Turku PET Centre Modelling report TPCMOD0029 2004-08-19
Appendix C
Marketta Hiissa

Pseudo-code for the Markov random field clustering algorithm

The idea of the algorithm

The aim of the clustering is to segment the dynamic PET image automatically, which then enables an automatic extraction of the reference tissue input function. The segmentation algorithm presented in the article of Chen et al. [Chen et al. 2001] uses a Markov random field model for the TAC class labels because this enables to incorporate spatial interaction between voxels in the segmentation process. The algorithm used in the segmentation is the Expectation Maximization (EM) algorithm.

The algorithm proceeds in principle in the same way as the algorithm of Ashburner et al., which is introduced in the Appendix A. The main difference is that in the method of Ashburner et al. it is assumed that image voxels are independent while the Markov random field model enables us to take into account the correlation of voxels close to each other. There are also some differences in the ways of computing cluster means and variances. TAC i is denoted a_i and its component in time frame j is denoted a_{ij}. Cluster label for TAC i is denoted z_i and it is assumed that the number of clusters is k. The algorithm proceeds as follows.

1. Define a neighbourhood for each voxel in the original data matrix and then create a matrix having voxels as rows and time frames as columns. The neighbourhood of a voxel is defined to be a 3×3 voxel grid.
2. Set initial belonging probabilities for each TAC and define the initial clusters. The initial belonging probabilities are set to be either zero ore one.
3. Compute means and variances for each cluster.
4. Compute probability densities for each TAC given a cluster, i.e. probability densities $p(a_i|z_i = k)$. These probabilities are based on the Gaussian distributions.
5. Compute probabilities for each class label z_i given the neighbourhood N_i of voxel i. The Markov random field model is used in calculating these probabilities.
6. Calculate new belonging probabilities and define new clusters by assigning each TAC to a cluster such that the belonging probabilities are maximized.
7. Compute the value of log likelihood and if it has increased significantly since the previous iteration, continue iterating from step two.

Pseudo-code

/* Define the neighbourhood of each element in the original data matrix. It is assumed that there are X rows and Y columns in the original data matrix. The neighbourhood is selected to be a 3×3 voxel grid.
The set $N_{(x,y,plane)}$ contains the neighbours of voxel in place *(x,y,plane)* in the data matrix. */

```
for each plane
    for x  = 1 to X
        for y   = 1 to Y
            N(x,y,plane)  = { (x-1,y-1), (x-1,y), (x-1,y+1),  (x,y-1),(x,y+1),
                              (x+1,y-1),(x+1,y),(x+1,y+1) }
```

```
        endfor
    endfor
endfor
```

/* It is assumed that matrix A has voxels as rows and time frames as columns and that the variable N_i includes the neighbourhood of a voxel that is on the i:th row in the matrix A.
It is also assumed that there are m rows (voxels) and n columns (time frames) in the matrix A and an element of A in place (i,j) is denoted a_{ij}. From now on, the input of the algorithm is the matrix A. */

/* Define the initial belonging probabilities. This is done in the same way as in the Ashburner's method */

```
    for i = 1 to m
        for t = 1 to k
            p_it = 0
        endfor
    endfor

    b = 1
    t = 1
    while t ≤ k
            e = t * m / k    // the result must be an integer
            for i = b to e
                p_it = 1
            endfor
            b = e + 1
            t = t + 1
    endwhile
```

/* Define the initial clusters */

```
    for i = 1 to m
          t = 1
          while p_it ≠ 1
                  t = t + 1
          endwhile
          cluster_i = t
    endfor

old_llh = 0
changed = true

while changed = = true
```

/* Compute the mean TACs for each cluster
The mean is a weighted mean of values a_{ij} and weighting factors are probabilities p_{it}
Note that in the algorithm of Ashburner et al. means are calculated in a little bit different way. */

```
                for t = 1 to k
```

```
        for j = 1 to n
            num = 0    //  the numerator of the expression for the mean
            denom = 0  //  the denominator of the expression
                i = 1
                while i ≤ m
                    num = num + a_ij * p_it
                    denom = denom + p_it
                    i = i + 1
                endwhile
                m_tj = num / denom
        endfor
    endfor
```

/* Compute variances for each time frame in each cluster Note that in the Ashburners's method it was assumed that the variance matrix is common to all clusters. */

```
for t = 1 to k
    for j = 1 to n
        num = 0    //  the numerator for the expression of the variance
        denom = 0  //  the denominator for the expression of the variance
        i = 1
        while i ≤ m
            num = num + p_it * (x_ij – m_tj)^2
            denom = denom + p_it
            i = i + 1
        endwhile
        v_tj = (1 / n) * num / denom
    endfor
endfor
```

/* Compute the conditional probability densities for each voxel given a cluster.
The conditional distribution $p(a_i|z_i = t)$ is modelled using a Gaussian distribution. */

```
for i = 1 to m
    for t = 1 to k
        denom = 1   // defines the denominator of the expression
        expon = 0
        j = 1
        while j ≤ n
            denom = sqrt ( denom * 2π * v_tj )
            expon = expon + (x_ij –m_tj )^2 / v_tj
            j = j + 1
        endwhile
            c_it = (1 / (denom) )* exp (-0.5 * expon)
    endfor
endfor
```

/* If we assume that image voxels are independent, we can compute probabilities $P(z_i = t)$ = g_t and new belonging probabilities p_{it} in the same way as in the

Ashburner's method. If we do not assume the independence of image voxels, we can use an approximate formula based on a Markov random field model to define the probabilities. */

/* Compute first d_{it}, the number of neighbours of voxel i that are in cluster t. */

```
for i  = 1 to m
    for t  = 1 to k
        d_it  = 0
    endfor
    for p in N_i   // for all voxels in the neighbourhood of voxel i
        h  = cluster_p
        d_ih  = d_ih + 1
    endfor
endfor
```

/* Compute the approximations for the probabilities $P(z_i = k)$.
Constant b is a parameter that controls the influence of neighbouring voxels. */

```
for i = 1 to m
    for t = 1 to k
        denom  = 0
        r  = 1
        while  r ≤ k
                denom  = denom + Exp[b * d_ir]
                r : = r + 1
        endwhile
        g_it  = Exp (b * d_it ) / denom
    endfor
endfor
```

/* Compute new belonging probabilities p_{it} */

```
for i  = 1 to m
    denom  = 0
    for t  = 1 to k
        r  = 1
        while  r ≤ k
                denom  = denom + c_ir * g_ir
                r  = r + 1
        endwhile
        p_it = c_it * g_it / denom
    endfor
endfor
```

/* Compute the value of the log likelihood */

```
llh = 0
i  = 1
while  i ≤ m
        llh_a  = 0
```

```
                t = 1
                  while t ≤ k
                    llh_a = llh_a + c_it * g_it
                    t = t + 1
                  endwhile
                  llh = llh + Log (llh_a)
                  i = i + 1
        endwhile
```

/* Create the clusters by assigning each voxel to a cluster such that the belonging probability p_{it} is maximised for each voxel. */

```
          for i = 1 to m
            cluster_i = 0
            max = 0
            t = 1
            while t ≤ k
                    if p_it > max
                        then cluster_i = t
                              max = p_it
                    endif
                    t = t + 1
          endwhile
        endfor
```

/* Make the comparisons to define whether to continue iterating or not. */

```
        if  llh − old_llh > 0.001
            then changed = true
                 old_llh = llh
            else changed = false
        endif

endwhile
```

Reference

J. L. Chen et al.: Markov Random Field Models for Segmentation of PET Images. In Insana, M. F. and Leahy, R. M., editors, *Proceedings of Information Processing in Medical Imaging* 2082, pages 468–474 (2001)