Solutions to exercises: week 3

Exercise 3.1
(a) Look up what psd means and realize that for a vector z, $z^T X^T X z$ is always nonnegative, because it equals the inner product of Xz with itself.
(b) $X^T X$ is psd and therefore has only nonnegative eigenvalues (it basically is a covariance matrix). Adding λI makes sure all eigenvalues become positive and therefore this sum of matrices is invertible.
(c) 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...
(d) The influence of the regularizer becomes smaller and smaller. In the limit, one find the optimal linear least squares fit in which the influence of the regularizer has completely vanished.
(e) In the limit, we get $w = 0$.

Exercise 3.2
Simplest solution[?]: generate a single point with a 1D input and a 1D output and show that the solution always goes through this point, irrespective of amount of regularization.

Exercise 3.3
(a) One should get the impression that regularization can have quite a beneficial effect.
(b) It may be possible to easily do this in a nifty way using `prcrossval`, but I didn't try. I just wrote some loops that make sure that I run over enough random training sets. For all training set sizes, I found a values around 3 to be fairly optimal.

Exercise 3.4
I don't know about you, but I would probably use cross validation for this.

Exercise 3.5
If we have a number of samples such that $X^T X$ is invertible, we just get the standard solution back, which is unique to start with and so also is the minimum norm. If $X^T X$ is not invertible, we give the following handwavy "limit" argument. The nonivertibility leads to a whole set of solutions W that all minimize $\|Xw - Y\|^2$, i.e., they all have the same total squared loss. The solution based on the pseudoinverse is obtained by considering the regularized problem $\|Xw - Y\|^2 + \lambda\|w\|^2$, where $\lambda > 0$ shrinks to 0. Among the set of all solutions W, the one with the minimum norm minimizes $\|Xw - Y\|^2 + \lambda\|w\|^2$ for any λ. The limit $\lambda \downarrow 0$ will therefore also give the minimum norm solution.

Exercise 3.7
(a) No, none of the entries ever become zero really. The probability that this happens is 0. In the limit, for λ larer and larger, w should of course shrink to 0 however.
(b) In this setting there will be a finite λ for which at least on of the entries (most often the second of course!) becomes zero. For an even larger λ, also the other entry will become 0.

Exercise 3.8
(a) I'm not going to be precise (intercept yes/no etc.). So, one way to describe it is that $H = \mathbb{R}^d$ and, with $D = (X, Y)$, the loss is $\|Xh - Y\|^2$. Finally, the regularization term is given by $\lambda\|h\|^2$.
(b) The hypothesis class is the set of all Gaussian distributions (in $\mathbb{R}^d$, say). Of course, you should be able to write more explicitly how that set looks like... With $D = X$, the loss is the negative(!) (log-)likelihood: $\prod_{i=1} h(x_i)$. We don't have a regularization term.

Exercise 3.9
(a) With no intercept, we have $\ell(x, y|h) = (h(x) - y)^2 = (w^T x - y)^2$ if we assume that h is parametrized by $w \in \mathbb{R}^d$.
(b) Given that $h \in H$ is a probabilistic model and (x, y) is a point from the training data, we could define the loss as $-\log h(x, y)$. Often, we make explicit that H can be parametrized (say through some θ) and we would consider the corresponding loss $-\log h(x, y|\theta)$.

(c) One example is AUC.
(d) I would go for 1NN. Or better 3NN.

Exercise 3.10
(a) Every term in the objective function can be written as $\log_2(1 + e^{-y_i x_i^T w})$, so we have $v(a) = \log_2(1+e^{-a})$ (or some equivalent expression of course). Note that this also shows that the definition of a classifier in terms of its hypothesis class, its loss, and its regularization function is not unique: there are multiple formulations that lead to the same classifier.
(b) $v(a) = (1 - a)^2$.
(c) This may not be obvious. Answer: $v(a) = \max(0, 1 - a)$.
(d) I don't think so. But if you have an idea, I would be happy to discuss (ML).

Exercise 3.12
(b) With this w_{ML}, the estimator for σ becomes $\frac{1}{N}\sum(w_{\mathrm{ML}}^T x_i - y_i)^2$.

Exercise 3.13
(a) Python code may look like:
```
N = 30
e = np.zeros(N)
t = pr.gendath([500,500])
for i in range(N):
    a = pr.gendath([20,20])
    e[i] = t*pr.ldc(a)*pr.testc()
print(e)
print(np.std(e))
```
Each time we draw a new, small training set. So the variation is in the variability of the training set.
(b) Here we draw a new test set each time. The test set is relatively large, so the variability in the error should be much smaller.

Exercise 3.14
(a)
```
a = pr.gendath([1000,1000])
noise = np.random.randn(2000,60)
a.concatenate(noise,axis=1)
pr.cleval(a,pr.nmc(),[64,128,256,512])
pr.cleval(a,pr.ldc(),[64,128,256,512])
pr.cleval(a,pr.qdc(),[64,128,256,512])
plt.legend()
```
Test curves go down, training curves go up. Both curves should converge in the end. Where the curves converge depend a bit on what classifier you use. More flexible classifiers get a lower asymptotic error.
(b) Simpler classifiers work generally better when sample sizes are small. None of the classifiers is best.
(c) Training and test error converge to same value, QDC < LDC < NMC.

Exercise 3.15
(a) The 1-NN classifier has a zero apparent error!

Exercise 3.16
(a) The curves will intersect in a different kind of way because of the difference in data distributions and the way the classifiers fit these distributions.

Exercise 3.17
(a) The classifier fits better, though still badly, to the data when there are fewer data points. As a result we see the not-so-well-known dipping phenomenon.

Exercise 3.18

(a) We also want to know the solution.

Exercise 3.19
(a) When you do not change the training set, nothing will change. In the function `clevalf`each time the first 4, 8, 16, 32 or 64 features are used, so there is no randomization is used here.
(b) You only get the first part of the learning curve, because less data is available. In particular when you use only 40% of the data, you tend to see only overfitting on the training set.

Exercise 3.20
(b) Typically you get a less biased estimate of the error when you increase n, but the variance typically increases a bit with larger n.
(c) When you have a small n, the variance is a bit smaller
(d) For a larger dataset, the bias and the variance are a bit smaller

Exercise 3.21
(a) The `lab`contains the original, true, labels, while `lab2`contains the labels that are predicted by classifier `w`.

Exercise 3.22
(a) The error using the LDA on the Zernike dataset is around 20%, while on the Karhunen-Loeve moments it is just 5%.
(b) There is a lot of confusion between classes 6 and 9 in the Zernike dataset. This is not surprising, because the Zernike moments are rotationally invariant.

Exercise 3.23
(a) When a correct classification incurs no cost, we know that $a = d = 0$. If misclassifying class 1 objects to class 2, is twice as bad as vice-versa, then $b = 2c$.