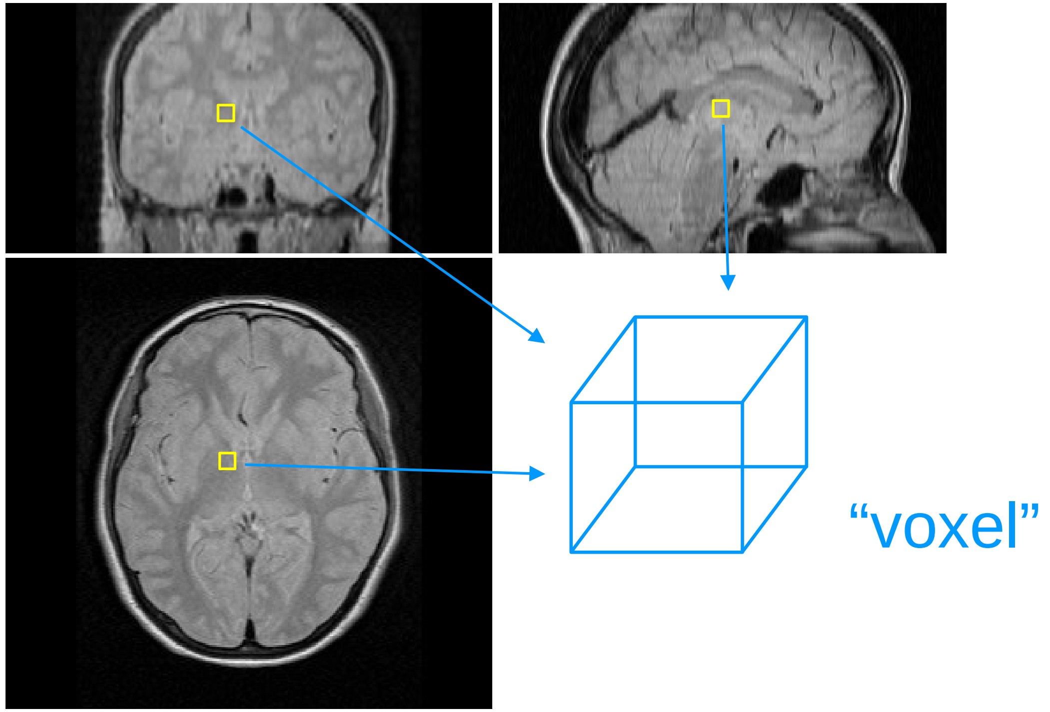
Model-based Segmentation: Part I

Medical Image Analysis
Koen Van Leemput
Fall 2023

Aalto-yliopisto
Aalto-universitetet
Aalto University

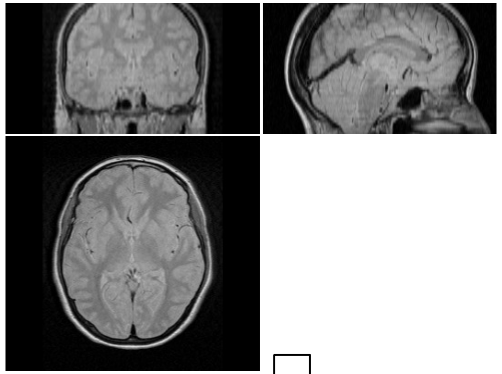
Voxel-based segmentation



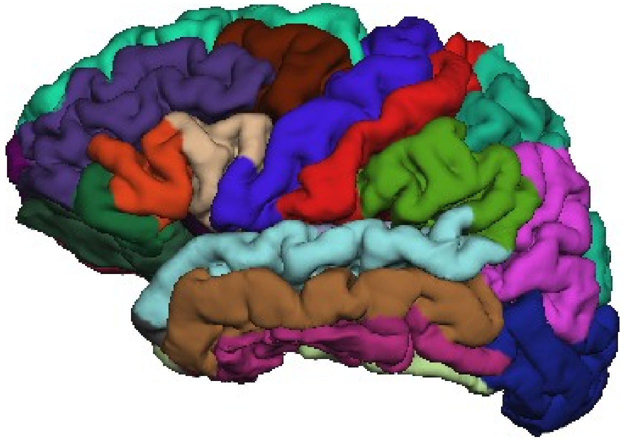
Determine to which anatomical structure each voxel in the image belongs:

- Think "LEGO bricks"
- Outer surfaces can easily be extracted if needed

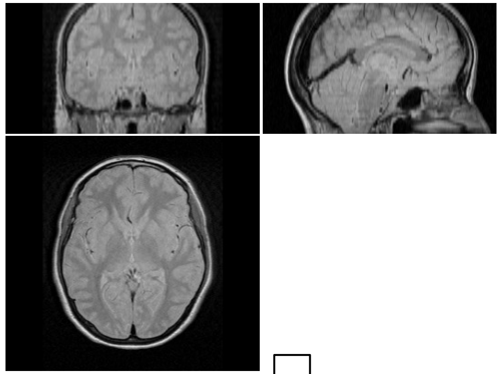
Voxel-based segmentation



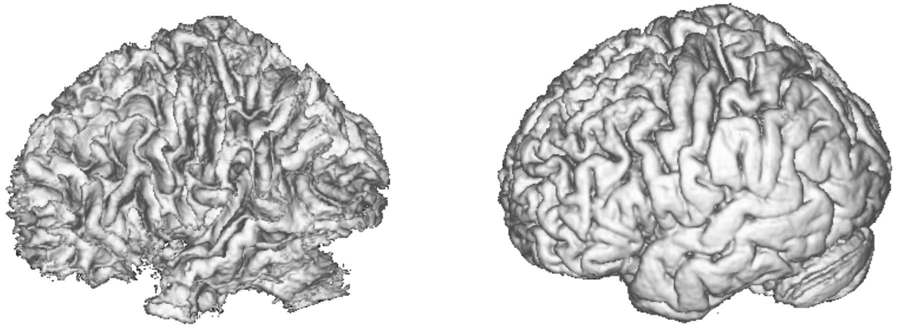
Automated
computational
methods



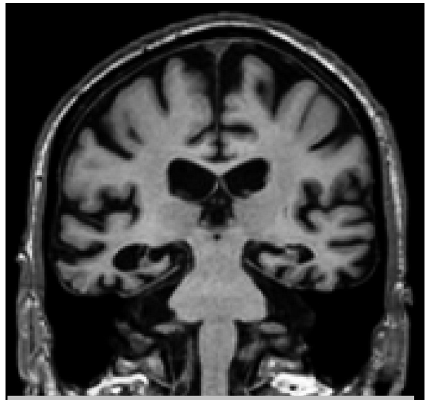
This and next lecture



Automated computational methods



The problem to be solved



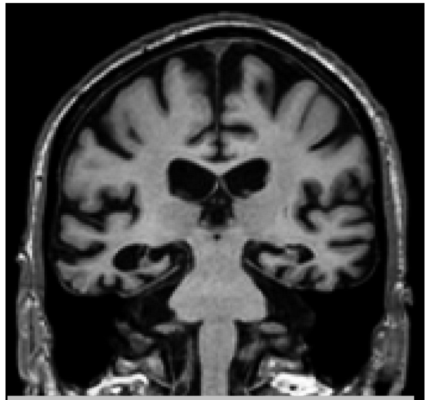
MRI image $\mathbf{d}$

N voxels

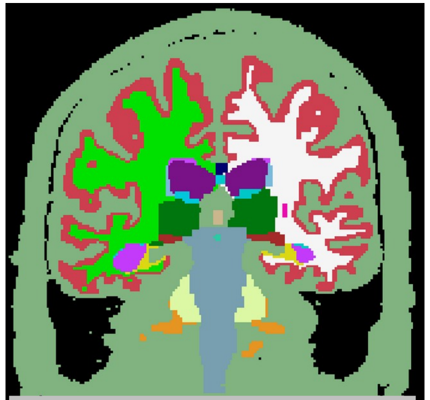
$$\mathbf{d} = (d_1, \ldots, d_N)^{\mathrm{T}}$$

d_n: intensity in voxel n

The problem to be solved



MRI image $\mathbf{d}$



Label image $\mathbf{l}$

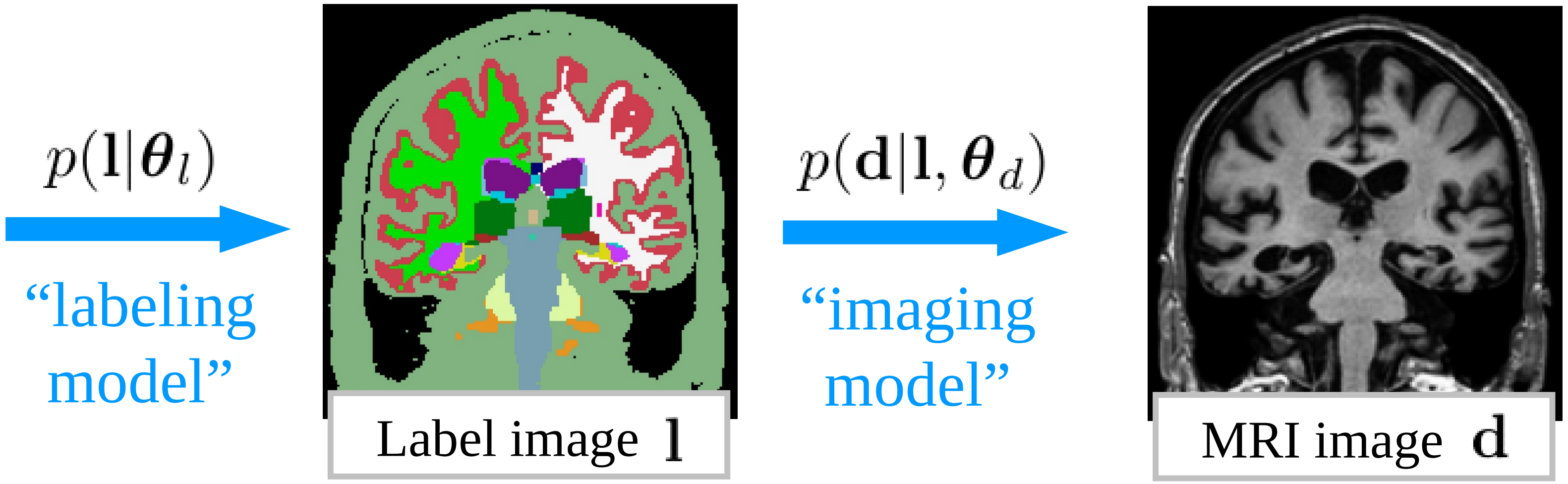
$$\mathbf{l} = (l_1, \ldots, l_N)^{\mathrm{T}}$$

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

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 $\boldsymbol{\theta} = (\boldsymbol{\theta}_l^{\mathrm{T}}, \boldsymbol{\theta}_d^{\mathrm{T}})^{\mathrm{T}}$

– Appropriate values $\hat{\boldsymbol{\theta}}$ are assumed to be known for now...

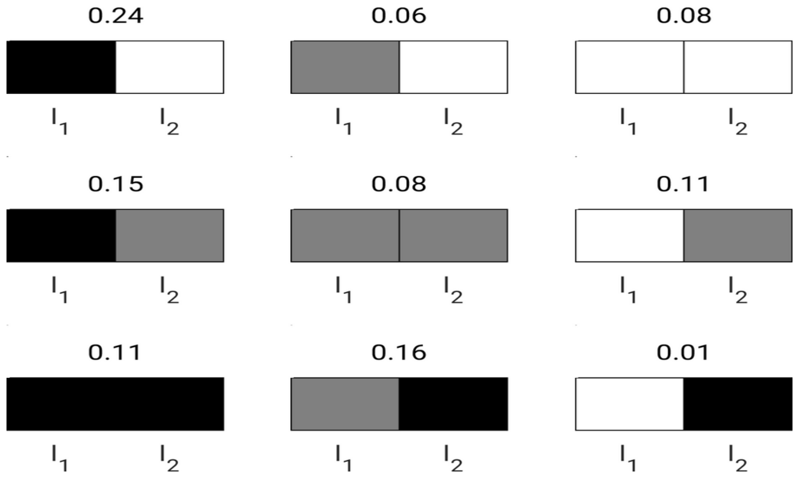
Toy example

$N = 2$ voxels

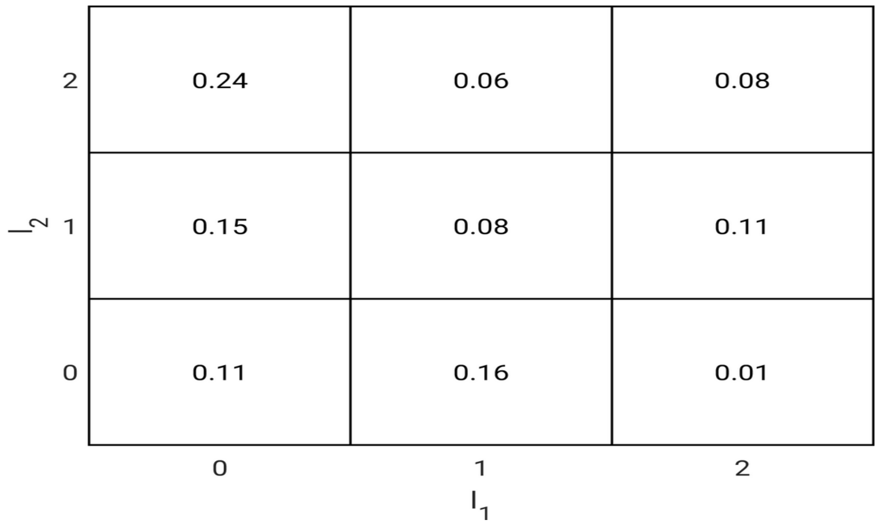
$K = 3$ classes

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

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



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



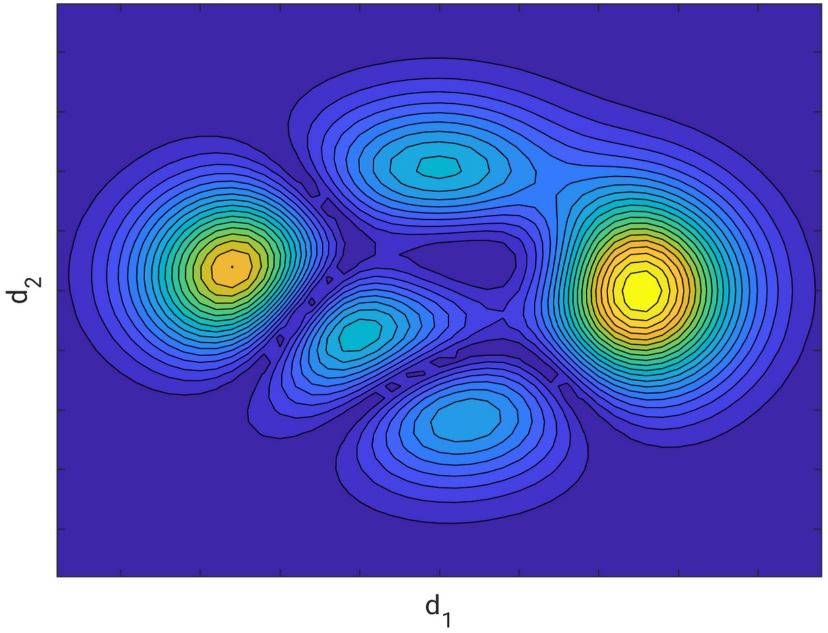
Toy example

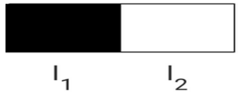
$N = 2$ voxels

$K = 3$ classes

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

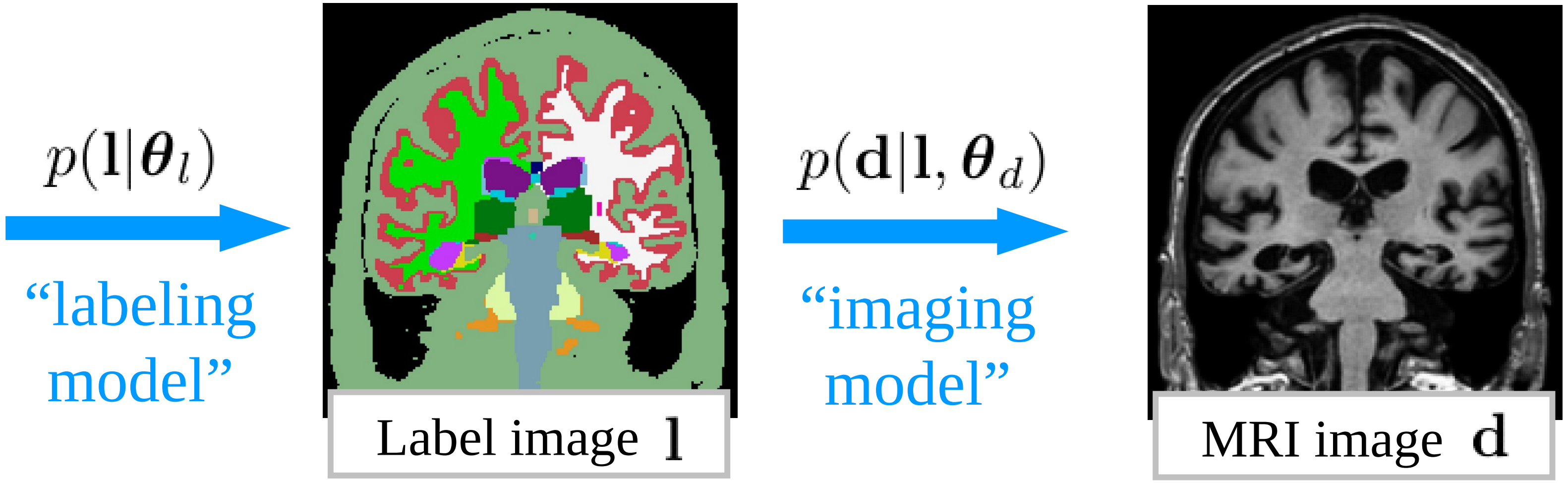
$$p(\mathbf{d}|\mathbf{l}) = p(d_1, d_2 | l_1, l_2)$$





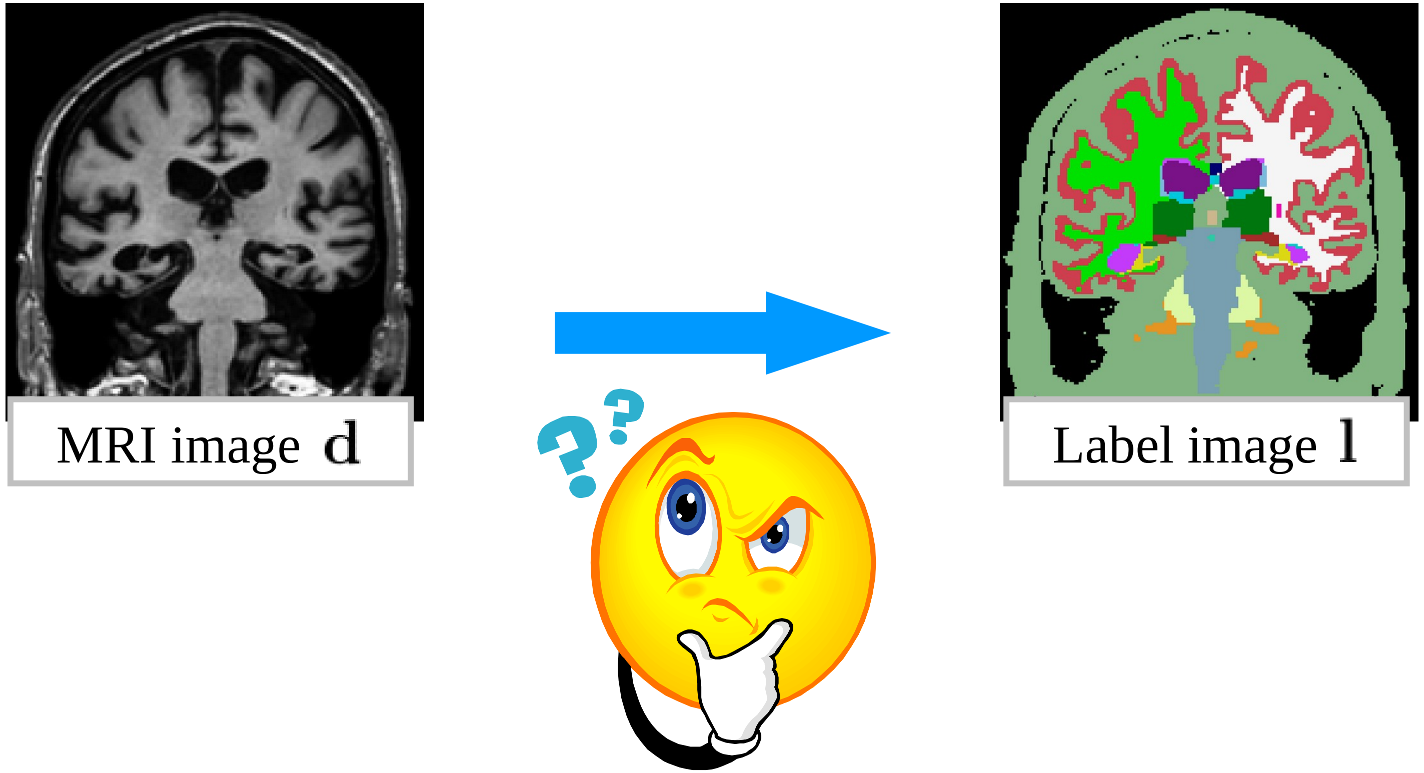
One solution: generative modeling

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

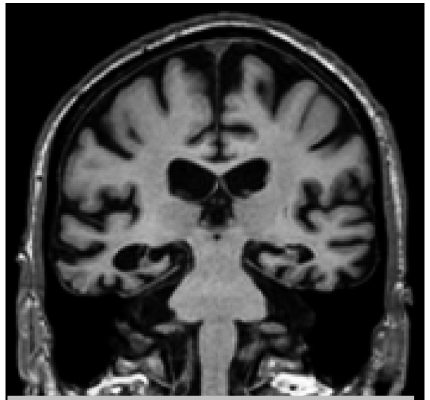


- The model depends on some parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_l^{\mathrm{T}}, \boldsymbol{\theta}_d^{\mathrm{T}})^{\mathrm{T}}$
- Appropriate values $\hat{\boldsymbol{\theta}}$ are assumed to be known for now...

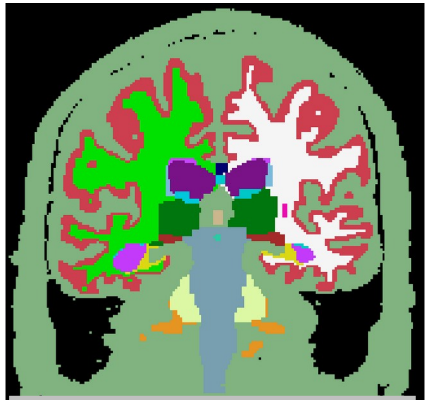
Segmentation = inverse problem



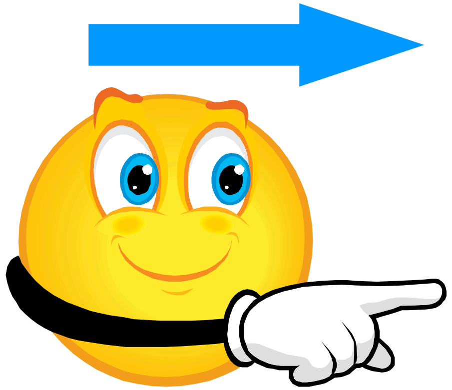
Segmentation = inverse problem



MRI image $\mathbf{d}$



Label image $\mathbf{l}$

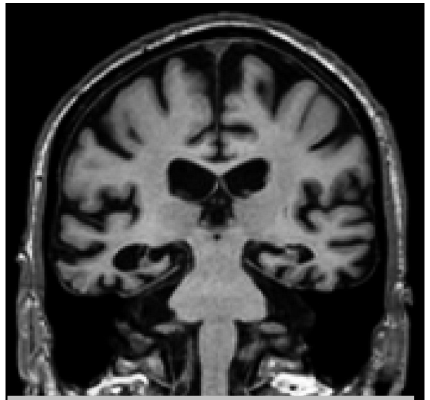


$$\hat{\mathbf{l}} = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

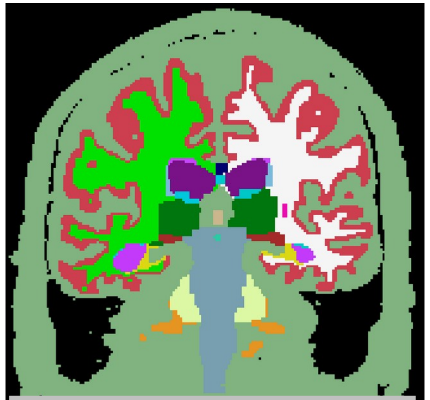
The posterior distribution $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$ is given by Bayes rule:

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

Segmentation = inverse problem



MRI image $\mathbf{d}$



Label image $\mathbf{l}$

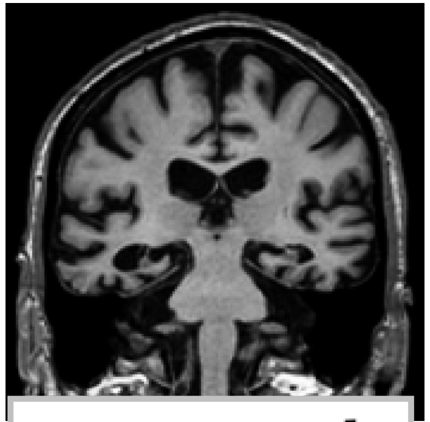
$$\hat{\mathbf{l}} = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

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

labeling model

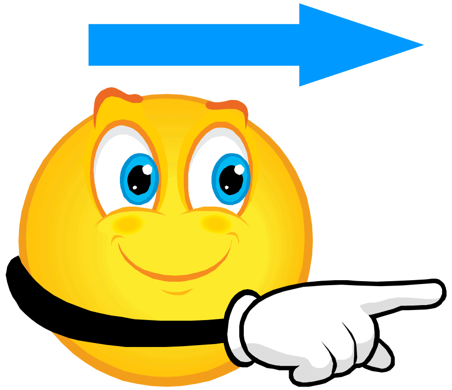
$$p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}}) = \frac{p(\mathbf{d}|\mathbf{l}, \hat{\boldsymbol{\theta}}_d)\, p(\mathbf{l}|\hat{\boldsymbol{\theta}}_l)}{p(\mathbf{d}|\hat{\boldsymbol{\theta}})}$$

Segmentation = inverse problem



MRI image $\mathbf{d}$

Label image $\mathbf{l}$



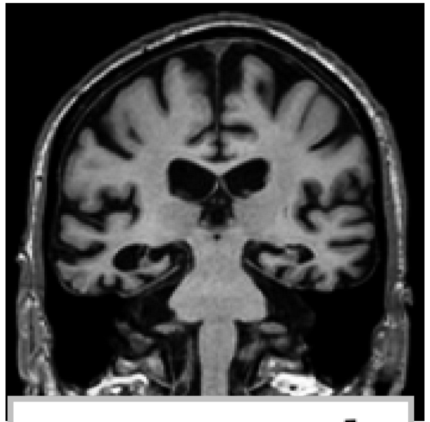
$$\hat{\mathbf{l}} = \arg\max_{\mathbf{l}} p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$$

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



MRI image $\mathbf{d}$

Label image $\mathbf{l}$

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

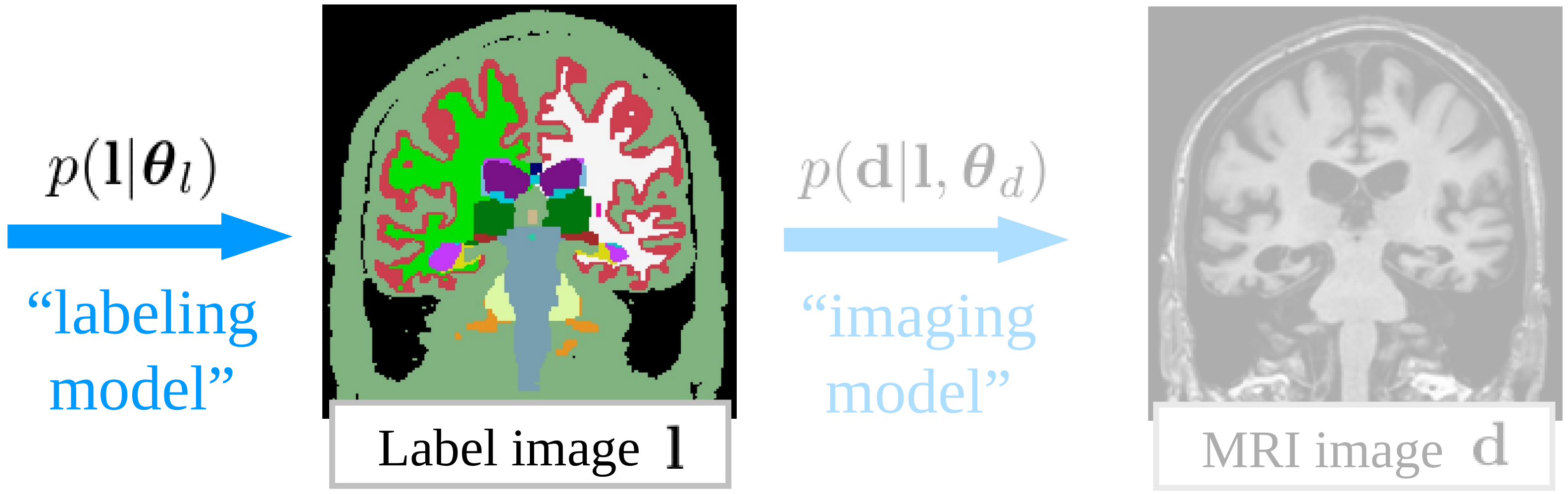
The posterior distribution $p(\mathbf{l}|\mathbf{d}, \hat{\boldsymbol{\theta}})$ is given by Bayes rule:

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

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

(but not needed)

Gaussian mixture model



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

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

Toy example

$N = 2$ voxels

$K = 3$ classes

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

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

l_2 \ l_1	0	1	2
2	0.1	0.15	0.25
1	0.06	0.09	0.15
0	0.04	0.06	0.1

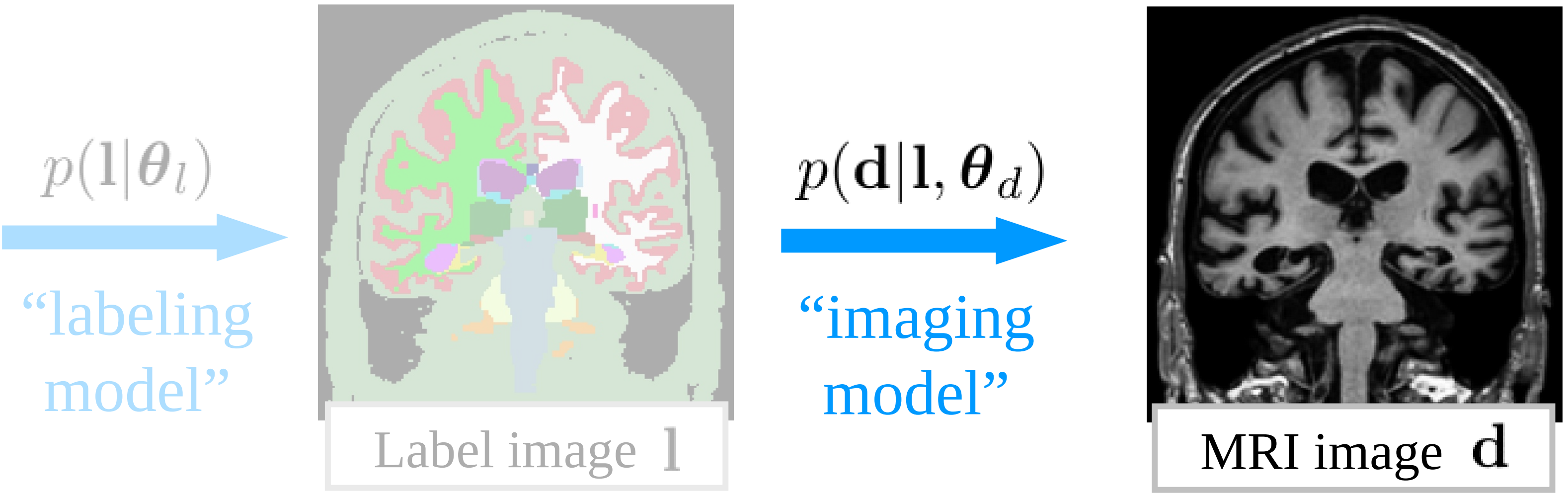
=

l_2	
2	0.5
1	0.3
0	0.2

·

l_1	0	1	2
	0.2	0.3	0.5

Gaussian mixture model



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

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

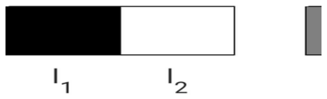
$$\mathcal{N}(d|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(d-\mu)^2}{2\sigma^2}\right]$$

Toy example

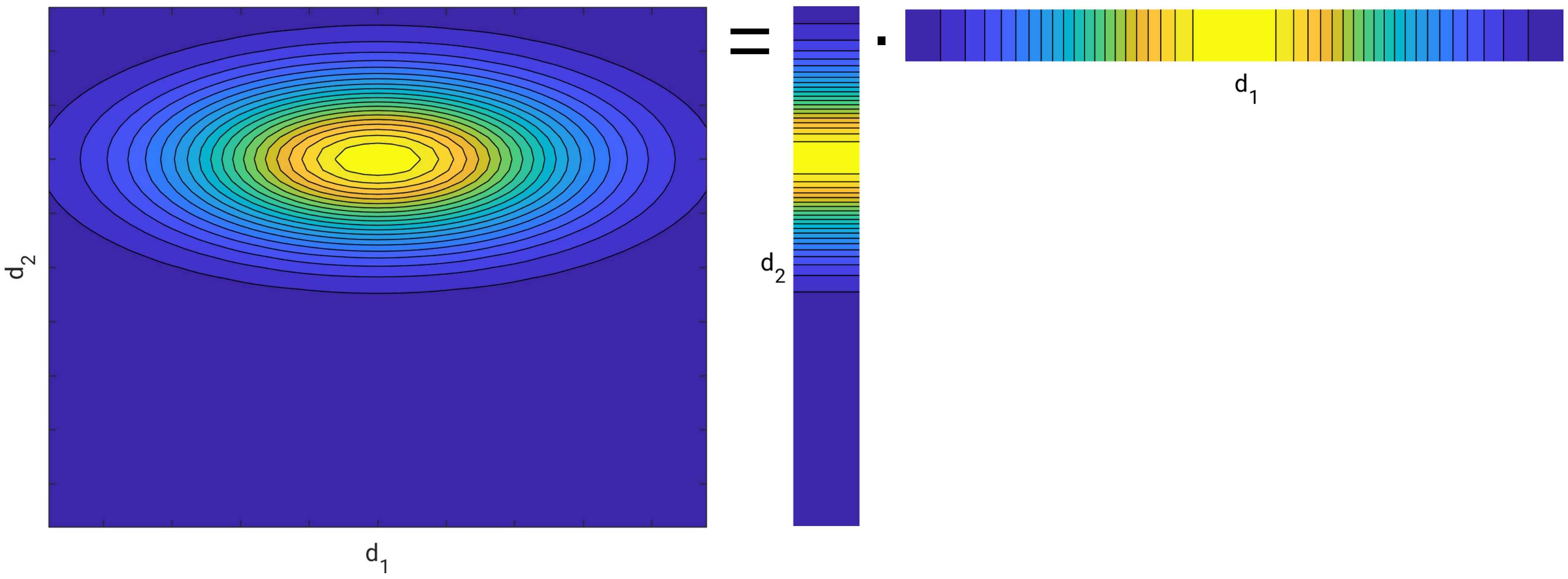
$N = 2$ voxels

$K = 3$ classes

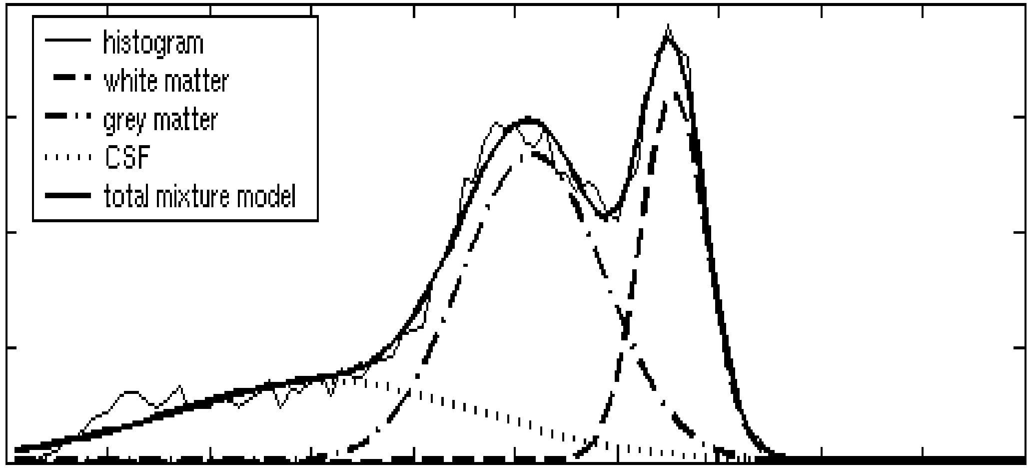
$$\mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$$



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



Gaussian mixture model



$K = 3$ labels

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

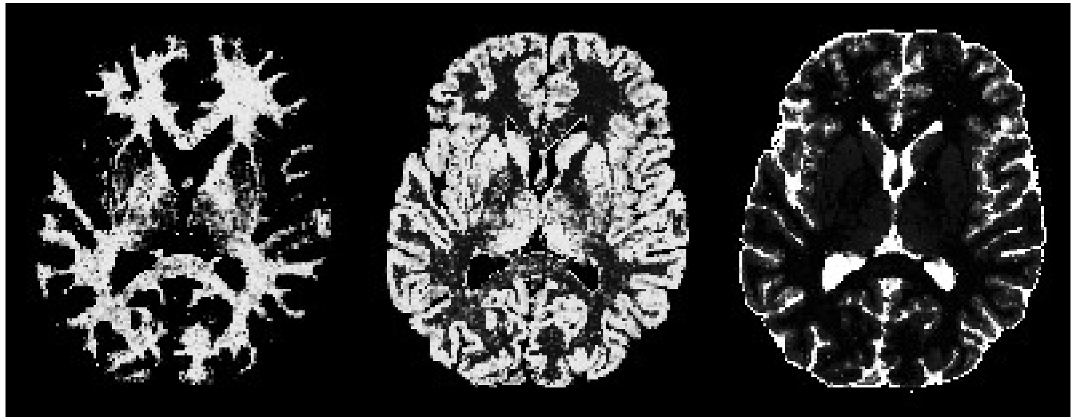
$$\boldsymbol{\theta} = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots \sigma_K^2, \pi_1, \ldots, \pi_K)^{\mathrm{T}}$$

Posterior probability distribution

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

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

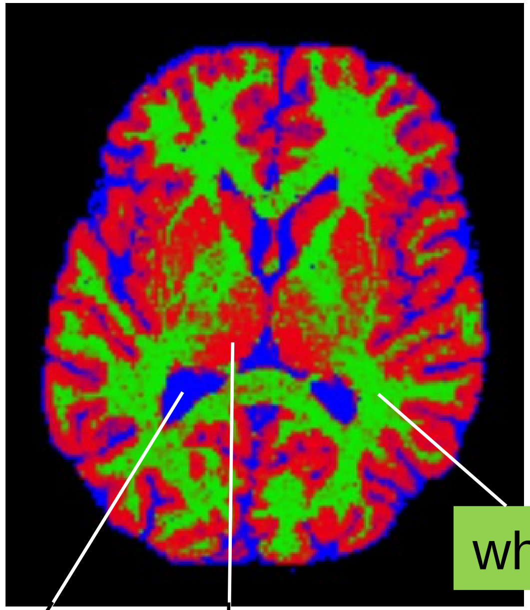
$$= \prod_n p(l_n|d_n,\hat{\boldsymbol{\theta}})$$

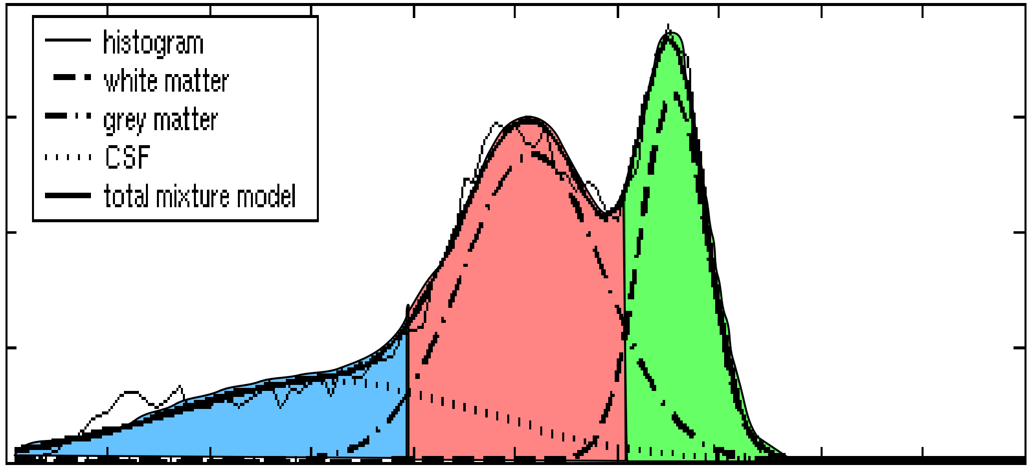


$$p(l_n|d_n,\hat{\boldsymbol{\theta}}) = \frac{\mathcal{N}(d_n|\hat{\mu}_{l_n},\hat{\sigma}^2_{l_n})\hat{\pi}_{l_n}}{\sum_k \mathcal{N}(d_n|\hat{\mu}_k,\hat{\sigma}^2_k)\hat{\pi}_k}$$

Maximum a posteriori segmentation

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

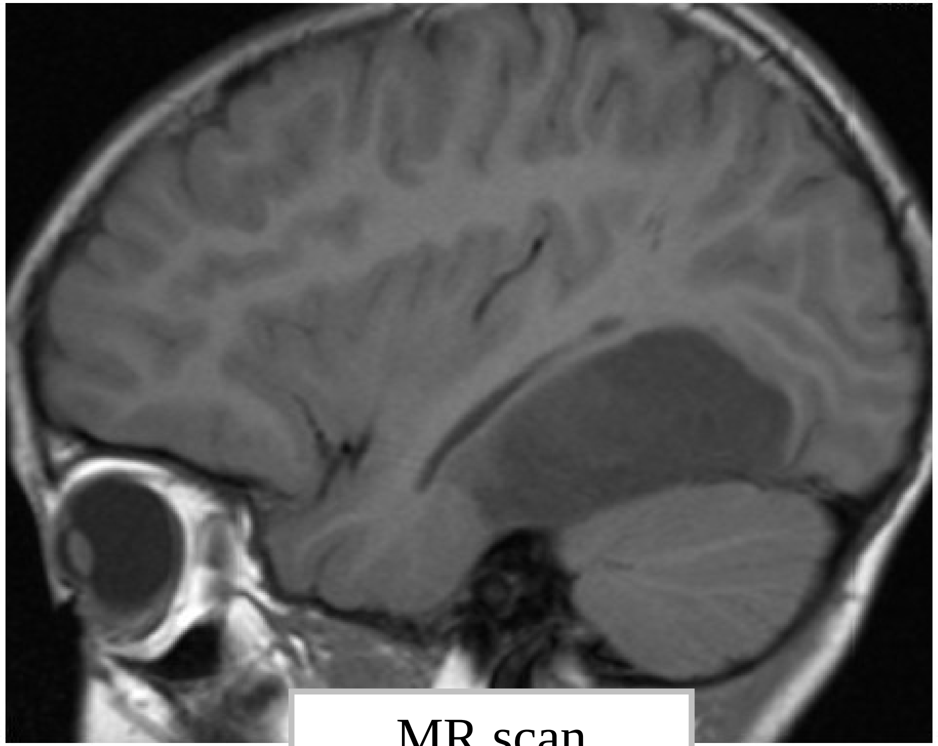




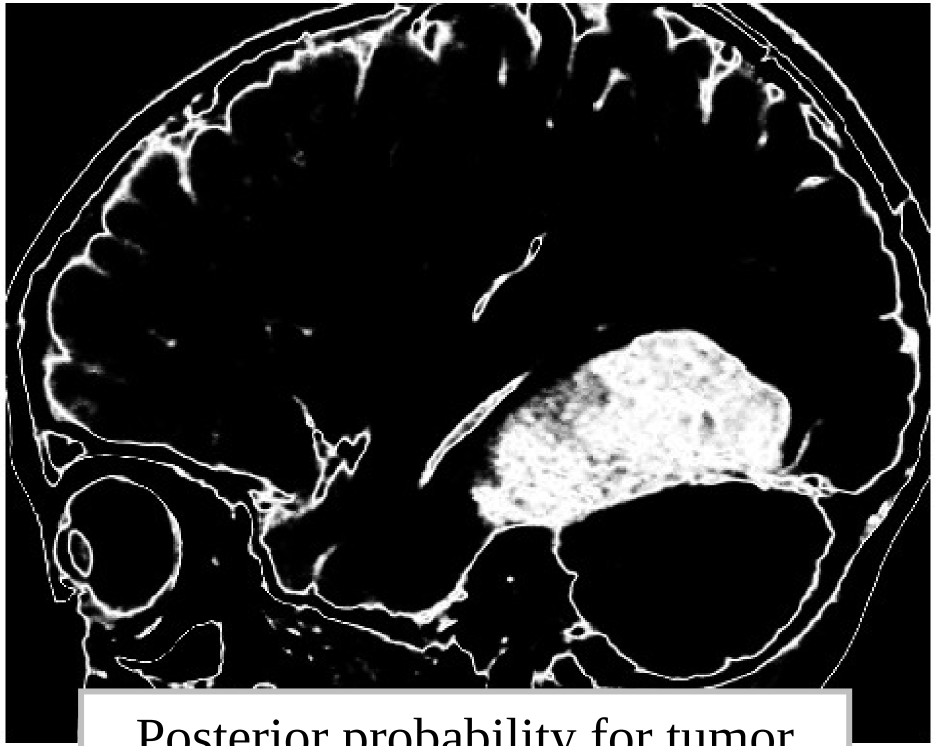
Problem solved?

Two-component Gaussian mixture model:

tumor vs. "other"



MR scan

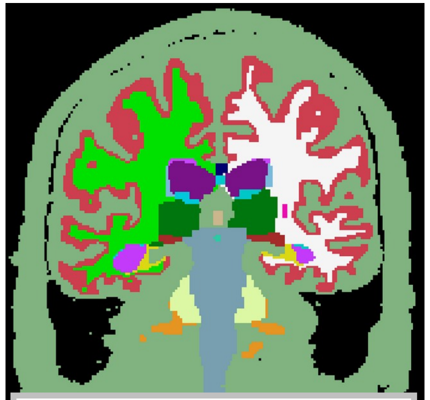


Posterior probability for tumor

Gaussian mixture model

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

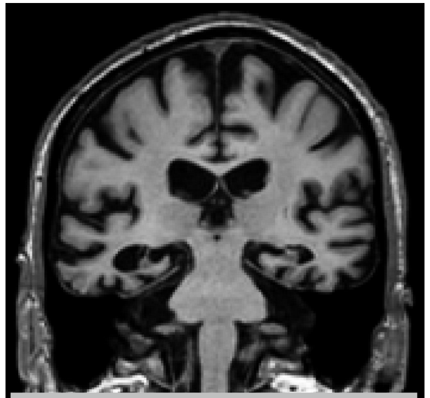
"labeling model"



Label image $\mathbf{l}$

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

"imaging model"



MRI image $\mathbf{d}$

Gaussian mixture model

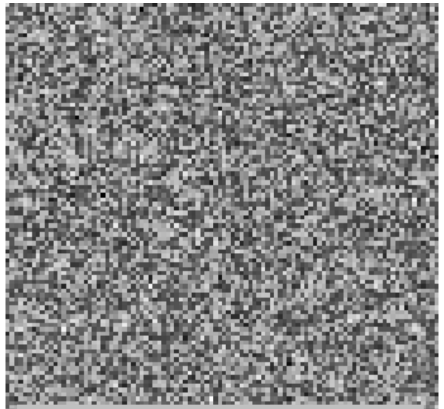
$p(\mathbf{l}|\boldsymbol{\theta}_l)$

“labeling model”

Label image $\mathbf{l}$

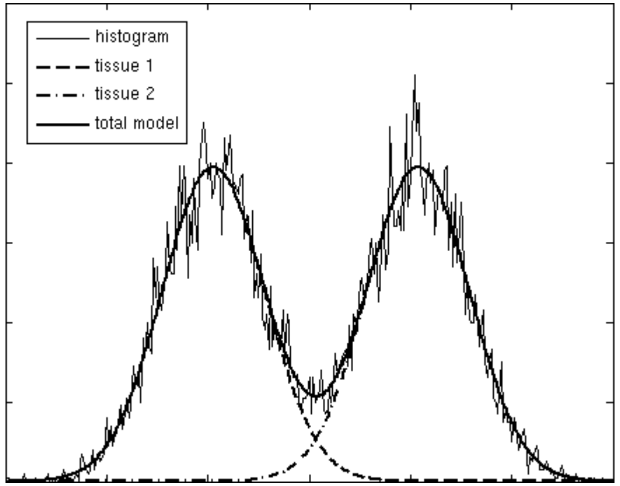
$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d)$

“imaging model”



MRI image $\mathbf{d}$

$$\mu_1 = 70, \mu_2 = 90$$
$$\sigma_1 = 5, \sigma_2 = 5$$
$$\pi_1 = 0.5, \pi_2 = 0.5$$



Gaussian mixture model

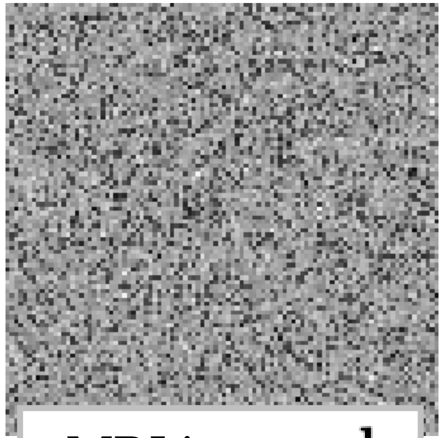
$p(\mathbf{l}|\boldsymbol{\theta}_l)$

"labeling model"

Label image $\mathbf{l}$

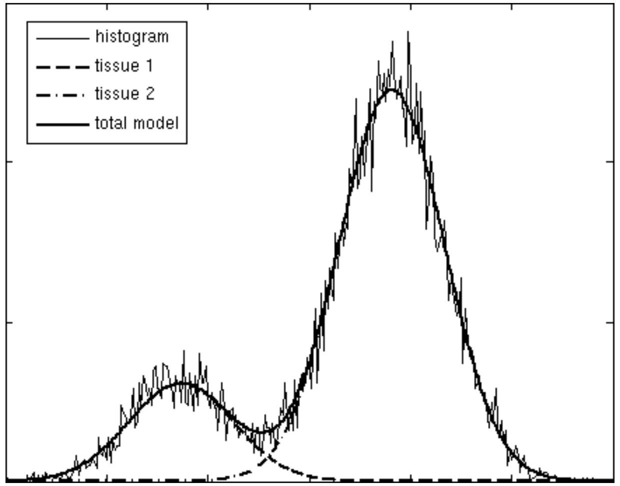
$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d)$

"imaging model"



MRI image $\mathbf{d}$

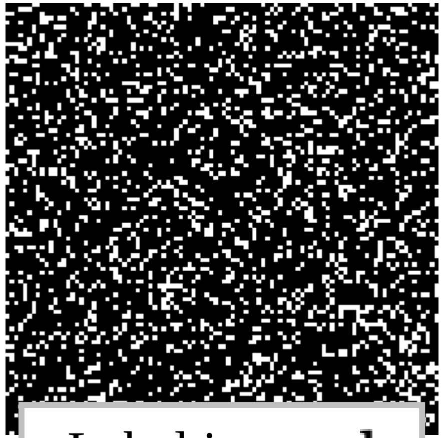
$$\mu_1 = 70, \mu_2 = 90$$
$$\sigma_1 = 5, \sigma_2 = 5$$
$$\pi_1 = 0.2, \pi_2 = 0.8$$



Gaussian mixture model

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

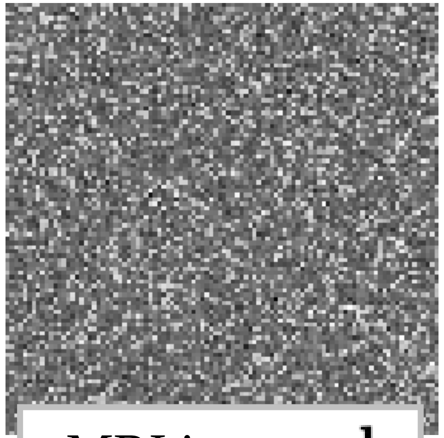
"labeling model"



Label image $\mathbf{l}$

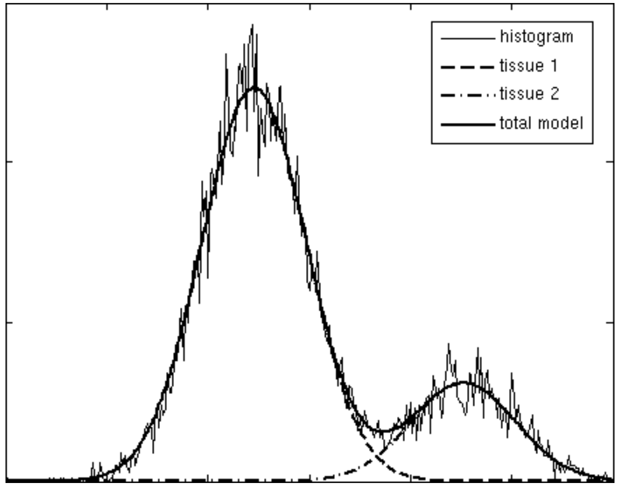
$p(\mathbf{d}|\mathbf{l},\boldsymbol{\theta}_d)$

"imaging model"



MRI image $\mathbf{d}$

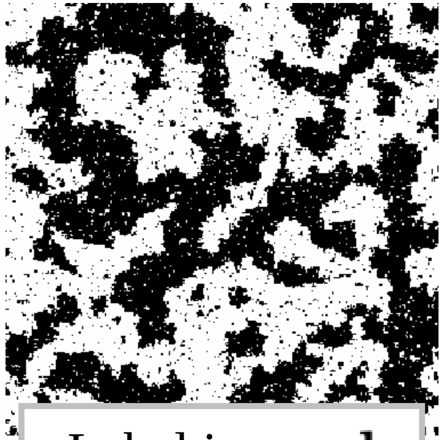
$$\mu_1 = 70, \mu_2 = 90$$
$$\sigma_1 = 5, \sigma_2 = 5$$
$$\pi_1 = 0.8, \pi_2 = 0.2$$



Markov random field model

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

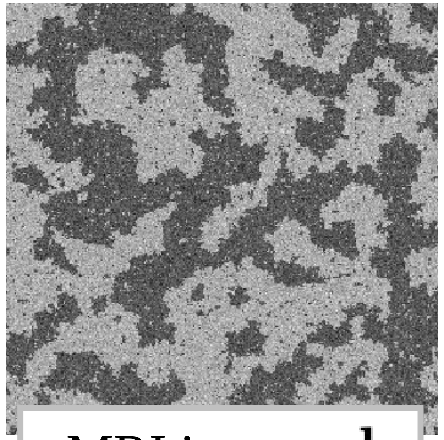
"labeling model"



Label image $\mathbf{l}$

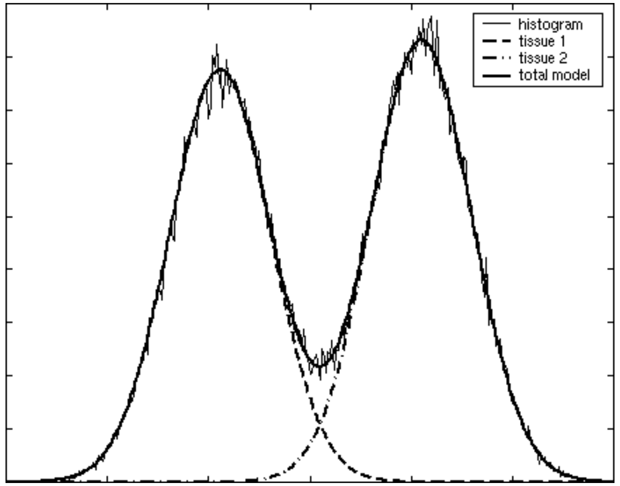
$p(\mathbf{d}|\mathbf{l}, \boldsymbol{\theta}_d)$

"imaging model"



MRI image $\mathbf{d}$

$$\mu_1 = 70, \mu_2 = 90$$
$$\sigma_1 = 5, \sigma_2 = 5$$



Markov random field model

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

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

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

- $Z(\boldsymbol{\theta}_l) = \sum_{\mathbf{l}} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$ is a normalizing constant

Markov random field model

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

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

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

sum over all neighboring voxels

– $Z(\boldsymbol{\theta}_l) = \sum_{\mathbf{l}} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$ is a normalizing constant

Markov random field model

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

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

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

zero if labels are the same,

one otherwise

- $Z(\boldsymbol{\theta}_l) = \sum_{\mathbf{l}} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$ is a normalizing constant

Markov random field model

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

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

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

Parameter controling

strength of penalization

- $Z(\boldsymbol{\theta}_l) = \sum_{\mathbf{l}} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$ is a normalizing constant

Markov random field model

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

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

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

- $Z(\boldsymbol{\theta}_l) = \sum_{\mathbf{l}} \exp(-U(\mathbf{l}|\boldsymbol{\theta}_l))$ is a normalizing constant

Not needed in practice

Markov random field model

- Slightly more general:

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

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

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

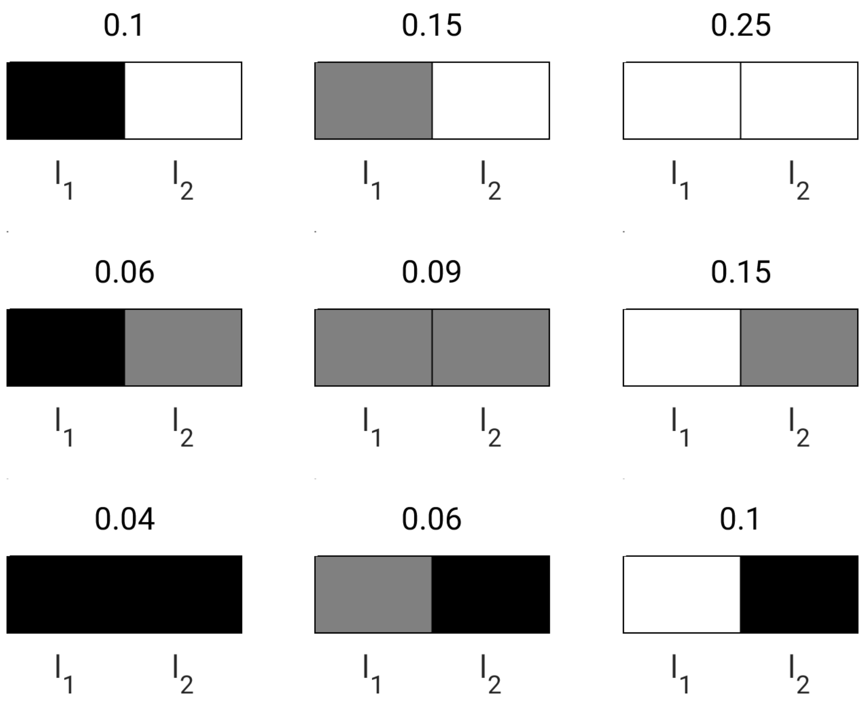
Toy example

$N = 2$ voxels

$K = 3$ classes

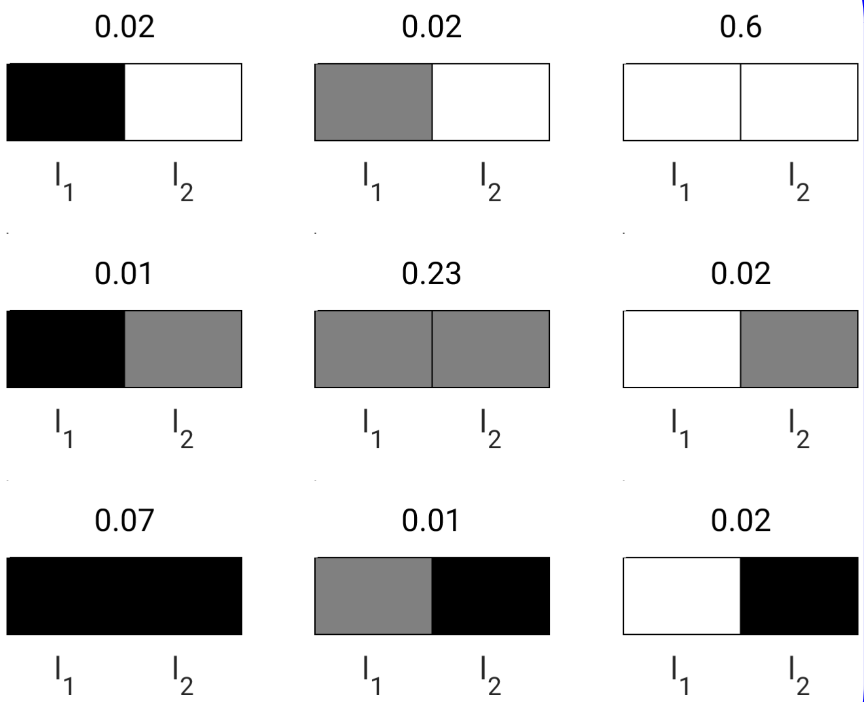
$$\mathbf{l} = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix}$$

$\beta = 0$



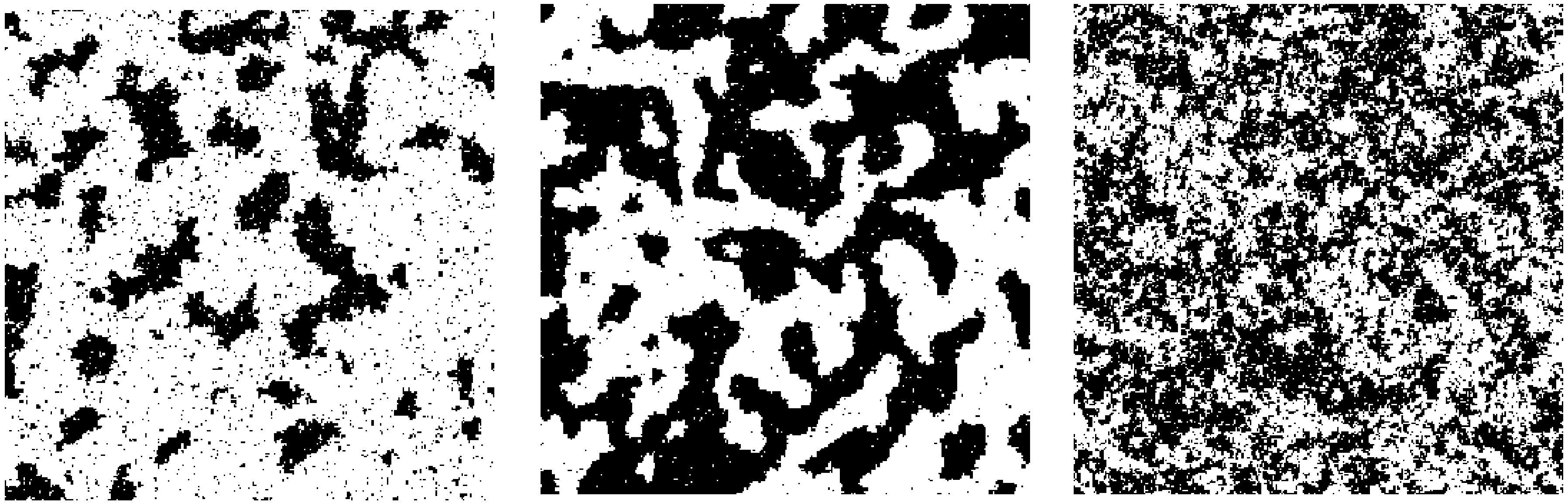
$$p(\mathbf{l}) = p(l_2|\cancel{l_1})p(l_1)$$

$\beta = 3.0$



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

Samples



Different values for model parameters $\boldsymbol{\theta}_l = (\beta, \pi_1, \ldots, \pi_K)^{\mathrm{T}}$

Why exactly this model?

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

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

Why exactly this model?

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

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

All labels except the one of voxel i

Why exactly this model?

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

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

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

- Quantify the difference between the two distributions using the "Kullback-Leibler divergence"

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

Toy example

$N = 2$ voxels

$K = 3$ classes

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

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

l_2 \ l_1	0	1	2
2	0.24	0.06	0.08
1	0.15	0.08	0.11
0	0.11	0.16	0.01

$\simeq$

l_2	
2	??
1	??
0	??

$\cdot$

l_1	0	1	2
	??	??	??

Mean field approximation

– Solution for one voxel i:

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

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:

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

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

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

Mean field approximation

– Solution for one voxel i:

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

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

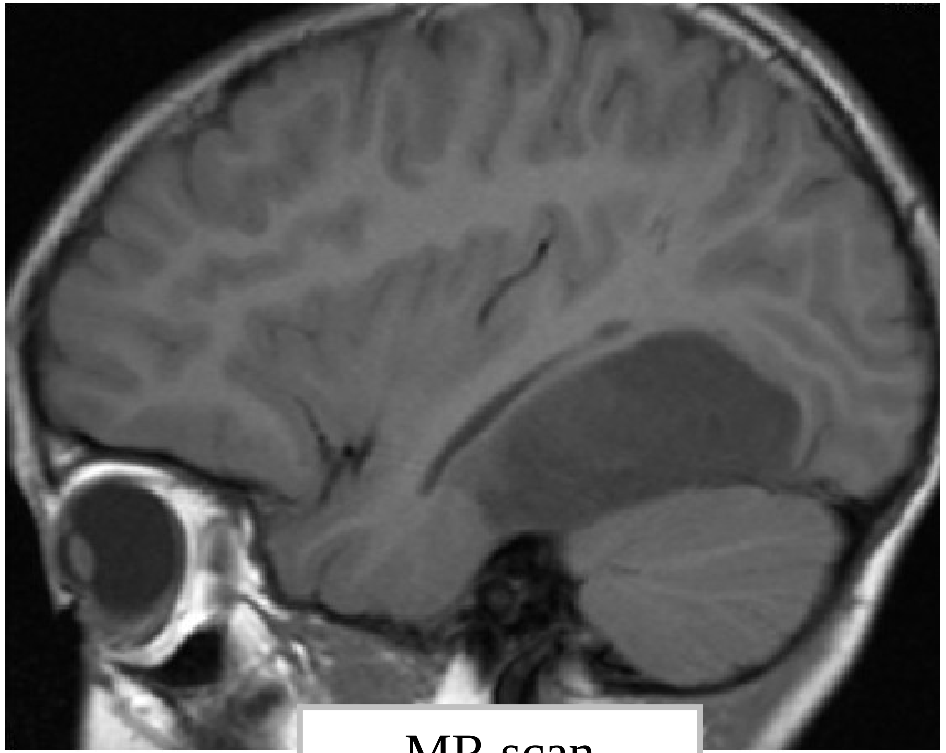
Influenced by the result
in neighboring voxels:
spatial context!!!!

– Need to iterate across all voxels

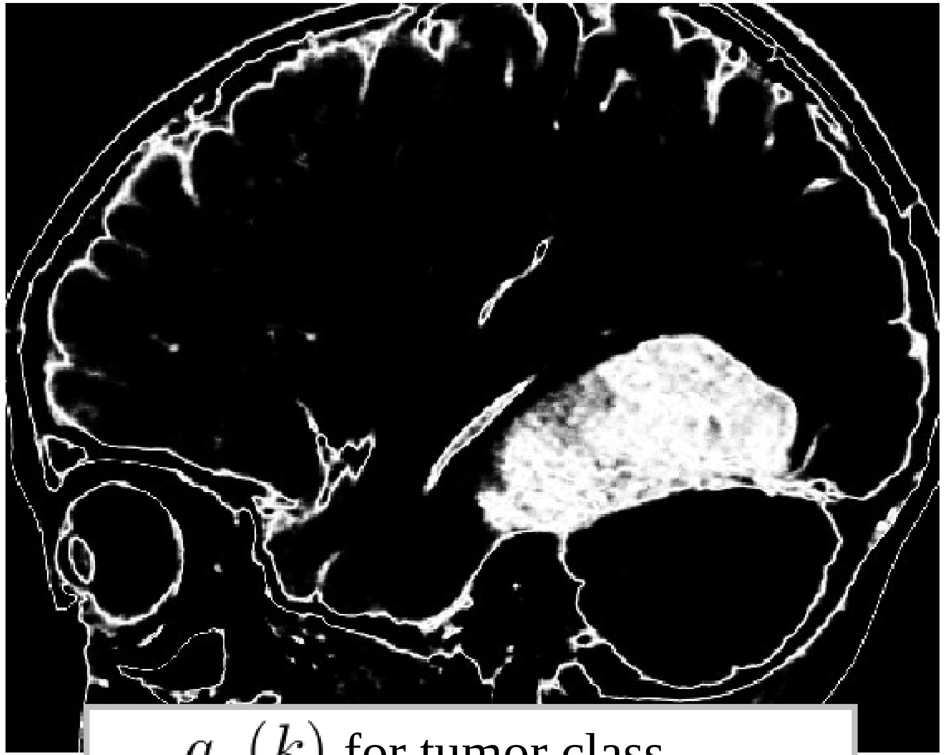
Example

Two-component Gaussian mixture model:

tumor vs. “other”



MR scan



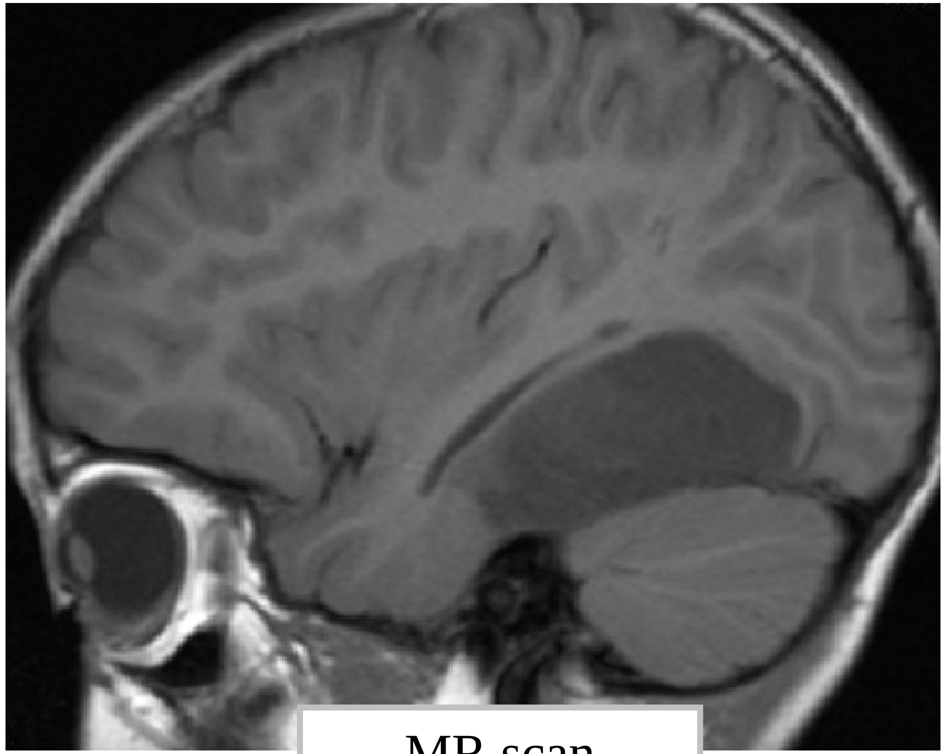
$q_n(k)$ for tumor class

$$\beta = 0$$

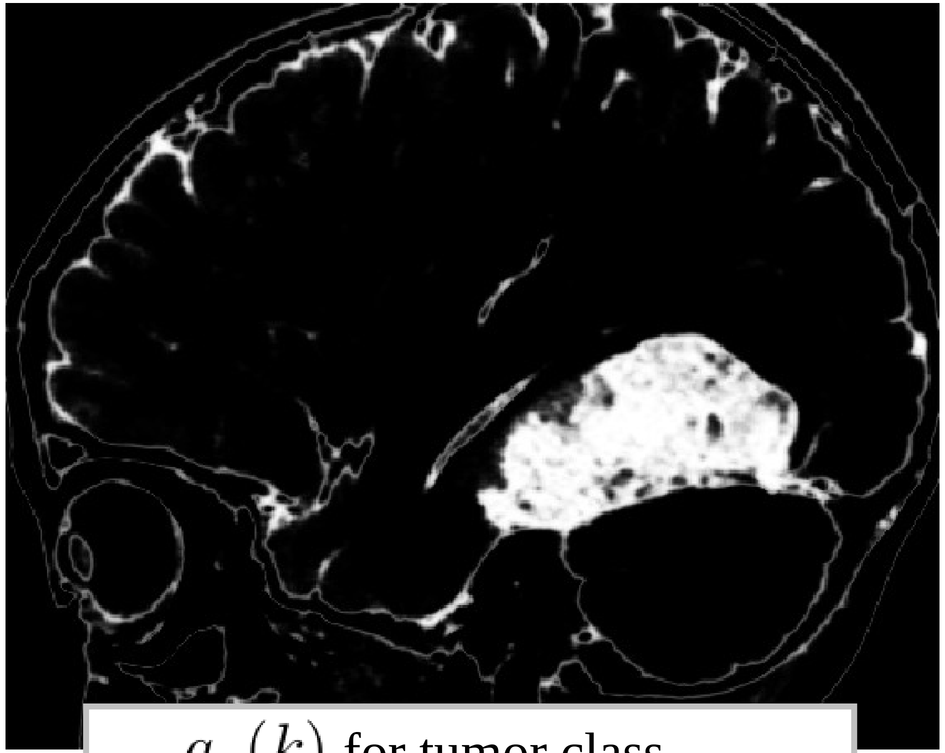
Example

Two-component Gaussian mixture model:

tumor vs. "other"



MR scan



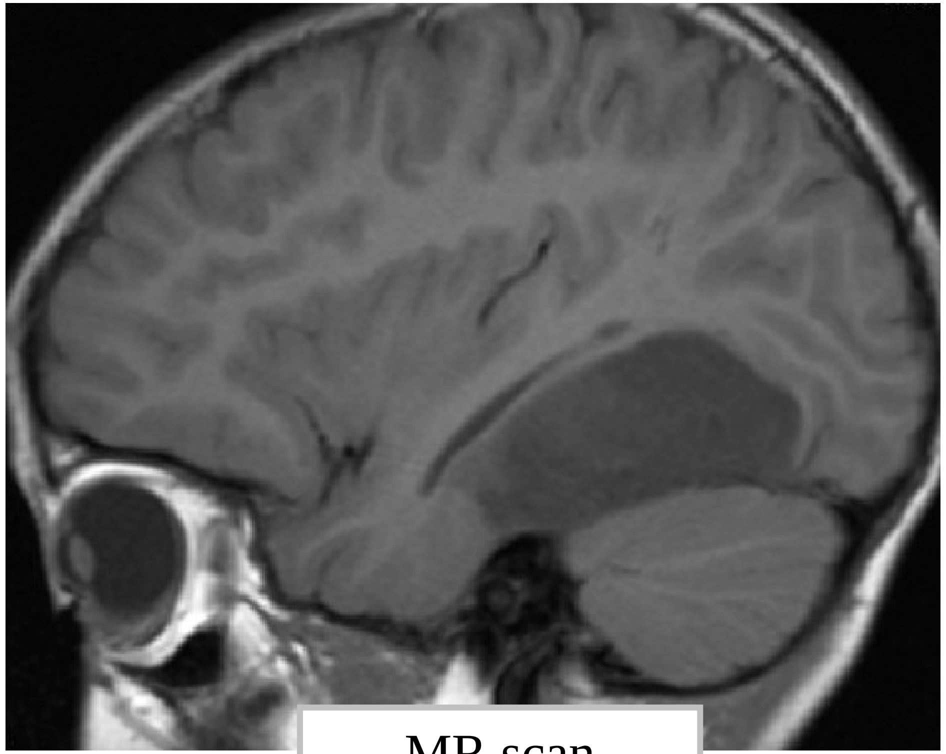
$q_n(k)$ for tumor class

$\beta = 0.25$

Example

Two-component Gaussian mixture model:

tumor vs. "other"



MR scan

$q_n(k)$ for tumor class

$$\beta = 0.55$$