# 1. Errata from NETLAB book

## 1.1 Errata in third printing (2004).

• Page 82, equation 3.10: should read

$$p(\boldsymbol{x}|j) = \frac{1}{(2\pi)^{d/2}\prod_{i=1}^{d}\sigma_{j,i}} \exp\left\{-\sum_{i=1}^{d}\frac{(x_i - mu_{j,i})^2}{2\sigma_{j,i}^2}\right\}.$$

• Page 242, lines 3 and 4 of listing of `gmmsamp` should read

```
3  mix.U(:, :, j)* ...
4  (diag(mix.lambda(j, :))-(mix.covars(j)*eye(mix.ppca_dim)))* ...
5  (mix.U(:, :, j)');
```

• Page 385, line 18 of `gperr` should read:

```
18      e2 = (sum(eprior, 1))*(1./net.pr_var);
```

## 1.2 Errata from second printing (2003), fixed in third printing (2004).

• Page 42, equation 2.14: should read

$$\frac{(x-b)^2}{2}f''(b) \leq \epsilon|f(b)|$$

$$\iff |x-b| \leq \sqrt{\frac{2\epsilon|f(b)|}{|f''(b)|}} = b\sqrt{\epsilon}\sqrt{\frac{2|f(b)|}{b^2|f''(b)|}}.$$

• Page 43, equation 2.17: should read

$$x = b - \frac{1}{2}\frac{(b-a)^2\left[f(b)-f(c)\right] - (b-c)^2\left[f(b)-f(a)\right]}{(b-a)\left[f(b)-f(c)\right] - (b-c)\left[f(b)-f(a)\right]}.$$

- Page 74, equation 2.74: should read

$$\nabla P_Q(\boldsymbol{x}) = \nabla f + \rho(\hat{\boldsymbol{g}}^T \nabla \hat{\boldsymbol{g}}).$$

  Line 10 in the definition of the function `gradcon` should read

```
10  gc = fgrad + rho*(g*ggrad');
```

  (This has also been corrected in the software download).
- Page 88, equation 3.15: should read

$$((\boldsymbol{x} - \boldsymbol{\mu}_j)\boldsymbol{C}^{-1}) .* ((\boldsymbol{x} - \boldsymbol{\mu}_j)\boldsymbol{C}^{-1})$$

- Page 133, para. 2, line -4: should read 'to initialise the parameter $\boldsymbol{w}$ reasonably close to the maximum.'
- Page 276, listing of `canvar`: lines 14–40 should read

```
14  S_W = zeros(nin, nin);
15  N_c = zeros(1, nclasses);
16  for c = 1:nclasses
17    N_c(c) = sum((t(:, c) == 1), 1);
18    if (N_c(c) > 1)
19      S_W = S_W + ((N_c(c)-1)/ndata) * cov(x(find(t(:, c)==1), :));
20    end
21  end
22  % Between class covariance matrix
23  Sigma = cov(x, 1);  % Ensure normalisation by ndata
24  S_B = Sigma - S_W;
25  % Now compute canonical variates using generalised eigenvalues
26  [temp_cvecs, temp_cvals] = eig(S_B, S_W);
27  % Sort values and vectors into descending order
28  Tc = diag(temp_cvals);
29  [cvals perm] = sort(-Tc);
30  cvals = -cvals(1:N);
31  if cvals == Tc(1:N)
32    % Originals were in order
33    cvecs = temp_cvecs(:, 1:N);
34    return
35  else
36    % Need to reorder the eigenvectors
37    for i=1:N
38       cvecs(:,i) = temp_cvecs(:,perm(i));
39    end
40  end
```

  The significant changes are in lines 16–23, where we ensure that the covariance matrices are scaled by $n$.
- Page 328, equation 9.5: should read

$$E(\boldsymbol{y}|\boldsymbol{x}^*, \mathcal{D}) = \int \boldsymbol{y}\, p(\boldsymbol{y}|\boldsymbol{x}^*, \mathcal{D})\, \mathrm{d}\boldsymbol{y},$$

- Page 340, equation 9.21: should read

$$\frac{\partial E_W}{\partial w_i} = \alpha w_i,$$

## 1.3 Errata from first edition, first printing (2002), fixed in second printing

- Page IX, para. 3, line 3: should read 'The attraction of this model, which is discussed in Chapter 10, ...'.
- Page IX, para. 4, line 4: should read 'This free software is covered by a very open license based on that used for BSD.'
- Page X1, para. 2, line 5: Markus Svensén's name is mis-spelled. Sorry, Markus.
- Page 53, equation (2.30): should read

$$\lambda_j = \begin{cases} 0 & \boldsymbol{g}_j^T H \boldsymbol{d}_j = 0 \\ \frac{\boldsymbol{g}_j^T \boldsymbol{g}_j}{\boldsymbol{g}_j^T H \boldsymbol{d}_j} & \text{otherwise.} \end{cases}$$

- Page 74, equation (2.4) should read

$$\nabla P_Q(\boldsymbol{x}) = \nabla f + \rho(\hat{\boldsymbol{g}}^T \nabla \hat{\boldsymbol{g}}).$$

  and, correspondingly, in the second program, line 10: should read

```
10  gc = fgrad + rho*(g*ggrad');
```

- Page 74, third program, line 1: should read

```
 1  function [y, x] = optcon(x, fn, gradf)
```

- Page 94, line 40 of the program should read

```
40          mix.covars(:,:,j) = mix.covars(:,:,j) + ...
```

- Page 118, equation (4.4) should read

$$y_j = \frac{\exp(a_j)}{\sum_{j'} \exp(a_{j'})}.$$

- Page 147, equation (4.60) should read

$$\boldsymbol{w}_1 = \sum_{i=1}^{r} \left( \frac{\sigma_i t_i}{\sigma_i^2 + \alpha^* \gamma_i^2} \right) \boldsymbol{v}_i.$$

- Page 151, equation (5.7) should read

$$y_k = \frac{\exp(a_k^{(2)})}{\sum_{k'} \exp(a_{k'}^{(2)})}. \tag{1.1}$$

- Page 241, first code fragment, lines 16–22 should be replaced by

```
16      for j = 1:mix.ncentres
17        c = x(find(post(:,j)),:);
18        diffs = c - (ones(size(c, 1), 1) * mix.centres(j, :));
19        [tempcovars, tempU, templambda] = ...
20          ppca((diffs'*diffs)/size(c, 1), mix.ppca_dim);
21        if length(templambda) ~= mix.ppca_dim
22          error('Unable to extract enough components');
23        else
24          mix.covars(j) = tempcovars;
25          mix.U(:, :, j) = tempU;
26          mix.lambda(j, :) = templambda;
27        end
28      end
```

- Page 242, equation (7.30) should read

$$\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 \boldsymbol{I} = \boldsymbol{U}_q (\boldsymbol{\Lambda}_q - \sigma^2 \boldsymbol{I}) \boldsymbol{U}_q^T + \sigma^2 \boldsymbol{I}.$$

  and the code fragment just above it should read

```
1  case 'ppca'
2    covar = mix.covars(j) * eye(mix.nin) + ...
3      mix.U(:, :, j)*(diag(mix.lambda(j, :)) - ...
4      diag(mix.covars(j))) * (mix.U(:, :, j)');
```

- Page 244, equation (7.32) should read

$$p(\boldsymbol{x}|\boldsymbol{W}, \sigma) = \int p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{W}, \sigma) p(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}.$$

- Page 277, caption to Figure 7.15 should read 'Demonstration of canonical variates. The two classes are indicated by circles and crosses respectively. The canonical variate (*dashed line*) and first principal component (*solid line*) are clearly contrasted.'

- Page 329, line 1 should read 'If we use a weight prior $p(\boldsymbol{w}_{MP}^{(i)}|\mathcal{M}_i)$ that is uniform over some large region ...'.
- Page 353, code fragment for `fevbayes`, lines 8–13 should be replaced by

```
 8  ntest = size(x_test, 1);
 9  var = zeros(ntest, 1);
10  for idx = 1:1:net.nout,
11    for n = 1:1:ntest,
12      grad = squeeze(g(n,:,idx));
13      var(n,idx) = grad*invhess*grad';
14    end
15  end
```

- Page 394, para. 2, line 2: should read 'an iterative algorithm, ...'.