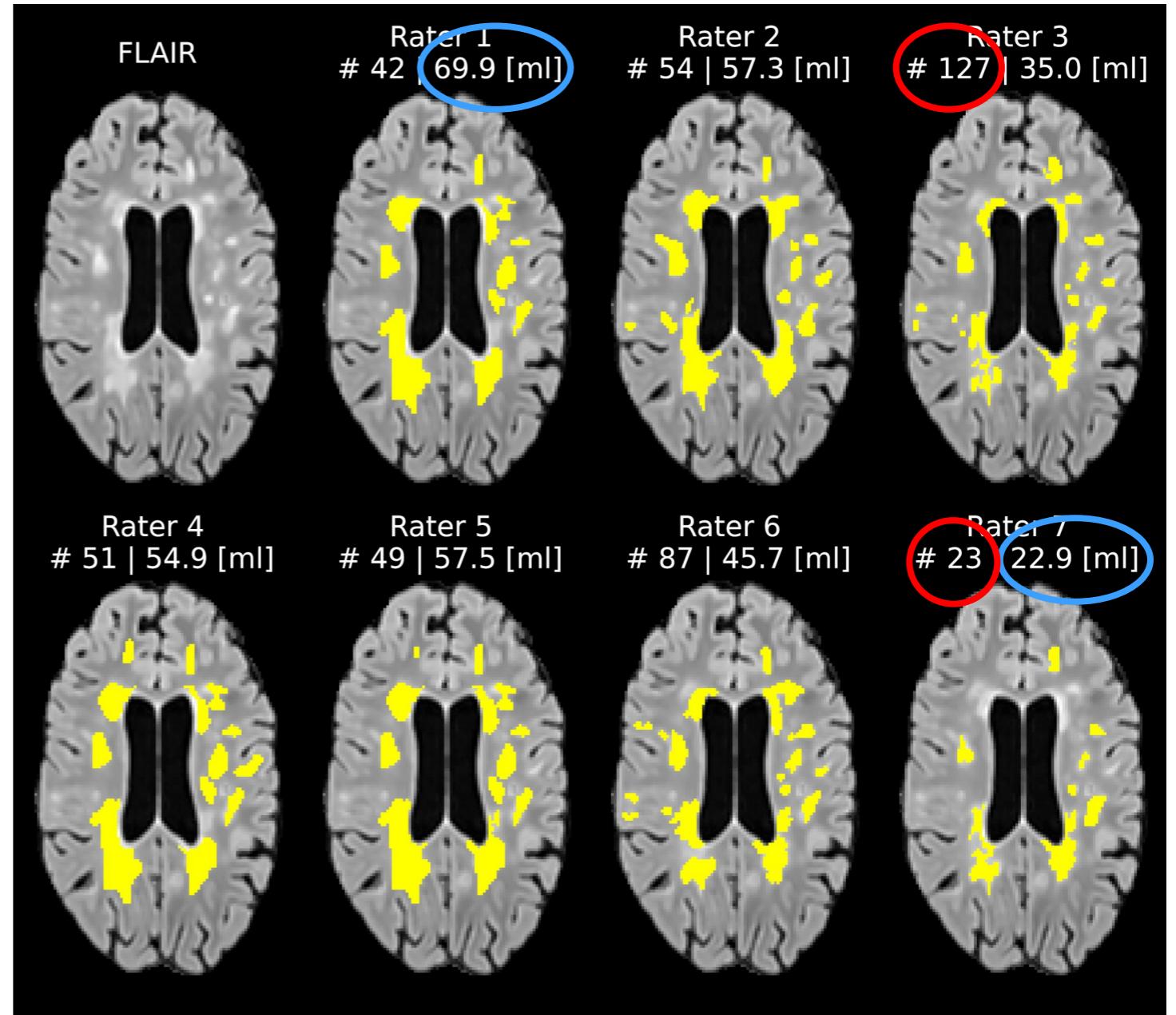
Exposing the “unseeable”



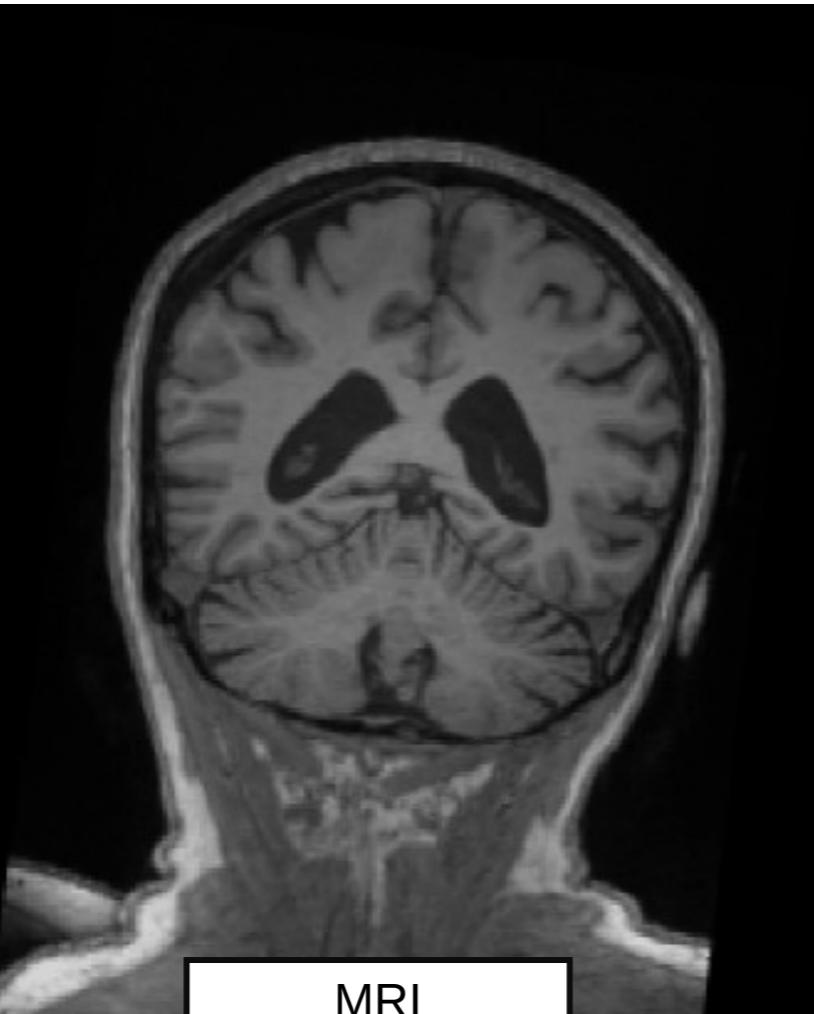
Measuring more consistently

Quantifying lesions in multiple sclerosis (MS):

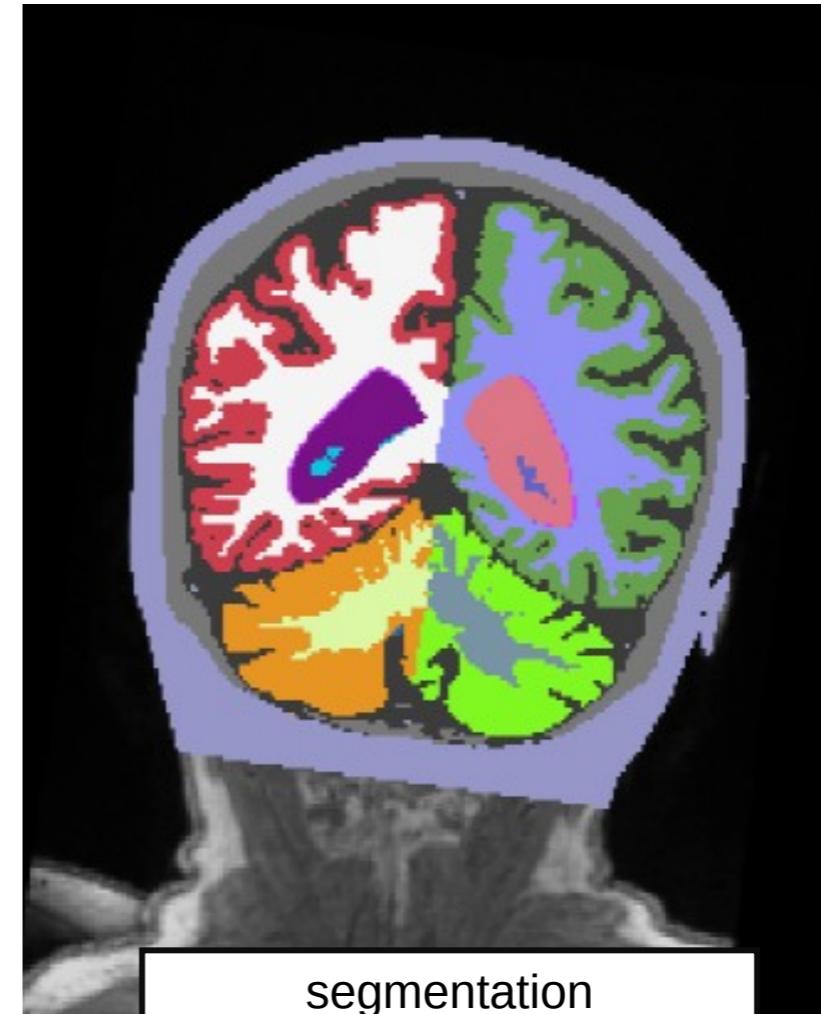
- number (#)
- volume (ml)



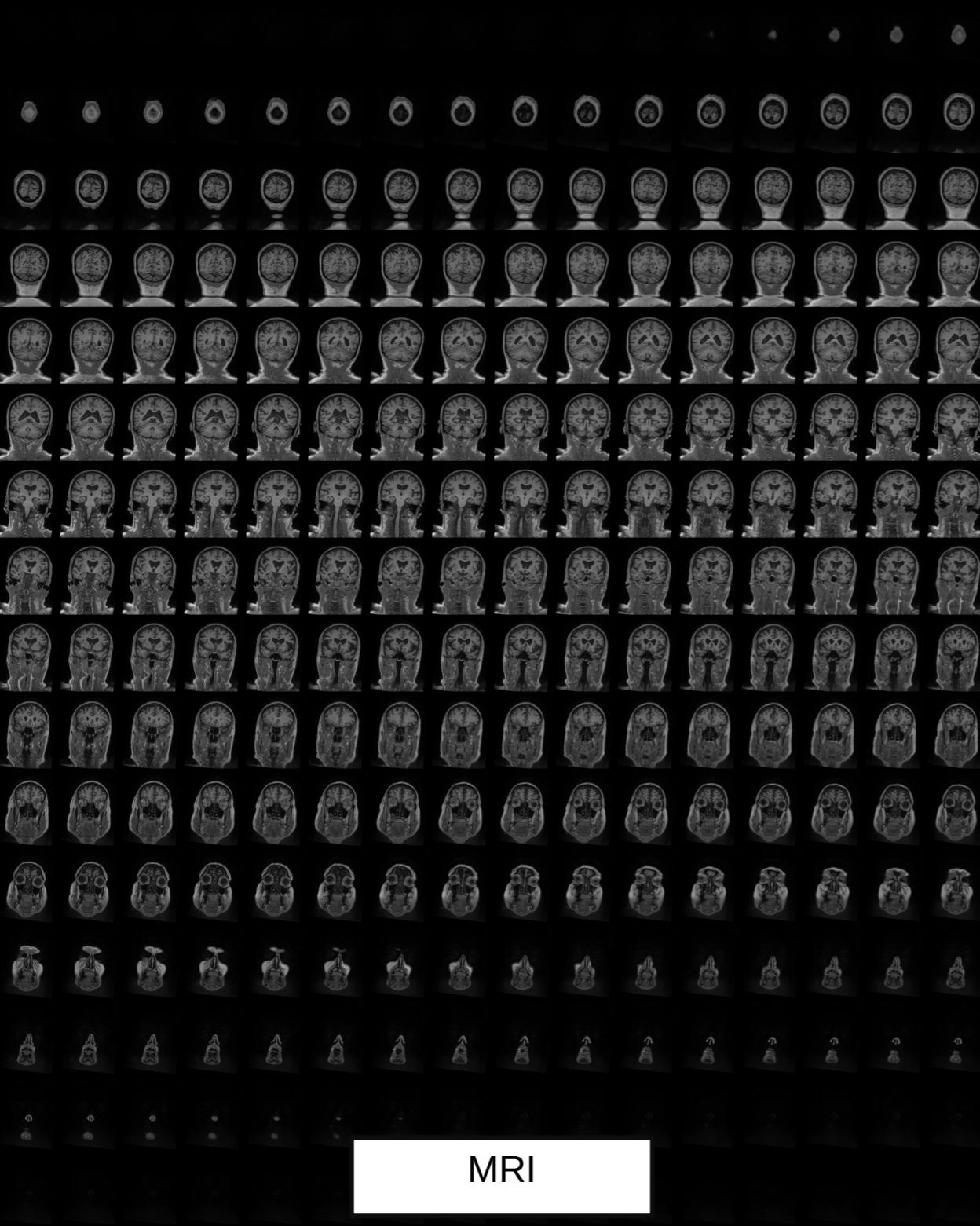
Analyzing images faster



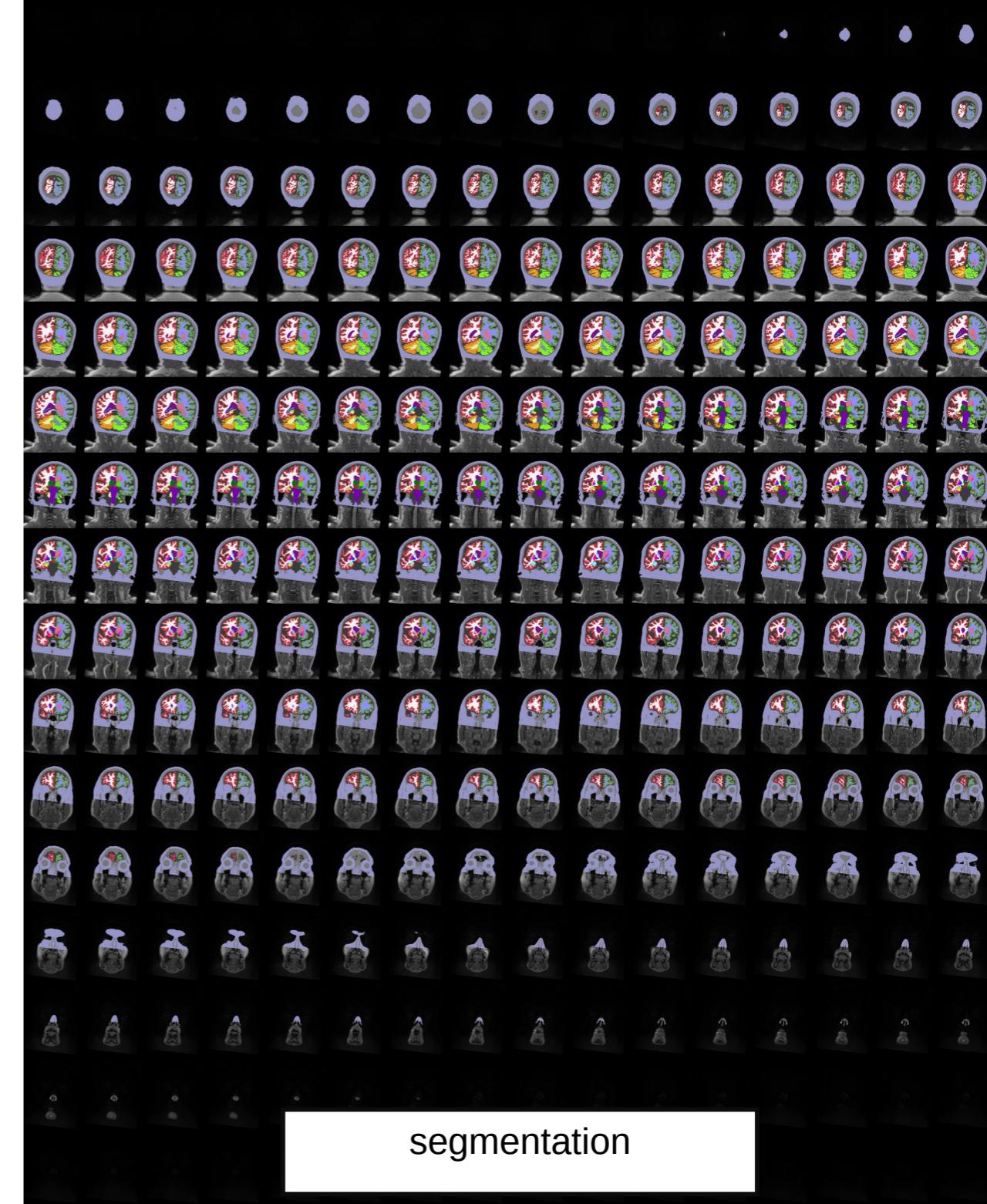
MRI



segmentation

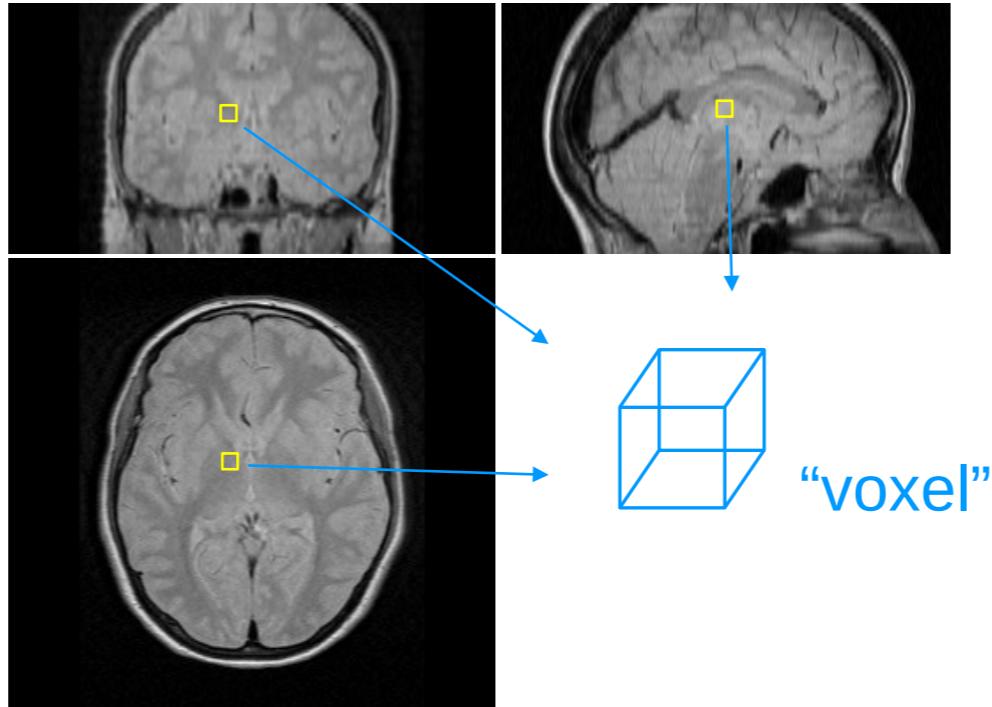


MRI



segmentation

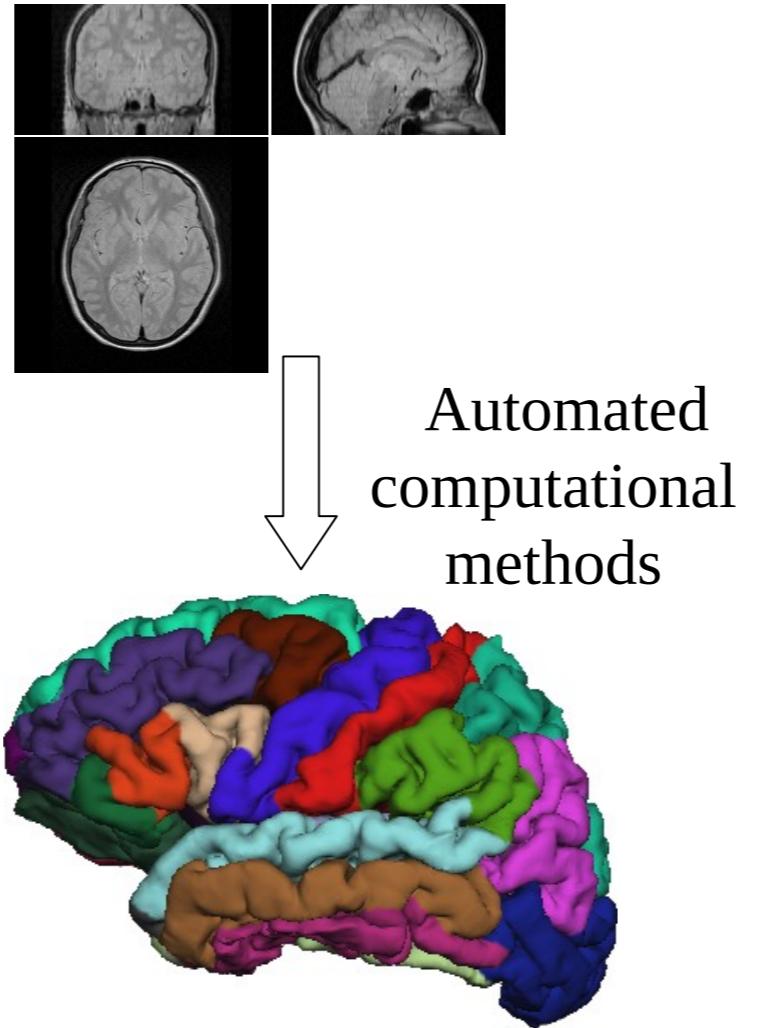
Voxel-based segmentation



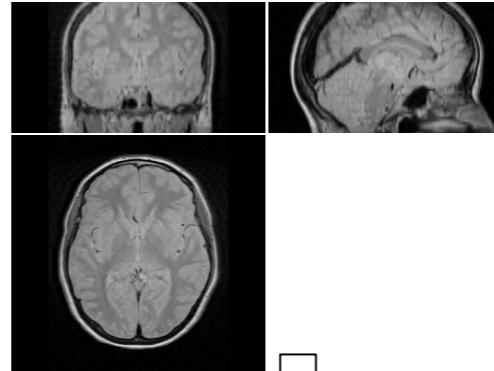
Determine which anatomical structure each voxel belongs to:

- think “LEGO bricks”
- outer surfaces can easily be extracted if needed

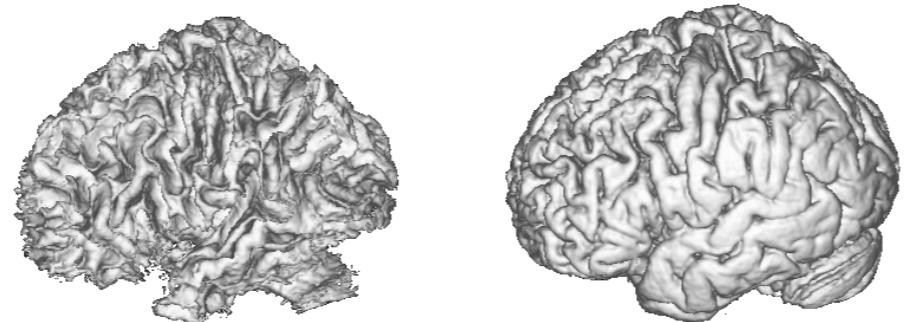
This and next lecture



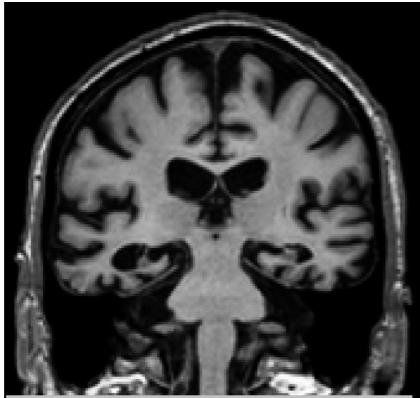
This and next lecture



Automated
computational
methods



The problem to be solved

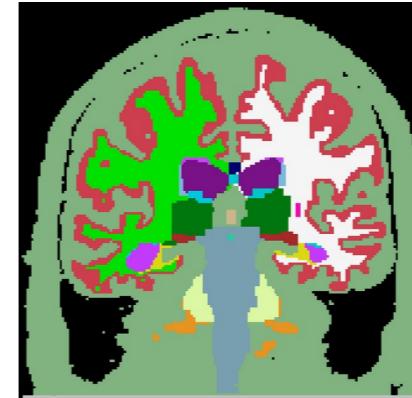


MRI image \mathbf{d}

N voxels

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

d_n : intensity in voxel n



Label image \mathbf{l}

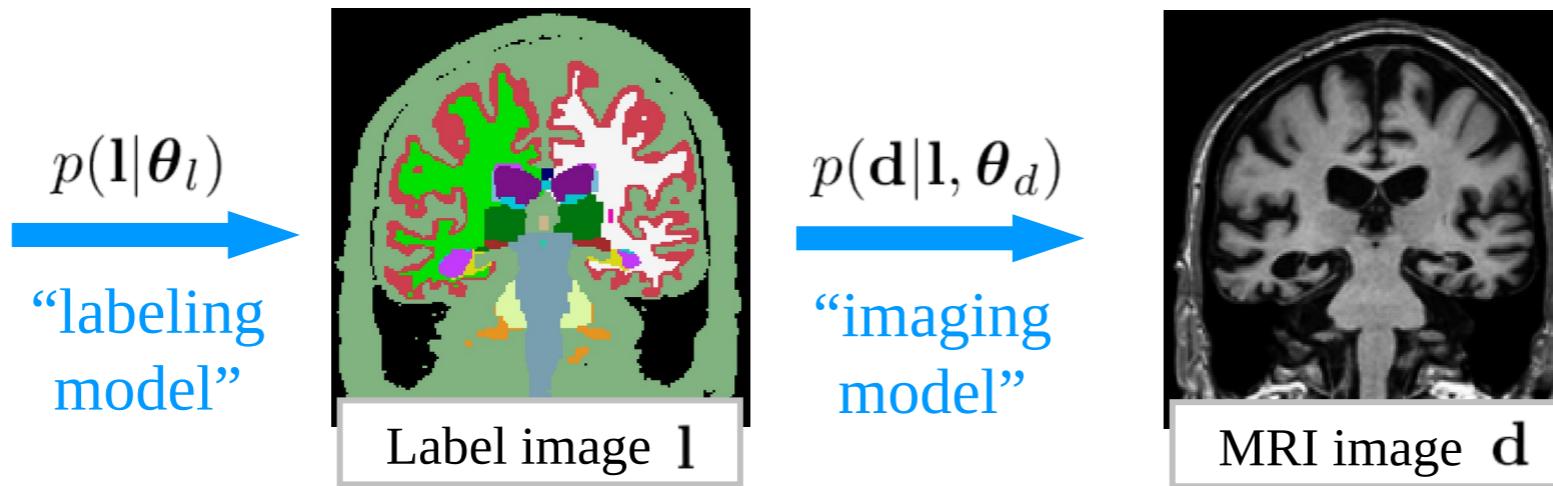
$$\mathbf{l} = (l_1, \dots, l_N)^T$$

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

K : number of classes

One solution: generative modeling

- Formulate a statistical model of how a medical image is formed



- The model depends on some parameters $\theta = (\theta_l^T, \theta_d^T)^T$
- Appropriate values $\hat{\theta}$ are assumed to be known for now...

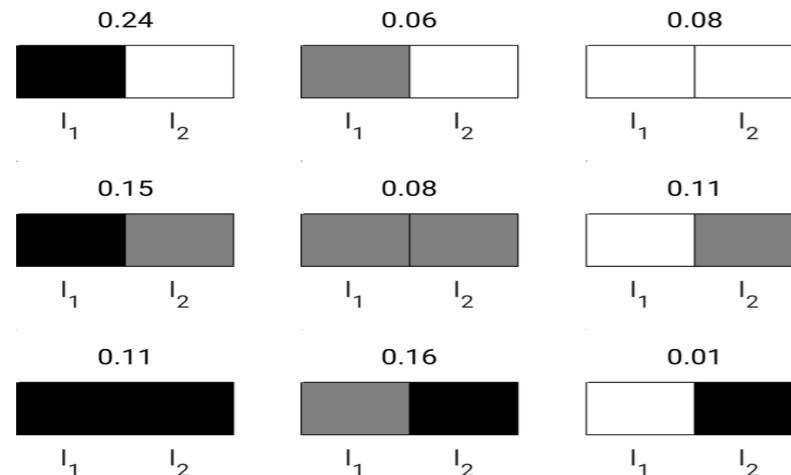
Toy example

$N = 2$ voxels

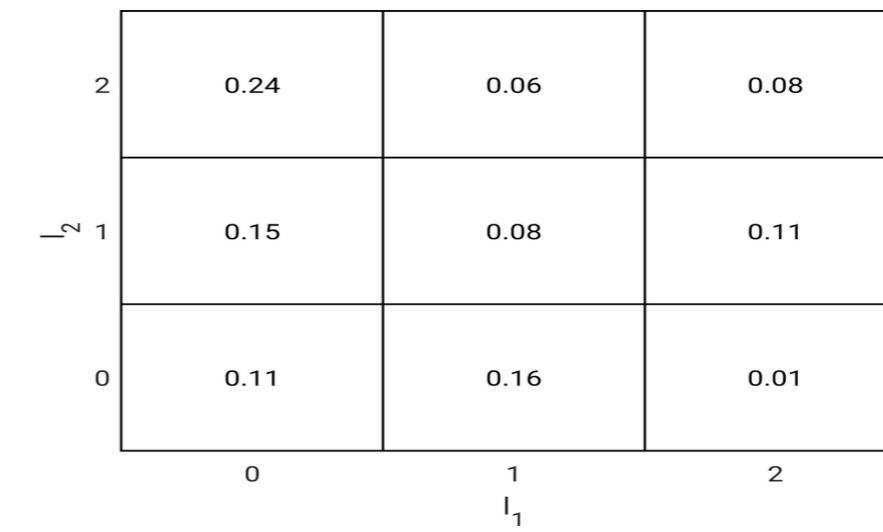
$K = 3$ classes

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix}$$

$$p(\mathbf{l}) = p(l_1, l_2)$$



$$p(\mathbf{l}) = p(l_1, l_2)$$



labeling model

Toy example

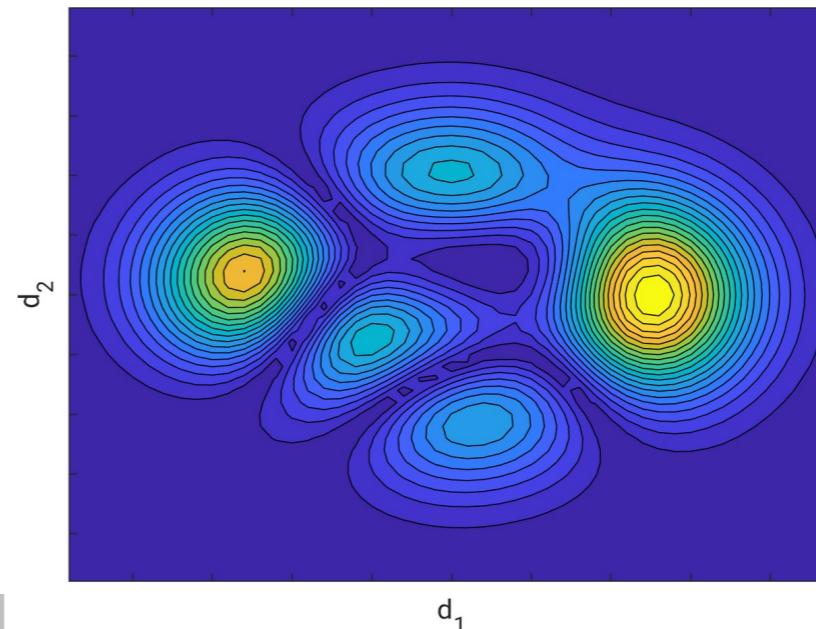
$N = 2$ voxels

$K = 3$ classes

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

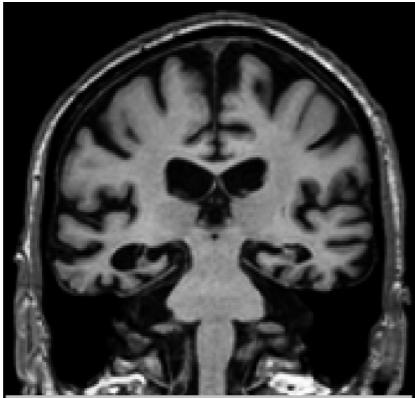


$$p(\mathbf{d}|\mathbf{l}) = p(d_1, d_2 | l_1, l_2)$$

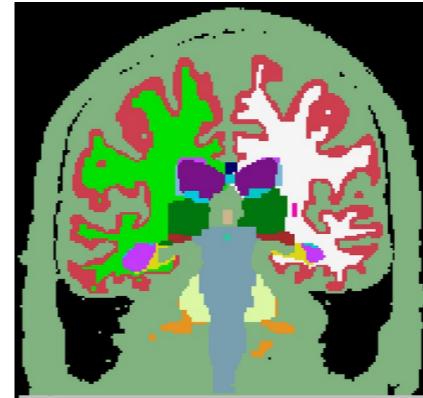


imaging model

Segmentation = inverse problem

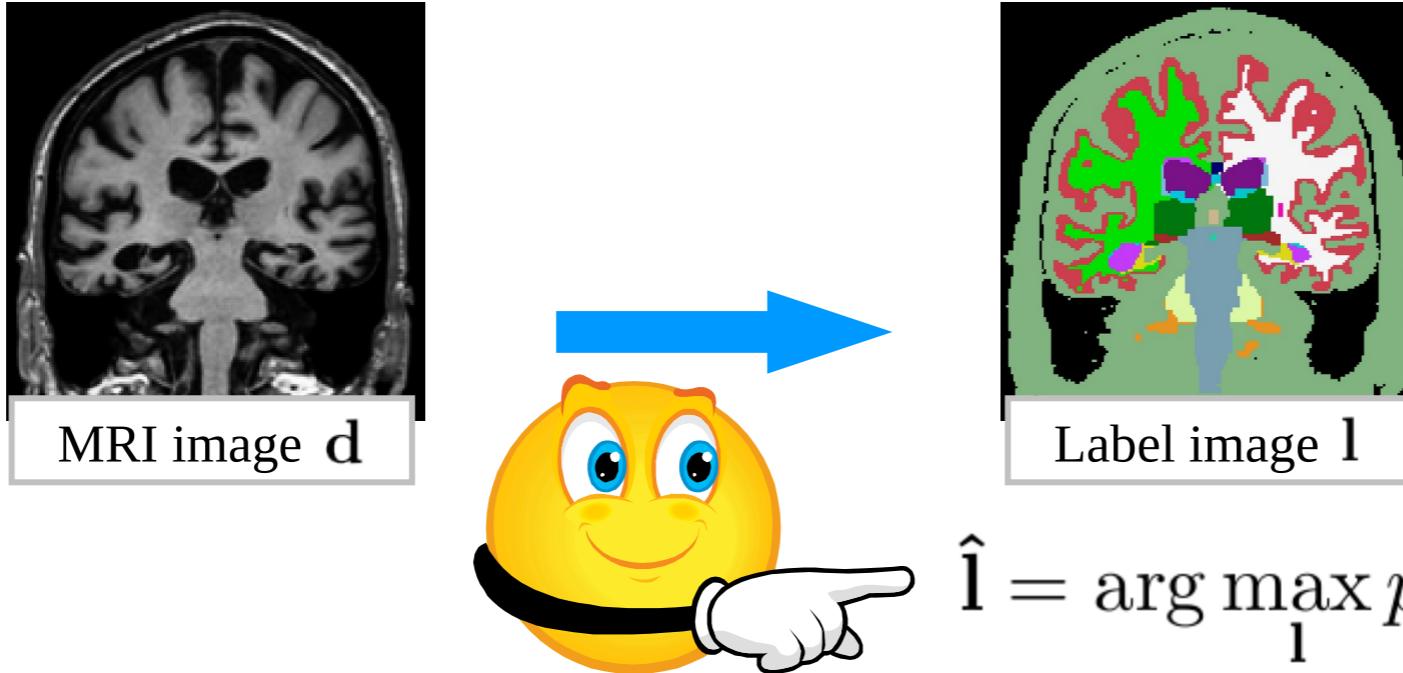


MRI image **d**



Label image **l**

Segmentation = inverse problem

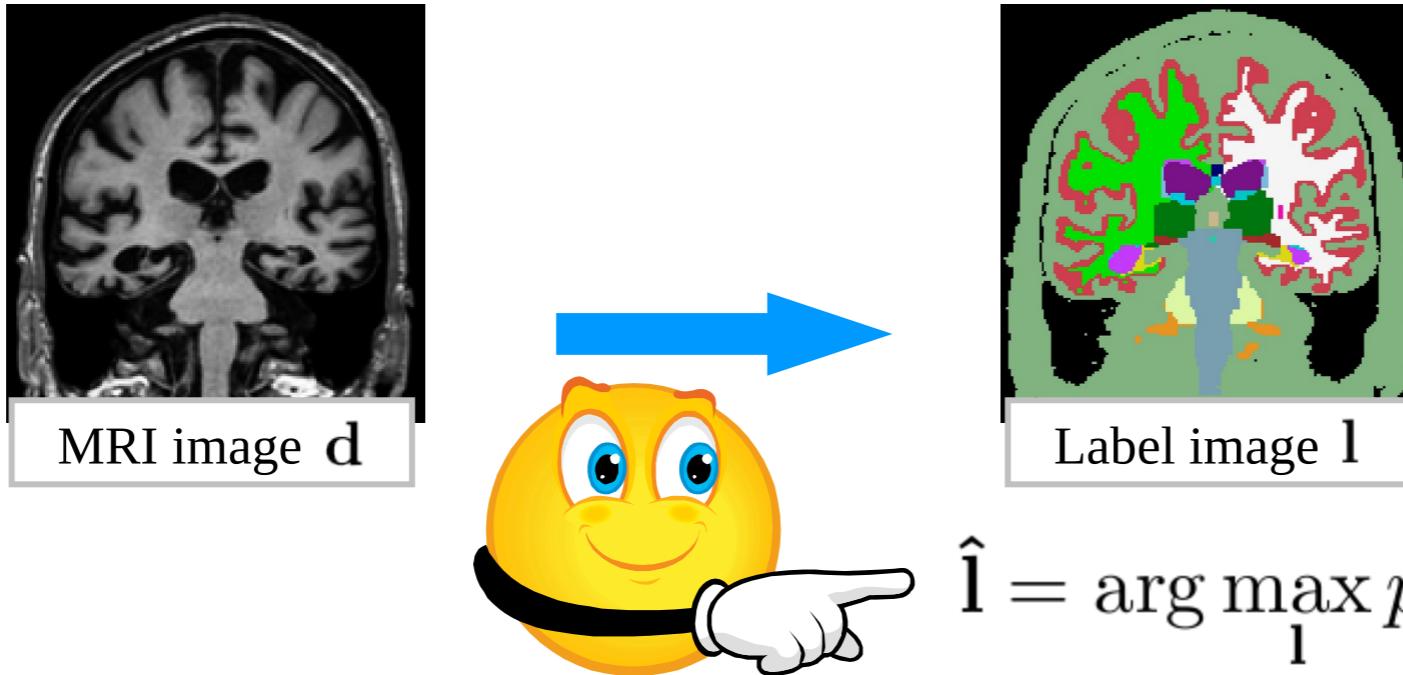


$$\hat{\mathbf{l}} = \arg \max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

The posterior distribution $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$ is given by Bayes rule:

$$p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l)}{p(\mathbf{d}|\hat{\boldsymbol{\theta}})}$$

Segmentation = inverse problem



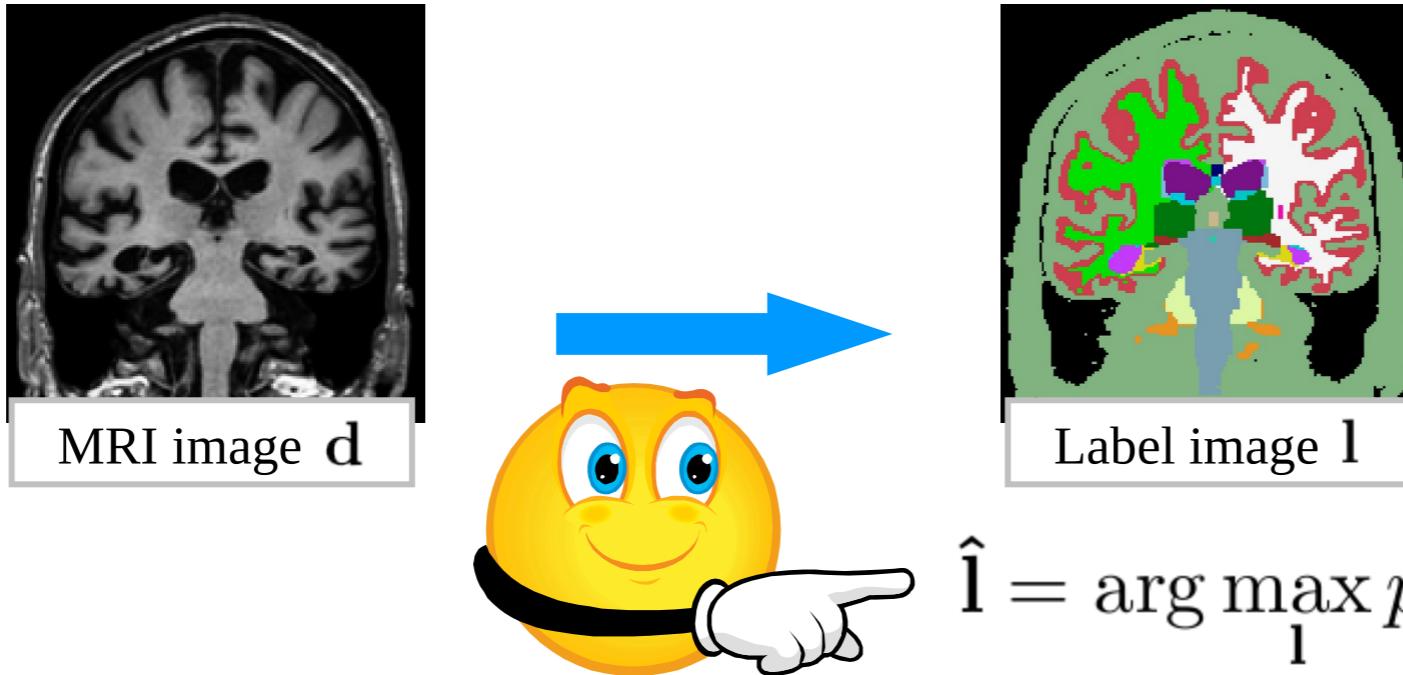
$$\hat{\mathbf{l}} = \arg \max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

The posterior distribution $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$ is given by Bayes rule:

$$p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l)}{p(\mathbf{d}|\hat{\boldsymbol{\theta}})}$$

labeling model

Segmentation = inverse problem



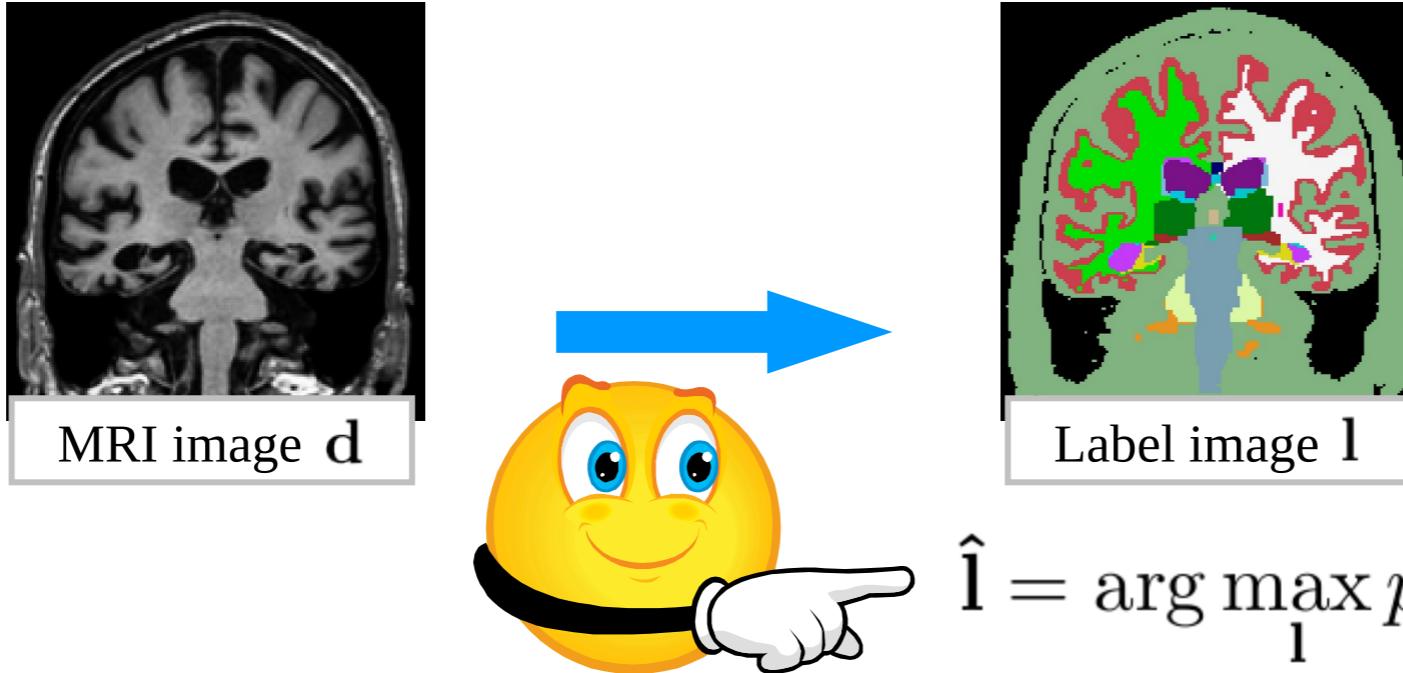
$$\hat{\mathbf{l}} = \arg \max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

The posterior distribution $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$ is given by Bayes rule:

imaging model

$$p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l)}{p(\mathbf{d}|\hat{\boldsymbol{\theta}})}$$

Segmentation = inverse problem

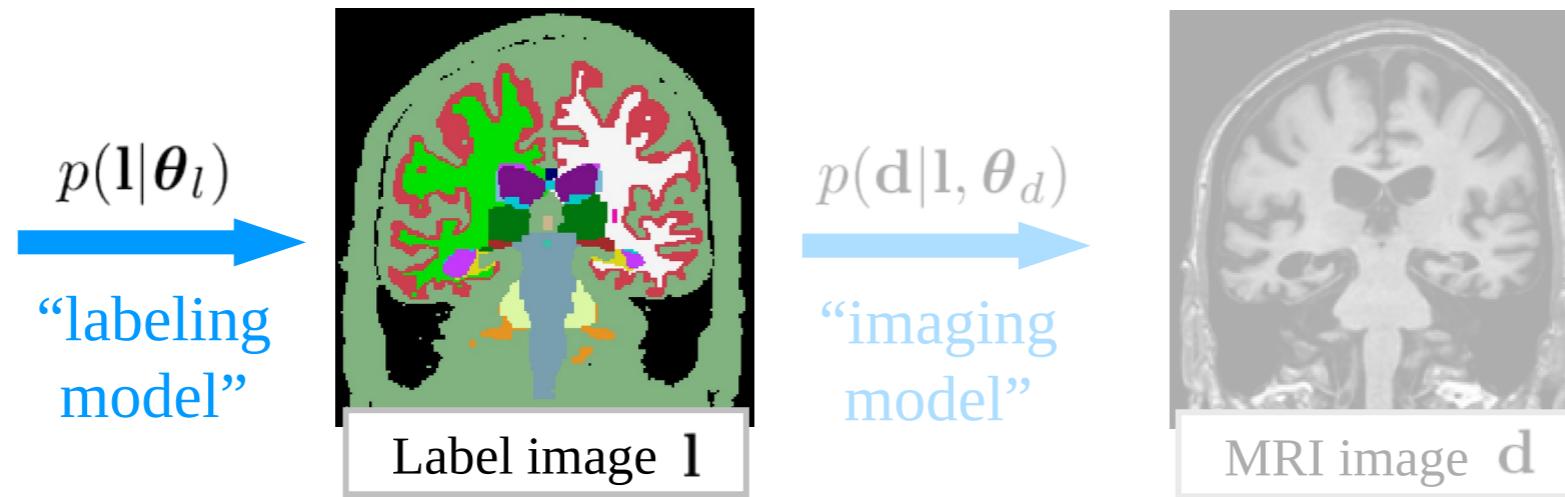


$$\hat{\mathbf{l}} = \arg \max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

The posterior distribution $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$ is given by Bayes rule:

$$p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l)}{p(\mathbf{d}|\hat{\boldsymbol{\theta}})} = \sum_{\mathbf{l}} p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l) \quad (\text{but not needed})$$

Gaussian mixture model



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

$$p(l|\theta_l) = \prod_n \pi_{l_n} , \quad \theta_l = (\pi_1, \dots, \pi_K)^T$$

Gaussian mixture model

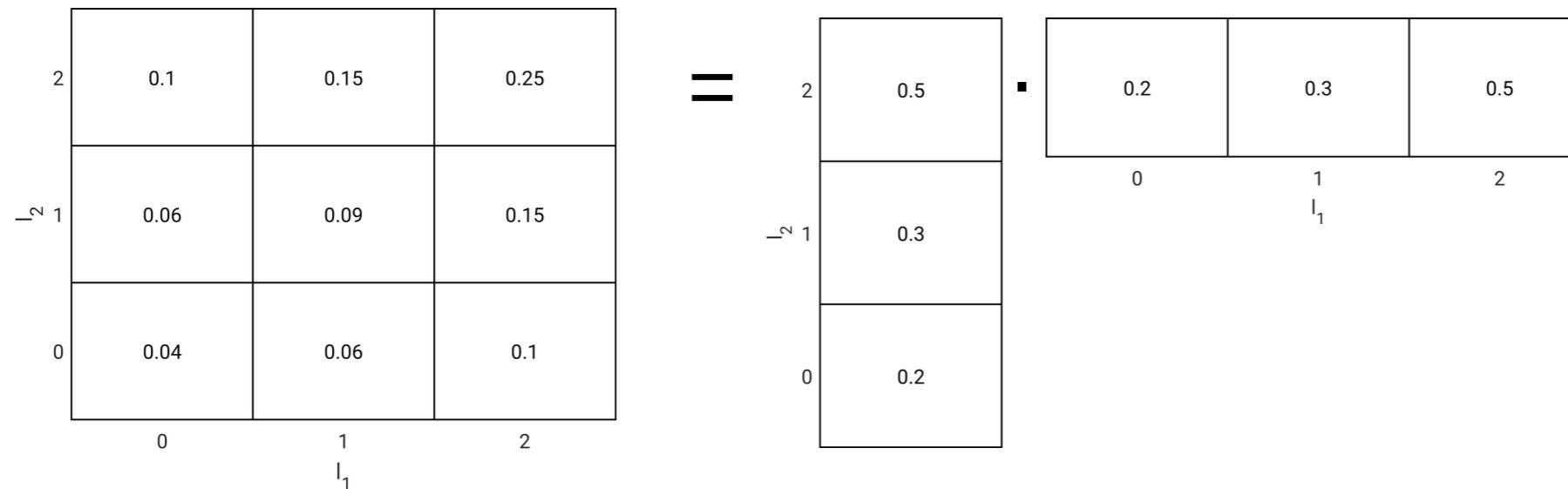
$N = 2$ voxels

$K = 3$ classes

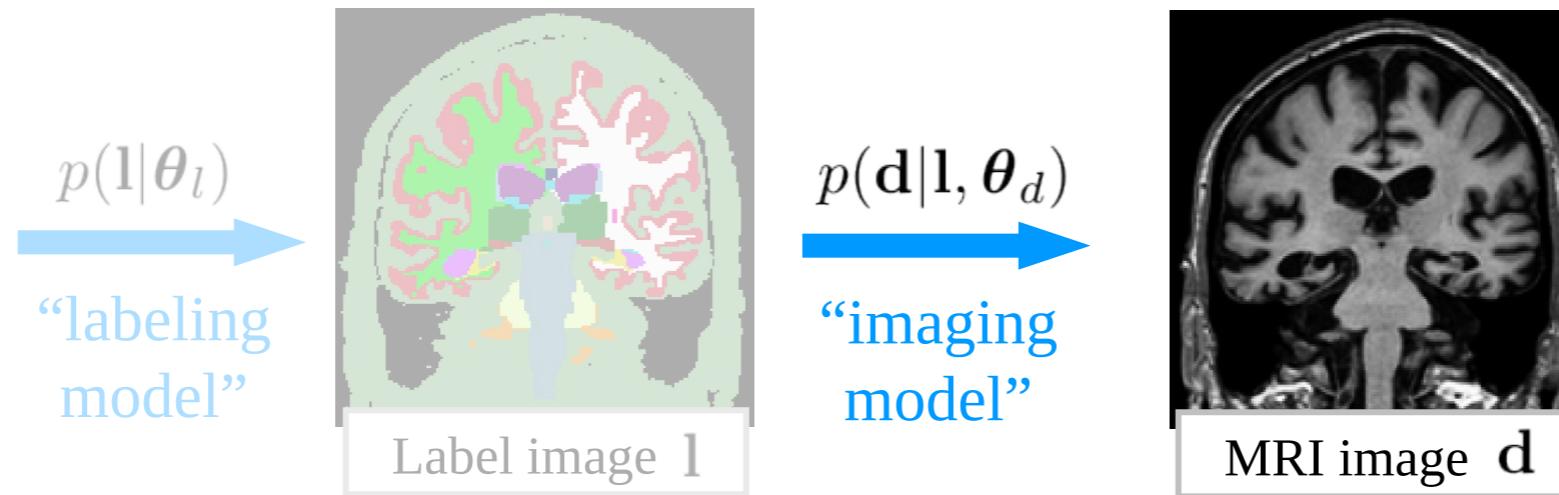
$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix}$$

independence between voxels

$$p(\mathbf{l}) = p(l_1, l_2) = p(l_2 | l_1)p(l_1)$$



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

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

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

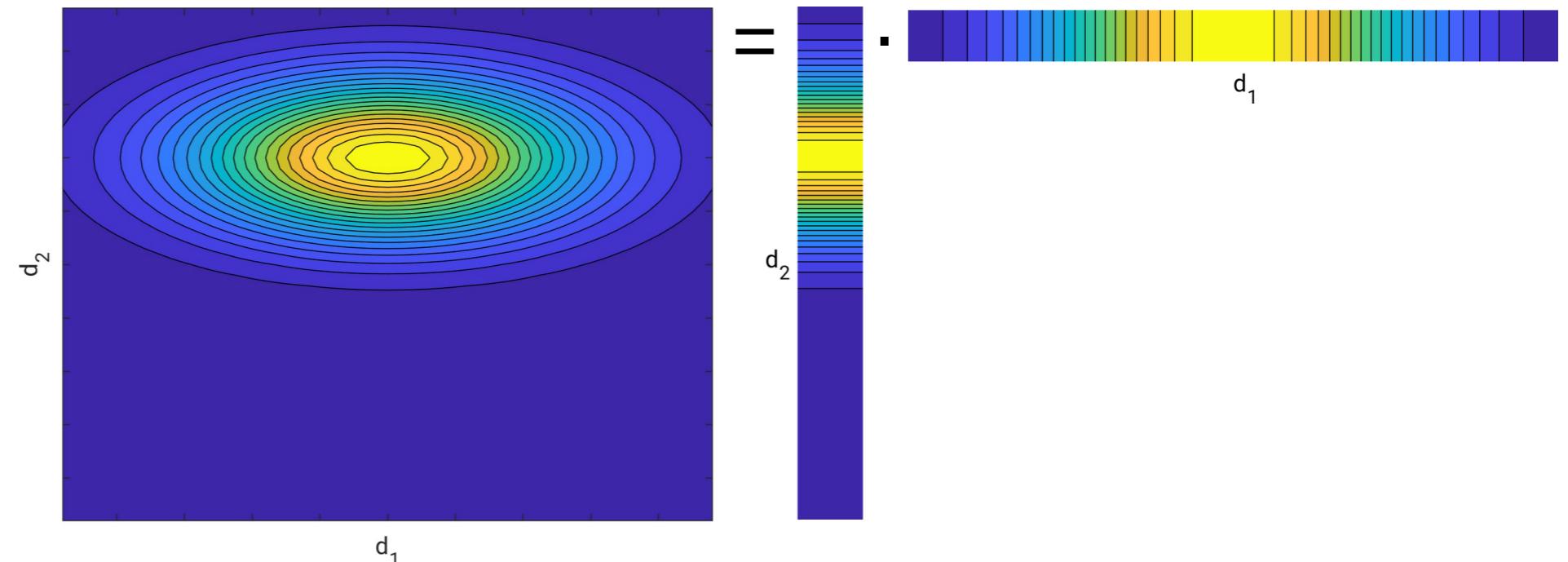
Toy example

$N = 2$ voxels

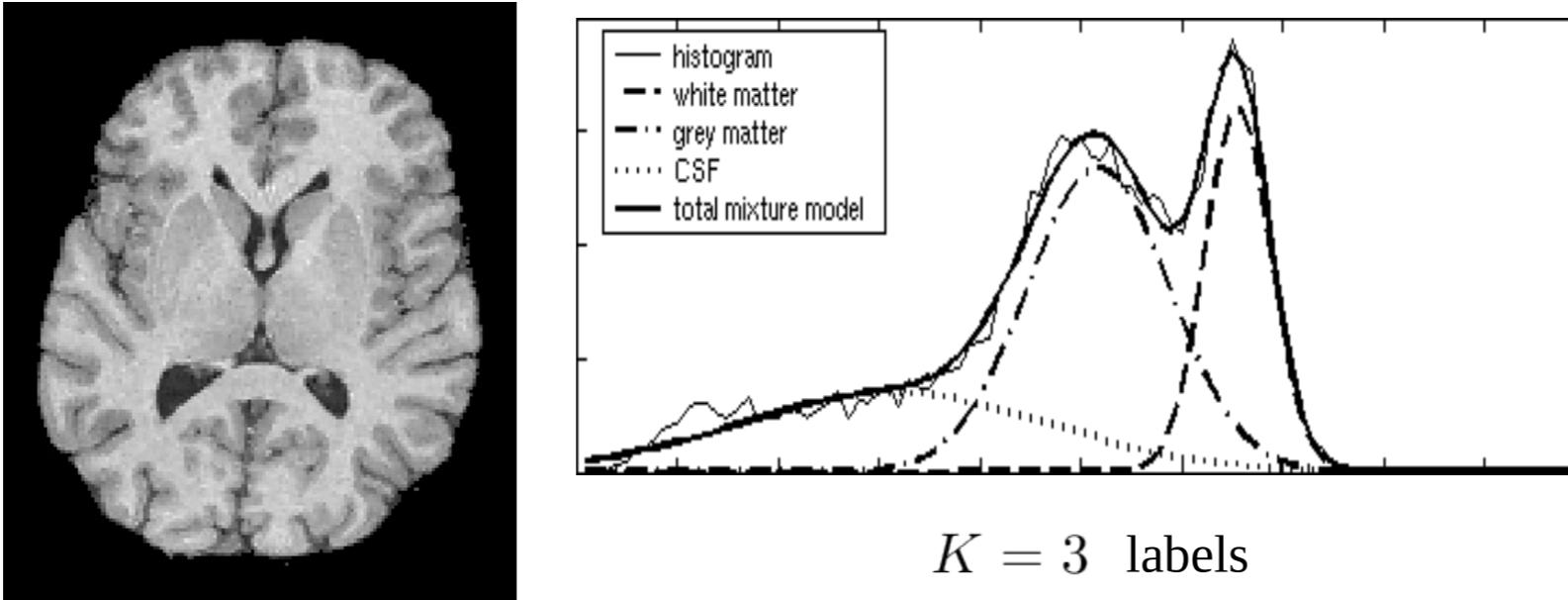
$K = 3$ classes

$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

$$p(\mathbf{d}|l) = p(d_1, d_2 | l_1, l_2) = p(d_2 | l_1, l_2, d_1) p(d_1 | l_1, l_2)$$



Gaussian mixture model



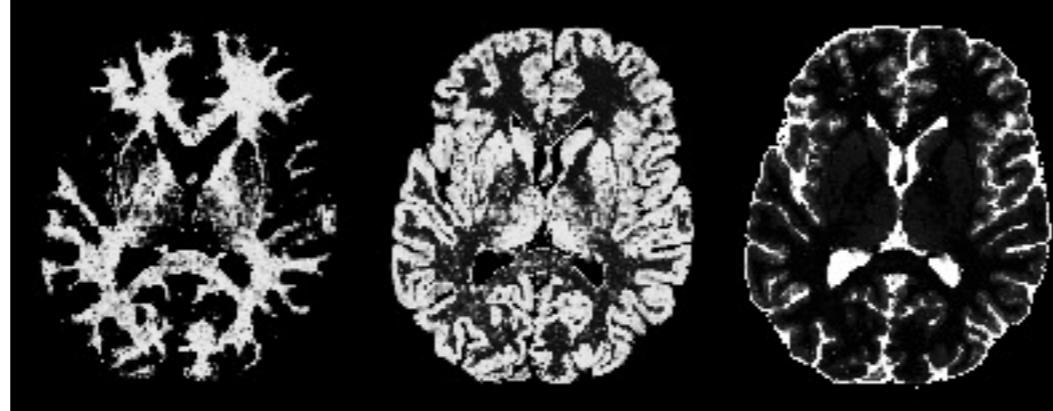
$$p(\mathbf{d}|\boldsymbol{\theta}) = \prod_n \left(\sum_k \mathcal{N}(d_n | \mu_k, \sigma_k^2) \pi_k \right)$$

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

Posterior probability distribution



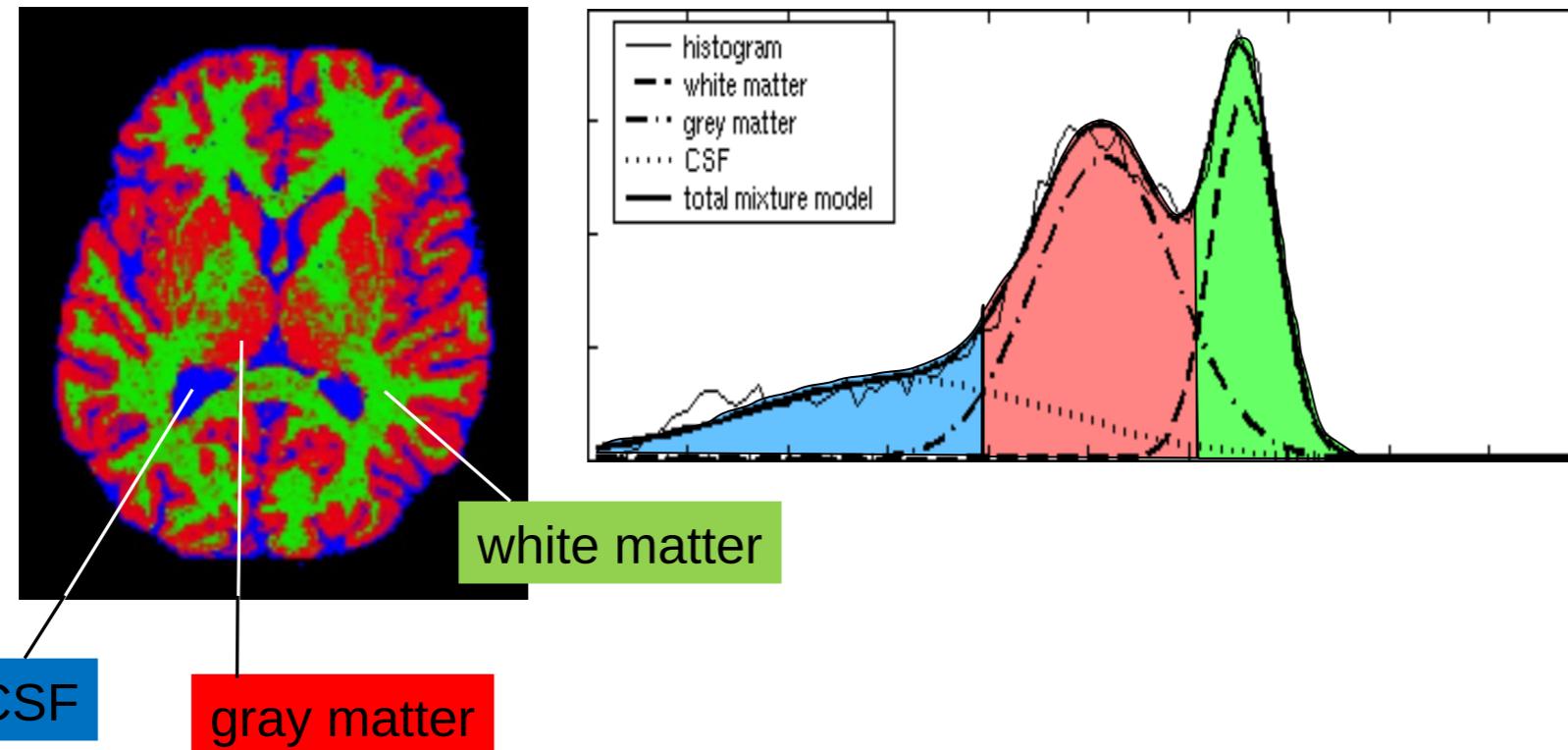
$$\begin{aligned} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) &= \frac{p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l)}{p(\mathbf{d}|\hat{\boldsymbol{\theta}})} \\ &= \frac{\prod_n \mathcal{N}(d_n | \hat{\mu}_{l_n}, \hat{\sigma}_{l_n}^2) \prod_n \hat{\pi}_{l_n}}{\prod_n \sum_k \mathcal{N}(d_n | \hat{\mu}_k, \hat{\sigma}_k^2) \hat{\pi}_k} \\ &= \prod_n p(l_n | d_n, \hat{\boldsymbol{\theta}}) \end{aligned}$$



$$p(l_n | d_n, \hat{\boldsymbol{\theta}}) = \frac{\mathcal{N}(d_n | \hat{\mu}_{l_n}, \hat{\sigma}_{l_n}^2) \hat{\pi}_{l_n}}{\sum_k \mathcal{N}(d_n | \hat{\mu}_k, \hat{\sigma}_k^2) \hat{\pi}_k}$$

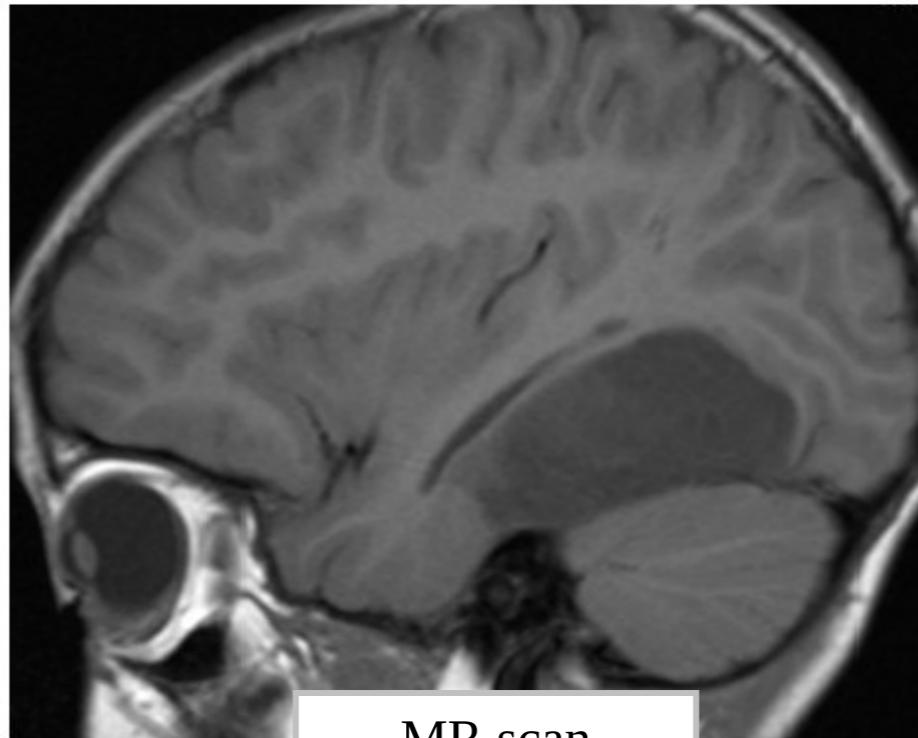
Maximum a posteriori segmentation

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

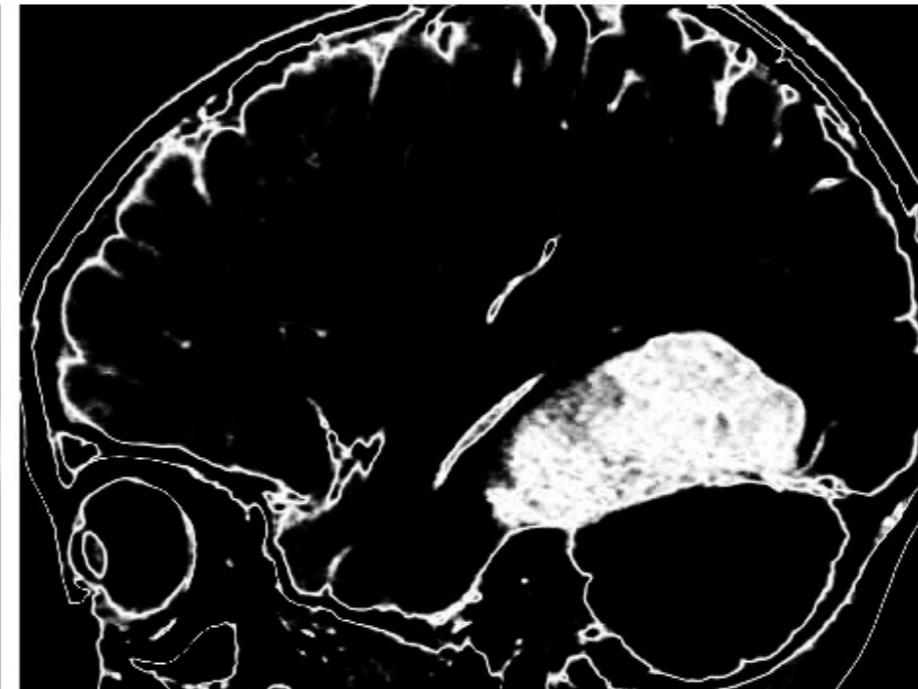


Problem solved?

Two-component Gaussian mixture model:
tumor vs. “other”

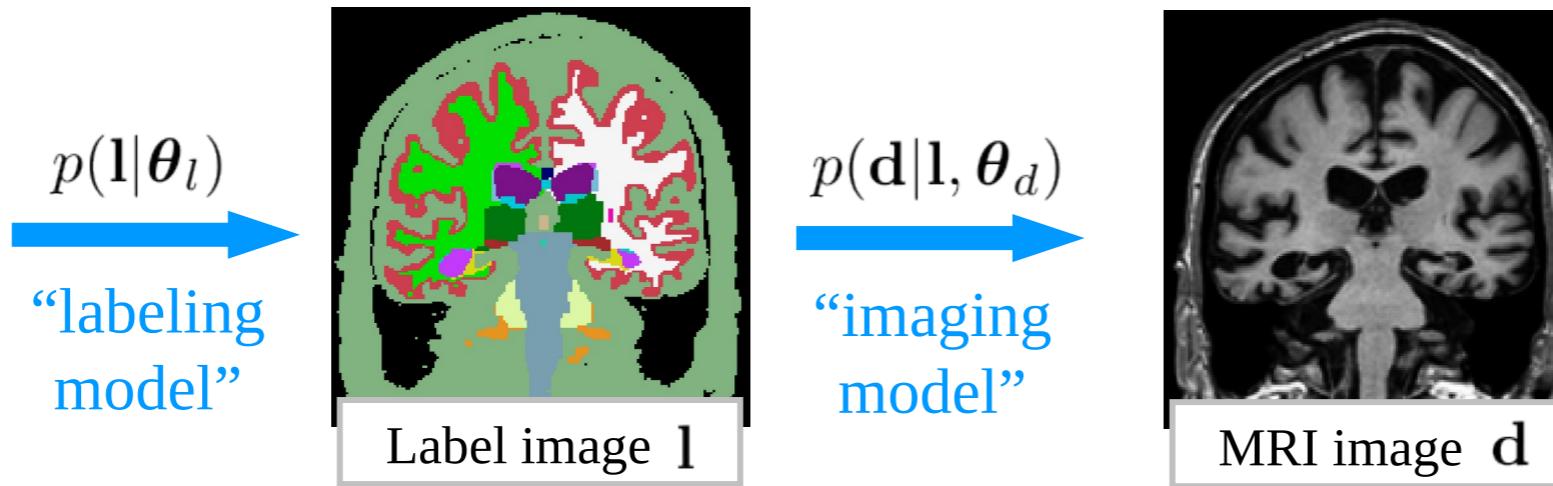


MR scan



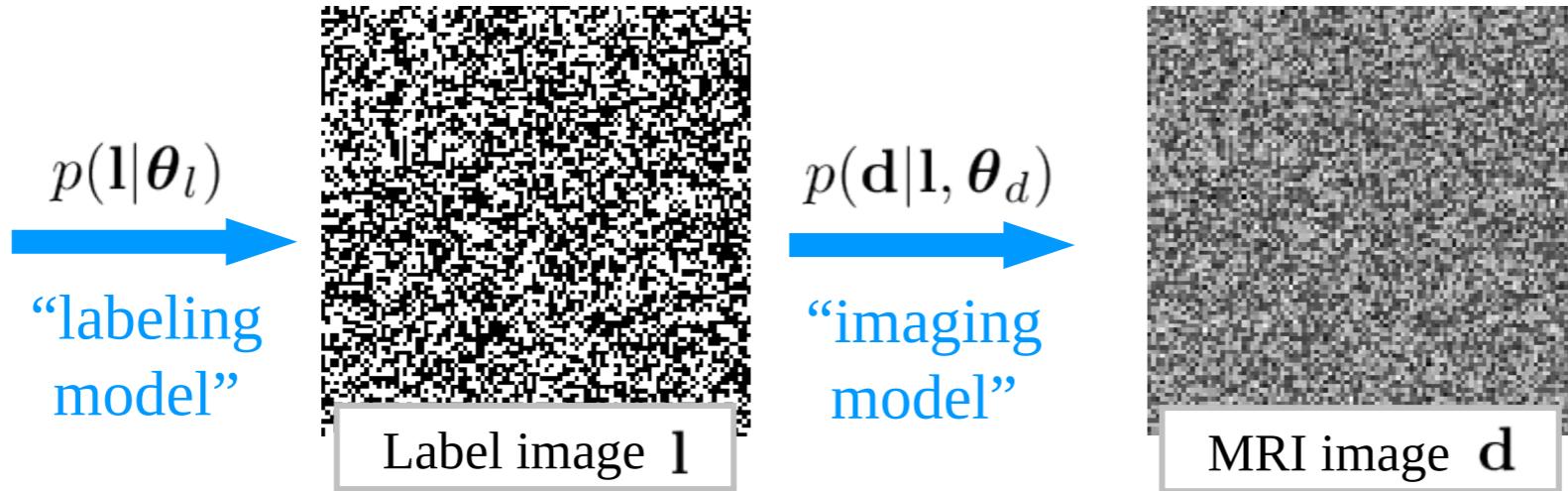
Posterior probability for tumor

Gaussian mixture model



which part of the model needs fixing?

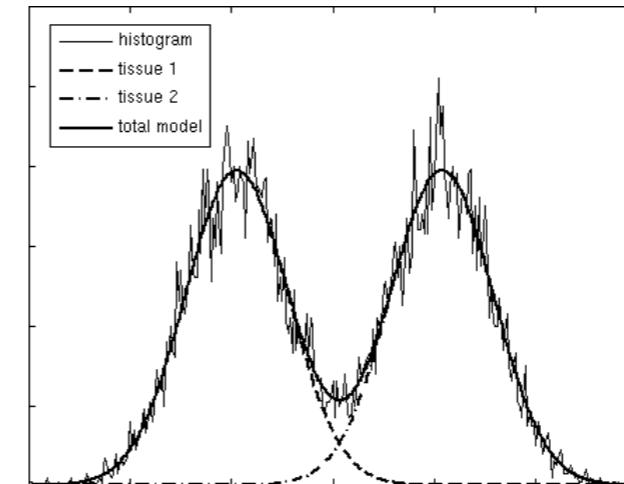
Gaussian mixture model



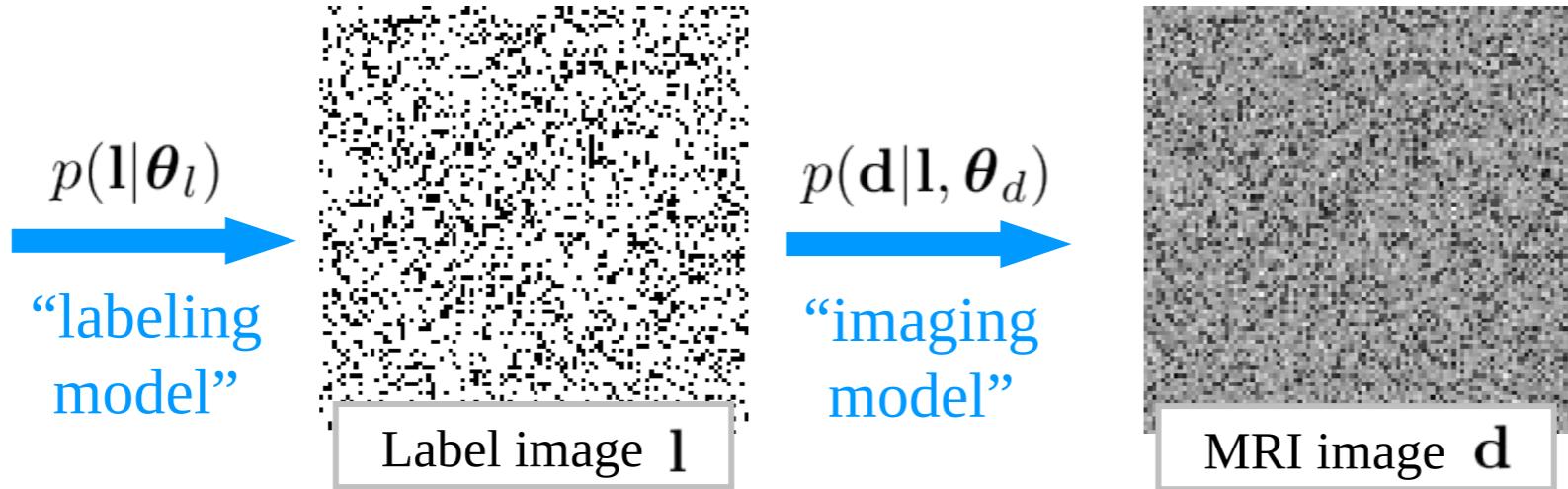
$$\mu_1 = 70, \mu_2 = 90$$

$$\sigma_1 = 5, \sigma_2 = 5$$

$$\pi_1 = 0.5, \pi_2 = 0.5$$



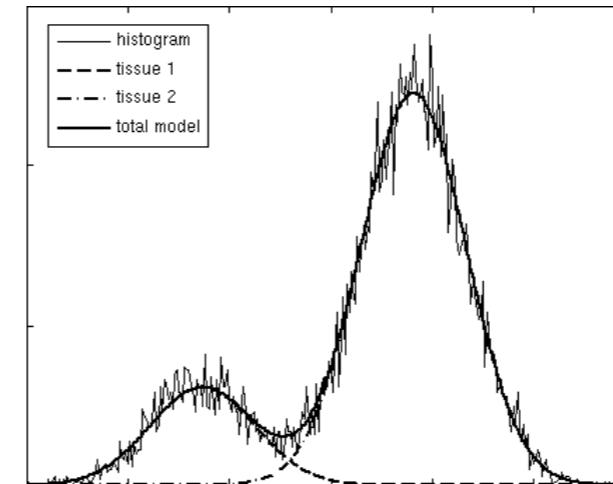
Gaussian mixture model



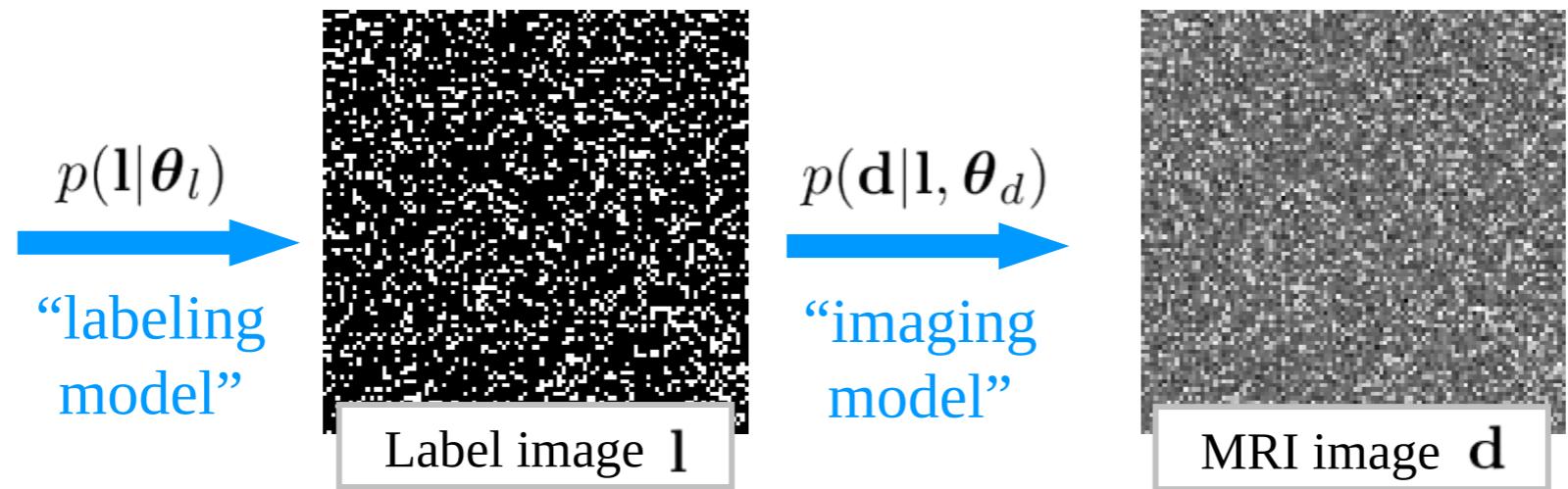
$$\mu_1 = 70, \mu_2 = 90$$

$$\sigma_1 = 5, \sigma_2 = 5$$

$$\pi_1 = 0.2, \pi_2 = 0.8$$



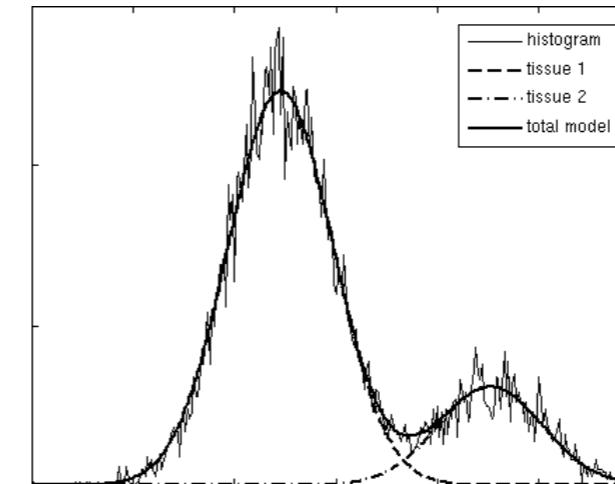
Gaussian mixture model



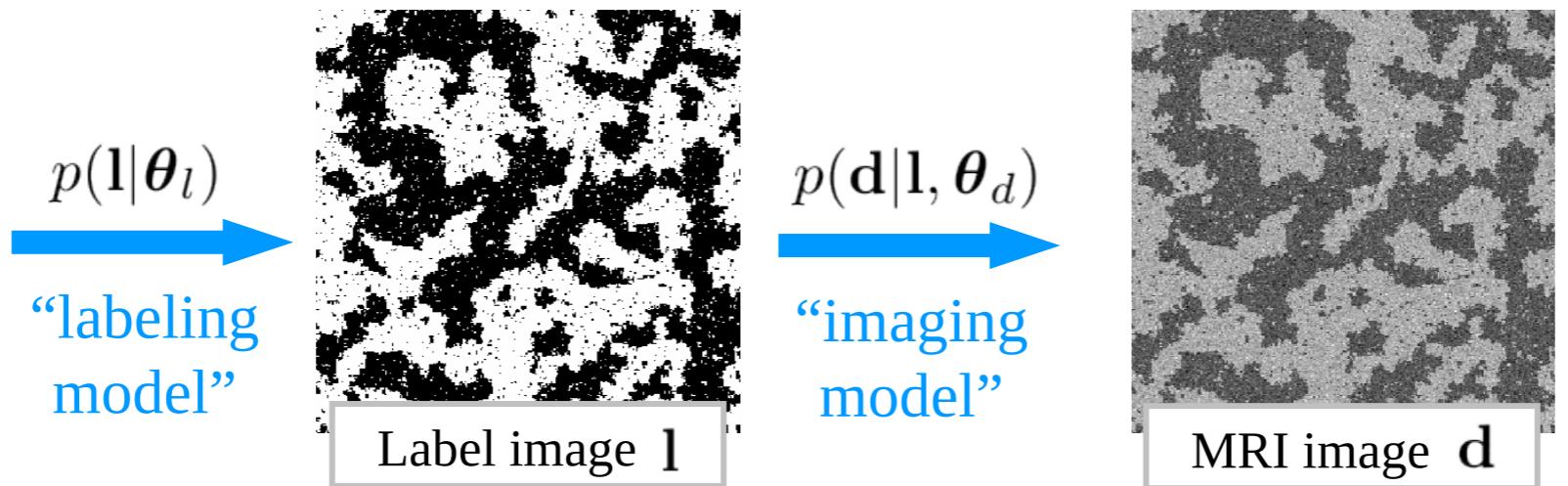
$$\mu_1 = 70, \mu_2 = 90$$

$$\sigma_1 = 5, \sigma_2 = 5$$

$$\pi_1 = 0.8, \pi_2 = 0.2$$

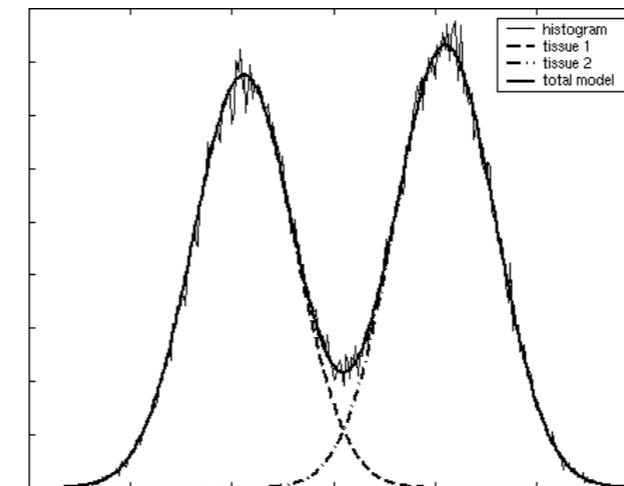


Markov random field model



$$\mu_1 = 70, \mu_2 = 90$$

$$\sigma_1 = 5, \sigma_2 = 5$$



Markov random field model

- Prior that prefers voxels with the same label to be spatially clustered

$$p(\mathbf{l}|\boldsymbol{\theta}_l) = \frac{1}{Z(\boldsymbol{\theta}_l)} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$$

strength of penalty

$$U(\mathbf{l}|\boldsymbol{\theta}_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j)$$

zero if labels are the same, one otherwise

sum over all neighboring voxels

- $Z(\boldsymbol{\theta}_l) = \sum_{\mathbf{l}} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$ is a normalizing constant (not needed in practice)

Markov random field model

- Slightly more general:

$$p(\mathbf{l}|\boldsymbol{\theta}_l) = \frac{1}{Z(\boldsymbol{\theta}_l)} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$$

$$U(\mathbf{l}|\boldsymbol{\theta}_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j) - \sum_i \log(\pi_{l_i})$$

- $\boldsymbol{\theta}_l = (\beta, \pi_1, \dots, \pi_K)^T$ are the model parameters
- Reduces to Gaussian mixture model prior $p(\mathbf{l}|\boldsymbol{\theta}_l) = \prod_n \pi_{l_n}$ for $\beta = 0$!

Toy example

$N = 2$ voxels

$K = 3$ classes

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix}$$

$$\beta = 0$$

0.1



l_1

0.06



l_1

0.04



l_1

0.15



l_1

0.09



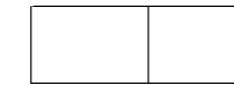
l_1

0.06



l_1

0.25



l_1

0.15



l_1

0.1



l_1

$$p(\mathbf{l}) = p(l_2|l_1)p(l_1)$$

$$\beta = 3.0$$

0.02



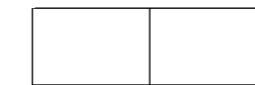
l_1

0.02



l_1

0.6



l_1

0.01



l_1

0.23



l_1

0.02



l_1

0.07



l_1

0.01



l_1

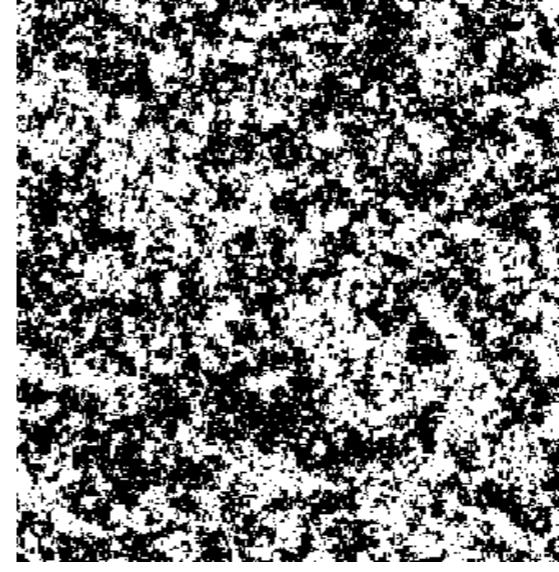
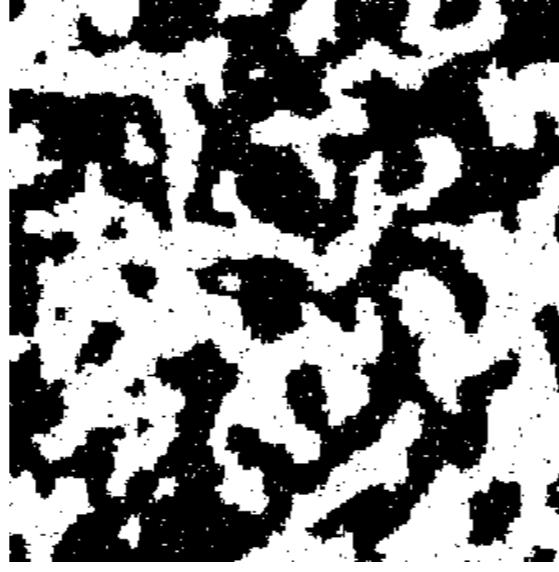
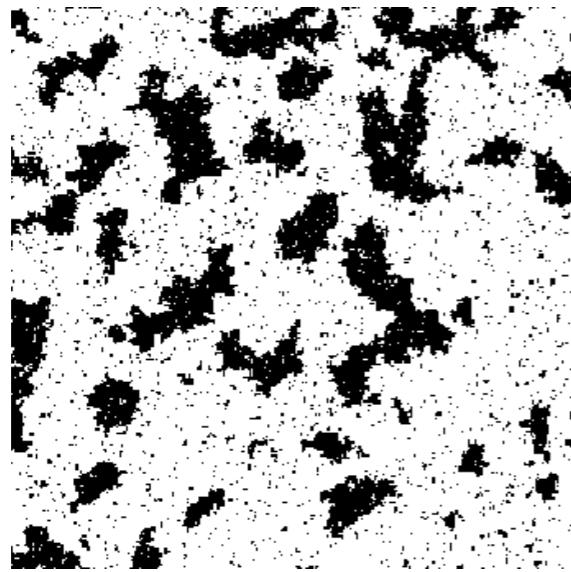
0.02



l_1

$$p(\mathbf{l}) = p(l_1, l_2)$$

Samples



Different values for model parameters $\theta_l = (\beta, \pi_1, \dots, \pi_K)^T$

Why exactly this model?

- Long-range statistical dependencies between voxels
- Local computations (efficient!):

$$\begin{aligned} p(l_i | \mathbf{l}_{\setminus i}) &= \frac{p(\mathbf{l})}{p(\mathbf{l}_{\setminus i})} \\ &= \frac{p(\mathbf{l})}{\sum_{l_i} p(\mathbf{l})} \\ &= \frac{\exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))}{\sum_{l_i} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))} \\ &= \frac{\pi_{l_i} \cdot \exp \left(-\beta \sum_{j \in \mathcal{N}_i} \delta(l_i \neq l_j) \right)}{\sum_k \pi_k \cdot \exp \left(-\beta \sum_{j \in \mathcal{N}_i} \delta(l_j \neq k) \right)} \end{aligned}$$

all labels except the one of voxel i

neighbors of voxel i

Mean field approximation

- In the Gaussian mixture model, the posterior was of the form

$$p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) = \prod_n p(l_n|d_n, \hat{\boldsymbol{\theta}})$$

- With the Markov random field model, the posterior no longer “factorizes” that way
- For a 2-label model in a standard 256x256x128 MR scan, there are over $10^{1000000}$ unique label images with each its own posterior probability!
- Solution: approximate $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$

Mean field approximation

- Approximate $p(\mathbf{l}|\mathbf{d}, \hat{\theta})$ with something of the form

$$q(\mathbf{l}) = \prod_n q_n(l_n)$$

- Find the voxel-wise distributions $q_n(k)$ that minimize the difference between $q(\mathbf{l})$ and $p(\mathbf{l}|\mathbf{d}, \hat{\theta})$
- Quantify the difference between the two distributions using the “Kullback-Leibler divergence”

$$KL \left(q(\mathbf{l}) \parallel p(\mathbf{l}|\mathbf{d}, \hat{\theta}) \right) = - \sum_{\mathbf{l}} q(\mathbf{l}) \log \frac{p(\mathbf{l}|\mathbf{d}, \hat{\theta})}{q(\mathbf{l})}$$

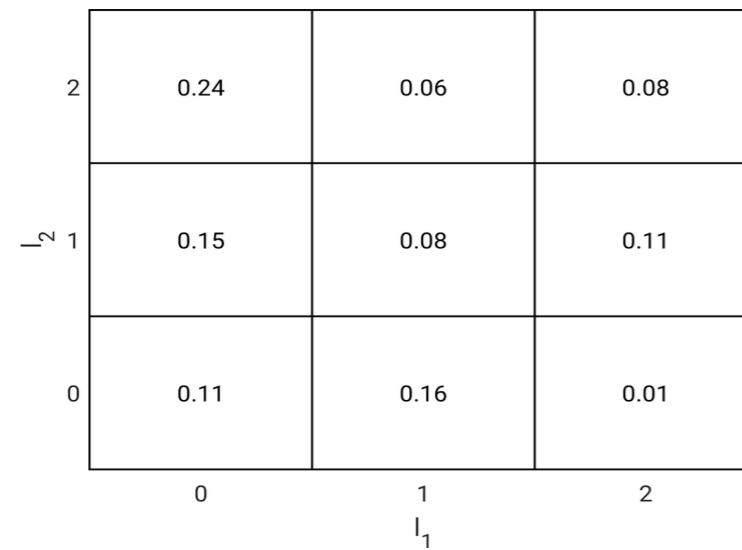
Toy example

$N = 2$ voxels

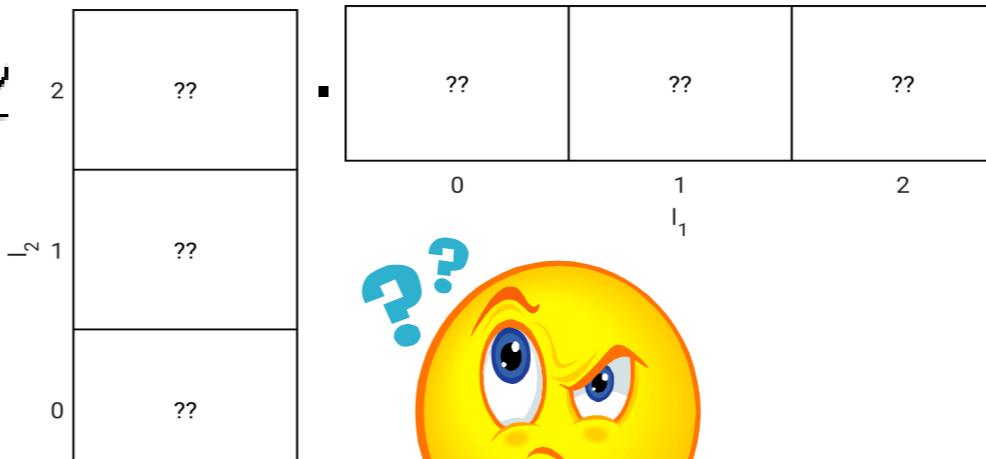
$K = 3$ classes

$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \quad \mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$

$$p(\mathbf{l}|\mathbf{d}) = p(l_1, l_2|d_1, d_2) \simeq q(l_1)q(l_2)$$



\approx



Mean field approximation

- Solution for one voxel i :

$$q_i(l_i) = \frac{\mathcal{N}(d_i | \hat{\mu}_{l_i}, \hat{\sigma}_{l_i}^2) \gamma_i(l_i)}{\sum_k \mathcal{N}(d_i | \hat{\mu}_k, \hat{\sigma}_k^2) \gamma_i(k)}$$

$$\text{where } \gamma_i(k) = \frac{\hat{\pi}_k \cdot \exp\left(-\beta \sum_{j \in \mathfrak{N}_i} (1 - q_j(k))\right)}{\sum_{k'} \hat{\pi}_{k'} \cdot \exp\left(-\beta \sum_{j \in \mathfrak{N}_i} (1 - q_j(k'))\right)}$$

Mean field approximation

- Solution for one voxel i :

$$q_i(l_i) = \frac{\mathcal{N}(d_i | \hat{\mu}_{l_i}, \hat{\sigma}_{l_i}^2) \gamma_i(l_i)}{\sum_k \mathcal{N}(d_i | \hat{\mu}_k, \hat{\sigma}_k^2) \gamma_i(k)}$$

$$\text{where } \gamma_i(k) = \frac{\hat{\pi}_k \cdot \exp\left(-\beta \sum_{j \in \mathfrak{N}_i} (1 - q_j(k))\right)}{\sum_{k'} \hat{\pi}_{k'} \cdot \exp\left(-\beta \sum_{j \in \mathfrak{N}_i} (1 - q_j(k'))\right)}$$

Influenced by the result
in neighboring voxels:
spatial context!!!!

Mean field approximation

- Solution for one voxel i :

$$q_i(l_i) = \frac{\mathcal{N}(d_i | \hat{\mu}_{l_i}, \hat{\sigma}_{l_i}^2) \gamma_i(l_i)}{\sum_k \mathcal{N}(d_i | \hat{\mu}_k, \hat{\sigma}_k^2) \gamma_i(k)}$$

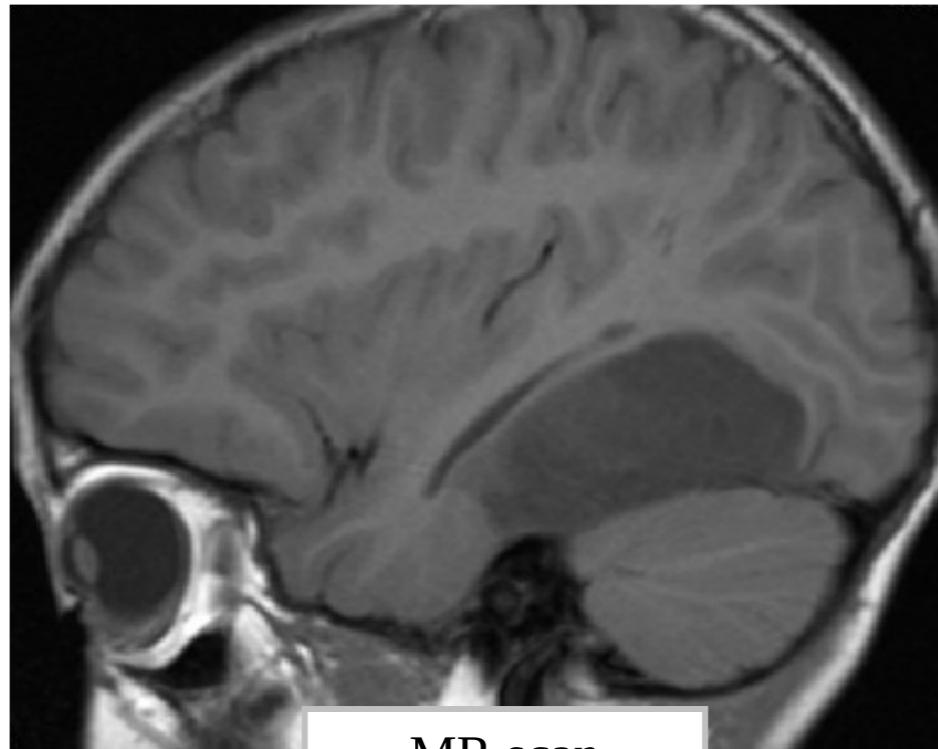
$$\text{where } \gamma_i(k) = \frac{\hat{\pi}_k \cdot \exp\left(-\beta \sum_{j \in \mathfrak{N}_i} (1 - q_j(k))\right)}{\sum_{k'} \hat{\pi}_{k'} \cdot \exp\left(-\beta \sum_{j \in \mathfrak{N}_i} (1 - q_j(k'))\right)}$$

Influenced by the result
in neighboring voxels:
spatial context!!!!

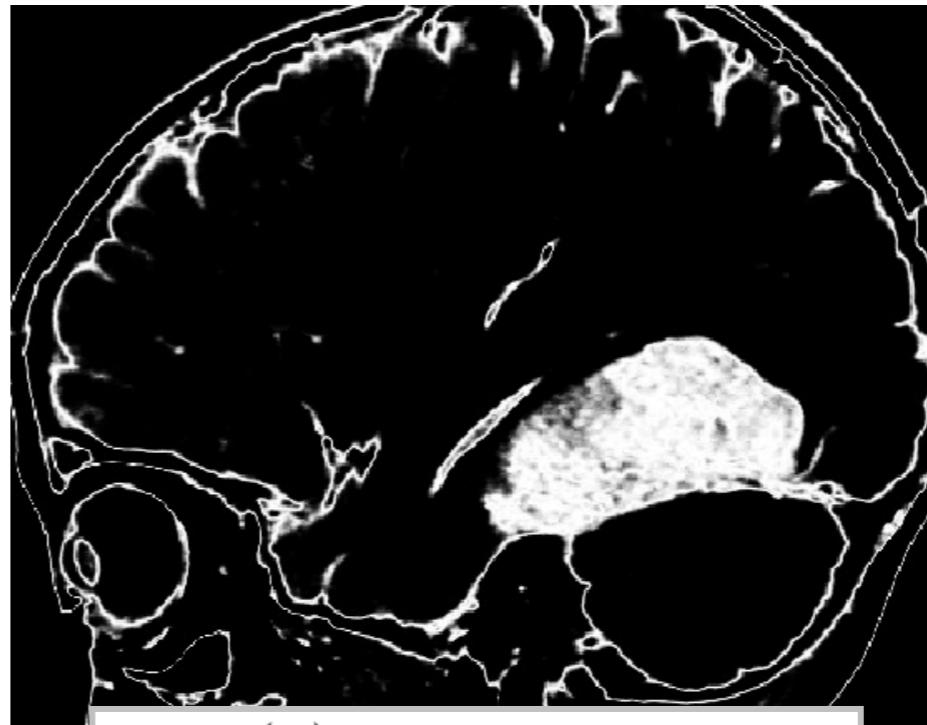
- Need to iterate across all voxels

Example

Two-component Gaussian mixture model:
tumor vs. “other”



MR scan

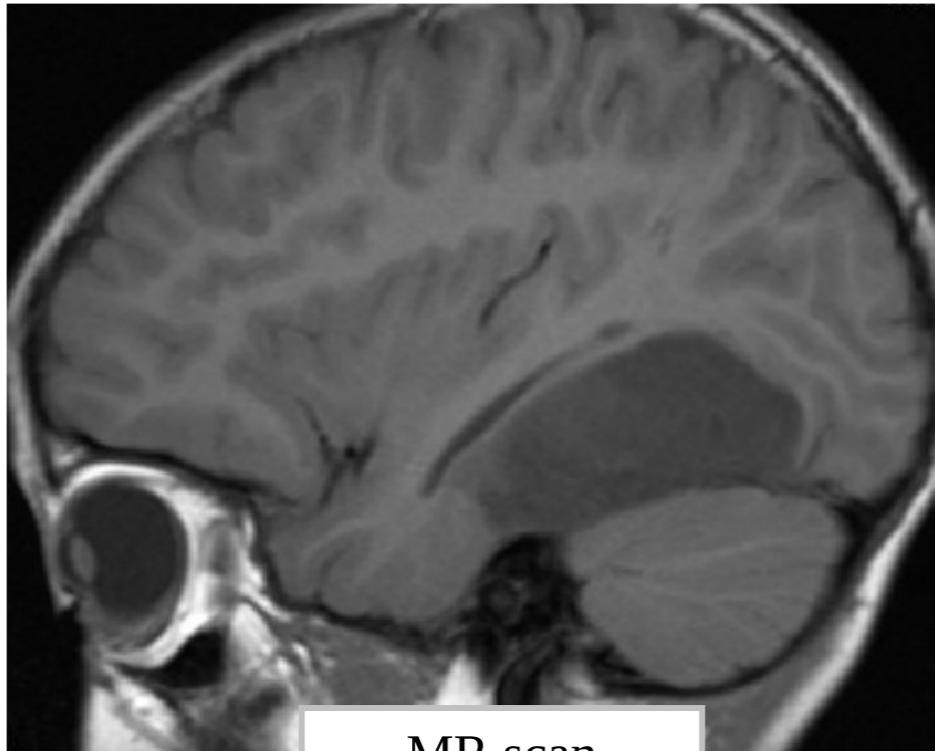


$q_n(k)$ for tumor class

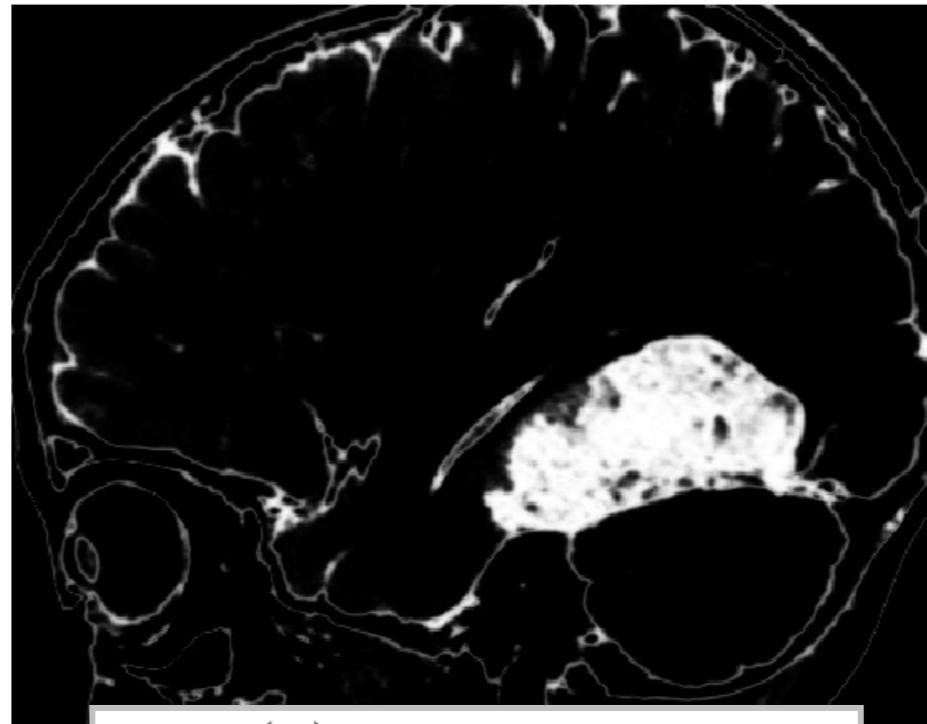
$$\beta = 0$$

Example

Two-component Gaussian mixture model:
tumor vs. “other”



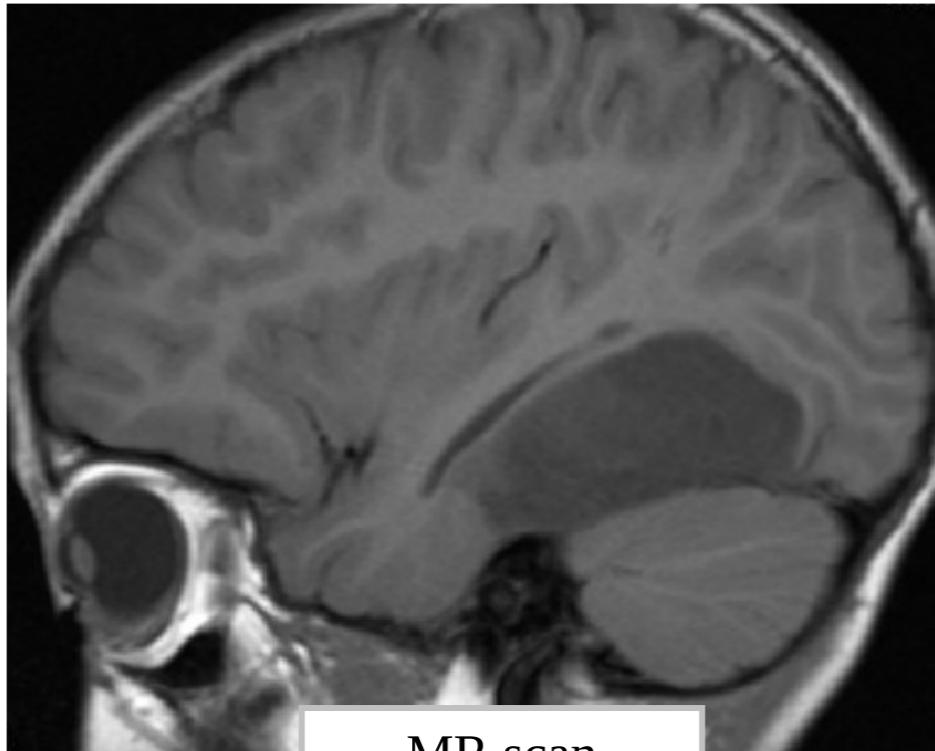
MR scan



$$q_n(k) \text{ for tumor class}$$
$$\beta = 0.25$$

Example

Two-component Gaussian mixture model:
tumor vs. “other”



MR scan



$q_n(k)$ for tumor class
 $\beta = 0.55$