Generative models for segmentation I

Course 22525

Koen Van Leemput
DTU HealthTech
Technical University of Denmark
Course structure

Fitting functions

Registration

Segmentation
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 week and next week

Automated computational methods
The problem to be solved

\[ N \text{ voxels} \]

\[ d = (d_1, \ldots, d_N)^T \]

\[ d_n: \text{ intensity in voxel } n \]
The problem to be solved

MRI image $d$

Label image $l$

$l = (l_1, \ldots, l_N)^T$

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

$K$: number of classes
One solution: generative modeling

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

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

\[ N = 2 \text{ voxels} \]
\[ K = 3 \text{ classes} \]

\[ l = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \]

\[ p(l) = p(l_1, l_2) \]

\[
\begin{array}{c c c}
0.24 & 0.06 & 0.08 \\
I_1 & I_2 & I_1 \\
0.15 & 0.08 & 0.11 \\
I_1 & I_2 & I_1 \\
0.11 & 0.16 & 0.01 \\
I_1 & I_2 & I_1 \\
\end{array}
\]

\[
\begin{array}{c c c}
0.24 & 0.06 & 0.08 \\
0.15 & 0.08 & 0.11 \\
0.11 & 0.16 & 0.01 \\
\end{array}
\]
Toy example

\[ N = 2 \text{ voxels} \]
\[ K = 3 \text{ classes} \]

\[ \mathbf{d} = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \]

\[ p(\mathbf{d} | \mathbf{l}) = p(d_1, d_2 | l_1, l_2) \]
One solution: generative modeling

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

\[ p(l|\theta_l) \quad \text{“labeling model”} \]

\[ p(d|l, \theta_d) \quad \text{“imaging model”} \]

- 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...
Segmentation = inverse problem

MRI image \text{d}

Label image \text{l}
Segmentation = inverse problem

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

$$p(l|d, \hat{\theta}) = \frac{p(d|l, \hat{\theta}_d)p(l|\hat{\theta}_l)}{p(d|\hat{\theta})}$$
Segmentation = inverse problem

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

$$\hat{l} = \arg \max_l p(l|d, \hat{\theta})$$

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

- **MRI image** $d$
- **Label image** $l$
Segmentation = inverse problem

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

The imaging model

$$p(l|d, \hat{\theta}) = \frac{p(d|l, \hat{\theta}_d)p(l|\hat{\theta}_l)}{p(d|\hat{\theta})}$$
Segmentation = inverse problem

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

\[
p(l|d, \hat{\theta}) = \frac{p(d|l, \hat{\theta}_d)p(l|\hat{\theta}_l)}{p(d|\hat{\theta})} = \sum_l p(d|l, \hat{\theta}_d)p(l|\hat{\theta}_l)
\]  

(but not needed)
Gaussian mixture model

- Assign a label to each voxel independently
- Probability of assigning label $k$ is $\pi_k$

$$p(1|\theta_l) = \prod_n \pi_{l_n}, \quad \theta_l = (\pi_1, \ldots, \pi_K)^T$$
Toy example

\[ N = 2 \text{ voxels} \]
\[ K = 3 \text{ classes} \]

\[ l = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \]

\[ p(l) = p(l_1, l_2) = p(l_2|l_1)p(l_1) \]
Gaussian mixture model

$p(1|\theta_l)$

"labeling model"

\[ p(d|l, \theta_d) = \prod_{n} \mathcal{N}(d_n|\mu_{l_n}, \sigma_{l_n}^2) , \quad \theta_d = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2)^T \]

Drawn the intensity in each voxel with label \( k \) from a Gaussian distribution with mean \( \mu_k \) and variance \( \sigma_k^2 \)

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

Label image \( l \)

MRI image \( d \)
Toy example

\(N = 2\) voxels

\(K = 3\) classes

\[
d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}
\]

\[
p(d|l) = p(d_1, d_2|l_1, l_2) = p(d_2|l_1, l_2, d_1)p(d_1|l_1, l_2)
\]
Gaussian mixture model

\[ p(d|\theta) = \prod_n \left( \sum_k N(d_n | \mu_k, \sigma_k^2) \pi_k \right) \]

\[ \theta = (\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, \pi_1, \ldots, \pi_K)^T \]
Posterior probability distribution

\[
p(l|d, \hat{\theta}) = \frac{p(d|l, \hat{\theta}_d)p(l|\hat{\theta}_l)}{p(d|\hat{\theta})} = \frac{\prod_n \mathcal{N}(d_n|\hat{\mu}_{l_n}, \hat{\sigma}^2_{l_n}) \prod_n \hat{\pi}_{l_n}}{\prod_n \sum_k \mathcal{N}(d_n|\hat{\mu}_k, \hat{\sigma}^2_k) \hat{\pi}_k} = \prod_n p(l_n|d_n, \hat{\theta})
\]

\[
p(l_n|d_n, \hat{\theta}) = \frac{\mathcal{N}(d_n|\hat{\mu}_{l_n}, \hat{\sigma}^2_{l_n}) \hat{\pi}_{l_n}}{\sum_k \mathcal{N}(d_n|\hat{\mu}_k, \hat{\sigma}^2_k) \hat{\pi}_k}
\]
Maximum a posteriori segmentation

\[ \hat{l} = \arg \max_l p(l | d, \hat{\theta}) = \arg \max_{l_1, \ldots, l_I} p(l_n | d_n, \hat{\theta}) \]
Problem solved?

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

MR scan

Posterior probability for tumor
Gaussian mixture model

\[ p(1|\theta_l) \]

“labeling model”

Label image \( I \)

\[ p(d|l, \theta_d) \]

“imaging model”

MRI image \( d \)
Gaussian mixture model

\[ p(1|\theta_l) \]

“labeling model”

\[ p(d|1, \theta_d) \]

“imaging model”

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

Label image 1

MRI image d
Gaussian mixture model

\[ p(l | \theta_l) \rightarrow \text{"labeling model"} \]

\[ \text{Label image } l \]

\[ p(d | l, \theta_d) \rightarrow \text{"imaging model"} \]

\[ \text{MRI image } d \]

\[ \mu_1 = 70, \mu_2 = 90 \]
\[ \sigma_1 = 5, \sigma_2 = 5 \]
\[ \pi_1 = 0.2, \pi_2 = 0.8 \]
Gaussian mixture model

$p(1|\theta_l)$
“labeling model”

Label image $l$

$p(d|l, \theta_d)$
“imaging model”

MRI image $d$

$\mu_1 = 70, \mu_2 = 90$
$\sigma_1 = 5, \sigma_2 = 5$
$\pi_1 = 0.8, \pi_2 = 0.2$
Markov random field model

$p(l | \theta_l)$

“labeling model”

Label image $l$

$p(d | l, \theta_d)$

“imaging model”

MRI image $d$

$\mu_1 = 70, \mu_2 = 90$

$\sigma_1 = 5, \sigma_2 = 5$
Markov random field model

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

\[ p(1|\theta_l) = \frac{1}{Z(\theta_l)} \exp(-U(1|\theta_l)) \]

\[ U(1|\theta_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j) \]

- \( Z(\theta_l) = \sum_1 \exp(-U(1|\theta_l)) \) is a normalizing constant
Markov random field model

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

\[ p(1|\theta_l) = \frac{1}{Z(\theta_l)} \exp(-U(1|\theta_l)) \]

\[ U(1|\theta_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j) \]

sum over all neighboring voxels

- \[ Z(\theta_l) = \sum_1 \exp(-U(1|\theta_l)) \] is a normalizing constant
Markov random field model

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

\[ p(1|\theta_l) = \frac{1}{Z(\theta_l)} \exp(-U(1|\theta_l)) \]

\[ U(1|\theta_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j) \]

zero if labels are the same,
one otherwise

- \[ Z(\theta_l) = \sum_1 \exp(-U(1|\theta_l)) \] is a normalizing constant
Markov random field model

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

\[
p(1|\theta_l) = \frac{1}{Z(\theta_l)} \exp(-U(1|\theta_l))
\]

\[
U(1|\theta_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j)
\]

Parameter controlling strength of penalization

- \(Z(\theta_l) = \sum_1 \exp(-U(1|\theta_l))\) is a normalizing constant
Markov random field model

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

\[ p(1|\theta_l) = \frac{1}{Z(\theta_l)} \exp(-U(1|\theta_l)) \]

\[ U(1|\theta_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j) \]

\[ Z(\theta_l) = \sum_1 \exp(-U(1|\theta_l)) \]

Not needed in practice
Markov random field model

- Slightly more general:

\[
p(1|\theta_l) = \frac{1}{Z(\theta_l)} \exp(-U(1|\theta_l))
\]

\[
U(1|\theta_l) = \beta \sum_{(i,j)} \delta(l_i \neq l_j) - \sum_{i} \log(\pi_{l_i})
\]

- \( \theta_l = (\beta, \pi_1, \ldots, \pi_K)^T \) are the model parameters

- Reduces to Gaussian mixture model prior \( p(1|\theta_l) = \prod_n \pi_{l_n} \) for \( \beta = 0 \)
$N = 2$ voxels

$K = 3$ classes

\[ l = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \]

\[ p(l) = p(l_2 | l_1) p(l_1) \]

\[ \beta = 0 \]

\[ \beta = 3.0 \]
Different values for model parameters $\theta_l = (\beta, \pi_1, \ldots, \pi_K)^T$
Why exactly this model?

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

\[
\begin{align*}
p(l_i|\mathbf{1}_{\setminus i}) &= \frac{p(1)}{p(1|\setminus i)} \\
&= \frac{p(1)}{\sum_{l_i} p(1)} \\
&= \frac{\exp(-U(1|\mathbf{\theta}_l))}{\sum_{l_i} \exp(-U(1|\mathbf{\theta}_l))} \\
&= \pi_{l_i} \cdot \exp \left( -\beta \sum_{j \in \mathcal{N}_i} \delta(l_i \neq l_j) \right) \\
&= \sum_k \pi_k \cdot \exp \left( -\beta \sum_{j \in \mathcal{N}_i} \delta(l_j \neq k) \right)
\end{align*}
\]
Why exactly this model?

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

\[
p(l_i | 1_{\setminus i}) = \frac{p(1)}{p(1_{\setminus i})}
= \frac{p(1)}{\sum_{l_i} p(1)}
= \frac{\exp(-U(1|\theta_l))}{\sum_{l_i} \exp(-U(1|\theta_l))}
= \pi_{l_i} \cdot \exp\left( -\beta \sum_{j \in n_i} \delta(l_i \neq l_j) \right)
= \frac{\pi_{l_i} \cdot \exp\left( -\beta \sum_{j \in n_i} \delta(l_i \neq l_j) \right)}{\sum_k \pi_k \cdot \exp\left( -\beta \sum_{j \in n_i} \delta(l_j \neq k) \right)}
\]
Why exactly this model?

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

\[
\begin{align*}
    p(l_i | 1_{\setminus i}) &= \frac{p(1)}{p(1_{\setminus i})} \\
    &= \frac{p(1)}{\sum_{l_i} p(1)} \\
    &= \frac{\exp(-U(1|\theta_l))}{\sum_{l_i} \exp(-U(1|\theta_l))} \\
    &= \frac{\pi_{l_i} \cdot \exp\left(-\beta \sum_{j \in n_i} \delta(l_i \neq l_j)\right)}{\sum_k \pi_k \cdot \exp\left(-\beta \sum_{j \in n_i} \delta(l_j \neq k)\right)}
\end{align*}
\]
Mean field approximation

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

\[ p(l|d, \hat{\theta}) = \prod_n p(l_n|d_n, \hat{\theta}) \]

- 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 $p(l|d, \hat{\theta})$
Mean field approximation

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

$$q(l) = \prod_n q_n(l_n)$$

- Find the voxel-wise distributions $q_n(k)$ that minimize the difference between $q(l)$ and $p(l|d, \hat{\theta})$

- Quantify the difference between the two distributions using the “Kullback-Leibler divergence”

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

\(N = 2\) voxels

\(K = 3\) classes

\[
l = \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} \quad d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}
\]

\[
p(l|d) = p(l_1, l_2|d_1, d_2) \approx q(l_1)q(l_2)
\]
Mean field approximation

- Solution for one voxel $i$:

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

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

Influenced by the result in neighboring voxels: spatial context!!!!
Mean field approximation

- Solution for one voxel $i$:

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

where

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

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

$\eta_n(k)$ for tumor class

$\beta = 0.55$