Past Papers unofficial solutions

While studying and unable to have the solutions for these papers, I decided to create this document which will (hopefully) contain all (unofficial) solutions from every question from every past paper in the MLPR course. There are only a few days till the exam, so I won't be writing answers in details; rather a sentence or a phrase to which you can use to expand your solution for that question.

These answers have been written by me or with some assistance from someone who knew the topic or material more than I do. Some questions contain multiple possible answers. It's important to note that **these answers are not official and not fact checked with a course organizer**. Take them with a grant of salt.

I spoke with Dr. Arno, and he likes the idea and even encouraged me (and everyone) to share past paper solutions with each other. Discussion with your peers is a great way of learning. Your peer may know something that you don't and vice versa.

If you have any suggestion or any edits I should make, please email me and I will immediately act on it.

2021 Dec

1. A) *(Pick one solution)*
   i. Gaussian Elimination -> epoch form -> # of Non 0 rows Σ(2), so not full rank $\therefore$ Σ(2) causes the problem.
   ii. Cholesky Decomposition – numpy.linlang.cholesky(Σ) -> Σ(2) throws an error.

2. A) *(Any two solutions)*
   i. **(possibly wrong)** L2 Regularization to the diagonal of the matrix – makes it positive definite.
   ii. **(possibly wrong)** Use Partial Eigenvalue Decomposition on Σ(2) = QΛQ^T. replace negative eigenvalues with 0 -> reconstruct Σ(2).
   iii. Modify training data.
   iv. Remove 3rd feature.

   B)
   i. **(mod required)** Function based on description (large values close to edges): sigmoid with range (3,5): $\mathrm{a}_{t+1} = 3 + \sigma\big(\eta \cdot \nabla_{\mathrm{a}} \mathrm{c}(\mathrm{a}_t)\big) \cdot 2$, with cost function $c(a) = \frac{1}{2m}\sum_{i=1}^{m}\big(y_i - f(x_i;a)\big)^2$ where the partial derivative is: $\frac{\partial c}{\partial a} = -\frac{1}{m}\sum_{i=1}^{m}\big(y_i - f(x_i;a)\big)\frac{\partial f(x_i;a)}{\partial a}$. Use a small learning rate, $\eta\ =\ 0.001$.
   ii. ☹
   iii. Apply Gaussian Naives optimization and then RMSE (check Assignment, Q5).

3. A)
    i. D features, N data points -> O(D*N)
    ii. Full Covariance -> O(N*D^2)

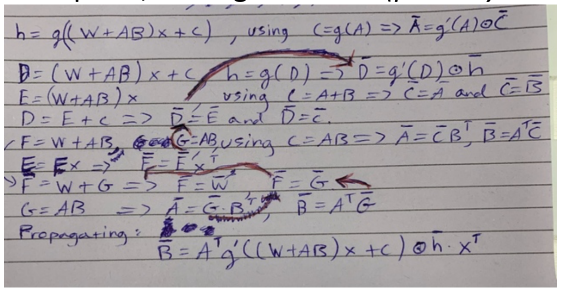
   B)
    i. Pro: Faster computationally.
       Pro: Suitable for large databases, thanks to the use of mini batches.
       Con: Harder to implement.
       Con: Higher Computational Complexity (greater Big "Oh").

4. A)
    i. W: H x D (H hidden neurons, D dimensional layer x) order matters!
    ii. b: H x 1

   B) Matrix A: H x K
   Matrix B: K x D
   Bias c : H
   Extra parameters per country: H x K + K x D + H.

   C) `X@(W+A@B).T+c[None,:]`

   D) Less risk on overfitting, less likely to memorize training data, less memory consumption, more generative. *(pick any two of these reasons)*

   E)


$h = g((W+AB)x+c)$, using $C = g(A) \Rightarrow \bar{A} = g'(A) \odot \bar{C}$

$D = (W+AB)x+c$ $\quad h = g(D) \Rightarrow \bar{D} = g'(D) \odot \bar{h}$

$E = (W+AB)x$ $\quad$ using $C = A+B \Rightarrow \bar{C} = \bar{A}$ and $\bar{C} = \bar{B}$

$D = E + c \Rightarrow \bar{D} = \bar{E}$ and $\bar{D} = \bar{c}$.

$F = W + AB$, $G = AB$, using $C = AB \Rightarrow \bar{A} = \bar{C}B^T$, $\bar{B} = A^T\bar{C}$

$E = Fx \Rightarrow \bar{F} = \bar{E}x^T$

$F = W + G \Rightarrow \bar{F} = \bar{W}$ $\quad \bar{F} = \bar{G}$

$G = AB \Rightarrow \bar{A} = \bar{G} \cdot B^T$, $\bar{B} = A^T\bar{G}$

Propagating: $\bar{B} = A^T g'((W+AB)x+c) \odot \bar{h} \cdot x^T$

   F) Initialize AB to be a zero matrix by setting the matrix B to zero. Replace bias c to b. For the first epoch, the model will be back to its original form. The gradients from B will slowly be generated after training.

   G) Early Stopping (to prevent overfitting).

5. A) ☹

   B) **(possible idea)** take k(x1, x2) = E[f1 * f2] – E[f1] * E[f2], where f = a+bx+cx^2. No, we can't have a gaussian process k(x1, x2) < 0.

2021 Dec Comments

Harder than expected. The exam was open book, so questions are expected to be harder and require higher levels of technical knowledge to answer them. Don't let that bring you down. ☺ Keep in mind, that the above answers are **NOT** fact checked by any organizers of this course, so mistakes are expected to exist. The answers will be updated and checked daily by myself and my peers until the day of the examination.

2019 Dec

i. A)
$\mathbb{E}_{p(\mathbf{x},y)}[L(y, f(\mathbf{x}))]$ Where: L = loss, y = training result, f(x) = model function, p = true data distribution.
OR $\int L(y, f(\mathbf{x}))p(\mathbf{x}, y)\mathrm{d}\mathbf{x}\mathrm{d}y$ Where: L = loss, y = training result, f(x) = model function, p = true data distribution.
OR The expected loss that the model (function) will achieve/yield/score on future/unseen test datapoints.

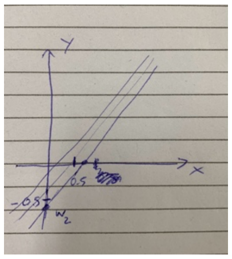
ii. $\frac{1}{M}\sum_{m=1}^{M} L(y^{(m)}, f(\mathbf{x}^{(m)}))$

iii. $\sqrt{\frac{1}{M}} \sum_{m=1}^{M} (Lm - L)^\wedge 2$

B)
i. –
ii. –

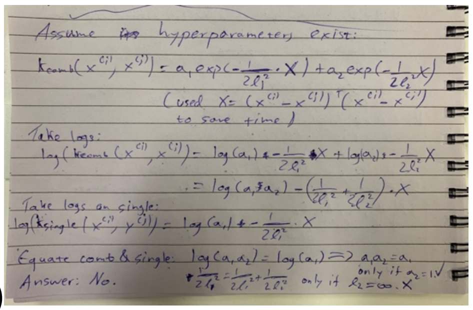
C)
i. Generalization error doesn't predict risk to overfitting. Model could be prone to overfitting.
ii. Provide validation data sets to test the model and check if it overfits.

2. A) W: K x D, c: D x 1, v: 1 x K, b: 1 x 1.
B) Gradients will be zero and the model will not learn.
C) Prevents from the gradients (derivatives) from growing too large, causing optimization issues.

3. A) i) 

3 lines close to the means on each axis, each liine close to each other (small variance), straight lines (linear function). Not necessarily parallel, small angle should be considered, given the means.
ii) Increase the variance, so there's less emphasis on the mean.
B) i) We have prior knowledge, based on the parameters (mean and variance). Very efficient with less data, just like in our example.
ii) Minimizing the expected square error, will essentially get closer to the meam of the samples. Similarly, the predictive distribution, will choose a predictive value closer to the mean (mode) of the gaussian. Therefore, both ideas estimate the y in a similar way.
C) i) α1,α2,l1,l2. α1 > 0,α2 > 0, l != 0 1,l2 != 0.
ii) α1,α2, give more emphasis on one kernel over another. The l (lengthscale) changes

the variation of the kernel.

iii) 

Assume ~~[illegible]~~ hyperparameters exist:

$k_{comb}(x^{(i)}, x^{(j)}) = a_1 \exp(-\frac{1}{2\ell_1^2} \cdot X) + a_2 \exp(-\frac{1}{2\ell_2^2} X)$

(used $X = (x^{(i)} - x^{(j)})^T (x^{(i)} - x^{(j)})$ to save time)

Take logs:

$\log(k_{comb}(x^{(i)}, x^{(j)})) = \log(a_1) + -\frac{1}{2\ell_1^2} X + \log(a_2) + -\frac{1}{2\ell_1^2} X$

$= \log(a_1 a_2) - (\frac{1}{2\ell_1^2} + \frac{1}{2\ell_2^2}) \cdot X$

Take logs on single:

$\log(k_{single}(x^{(i)}, y^{(j)})) = \log(a_1) + -\frac{1}{2\ell_1^2} \cdot X$

Equate comb & single: $\log(a_1 a_2) = \log(a_1) \Rightarrow a_1 a_2 = a_1$ only if $a_2 = 1$ ✓

Answer: No.

$\frac{1}{2\ell_1^2} = \frac{1}{2\ell_1^2} + \frac{1}{2\ell_2^2}$ only if $\ell_2 = \infty$. ✗

4. A) B (dash) = A^T x C (dash) -> 1st Standard Result)
   B) O(N x K x D^2)
   C) PCA, pro: doesn't require data labels, con: error when re-assembling data (some data loss).

5. A) $p(y = 1 \mid x, D) \approx S1\sum s = 1S\sigma(wsTx)$
   B) ☹
   C) SVI, Laplace, Sampling, Monte Carlo, Sequential Monte Carlo (any two).

2019 Dec Comments

1)a) has been fact checked from the last exam preparation lecture.