Solutions to exercises: week 2

Exercise 2.1
(a) A solution can be found by considering the gradient and equating that to 0. Additionally, one needs to realize (though this may not be that trivial) that the function we are minimizing is convex. Maybe we would need a more formal argument for the latter as well...
(b) We can say that we need at least d observations to make sure that the inverse exists. If, however, the subspace spanned by all data contains the origin, we need at least $d+1$ observations. Is this enough for $X^T X$ to be invertible? No, in general we can state that we need the data together with the origin to (linearly) span the whole feature space. Somewhat formally : $(X^T X)^{-1}$ exists if and only if $\dim(\text{span}(\{0, x_1, \ldots, x_N\})) = d$. Note that $d+1$ observations is not sufficient for invertibility, as feature vectors can be linearly dependent.

Exercise 2.2
(a) A solution can be found by considering the gradient and equating that to 0. Additionally, one needs to realize (though this may not be that trivial) that the function we are minimizing is convex. Maybe we would need a more formal argument for the latter as well...
(b) If the data spans the space, we know that $(Z^T Z)^{-1}$ exists and vice versa.
(c) Let's keep it simple and take $X = 1$.

Exercise 2.3
(a) All straight lines but the vertical one that pass through the point (π, e) will minimize the objective function.
(b) One solution is $\{ax + e - \pi a | a \in \mathbb{R}\}$.
(c) Take all 2D planes that go through the points $(1, 1, 1)$ and $(-2, 1, -1)$ except for the one that is perpendicular to the 2D X-plane.
(d) We have a perfect fit, so it's 0!
(e) Take all 2D planes that go through the points $(0, 0, 0)$ and $(1, 1, \pi)$ (except for the one that is perpendicular to the 2D X-plane).
(f) Of course, there is no unique answer to this question. Nevertheless, an argument one can make is that one does not want the solution to be unnecessarily tilted, increasing, or decreasing. In the 1D example this means we would choose a constant solution that goes right through the single training point, i.e., set $a = 0$ and the intercept to e. In the third example, where we forgot about the intercept, out of symmetry considerations, we could decide that if we look in directions perpendicular to the line $x_1 + x_2 = 0$, the regression fit should stay constant. That is, we choose the solution $\frac{\pi}{2}(x_1 + x_2)$, because why would we tilt it more or less to either side of that line? Similarly, but maybe slightly more difficult to see, we would prefer the "flattest" solution $1\frac{1}{2}x_1 - \frac{1}{2}$ for our 2D example with intercept. Interestingly, following this argument, we prefer to pick the solution for which the norm of the (non-intercept) weights is smallest.

Exercise 2.4
(a) We only need to determine the slope, which is easily determined to be $6/14$. So, the function is $6/14x$.
(b) Now we have to invert a matrix! We have $X^T X = (140; 04)$, while $X^T Y = (6; 7)$. So, solution is $(140; 04)^{-1}(14; 7) = (6/14; 7/4)\ldots$ and the intercept equals $7/4$.
(c) There are four points, so one needs at least a third-order polynomial to fit these. So the minimum degree is 3. Whether or not there is an intercept present is of no consequence.
Note that this last part is a bit tricky: the answer assumes that we do not need an explicit intercept anymore as it is automatically modelled by the 0th degree monomial anyway. If, however, we mean that such 0th degree are not at all allowed, we need an additional degree of freedom to fit four points, which means we need polynomials of degree 4.

Exercise 2.6
My quick-and-dirty solution would be two-fold. First, the encoding into months and days is not nice and I would like a more linear kind of time-scale. So, I propose to first transform that into a

1D representation t by something like $t = 30(x_1 - 1) + (x_2 - 1)$. This makes t fairly linear over one year with a minimum of 0 and a maximum of 360. Now, I would expect some periodicity in the signal. So, I actually want to move away from the linear t. I would expect one max and one min temperture in the year, so a first order approximation with a (co)sinus should be a good first attempt. Based on this, I would use as final 2D input: $(\sin(t), \cos(t))$.

Exercise 2.7
(b) Big error, the fit is basically a constant at 0.
(c) The fit again is basically constant at 0... for all degrees that are not insanely large.
(d) The issues is that `linearr`does not take into account any cross-terms!
(e) The function $y = x_1x_2$ largely behaves like $y = 50\sin(x_1)\sin(x_2)$ where the x data is sampled. But for $y = x_1x_2$ the failure of second and higher order regression should be more apparent as it is actually a second degree polynomial. So, indeed, the issues is that `linearr`does not take into account any cross-terms, i.e., x_ix_j, $x_i^2x_k^5$, $x_ix_jx_k$, etc.

Exercise 2.8
(a) $\binom{m+d-1}{m} = \frac{(m+d-1)!}{(d-1)!m!}$.
(b) If $d > 1$ then yes. The order of polynomial growth for the number of features equals the dimensionality d. You can check this experimentally.
(c) Sure. In bioinformatics one is dealing with gene expression data in which easily $d \leq 10,000$ and so $m = 3$ becomes unmanagable already. Worse even, are image classification problems in which in which one cannot afford to subsample the image. Nowadays one easily has $d > 1,000,000$ and so $m = 2$ already becomes infeasible.

Exercise 2.9
(a) We have already shown that $w = (X^TX)^+X^TY$ solves the regression task. Realizing that $(X^TX)^+X^TY = 2(X^TX)^+N\frac{1}{2N}X^TY = 2(\frac{1}{N}X^TX)^+\frac{1}{2N}X^TY$ and then working out the two components gets you to the solution.
(b) What we need to show is that the regressor w learned before the transformation applied to an untransformed x gives the same output as the regressor trained on the transformed data applied to the transformed x.

Exercise 2.10
(a) Let us use the notation $N(a|m,s)$ for the normal distribution with mean m and variance s^2 for the variable a, then we simply have $p(x,y|w) = N(x|\nu,\tau)N(y|x^Tw + w_0, \sigma)$.
(b) We just get the standard linear least squares solution back.
(c) When taking the derivative of the log-likelihood to w, the θ disappears from the equations and vice versa.
(d) We already determined $\hat{w}$ and $\hat{w}_0$ in the first part of this exercise. In addition, $\hat{\sigma} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i^T\hat{w} - \hat{w}_0)^2}$.

Exercise 2.11
(a) These should be step functions that step from -1 to +1 at the set parameter value.
(c) For all a from the intervals $(\infty, 1$ and $[5,6)$. The error rate is 2/5 for all choices.
(d) $a \in [2, \pi)$.

Exercise 2.12
(b) For H_0, we have that the optimal a should be from the set $\{a = c(\sin(\alpha), \cos(\alpha))|\alpha \in (\frac{\pi}{4}, \frac{3\pi}{4}), c \in (0,\infty)\}$ and the corresponding error rate is 0.
(c) First off, the optimal error is again 0. The three corner points are classifiers that are on the brink of making an error. They are the lines that go through the three pairs of two points. Of course, the correct label should still be assigned to the different halves of the feature space. All in all, the specific solutions for a are $(0,-1)$, $(\frac{1}{2}, \frac{1}{2})$, $(-\frac{1}{2}, -\frac{1}{2})$.
(d) We cannot find a classifier that makes 0 error now. The best we can do is an error rate of 1/3. Also, we only have a single corner now, which is the line that goes through the two negative points and classifies everything above as +1, i.e., $a = (0,-1)$.

(e) E.g. $(0,2)$ is positive and $(0,1)$ is negative. Other way around: $(1,2)$ is negative and $(1,1)$ is positive.
(f) No, this is not possible.

Exercise 2.13
(a) Let's say you found the decision boundary to be located at a, as for this example it is just one single point. In a way, any 0D "plane" $wx + w_0$ for which $wa + w_0 = 0$ works. E.g., one may take $w = 1$ and $w_0 = -a$. The only option that needs to be excluded is the choice $w = 0$.

Exercise 2.14
(a) We start with $\mathbf{w}_0 = [0,0]^T$. Let's call the first class ω_1 and the second class ω_2. Now object $\mathbf{x}_1 = [0,0]^T$ goes wrong ($\mathbf{w}_0^T\mathbf{x}_1 \leq 0$), so we update $\mathbf{w}_1^T = [0,0]^T + 1 \cdot [0,0]^T = [0,0]^T$. Next, $\mathbf{w}_1^T\mathbf{x}_2 \leq 0$, so is also wrong, and we update $\mathbf{w}_2^T = [0,0]^T + 1 \cdot [0,1]^T = [0,1]^T$. Next, $\mathbf{w}_2^T\mathbf{x}_3 \geq 0$, so is also wrong, and we update $\mathbf{w}_3^T = [0,1]^T - 1 \cdot [1,0]^T = [-1,1]^T$. And $\mathbf{w}_3^T\mathbf{x}_4 \geq 0$, so we update $\mathbf{w}_3^T = [-1,1]^T - 1 \cdot [1,1]^T = [-2,0]^T$.
In the next round $\mathbf{x}_1 = [0,0]$ is still wrong, but updating with $[0,0]$ does not change the classifier. But after updating with $\mathbf{x}_2$ we get $\mathbf{w}_5 = [-2,1]^T$. With this weight vector also $\mathbf{x}_3$ and $\mathbf{x}_4$ are correctly classified. Only for $\mathbf{x}_0$ we will always get a 0 output (because we did not use a bias term). So this will be our solution: $\mathbf{w} = [-2,1]^T$.
If we want to make a solution including the bias term, we start with the weight vector $\mathbf{w} = [0,0,0]^T$:
for $\mathbf{x}_1 = [0,0,1]^T$: $\mathbf{w}^T\mathbf{x} = 0 \rightarrow$ incorrect$\rightarrow$ $\mathbf{w}_2 = [0,0,0] + [0,0,1]$.
for $\mathbf{x}_2 = [0,1,1]^T$: $\mathbf{w}^T\mathbf{x} = 1 \rightarrow$ correct
for $\mathbf{x}_3 = [1,0,1]^T$: $\mathbf{w}^T\mathbf{x} = 1 \rightarrow$ incorrect$\rightarrow$ $\mathbf{w}_3 = [0,0,1] - [1,0,1] = [-1,0,0]$.
for $\mathbf{x}_4 = [1,1,1]^T$: $\mathbf{w}^T\mathbf{x} = -1 \rightarrow$ correct
for $\mathbf{x}_1 = [0,0,1]^T$: $\mathbf{w}^T\mathbf{x} = 0 \rightarrow$ incorrect$\rightarrow$ $\mathbf{w}_5 = [-1,0,0] + [0,0,1] = [-1,0,1]$.
for $\mathbf{x}_2 = [0,1,1]^T$: $\mathbf{w}^T\mathbf{x} = 1 \rightarrow$ correct
for $\mathbf{x}_3 = [1,0,1]^T$: $\mathbf{w}^T\mathbf{x} = 0 \rightarrow$ incorrect$\rightarrow$ $\mathbf{w}_7 = [-1,0,1] - [1,0,1] = [-2,0,0]$.
for $\mathbf{x}_4 = [1,1,1]^T$: $\mathbf{w}^T\mathbf{x} = -2 \rightarrow$ correct
for $\mathbf{x}_1 = [0,0,1]^T$: $\mathbf{w}^T\mathbf{x} = 0 \rightarrow$ incorrect$\rightarrow$ $\mathbf{w}_9 = [-2,0,0] + [0,0,1] = [-2,0,1]$.
for $\mathbf{x}_2 = [0,1,1]^T$: $\mathbf{w}^T\mathbf{x} = 1 \rightarrow$ correct
for $\mathbf{x}_3 = [1,0,1]^T$: $\mathbf{w}^T\mathbf{x} = -1 \rightarrow$ correct
for $\mathbf{x}_4 = [1,1,1]^T$: $\mathbf{w}^T\mathbf{x} = -1 \rightarrow$ correct
for $\mathbf{x}_1 = [0,0,1]^T$: $\mathbf{w}^T\mathbf{x} = 1 \rightarrow$ correct
Done!

Exercise 2.15
(a) `X=double(a); y = 2*getnlab(a)-3`
(b) Something like: `[n,dim] = size(x);`

```
x = [x ones(n,1)];
w = [0 0 0];
rho = 1;
t = 0;
I = 1:n;
while  isempty(I)
  I = find(y.*(x*w')<=0);
  if isempty(I), break; end
  w = w + rho*mean( bsxfun(@times,y(I),x(I,:)),1);
  t = t+1;
end
scatterd(a); hold on;
plot([0 w(1)],[0 w(2)]);
V = axis;
x2 = V(3:4);
x1 = -(x2*w(2)+w(3))/w(1);
hold on; plot(x1,x2,'r-')
```

(c) It should separate the classes!
(d) The weight vector will be updated indefinitely, and it will not converge.

Exercise 2.16

For Fisher, we get something like: `x = +a;`

```
y = 2*getnlab(a) - 3;
[n,dim] = size(x);
x = [x ones(n,1)];
w_hat = inv(x'*x)*x'*y
V = axis; x2 = V(3:4);
x1 = -(x2*w_hat(2)+w_hat(3))/w_hat(1);
hold on; plot(x1,x2,'r-')
w = fisherc(a); % the prtools way:
plotc(w,'k--');
```