CS736 Assignment 2

Adish Shah, Prathamesh Pilkhane and Padakanti Akshay Kiran

March 2022

Contents

1 Q1

1.1 Choice of q

We have chosen q based on the most closest image which gave good segmentation image,

$$q = 2.45$$

1.2 Neighbourhood mask w

We use the Gaussian filter mask of size 9X9 with standard deviation equal to 5. A higher dimension helps in smoothening the image and reduce the effects of bias by convolution. Note: The above

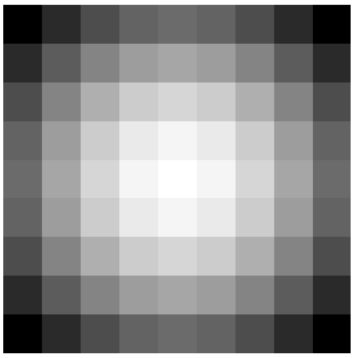
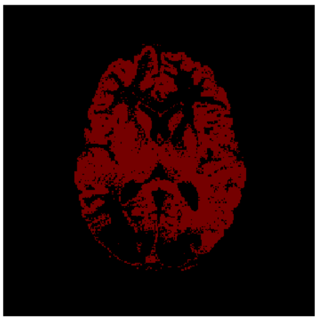


Figure 1: Weights

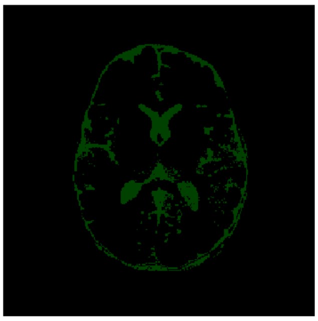
image has been made considering the brightest pixel corresponding to max value. If we make the image with 0 as white and 1 as black then these weights become more of a black image.

1.3 Initial Segmentations

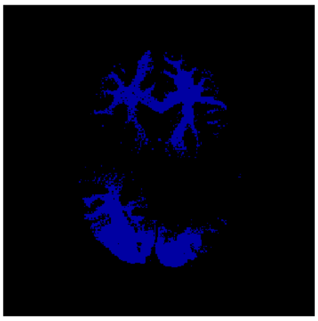
For initializing the segmentations and means, we apply a hard k-means on the masked image. Although the hard k-means does not provide the best possible segmentation (as image has bias) , we however get a decent choice and a rough estimate of the values present.



(a) Initial Class 1 Segmentation



(b) Initial Class 2 Segmentation



(c) Initial Class 3 Segmentation

Figure 2: Three Classes of Segmentation

1.4 Initial Means

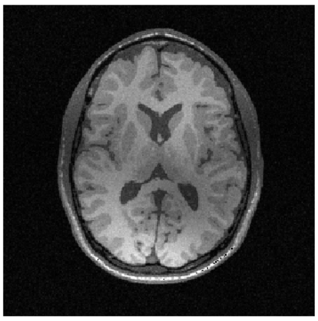
As mentioned above, the initial means are based on calculating the means as obtained from the hard k-means on the masked image. The above segmentation is based on the means as follows (in order) :

$$C = [0.4577 \quad 0.2539 \quad 0.6367]$$

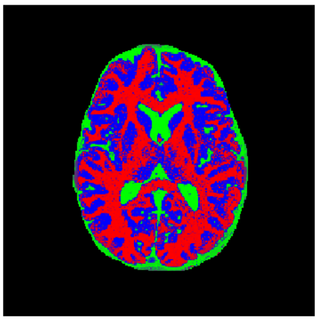
1.5 The Objective Values obtained in each iteration

The objective value for one of the runs of the algorithm are presented below: [328.5070 , 62.2553 , 44.5831 , 42.6225 , 42.2889 , 42.1970 , 42.1462 , 42.0864 , 42.0030 , 40.35895 , 41.7423 , 41.5579 , 41.3318 , 41.0596 , 40.7334 , 40.3391 , 39.8549 , 39.2474 , 38.4685 , 37.4561 , 36.1519 , 34.5615 , 32.8295 , 31.1663 , 29.8979 , 29.1066 , 28.5953 , 28.3149 , 28.1653 , 28.0862 , 28.0448 , 28.0252 , 28.0252]

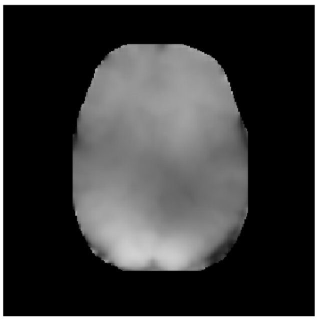
1.6 Images



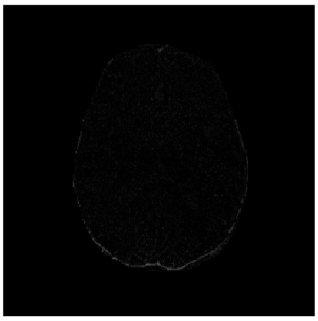
(a) Original Image



(b) Obtained segmentation



(c) Bias Field Estimate



(d) Residual Image

(e) Negative of Residual Image

Figure 3: Images obtained

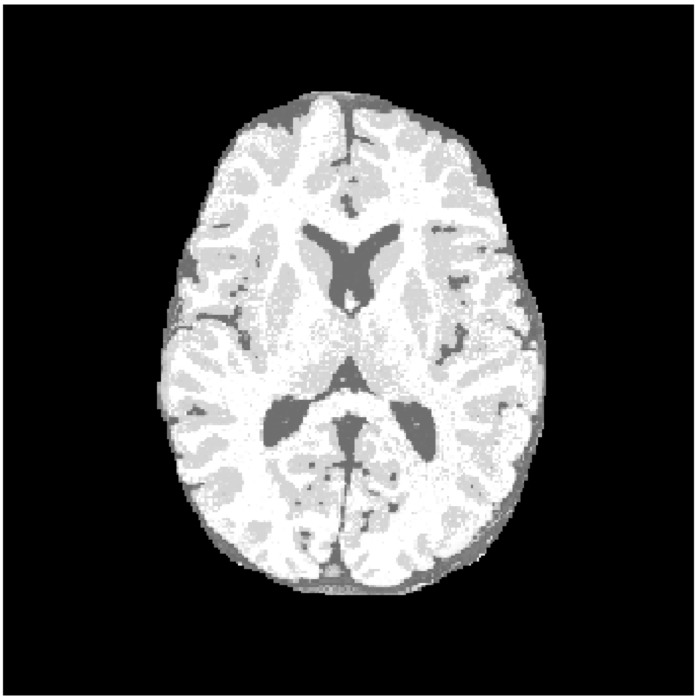


Figure 4: Bias Removed Image

1.7 Obtained Class Means

$$C_{\text{final}} = [0.8333, 0.4308, 1.0165]$$

Note, the mean of greater than 1 in a black and white image is due to the fact that the solution is not completely unique.
All the above quoted values are based on the implementation as discussed in class. These are not based on the suggested modification

1.8 Does the formulation give unique solution?

No, the formulation does not give a unique solution over C's and the bias values while however maintaining the same value of their product.
Proposed Solution : Since the formulation is not constrained, we add a constraint in the form of regularization on the bias field. Using a ridge regression, we can ensure that we get a minimum bias value, and corresponding a unique class mean values. Thus our J function now becomes

$$J = \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^{q} \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2 + \lambda_{jk} b_i^2$$

Thus , the only changes that occur are in d_{jk} and estimate of b_i as follows:

$$d_{jk} = \sum_{i=1}^{N} w_{ij}(y_j - c_k b_i)^2 + \lambda_{jk} b_i^2$$

,

$$b_i = \frac{\sum_{j=1}^{N} w_{ij} y_j \sum_{k=1}^{K} u_{jk}^{q} c_k}{\sum_{j=1}^{N} w_{ij} \sum_{k=1}^{K} u_{jk}^{q} c_k^2 + \sum_{j=1}^{N} \sum_{k=1}^{K} u_{jk}^{q} \lambda_{jk}}$$

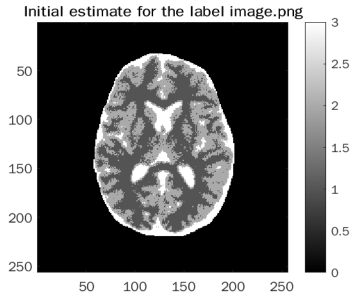
2 Q2

2.1 Chosen value of β

After manually tuning the β values, we chose the value of $\beta = 0.35$

2.2 Initial estimate for the label image x

We used K-means with K = 3 for label initialization. The motivation for this is that it gives a quick division of the values into 3 classes.



2.3 Initial estimates for Gaussian parameters

We used the label initialization to get means and variances. The initial class means are obtained using K-means and the initial standard deviation estimates are the average distances of any pixel value from the respective class means. This gives us a quick and good estimate of the optimal standard deviations.

2.4 Log posterior values

2.4.1 For the chosen β value

```
*** Starting modified ICM with beta = 0.350000 ***
Iteration 1: Log posterior before = 30485.634868
Iteration 1: Log posterior after = 31814.432866
Iteration 2: Log posterior before = 31926.111165
Iteration 2: Log posterior after = 32112.055738
Iteration 3: Log posterior before = 32207.237137
Iteration 3: Log posterior after = 32333.343951
Iteration 4: Log posterior before = 32390.451611
Iteration 4: Log posterior after = 32410.998630
Iteration 5: Log posterior before = 32438.794275
Iteration 5: Log posterior after = 32493.572927
Iteration 6: Log posterior before = 32502.480264
Iteration 6: Log posterior after = 32487.300660
```

2.4.2 For $\beta = 0$

```
*** Starting modified ICM with beta = 0.000000 ***
Iteration 1: Log posterior before = 35920.434868
Iteration 1: Log posterior after = 36321.566497
Iteration 2: Log posterior before = 36410.323969
Iteration 2: Log posterior after = 36563.648048
```

```
Iteration 3: Log posterior before = 36515.118912
Iteration 3: Log posterior after = 36605.971540
Iteration 4: Log posterior before = 36539.095232
Iteration 4: Log posterior after = 36592.323561
Iteration 5: Log posterior before = 36501.416814
Iteration 5: Log posterior after = 36533.674405
Iteration 6: Log posterior before = 36434.059587
Iteration 6: Log posterior after = 36457.048592
Iteration 7: Log posterior before = 36361.079212
Iteration 7: Log posterior after = 36372.608480
Iteration 8: Log posterior before = 36287.951860
Iteration 8: Log posterior after = 36294.530336
Iteration 9: Log posterior before = 36228.886900
Iteration 9: Log posterior after = 36232.161161
Iteration 10: Log posterior before = 36184.501185
Iteration 10: Log posterior after = 36186.001351
Iteration 11: Log posterior before = 36153.498537
Iteration 11: Log posterior after = 36154.521189
Iteration 12: Log posterior before = 36133.656769
Iteration 12: Log posterior after = 36134.298002
Iteration 13: Log posterior before = 36121.321450
Iteration 13: Log posterior after = 36121.711128
Iteration 14: Log posterior before = 36113.881495
Iteration 14: Log posterior after = 36114.175475
Iteration 15: Log posterior before = 36109.613773
Iteration 15: Log posterior after = 36109.735806
Iteration 16: Log posterior before = 36107.039852
Iteration 16: Log posterior after = 36107.114492
Iteration 17: Log posterior before = 36105.545271
Iteration 17: Log posterior after = 36105.597131
Iteration 18: Log posterior before = 36104.677209
Iteration 18: Log posterior after = 36104.693527
Iteration 19: Log posterior before = 36104.144361
Iteration 19: Log posterior after = 36104.168447
Iteration 20: Log posterior before = 36103.849148
Iteration 20: Log posterior after = 36103.851306
```

2.5 Images

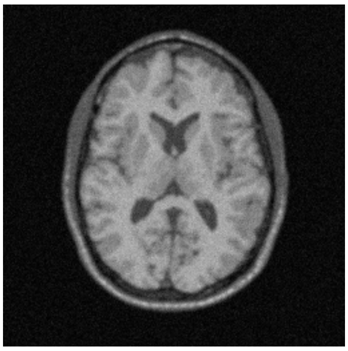


Figure 5: Corrupted Image

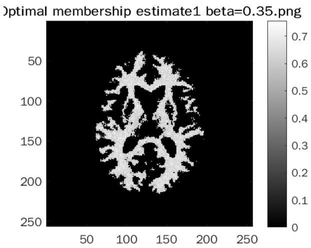


Figure 6: Optimal class membership estimate 1 for beta = 0.35

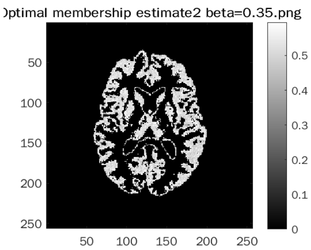


Figure 7: Optimal class membership estimate 2 for beta = 0.35

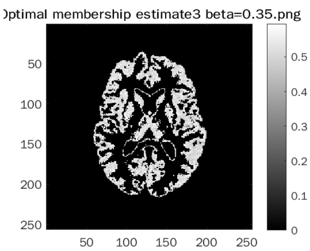


Figure 8: Optimal class membership estimate 3 for beta = 0.35

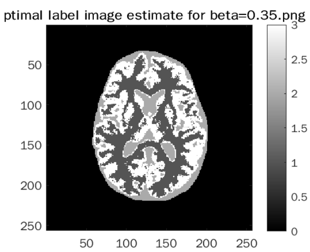


Figure 9: Optimal label image estimate for beta = 0.35

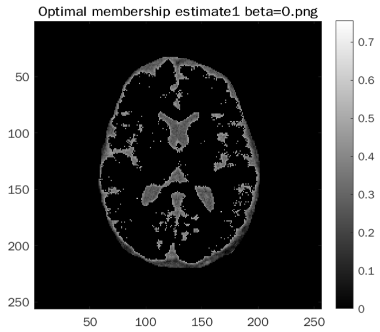


Figure 10: Optimal class membership estimate 1 for beta = 0

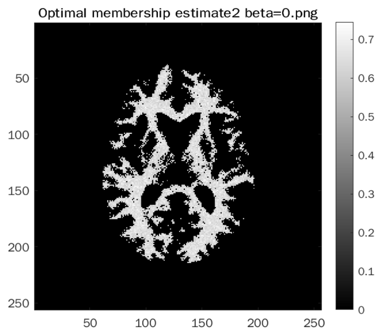


Figure 11: Optimal class membership estimate 2 for beta = 0

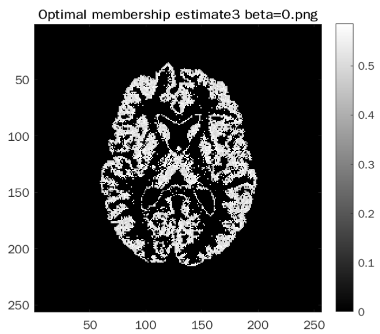


Figure 12: Optimal class membership estimate 3 for beta = 0

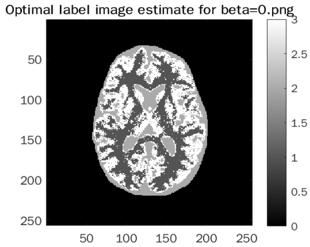


Figure 13: Optimal label image estimate for beta = 0

2.6 Optimal estimates for the chosen β

The optimal estimates for the class means for $\beta = 0.35$ are

$$C = [0.632014 \quad 0.522715 \quad 0.308619]$$

3 Problem 3

Data y := y_n where n = 1,2,3 ... N
Given: Prior on θ P(θ)

Startegy: MAP Estimation, We need to find :
argmax(θ) L($\theta|y$) = argmax(θ) {P(y|θ)*P(θ)} = argmax(θ) {log(P(y|θ))+log(P(θ))} = argmax(θ) $\sum_{n=1}^{N} log(\sum_{k=1}^{K} w_k * G(y_n; \mu_k, C_k)) + log(P(\theta))$

3.1 E step

This step remains the same even after adding prior

$F(q, \theta) = logP(y|\theta) + log(P(\theta)) - KL(P(x|y, \theta^i)P(x|y, \theta))$
So, evaluated at θ^i, $F(q, \theta^i) = log(P(y|\theta^i)) + log(P(\theta^i)) = log(L(\theta^i|y))$
So, F(.) touches log-MAP function at θ^i

3.2 M step

rewrite $F(q, \theta) = E_{q(.)}[logP(x, y|\theta)] + H(q) + log(P(\theta))$, where H(q) = entropy of q(.) is not a function of θ

Call $E_{q(.)}[logP(x, y|\theta)] + log(P(\theta))$ as $Q(\theta; \theta_i)$ function

Thus the quantity to be maximised is: (simplification is same as slides)
$Q(\theta; \theta^i) = \sum_{n=1}^{N} \sum_{k=1}^{K} P(z_n = k|y_n, \theta^i) * log(P(y_n|z_n = k, \theta)P(z_n = k|\theta)) + log(P(\theta))$

3.3 Using GMM

Goal: fit a GMM with K components
$P(x) = \sum_{k=1}^{K} w_k * G(x; \mu_k, C_k)$ Parameters $\theta = \{w_k, \mu_k, C_k\}_{k=1}^{K}$

3.3.1 Prior models for GMM parameters

3(i) a) ~~Prior~~ Prior for means

→ We can assume that prior for means will be a MVG with some mean $\overline{\Pi}$ and covariance $\overline{C}$, where $\overline{\Pi}$ and $\overline{C}$ are hyper-parameters.

$$\text{Let} \quad P(\mu_i) = G(\mu_i ; \overline{\Pi}, \overline{C}) \quad \text{for all } \mu_1, \mu_2 \ldots \mu_K$$

b) Prior for weights

→ $Q(\theta, \theta^i) = \sum q(x) \log P(x, y \mid \theta) + \log P(\theta)$

$= \sum q(x) \left(\log \left(P(x, y \mid \theta) \times P(\theta) \right) \right)$ $\left(\text{Since } \sum q(x) = 1 \right)$

$= \sum_n \sum_k \gamma_{nk} \left(-0.5 \log |C_k| - 0.5 (y_n - \mu_k)^T C_k^{-1} (y_n - \mu_k) + \log w_k \right) + \log P(\theta)$ ①

Thus,

$$\arg\max_\theta \left(Q(\theta, \theta^i) \right) = \arg\max_\theta e^{Q(\theta, \theta^i)}$$

$$= \prod_{n,k} \left(G(y_n, \mu_k, C_k)^{\gamma_{nk}} \right) \times \left(\prod_n \prod_k w_k^{\gamma_{nk}} \right) \times P(\theta)^{\sum_n \sum_k \gamma_{nk}}$$

→ Since it is proportional to $\prod_n \prod_k w_k^{\gamma_{nk}}$, it is ~~multinomial~~ dirichlet distribution in-terms of weights.

→ Thus we should assume a dirichlet prior on weights ~~as~~ which is it's conjugate prior for multinomial distribution.

$$P(W) = \text{Dir}(W; \alpha) \quad \left(\text{where } W \in \mathbb{R}^K \text{ is } [w_1, w_2 \ldots w_K] \right)$$

(α is hyper-parameter)

c) ~~Prior~~ Prior for Covariance matrices

→ we got $\arg\max_\theta \left(e^{Q(\theta, \theta^i)} \right) \propto \prod_n \prod_k G(y_n, \mu_k, C_k)^{\gamma_{nk}}$

$$\propto \prod_n \prod_k \frac{1}{(C_k)^{\frac{\gamma_{nk}}{2}}} \times e^{-0.5 \left((y_n - \mu_k)^T C_k^{-1} (y_n - \mu_k) \right) \gamma_{nk}}$$

→ We can thus use Inverse Wishart distribution (conjugate prior for Covariance matrices in ~~MVG~~ MVG. ~~Thus~~

Thus $P(C_i) = f_x(C_i; \psi, \nu) = \dfrac{|\psi|^{\nu/2}}{2^{\nu p/2}\,\Gamma_p(\nu/2)} \times |C_i|^{-\frac{(\nu+p+1)}{2}} e^{-\frac{1}{2}\operatorname{tr}(\psi C_i^{-1})}$

→ where ψ, ν are hyper parameters, p is dimension C_i and ψ. Γ_p is multivariate gamma function.

3(ii) E-step:-

~~E. We / need / to maxi~~

→ $Q(\theta, \theta^i) = \sum_n \sum_k \gamma_{nk} \left(-0.5\log|C_k| - 0.5 (y_n - u_k)^T C_k^{-1} (y_n - u_k) + \log w_k + \log P(\theta)\right)$

(from ①)

Now $\gamma_{nk} = P(y_n | z_n = k, \theta^i)\, P(z_n = k | \theta^i)$

$= \dfrac{G(y_n | u_k^i, C_k^i)\, w_k}{\sum_k G(y_n | u_k^i, C_k^i)\, w_k^i}$ (same as before)

M-step:

→ i) Update for weights

→ We know solution without prior as:-

$$w_k = \frac{\gamma_k}{\sum_k \gamma_k}$$

Since ~~$P(w_k)$~~ $P(W) = \mathrm{Dir}(W; \alpha)$

We have $\operatorname{argmax}_\theta (Q(\theta, \theta^i)) \propto \prod_k w_k^{\gamma_k + \alpha_k}$

Thus new update for W_k is:-

$$\boxed{w_k = \frac{\gamma_k + \alpha_k}{\sum(\gamma_k + \alpha_k)}}$$

2) Updates for means

$P(\theta)$ has the form $P(\mu_k) \propto e^{-(\bar{\mu}-\mu_k)^T \bar{C}^{-1}(\bar{\mu}-\mu_k)}$

Thus $Q(\theta,\theta') = \sum_n \sum_k \gamma_{nk} (y_n-\mu_k)^T C_k^{-1} (y_n-\mu_k) + \sum_k (\bar{\mu}-\mu_k)^T \bar{C}^{-1} (\bar{\mu}-\mu_k)$

$$\Rightarrow \frac{\partial Q(\theta,\theta')}{\partial \mu_k} = \sum_n \gamma_{nk} C_k^{-1} (y_n-\mu_k) + \bar{C}^{-1}(\bar{\mu}-\mu_k) = 0$$

$$\Rightarrow \sum_n \gamma_{nk} C_k^{-1} y_n + \bar{C}^{-1}\bar{\mu} = \left(\sum_n \gamma_{nk} C_k^{-1} + \bar{C}^{-1}\right) \mu_k$$

$$\Rightarrow \boxed{\mu_k = \left(\gamma_k C_k^{-1} + \bar{C}^{-1}\right)^{-1} \left(\sum_n \gamma_{nk} C_k^{-1} y_n + \bar{C}^{-1}\bar{\mu}\right)}$$

3) Updates for C_k

→ We can write

$$\underset{\theta}{\operatorname{argmax}} \left(Q(\theta,\theta')\right) = \underset{\theta}{\operatorname{argmax}} \left(e^{Q(\theta,\theta')}\right)$$

$$\propto \prod_k \frac{1}{(C_k)^{\lambda_k/2}} \times \prod_k e^{-0.5(\operatorname{tr}(C_k^{-1}A))} \times \prod_k P(C_k^{-1}) \rightarrow (\text{Prior})$$

(where $A = \sum_n \gamma_{nk}(y_n-\mu_k)^T(y_n-\mu_k)$)

without prior, we have $C_k = \frac{A}{\lambda_k}$

→ If we have $P(C_k) \propto \frac{1}{|C_k|^{+\frac{(v+p+1)}{2}}} e^{-1/2 \operatorname{tr}(\psi C_k^{-1})}$

$$\arg\max\left(Q(\theta,\theta')\right) \propto \prod_k \frac{1}{|C_k|^{\frac{(\lambda_k+(v+p+1))}{2}}} \times e^{-1/2 \operatorname{tr}(C_k^{-1}A + \psi C_k^{-1})}$$

Thus $\boxed{C_k = \frac{A+\psi}{\lambda_k + v+p+1}}$ (Since trace $AB = BA$)

→ Proof : that $\sum_n \gamma_{nk} (y_n - u_k)^T C_k^{-1} (y_n - u_k) = tr(C_k^{-1} A_k)$

where $A_k = \sum_n \gamma_{nk} (y_n - u_k)^T (y_n - u_k)^T$

→ By Cholesky decomposition, we have

$C_k = P_k D_k P_k^T$ for some orthogonal matrix P_k and diagonal matrix D_k with positive entries

now $C_k^{-1} = P_k D_k^{-1} P_k^T$

thus

$$\sum_n \gamma_{nk} (y_n - u_k)^T C_k^{-1} (y_n - u_k)$$

$= \sum_n X_{nk}^T D_k^{-1} X_{nk}$ where $X_{nk} = \sqrt{\gamma_{nk}}\, P_k (y_n - u_k)$

$= \sum_n \| \sqrt{D_k^{-1}} X_{nk} \|^2$ where $\sqrt{D_k^{-1}}$ is matrix with square root of elements of D_k^{-1}

$$= \sum_n tr\left(X_{nk} (\sqrt{D_k^{-1}}) (\sqrt{D_k^{-1}})^T X_{nk}^T\right)$$

$$= \sum_n tr\left(D_k^{-1} X_{nk} X_{nk}^T\right)$$

$$= tr\left(D_k^{-1} \sum_n X_{nk} X_{nk}^T\right)$$

$$= tr\left(D_k^{-1} \sum_n P_k (y_n - u_k)(y_n - u_k)^T P_k^T\right)$$

$$= tr\left(P_k D_k^{-1} P_k^T \sum_n \gamma_{nk} (y_n - u_k)(y_n - u_k)^T\right)$$

$= tr(C_k^{-1} A_k)$ (we have used $tr(AB) = tr(BA)$)