Statistical image analysis, 7.5hp **TMS016/MSA300**

Solutions to written exam **2018-05-30, 08:30 - 12:30**

1. (a) True. This does not hold for random fields in general, but for Gaussian random fields.

   (b) True. The graph defines the sparsity structure of the precision matrix.

   (c) False. $\{(i,j) : S_{ij} \subseteq A\}$ is called the erosion of A.

   (d) False. $E(N(A))$ is the first order moment measure.

   (e) True. Since the black curve is above the red curve, this indicates clustering (although it might not be statistically significant).

   (f) False. This is a requirement for convolutional neural networks, but not for feedforward neural networks in general.

2. (a) Let $\ell(Y; \boldsymbol{\theta}, \boldsymbol{\beta})$ be the log-likelihood for the model. The profile likelihood method uses the fact that we can compute the maximum likelihood estimate of $\boldsymbol{\beta}$ analytically if we fix $\boldsymbol{\theta}$. Let

$$\hat{\beta}(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\beta}} \ell(Y; \boldsymbol{\theta}, \boldsymbol{\beta}) = (\mathbf{B}^T \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} \mathbf{B})^{-1} \mathbf{B} \boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1} \mathbf{Y},$$

   where $\mathbf{Y}$ is a vector with the observations, with covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, and $\mathbf{B}$ is a matrix with the covariates evaluated at the measurement locations. We then define the profile log-likelihood as $\ell_p(Y; \boldsymbol{\theta}) = \ell(Y; \boldsymbol{\theta}, \hat{\boldsymbol{\beta}}(\boldsymbol{\theta}))$ which we maximize numerically with respect to $\boldsymbol{\theta}$ to find the maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{ML}$. The maximum likelihood estimate of $\boldsymbol{\beta}$ is finally given by $\hat{\beta}_{ML} = \hat{\beta}(\hat{\boldsymbol{\theta}}_{ML})$.

   (b) The advantage is that REML gives unbiased estimates of the covariance parameters $\boldsymbol{\theta}$. One disadvantage is that is typically has a higher variance.

   (c) The cliques are $\{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{1,4\}, \{2,3\}, \{3,4\}$.

   (c) The pseudo-likelihood is the product of all full contidionals,

$$L_p(\mathbf{Z}) = \prod_{i=1}^{4} \pi(Z_i | Z_{-i}).$$

   The reason for why we used this function is that this is easy to compute (and that pseudo-maximum likelihood share some of the good properties of regular maximum likelihood), whereas the regular likelihood has a normalizing constant which depends on the parameters that in general is intractable for large images.

3. (a) Since this is an intensity-based segmentation problem, we could use for example the K-means algorithm or a Gaussian mixture model with two classes. However, because of the large amount of noise in the data, it would likely be beneficial to take the spatial structure into account by using a Markov random field mixture model to perform the segmentation.

   (b) For the Gaussian mixture model, we have the prior class probabilities π_1 and π_2 for the two classes, which usually are estimated from the data. To take the pore volume faction into account we can simply fix these to $\pi_1 = 0.3$ and $\pi_2 = 0.7$ instead of estimating them from data. Similarly for the Markov random field mixture model, we can fix the α_k parameters to take the pore volume faction into account.

4. (a) We could for example use image moments, or look at the amount of horizontal and vertical edges in the images by using edge detecting filters. For the K nearest neighbor approach, we compute the features for the training data as well as the new image. Let $\mathbf{Y}_1, \ldots, \mathbf{Y}_{1000}$

denote the vectors of features for the training data, and let $\mathbf{Y}_0$ denote the vector of features for the new image. We then compute the distances

$$d_i = d(\mathbf{Y}_i, \mathbf{Y}_0),$$

where $d(X, Y)$ denotes a chosen distance function, such as the euclidian distance in $\mathbb{R}^d$. We extract the training images with the K smallest distances d_i and classify the new image according to the class that is in majority among the K training images.

(b) To choose K, we can use cross-validation. We then divide the data into p batches of images. For each batch we predict the class of the images using the remaining $p-1$ batches as training data using different values of K. We then compute the total missclassification rate based on all the batches and choose K as the value that has the smallest total crossvalidation error.