Solutions to exercises of Day 2

Exercise 2.1

(c) `w1=gaussm(dataset(a1)); w2=gaussm(dataset(a2)); scatterd(a,'legend');`. Visualize using `plotm(w1,2);plotm(w2,2);`.
(d) Something like: `[getlab(b) +phat1 +phat2]`.
(e) 27/38 and 11/38, respectively.
(f) `P = [(27/38)*phat1 (11/38)*phat2]; [dummy, lab2] = max(P,[],2);` .

Exercise 2.2

Simply replace `gaussm` by `parzenm`.

Exercise 2.4

(a) The classification error is slightly higher on the training set than on the test set.
(b) Also the cross-validation error is higher than the classification error on the test set.

Exercise 2.5

One could select differentially expressed genes with `anova1`.

Exercise 2.6

(a) $p(\omega_2|\boldsymbol{x}) = 1 - p(\omega_1|\boldsymbol{x})$
(b)

$$\begin{aligned}
\log\left(\frac{p(\omega_1|\boldsymbol{x})}{p(\omega_2|\boldsymbol{x})}\right) &= \beta_0 + \beta^T\boldsymbol{x} && (1)\\
\frac{p(\omega_1|\boldsymbol{x})}{p(\omega_2|\boldsymbol{x})} &= \exp\left(\beta_0 + \beta^T\boldsymbol{x}\right) && (2)\\
\frac{1 - p(\omega_2|\boldsymbol{x})}{p(\omega_2|\boldsymbol{x})} &= \exp\left(\beta_0 + \beta^T\boldsymbol{x}\right) && (3)\\
\frac{1}{p(\omega_2|\boldsymbol{x})} - 1 &= \exp\left(\beta_0 + \beta^T\boldsymbol{x}\right) && (4)\\
\frac{1}{p(\omega_2|\boldsymbol{x})} &= \exp\left(\beta_0 + \beta^T\boldsymbol{x}\right) + 1 && (5)\\
p(\omega_2|\boldsymbol{x}) &= \frac{1}{1 + \exp\left(\beta_0 + \beta^T\boldsymbol{x}\right)} && (6)
\end{aligned}$$

Again $p(\omega_2|\boldsymbol{x})$ should not be zero.

Exercise 2.7

(a) Two overlapping circular-shaped distributions with a distance of 1 between their means.
(b) Each of the classes has a Gaussian class-conditional probability distribution with equal covariance matrices. The decision boundary will therefore be linear. The logistic classifier also gives a linear decision boundary, so it is a good candidate model. However, due to the overlap between the classes the classification error can be substantial.
(c) The second dataset is practically linearly separable. The result is that the logistic classifier will give a very steep transition from $p(\omega_1|\boldsymbol{x}) = 0$ to $p(\omega_1|\boldsymbol{x}) = 1$. In the case the data is

overlapping, as in the first dataset, the slope is much less steep.

Exercise 2.8

(a) The decision boundary of the Gaussian classifier is quadratic which does not fit the banana data. So, many classification errors will be made. The Parzen or the kNN classifier should work much better. The difference between the latter two classifiers will not be large.
(b) First fit the models: `w1 = qdc(a);w2 = parzenc(a);w3 = knnc(a);`. Then plot the decision boundaries: `scatterd(a);w = {w1,w2,w3}; plotc(w1); plotc(w2,'red'); plotc(w3,'blue');`and estimate the classification error: `testc(a*w);`.

Exercise 2.9

Increasing these parameters gives smoother decision boundaries. This leads to increased classification error (on the training data, at least)

Exercise 2.10

(a) The second parameter indicates the dimensionality of the dataset, the third indicates the distance between the classes in the first dimension.
(b) The `knnc` should have zero error on the training set, but a very large error on the test set. This indicates that the classifier is overtrained.
(c) In the help is stated that all features, except for feature 1, are not informative. The class overlap is therefore constant for varying dimensionality and the Bayes error does not change.
(d) The performance on the test set in general deteriorates: the `knnc` is actually hampered by the extra, superfluous dimensions.
(e) The test error goes up for increasing dimensionality. A dimensionality of 2 or 3 is optimal.
Example code:

```
err = [];
dims = [2 3 5 10 25 100]; % array of dims to try
nrepeats = 25; % number of repeats
for i=dims
   total = 0;
   for j=1:nrepeats
     a = gendats([10 10],i,2); % generate training data
     b = gendats([1000 1000],i,2); % generate test data
     w1 = knnc(a,1); % train a one nearest-neighbor classifier
     total = total + +testc(b*w1); % calculate error on test data
   end
   err= [err total/nrepeats];
end
plot(dims,err) % plot test error as function of dimensionality
```

Exercise 2.11

(a) We have to recall the two equalities: $A\boldsymbol{x} = (\boldsymbol{x}^T A^T)^T$ and $(\Sigma_i^{-1})^T = \Sigma_i^{-1}$ (because Σ is a symmetric matrix). Using this we can already simplify $(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = \boldsymbol{x}^T\Sigma_i^{-1}\boldsymbol{x} - \boldsymbol{x}^T\Sigma_i^{-1}\boldsymbol{\mu} - \boldsymbol{\mu}^T\Sigma_i^{-1}\boldsymbol{x} + \boldsymbol{\mu}^T\Sigma_i^{-1}\boldsymbol{\mu}$. Rewrite: $\boldsymbol{x}^T\Sigma_i^{-1}\boldsymbol{\mu} = (\boldsymbol{x}^T\Sigma_i^{-1})\boldsymbol{\mu} = (\Sigma_i^{-1^T}\boldsymbol{x})^T\boldsymbol{\mu} = \boldsymbol{\mu}^T\Sigma_i^{-1}\boldsymbol{x}$ and it becomes $= \boldsymbol{x}^T\Sigma_i^{-1}\boldsymbol{x} - 2\boldsymbol{\mu}^T\Sigma_i^{-1}\boldsymbol{x} + \boldsymbol{\mu}^T\Sigma_i^{-1}\boldsymbol{\mu}$.
Now if we look at $g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$: $g_1(\boldsymbol{x}) - g_2(\boldsymbol{x}) = \log(p(\omega_1)) - \log(p(\omega_2)) - \frac{1}{2}\log\det(\Sigma_1) +$

$\frac{1}{2}\log\det(\Sigma_2) - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\Sigma_1^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T\Sigma_2^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) = \log(p(\omega_1)) - \log(p(\omega_2)) - \frac{1}{2}\log\det(\Sigma_1) + \frac{1}{2}\log\det(\Sigma_2) - \frac{1}{2}\boldsymbol{x}^T\Sigma_1^{-1}\boldsymbol{x} + \boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{x}^T\Sigma_2^{-1}\boldsymbol{x} - \boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu}_2$. Collecting the terms from the equation, we obtain for the quadratic term: $W = \frac{1}{2}(\Sigma_2^{-1} - \Sigma_1^{-1})$; for the linear term: $\boldsymbol{w}^T = \boldsymbol{\mu}_1^T\Sigma_1^{-1} - \boldsymbol{\mu}_2^T\Sigma_2^{-1}$; and for the bias: $w_0 = \log\frac{p(\omega_1)}{p(\omega_2)} + \frac{1}{2}\log\frac{\det\Sigma_2}{\det\Sigma_1} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma_1^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma_2^{-1}\boldsymbol{\mu}_2$.

Exercise 2.12

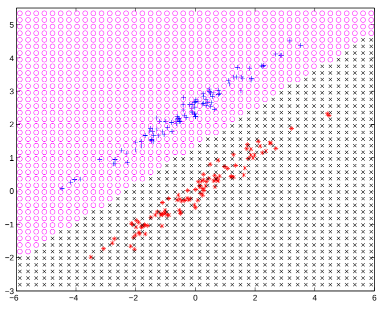
(a) We are using the results from the previous exercise, and substitute $\Sigma^{-1} = \Sigma_1^{-1} = \Sigma_2^{-1}$. The quadratic term disappears $W = 0$, and we only have $\boldsymbol{w}^T = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\Sigma^{-1}$ and $w_0 = \log\frac{p(\omega_1)}{p(\omega_2)} - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2$.

Exercise 2.13

(a) Define the function $h_i(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{\mu}_i)^2 = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{\mu}_i^T\boldsymbol{x} + \boldsymbol{\mu}_i^T\boldsymbol{\mu}_i$, similar to the Gaussian-based classifiers. The discriminant function becomes $h_1(\boldsymbol{x}) - h_2(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{\mu}_1^T\boldsymbol{x} + \boldsymbol{\mu}_1^T\boldsymbol{\mu}_1 - \boldsymbol{x}^T\boldsymbol{x} + 2\boldsymbol{\mu}_2^T\boldsymbol{x} - \boldsymbol{\mu}_2^T\boldsymbol{\mu}_2 = 2(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T\boldsymbol{x} + \boldsymbol{\mu}_1^T\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\mu}_2$.
(b) We simplify the covariance matrix even further by $\Sigma = \sigma^2\boldsymbol{I}$, and we get that $\Sigma^{-1} = \sigma^{-2}\boldsymbol{I}$. This results in $\boldsymbol{w}^T = \sigma^{-2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ and $w_0 = \log\frac{p(\omega_1)}{p(\omega_2)} - \frac{1}{2\sigma^2}(\boldsymbol{\mu}_1^T\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^T\boldsymbol{\mu}_2)$. This is identical to the nearest mean classifier, up to a scaling factor of $-\frac{1}{2\sigma^2}$.

Exercise 2.14

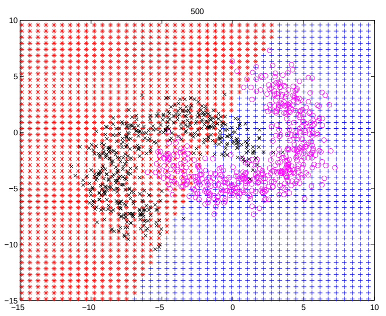
(a) Linear. The `ldc` will therefore fit best with a straight line separating the clusters:



With sufficient data a quadratic classifier also gives good results, but with a smaller training set the performance might become worse.

Exercise 2.15

(a) The class-conditional densities are clearly not Gaussian. See the figure below. Since this boundary is not the optimal boundary, the resulting error rate is relatively high, approximately 15%.



(b) The optimal decision boundary is non-linear and non-quadratic and can not be approximated by assuming two Gaussian classes.

Exercise 2.17

As explained in the lecture, for a two-class problem these classifiers are identical.

Exercise 2.18

(a) The inverse sigmoidal transforms the posterior probabilities `a*w` into the original values of the discriminant function. For *any* linear discriminant function this is proportional to the distance from x_i to the decision boundary:

$$d(x_i, \text{decision boundary}) = \frac{\mathbf{w}^T\mathbf{x}_i + w_0}{||\mathbf{w}||}.$$

(b) Code for computing the Fisher criterion given the distances to the decision boundary `d`:

```
% calculate the mapped mean for class 1
mu1 = mean(seldat(d,1));
% calculate the mapped mean for class 2
mu2 = mean(seldat(d,2));
% calculate the Fisher criterion J as described in the lecture
crit = (abs(mu1 - mu2)^2)/(sum((seldat(d,1)-mu1).^2)+sum((seldat(d,2)-mu2).^2));
```

(c) Using the code given under (b), you can also calculate the Fisher criterion for the `nmc`. Given the same data, the Fisher criterion for the Fisher classifier is systematically higher than for the nearest mean classifier.

Exercise 2.19

(c) The decision boundary clearly shows the axis-parallel splits of the feature space.
(d) For the default purity-based splitting criterion any degree of early pruning, that is `prune`> 0, is equally good and gives a test error of 0. The optimal model is a decision tree consisting of only two splits and sacrifices some performance on the training data for a simpler model with better performance on the test data. Other splitting criteria show similar results.
(e) The influence of the splitting criteria on the test error is substantial. The default purity-based splitting criterion gives a test error of approximately 30%, while the Fisher criterion gives a test error of about 3%.

Exercise 2.20

(c) The quadratic, linear and Fisher discriminant are not affected, the nearest mean and the Parzen density classifier are. This is because in the first three classifiers a covariance matrix

$\hat{\Sigma}$ is estimated. Therefore, the scale of the feature values is automatically estimated in these classifiers. Decision trees are also scale-insensitive by design, since the splitting criteria used are scale-insensitive. In principle, the 1-nearest neighbor classifier is not scale-insensitive but for this particular dataset scaling does not seem to influence performance.

(d) In many cases it is an advantage, because it means that you don't need to optimize the scaling of the features. The disadvantage is that the training set should be large enough to make the estimation of the scale reliable.