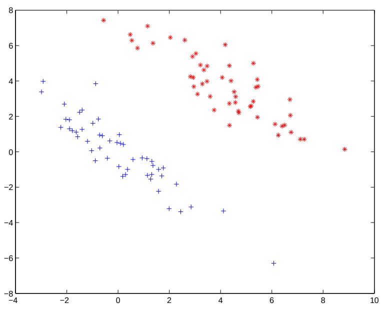
Solutions to exercises of Day 3

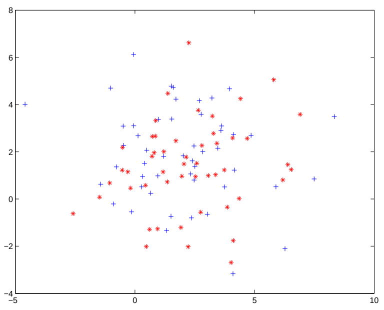
Exercise 3.1

(a) Yes, features 5 and 1.
(b) Different algorithms give different subsets. Feature 3 is always selected, though.

Exercise 3.2

(a) The scatterplot should look like the figure (a) below, not like (b):


(a)


(b)

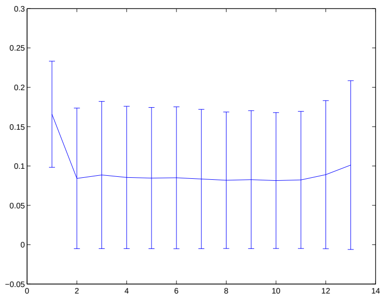
(b) Yes.

Exercise 3.3

(a) Either 2, or a range between 2 and 3-5. In either case, for more than (say) 6 features the error should increase. In figure (a) below, average and standard deviation over 5 experiments are shown.

Exercise 3.4

(a) Anything, as long as it is not feature 12, since it's quite politically incorrect. Features 2, 3, 6, 7, 10 and 13 look good.
(b) Somewhere around 3, most probably a range between 3 and 9 or so. It depends on the particular split, see the figure below.
(c) Best: 13, 10, as predicted. Worst: 12. The results can change quite a bit when repeated.
(d) They stay globally the same, taking into account that the results change quite a bit during repeated running.



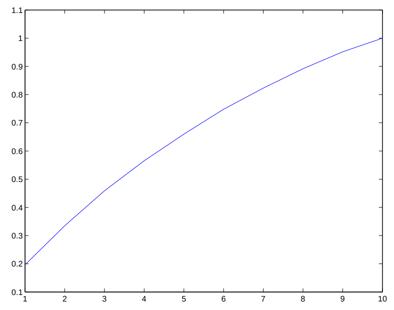
(e) Suddenly feature 12 is ranked higher. Also, the optimal number of features is much higher, sometimes even 13. This is caused by using the non-optimal `'maha-s'` criterion in conjunction with the 1-nearest neighbour classifier.

Exercise 3.5

(a) Over 2000, depending on the random splits of the cross-validation.
(b) If you pick $\Delta = 2.5$, you will end up with around 25 genes. The test set error may actually decrease.
(c) $\Delta = 3$ may give roughly 7 genes, with a still reasonable test error.
(d) In general, for high-dimensional data with small sample size such as microarray data, the simplest classifiers perform best.

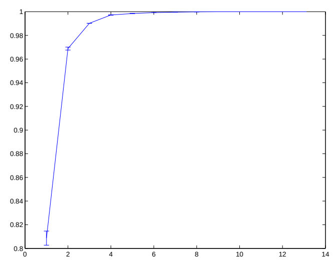
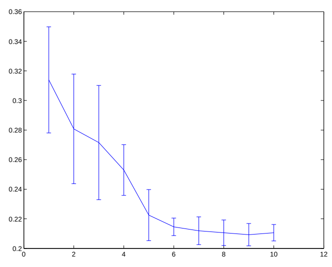
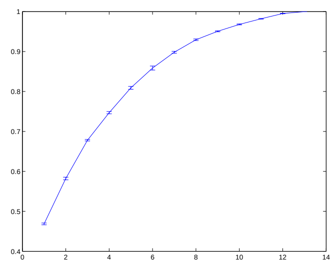
Exercise 3.6

(a) This plot tells us that PCA is not very informative in this case, as the explained variance is nearly the same for all principal components:



Exercise 3.7

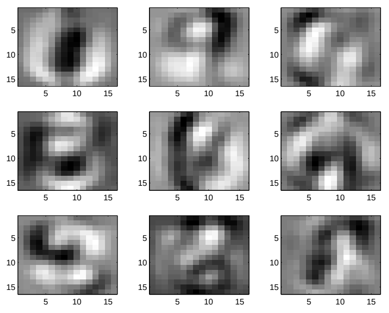
(b) They are higher, because they are not optimised with respect to classification (figure (b) below).
(c) The answer can vary quite a bit, but it will be more than 2. In figure (b) below, roughly 6 dimensions suffice.
(d) The intrinsic dimensionality seems to lie between 2 and 4 (figure (a) below, note the scale of the retained variance-axis).
(e) The first basis vector is (nearly) feature 10. `var(a)`shows that the variance of this feature is huge compared to that of the other features, so PCA will find this dimension as the first component.
(f) No, now the intrinsic dimensionality should be around 8 (figure (c)):

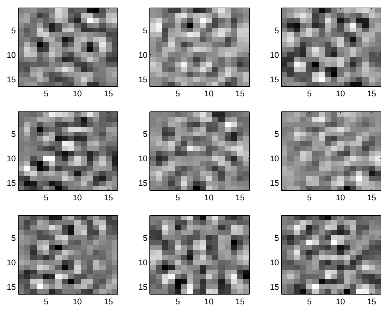

(a)

(b)

(c)

Exercise 3.8

(a) Around 50.
(b) First global blobs (figure (a) shows basis vectors 1-9), then later (figure (b), 51-59) higher

frequencies:


(a)

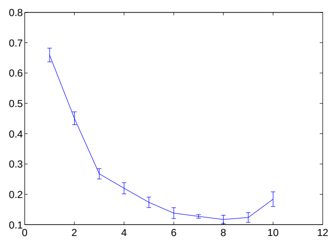

(b)

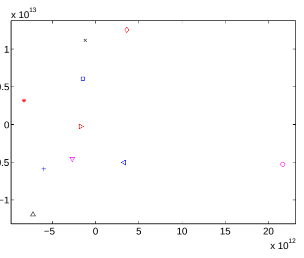
Exercise 3.9

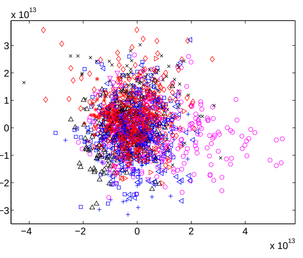
(a) Because the data has 3 classes, `fisherm` can map to at most 3-1 = 2 dimensions.

Exercise 3.10

(a) There are 10 classes, so at most 10-1 = 9 dimensions.
(b) Between 6 and 8, see figure (a) below.
(c) Performance becomes much worse, from less than 20% error to 50%.
(d) All samples in a single class are mapped onto the exact same point (the within-scatter is zero), so there are only 10 points, as in figure (b) below. This is possible since there are only a small number of points (200) in 256 dimensions.
(e) The test data is highly overlapping in the mapped space (figure (c) below). The solution found on the training data is overtrained: it fits the training data perfectly, but does not generalise.

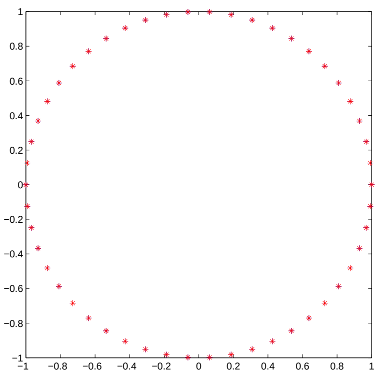

(a)

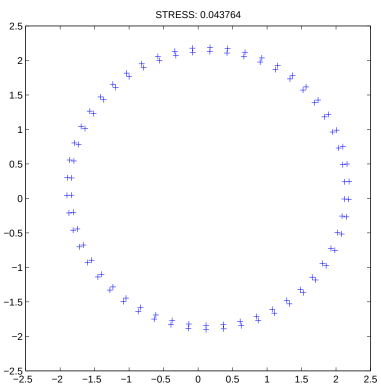

(b)


(c)

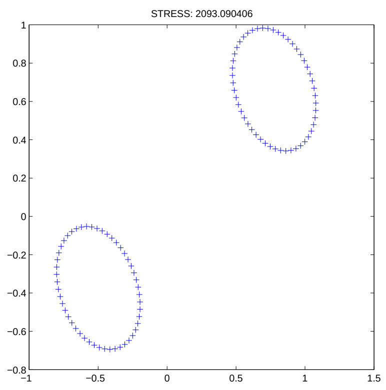
Exercise 3.11

(a) You will see one circle: the two circles have been plotted exactly on top of each other (figure (a) below).
(b) Two circles are visible. Initially, they are twisted a bit; they end up being of different size (figure (b)).
(c) For `opt.q = -2`, the circles are more distorted and further apart (figure (c)); for `opt.q = 2`, the circles are perfectly round but on top of each other (figure (d)). This follows from the definition of the stress measure. The resulting stress values are very different, because they are incomparable.

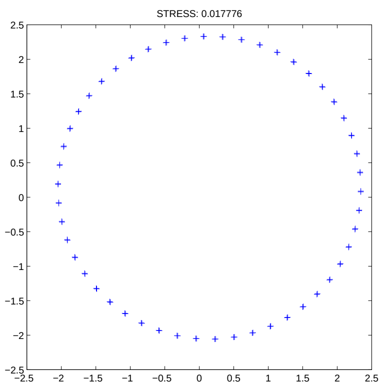

(a)



(b)

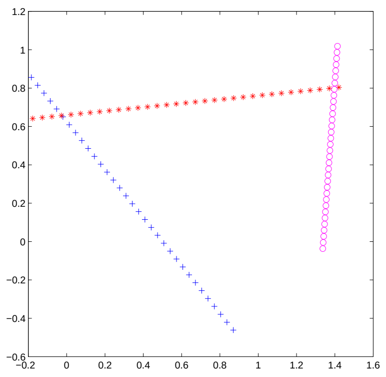


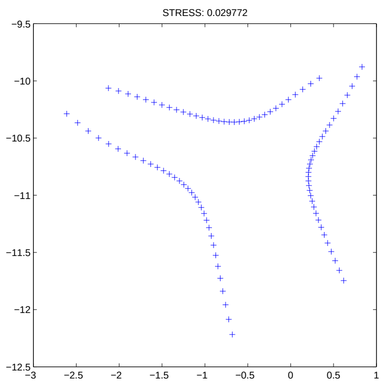
(c)



(d)

(d) The algorithm starts very wild, but should end up in the same configuration. It may hit a local optimum, and will take (much) longer to converge.
(e) PCA (figure (a)) lets the lines overlap. Obviously, MDS (figure (b)) distorts the fact that they are lines:

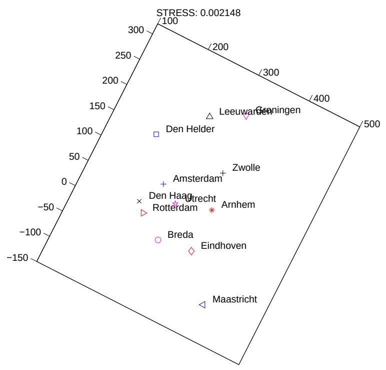

(a)



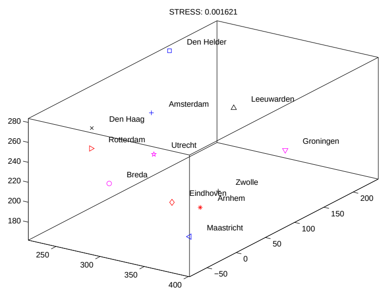
(b)

Exercise 3.12

(a) `opt.q = 0` should give reasonable results (figure (a) below).
(b) The cities do not lie in a plane (figure (b)), which the Netherlands should approximately be. This means there are residual stresses due to the distance measure used (over roads) which cannot be removed in 2D.



(a)



(b)

(c) There will be more emphasis on the applicable distances, sometimes resulting in suboptimal solutions. The algorithm may take a long time to converge.

Exercise 3.13

For $d = 1$, the stress is very large; for $d = 2$ and $d = 3$, it is getting smaller; for $d > 3$, it is the same as for $d = 3$. In other words, the distance matrix can be embedded reasonably well in 2D and perfectly in 3D and higher, which is to be expected.

Exercise 3.14

MDS should show better clusters than PCA.