Nimara Doumitrou-Daniil
Student, First Year
M.Sc. in Machine Learning
nimara@kth.se

Assignment 1

Machine Learning, advanced : November 2019 :
Due Sunday, December 8, 2019

Contents

Q1 **3**

Q2 **3**

Q3 **4**

Q4 **4**
Q4.1 4
Q4.2 4

Q5 **5**
Q5.1 5
Q5.2 7
Q5.3 7

Q6 **7**

Q7 **9**

Q8 **9**
Q8.1 9
Q8.2 9
Q8.3 9

Q9 **10**

Q10 **12**

Q10.1 . . . 12
Q10.2 . . . 13

Q11 **13**

Q12 **14**

Q13 **15**

Q14 **15**
Q14.1 . . . 15
Q14.2 . . . 16
Q14.3 . . . 16

Q15 **17**

Q16 **18**

Q17 **20**

Q18 **20**

Q19 **20**

Q20 **22**

Q21 **22**

Q22 **23**

Q23 **25**

Q24 **25**

Q1

Recall that in regression, we consider the following relationship between our target values $\mathbf{t}$ and $\mathbf{x}$:

$$\boldsymbol{t} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}$$

where $f(\boldsymbol{x})$ is the regression model and $\boldsymbol{\epsilon}$ is a random noise. We may then make the **assumption** that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 I)$, that is, the noise follows a normal (Gaussian) distribution with zero mean and a diagonal covariance matrix of equal variances (i.e. spherical) σ^2.

Then, when we condition our target t_i on f and x_i we have that it follows a Gaussian distribution ($f(\boldsymbol{x})$ is a constant value, since we conditioned on it, and thus the randomness derives from the Gaussian distribution of the noise), with:

$$E[\boldsymbol{t_i}|f, \boldsymbol{x_i}] = E[f(\boldsymbol{x_i})] + E[\boldsymbol{\epsilon}] = f(\boldsymbol{x_i}) + \mathbf{0} = f(\boldsymbol{x_i})$$

and

$$Var[\boldsymbol{t_i}|f, \boldsymbol{x_i}] = Var[f(\boldsymbol{x_i})] + Var[\boldsymbol{\epsilon}] = \mathbf{0} + \sigma^2 I = \sigma^2 I$$

Thus,

$$p(\boldsymbol{t_i}|f, \boldsymbol{x_i}) \sim N(f(\boldsymbol{x_i}), \sigma^2 I)$$

This distribution is sensible, since it only makes the following reasonable assumptions (from which it afterwards derives from):

- The uncertainty between the relationship of our features x and target t can be expressed through the form $\boldsymbol{t} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is random noise.
- The random noise follows a normal distribution. In fact, assuming a Gaussian noise is rather frequent throughout Machine Learning and Statistics literature. This distribution is common in Nature and plays a pivotal role in Statistics, as it exhibits many desirable properties. One especially important property is the Central Limit Theorem (CLT), which states that under certain conditions, the distribution of the average of independent random variables tends towards a Gaussian distribution. As such, we can imagine that our noise ϵ describes the average effect of independent factors (e.g. measurement errors) that we do not explicitly take into account in our regression model (as features).
- The random noise has zero mean and spherical covariance matrix.

The spherical covariance matrix in our data's likelihood derived from assuming a spherical covariance matrix in the noise. This simply states that the uncertainty between the x and t is the same in every dimension of the target. Equivalently, if $\boldsymbol{t} \in R^D$, then $\boldsymbol{\epsilon} \in R^D$, and $\epsilon_j \sim N(0, \sigma^2)$, $j \in \{1, 2, ..., D\}$, ϵ_j, ϵ_k independent for $j \neq k$. Thus, $\boldsymbol{t}_i$, $\boldsymbol{t}_j \in R$ are conditionally independent, given f and x.

Q2

The total likelihood derives from applying the multiplication rule of probability. Assuming independence makes the computation easier, since we can use the following formula:

$$p(A, B) = p(A)p(B)$$

In the absence of this, we need to use the more general product rule:

$$p(A, B) = p(A)p(B|A)$$

Thus, since $\boldsymbol{T} = [\boldsymbol{t_1}, ..., \boldsymbol{t_N}]$, we have:

$$p(\boldsymbol{T}|f, \boldsymbol{X}) = p(\boldsymbol{t_1}|f, \boldsymbol{X})p(\boldsymbol{t_2}|f, \boldsymbol{X}, \boldsymbol{t_1})...p(\boldsymbol{t_N}|f, \boldsymbol{X}, \boldsymbol{t_1}, \boldsymbol{t_2}..., \boldsymbol{t_{N-1}}) = \prod_{i=1}^{N} p(\boldsymbol{t_i}|f, \boldsymbol{X}, \boldsymbol{t_{<i}})$$

where $\boldsymbol{t_{<i}} = \boldsymbol{t_1}, ..., \boldsymbol{t_{i-1}}$

Since we did not assume that our data points are independent, we can not further simplify our formula, by stating that $\boldsymbol{t_i}$ is conditionally independent to $\boldsymbol{t_{<i}}$ or other data points $\boldsymbol{x_j}$ $j \neq i$.

Q3

The formula is easily derived using similar calculus to what we used in the first question. Since we considered a linear model:

$$\boldsymbol{t_i} = \boldsymbol{W}\boldsymbol{x_i} + \boldsymbol{\epsilon}$$

Thus, we see (as illustrated in question 1), that $(\boldsymbol{t_i}|\boldsymbol{W}, \boldsymbol{x_i})$ follows a Gaussian distribution with:

$$E[\boldsymbol{t_i}|\boldsymbol{W}, \boldsymbol{x_i}] = E[\boldsymbol{W}\boldsymbol{x_i}] + E[\boldsymbol{\epsilon}] = \boldsymbol{W}\boldsymbol{x_i} + \boldsymbol{0} = \boldsymbol{W}\boldsymbol{x_i}$$

and

$$Var[\boldsymbol{t_i}|\boldsymbol{W}, \boldsymbol{x_i}] = Var[\boldsymbol{W}\boldsymbol{x_i}] + Var[\boldsymbol{\epsilon}] = \boldsymbol{0} + \sigma^2 I = \sigma^2 I$$

Thus,

$$p(\boldsymbol{t_i}|\boldsymbol{W}, \boldsymbol{x_i}) = N(\boldsymbol{W}\boldsymbol{x_i}, \sigma^2 I)$$

Then,

$$p(\boldsymbol{T}|\boldsymbol{W}, \boldsymbol{X}) = \prod_{i=1}^{N} p(\boldsymbol{t_i}|\boldsymbol{W}, \boldsymbol{x_i}) = \prod_{i=1}^{N} N(\boldsymbol{W}\boldsymbol{x_i}, \sigma^2 I)$$

That is:

$$p(\boldsymbol{T}|\boldsymbol{W}, \boldsymbol{X}) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\sigma^2 I|}} \cdot e^{-\frac{1}{2}(\boldsymbol{t_i} - \boldsymbol{W}\boldsymbol{x_i})^T \sigma^{-2} I (\boldsymbol{t_i} - \boldsymbol{W}\boldsymbol{x_i})}$$

Q4

Q4.1

We have seen in this course (machine learning advanced D2434), as well as the previous one (machine learning D2421), that introducing this prior leads to ridge regression, which regularizes based on the L_2 norm of the weights. In order to regularize based on L_1 (lasso regression), we have seen in the previous course, that we need to impose a Laplace prior. As such, we should assume a matrix Laplace distribution on the prior $p(\boldsymbol{W})$ (see Lecture 7, slide 34 out of 108, Machine Learning).

The effect of imposing an L_1 regularization on our weights, means that we incentivize making w sparse (it will contain a lot of zero weight values). This is a characteristic of the Lasso regression, that is often used, due to the sparsity it imposes on the weights, as a feature selection technique.

Q4.2

We can see how the prior $p(W)$ acts as a regularization technique by doing the following:

$$p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T}) = \frac{1}{Z} p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{W}) p(\boldsymbol{W}) \iff$$

$$\log p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T}) = -\log Z + \log p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{W}) + \log p(\boldsymbol{W})$$

In the maximum a posteriori approach, we would seek to maximize $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T})$. It is clear how, then, $p(\boldsymbol{W})$ imposes a regularization on our maximization problem, forcing us to not only maximize the likelihood $p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{W})$, but also take into account the effect of $p(\boldsymbol{W})$. The term $-\log p(\boldsymbol{W})$ is called the *penalization term*. If we assume an L_2 based prior (Gaussian priors), then we end up with ridge regression, that imposes an L_2 based penalization penalty. We have seen this effect in "Lecture 2: Fundamentals of the probabilistic approach", slide 48 out of 100, where we examined maximum a posteriori and saw how this leads to ridge regression. Increasing $||w||_2$ leads to decreasing $P(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{t})$, and as such, this regularization incentivizes smaller weights. Similarly, the imposing L_1 regularization, formulates a penalty proportional to $||w||_1$, which incentivizes a sparser model. The penalization terms are:

- L_2: $\frac{1}{2\sigma^2}||\boldsymbol{w}||_2^2$, when $p(\boldsymbol{w}) = N(0, \sigma^2 I)$. Notice something elegant: The more "centered" our prior is (smaller σ), the higher the penalization ($\frac{1}{\sigma^2}$ increases) we impose for weights that steer further away from $\mathbf{0}$ (the mean of our normal distribution). For $p(\boldsymbol{w}) = N(\boldsymbol{w}_0, \sigma^2 I)$, the penalization term becomes $\frac{1}{2\sigma^2}||\boldsymbol{w} - \boldsymbol{w_0}||_2^2$
- L_1: $\frac{1}{b}||w||_1$, when $p(\boldsymbol{w}) = \prod_{i=1}^{q} \frac{1}{2\cdot b} e^{-\frac{|\boldsymbol{w}_i|}{b}}$. The penalization term shifts to $\frac{1}{b}||\boldsymbol{w} - \boldsymbol{w_0}||_1$ when $p(\boldsymbol{w}) = \prod_{i=1}^{q} \frac{1}{2b} e^{-\frac{|w_i - (w_0)_i|}{b}}$.

The effect of regularization is succinctly illustrated bellow:

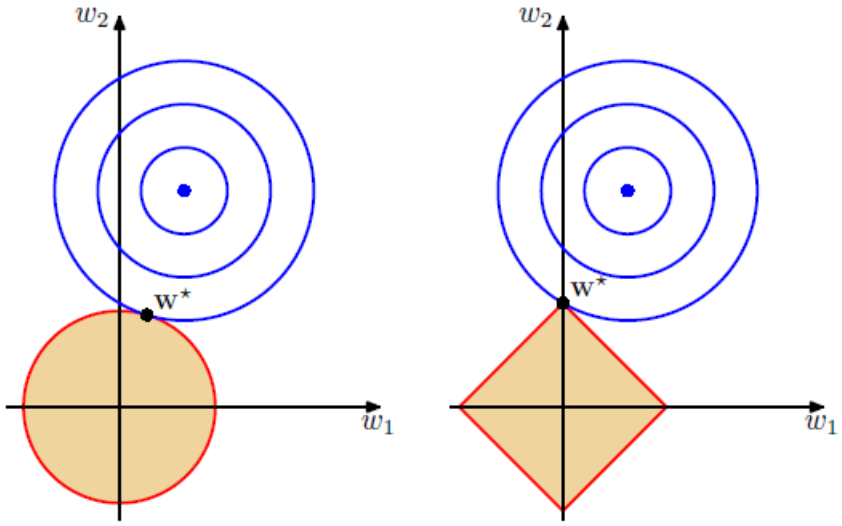


Figure 1: Regularizer illustration: effect of penalty term on unregularized square sum error (blue contour) $-\log P(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w} = (w_1, w_2))$. On the left: L_2 regularization (hence the circle), on the right: L_1 regularization [1]

The point in the center corresponds to the optimal $\boldsymbol{w} = \boldsymbol{w_{ML}}$ that minimizes $-\log P(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w} = (w_1, w_2))$ (thus maximizes $P(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w} = (w_1, w_2))$). The yellow shaded area correspond to the space $\{\boldsymbol{w} \in R^2 | \ ||w||_l \leq s\}$, $(l = 1, 2)$. The solution to this problem lies on the intersection of the yellow shaded area with the contour nearest to the optimal value. As we increase s (decrease penalization) $\boldsymbol{w}_{penalized} = \boldsymbol{w_{MAP}} \rightarrow \boldsymbol{w}_{ML}$, since the boundary grows, going ever nearer to $\boldsymbol{w_{ML}}$. Notice how, due to its "sharp corners", the intersection for L_1 is likely to occur on one of the axes. This is a simple explanation of the sparse quality of L_1.

Finally, the effect on the posterior is heavily linked to its effect on the Maximum a posteriori estimate $\boldsymbol{w_{MAP}}$. The posterior distribution peaks at $\boldsymbol{w_{MAP}}$ (by definition $\boldsymbol{w_{MAP}} = argmax(P[\boldsymbol{w}|\boldsymbol{X}, t])$). As such, it alters the posterior distribution, shifting the location of its peaks-center.

Q5

Q5.1

Since we assume conditional independence in the variables $\boldsymbol{t}$, we can derive $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T})$, by first computing $p(\boldsymbol{W_j}|\boldsymbol{X}, \boldsymbol{T_j})$ ($\boldsymbol{W_j}$, $\boldsymbol{T_j}$ the j-th row of W and T respectively) and then simply take the product of these probabilities to formulate the whole $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T})$. Let us work with row j, and for simplicity, ignore the index j. We will thus compute the posterior of a single row of weights $\boldsymbol{w}$. Now, our linear model is more familiar to us, as the target t is a scalar:

$$t = \boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + \epsilon$$

$\epsilon \sim N(0, \sigma^2)$. We know that the prior of each row weight is simply $p(w) = N(\boldsymbol{w_0}, \tau^2 I)$. Furthermore, our model is now of the form $t_i = \boldsymbol{w}^T \boldsymbol{x_i} + \epsilon$. As such, if $\boldsymbol{t}$ contains all the t_i from our data (this corresponds to a row from the initial $\boldsymbol{T}$, in the same way $\boldsymbol{w}$ is the corresponding row of $\boldsymbol{W}$), then ($\boldsymbol{X} = [\boldsymbol{x_1}, ..., \boldsymbol{x_N}]$):

$$\boldsymbol{t} = \boldsymbol{X^T w} + \epsilon$$

and as such (similar calculus that we used in previous questions)

$$p(\boldsymbol{t}|\boldsymbol{w}, \boldsymbol{X}) = N(\boldsymbol{X^T w}, \sigma^2 I)$$

Thus, our posterior is simply (ignoring the normalization term Z):

$$p(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}) \propto N(\boldsymbol{X^T w}, \sigma^2 I) * N(\boldsymbol{w_0}, \tau^2 I)$$

Let us for simplicity, focus on the exponent of the right hand side, since we know the overall result will be Gaussian. We have:

$$-\frac{1}{2}(\boldsymbol{t} - \boldsymbol{X}^T \boldsymbol{w})^T \sigma^{-2} I (\boldsymbol{t} - \boldsymbol{X}^T \boldsymbol{w}) - \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w_0})^T \tau^{-2} I (\boldsymbol{w} - \boldsymbol{w_0}) =$$

$$-\frac{1}{2\sigma^2}(\boldsymbol{t}^T \boldsymbol{t} - 2\boldsymbol{w}^T \boldsymbol{X t} + \boldsymbol{w}^T \boldsymbol{X X}^T \boldsymbol{w}) - \frac{1}{2\tau^2}(\boldsymbol{w}^T \boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{w_0} + \boldsymbol{w_0}^T \boldsymbol{w_0}) =$$

$$-\frac{1}{2}\boldsymbol{w}^T(\frac{1}{\sigma^2}\boldsymbol{X X}^T + \frac{1}{\tau^2} I)\boldsymbol{w} + \boldsymbol{w}^T(\frac{1}{\sigma^2}\boldsymbol{X t} + \frac{1}{\tau^2}\boldsymbol{w_0}) + (-\frac{1}{2\sigma^2}\boldsymbol{t}^T \boldsymbol{t} - \frac{1}{2\tau^2}\boldsymbol{w_0}^T \boldsymbol{w_0})$$

If we assume $(\boldsymbol{w}|\boldsymbol{X}, \boldsymbol{t}) \sim N(\boldsymbol{w}|\mu_w, \Sigma_w)$, and compare it to the general formula:

$$-\frac{1}{2}\boldsymbol{w}^T \Sigma_w^{-1} \boldsymbol{w} + \boldsymbol{w}^T \Sigma_w^{-1} \mu_w + \mu_w^T \Sigma_w^{-1} \mu_w$$

we derive that:

$$\mu_w = \Sigma_w(\frac{1}{\sigma^2}\boldsymbol{X t} + \frac{1}{\tau^2}\boldsymbol{w_0})$$

$$\Sigma_w^{-1} = \frac{1}{\sigma^2}\boldsymbol{X X}^T + \frac{1}{\tau^2} I$$

This holds for every row $\boldsymbol{w} = \boldsymbol{W_j}$ and its corresponding $\boldsymbol{t} = \boldsymbol{T_j}$, and as such the over all posterior probability (recall we assumed conditional independence in $\boldsymbol{t}$):

$$p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T}) = \prod_{j=1}^{D} p(\boldsymbol{W_j}|\boldsymbol{X}, \boldsymbol{T_j})$$

As such, each row W_j of matrix W follows a Gaussian distribution with:

$$\mu_{W_j} = \Sigma_{W_j}(\frac{1}{\sigma^2}\boldsymbol{X T_j} + \frac{1}{\tau^2}\boldsymbol{w_{0j}})$$

$$\Sigma_{W_j}^{-1} = \frac{1}{\sigma^2}\boldsymbol{X X}^T + \frac{1}{\tau^2} I$$

Notice how the variance is the same for each row, while only the mean changes. Equivalently, we can say that $(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T}) \sim MN(W_0', I, \frac{1}{\sigma^2}\boldsymbol{X X}^T + \frac{1}{\tau^2} I)$, where W_0' is a matrix with each row: $W_{0j}' = \Sigma_{W_j}(\boldsymbol{X T_j} + \boldsymbol{w_{0j}})$

Note: The results agree with Bishop's coursebook (2006, p. 153), with one slight difference being that he considers as $\boldsymbol{X}$ the transpose matrix of ours. That is, each row of his matrix containing the training data-features corresponds to a sample. On the other hand, our $\boldsymbol{X}$ has the samples per column. This was used so that it is consistent with $\boldsymbol{T}$, as given in this assignment.

Q5.2

In the maximum likelihood approach, we ignore the priors and simply focus on finding the weights $\boldsymbol{W_{ML}}$ that maximize the likelihood $p(\boldsymbol{T}|\boldsymbol{W}, \boldsymbol{X})$. Using our previous notation, we can deduce that (see Bishop [1] p. 142):

$$\boldsymbol{W_{ML}} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{T}^T$$

or equivalently, each row of this matrix is given by:

$$\boldsymbol{W_{MLj}} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{T_j}$$

There is an interesting connection between $\boldsymbol{W_{MLj}}$ and W'_{0j}. Namely, if we assume a very broad prior, that is $\tau^2 \to \infty$, then we have:

$$\Sigma_{W_j}^{-1} \to \frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{X}^T$$

$$W'_{0j} = \mu_{W_j} \to \Sigma_{W_j}\frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{T_j} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{T_j} = \boldsymbol{W_{MLj}}$$

Another interesting observation is that the mean of our distribution is in fact equal to $\boldsymbol{W}_{MAP}$ (see Bishop p. 145). This is perfectly reasonable. After all, the maximum a posteriori estimate chooses the weight that maximizes the posterior distribution. It is therefore the point in which the distribution peaks. Since our distribution is Gausian, the distribution peaks at its mean.

Q5.3

Constant Z is a normalization term, namely $Z = p(\boldsymbol{X}, \boldsymbol{T})$. In our calculation, we did not explicitly compute it, since we knew that the resulting distribution would be Gaussian, and so we merely needed to compute its covariance and mean. Though it is not required to explicitly compute it (as we saw in this exercise), we do need to take it into account implicitly, to ensure that $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{T})$ is indeed a probability density function. Lastly, if our distributions were not completely described from their mean and variance (as Gaussians), we would likely need to compute Z in order to fully describe our posterior.

Q6

An intuitive and effective way to look at the prior is through the marginal distribution, as described in section 6.4.2 (Bishop 2006):

$$p(\boldsymbol{t_i}) = \int p(\boldsymbol{t_i}|f_i)p(f_i|\boldsymbol{X}, \theta)df_i = N(\boldsymbol{t_i}|\boldsymbol{0}, C)$$

One can compute this integral by completing the squares, but it is easier to derive it from the formula $\boldsymbol{t_i} = \boldsymbol{t_i}(f_i)$:

$$\boldsymbol{t} = f + \boldsymbol{\epsilon}$$

Thus:

$$\mu_{\boldsymbol{t}} = E[f] + E[\epsilon] = 0$$

$$Var(\boldsymbol{t}) = Var[f] + Var[\epsilon] = k(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 I$$

That is, our covariance matrix C has:

$$C(i,j) = \underbrace{k(\boldsymbol{x_i}, \boldsymbol{x_j})}_{\text{randomness induced from f}} + \underbrace{\sigma^2\delta_{ij}}_{\text{randomness induced from noise}}$$

We therefore see the effect of our choice of prior on our marginal distribution: The uncertainty (randomness) induced from the kernel $k(\boldsymbol{X}, \boldsymbol{X})$ gets filtered through it, and manifests itself as one of the two types of randomness in our marginal distribution. We also see how, the uncertainty of data point

x_i ($C(i,j)$ $j \neq i$) is affected more by data points x_j that are closer (in respect to the kernel, that is $k(\boldsymbol{x_i}, \boldsymbol{x_j})$ is bigger) to it.

Note, how the final $p(\boldsymbol{T}) = \prod_{i=1}^{N} N(\boldsymbol{t_i}|\boldsymbol{0}, C)$.

But the effect of our prior is even better captured when we examine a new observation $\boldsymbol{x}_{N+1}$. Following Bishop's coursebook, in the case of one dimensional observations t (non vectors), we have that $p(t|\boldsymbol{t}) = N(\mu(\boldsymbol{x_{N+1}}), \sigma^2(\boldsymbol{x_{N+1}}))$ where

$$\mu(\boldsymbol{x_{N+1}}) = \boldsymbol{k}^T C_N^{-1} \boldsymbol{t},\ \boldsymbol{k}^T = (k(\boldsymbol{x_{N+1}}, \boldsymbol{x_1}), ..., k(\boldsymbol{x_{N+1}}, \boldsymbol{x_N}))$$

$$\sigma^2(\boldsymbol{x_{N+1}}) = c - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k},\ c = k(\boldsymbol{x_{N+1}}, \boldsymbol{x_{N+1}}) + \sigma^2$$

(C_N is the previous matrix C that derived from our training data). This generalizes for our $\boldsymbol{t} \in R^d$ (assuming conditional independence):

$$\boldsymbol{\mu}(\boldsymbol{x_{N+1}}) = \boldsymbol{k}^T C_N^{-1} \boldsymbol{T}$$

$$\sigma^2(\boldsymbol{x_{N+1}}) = c - \boldsymbol{k}^T C_N^{-1} \boldsymbol{k}$$

and finally

$$p(\boldsymbol{t}|\boldsymbol{T}) = N(\boldsymbol{t}|\boldsymbol{\mu}(\boldsymbol{x_{N+1}}), \sigma^2(\boldsymbol{x_{N+1}})I)$$

We thus see how, if $\boldsymbol{x_{N+1}}$ is unlike previously observed data points, then $\boldsymbol{k}$ is small (since the kernel measures similarity). Thus, $\boldsymbol{k}^T C_N^{-1} \boldsymbol{k}$ is small and therefore $\sigma^2(\boldsymbol{x_{N+1}})$ is large. That is, **the more different a new observation is than our data points, the more uncertain we are about the output value t**. This makes perfect sense!

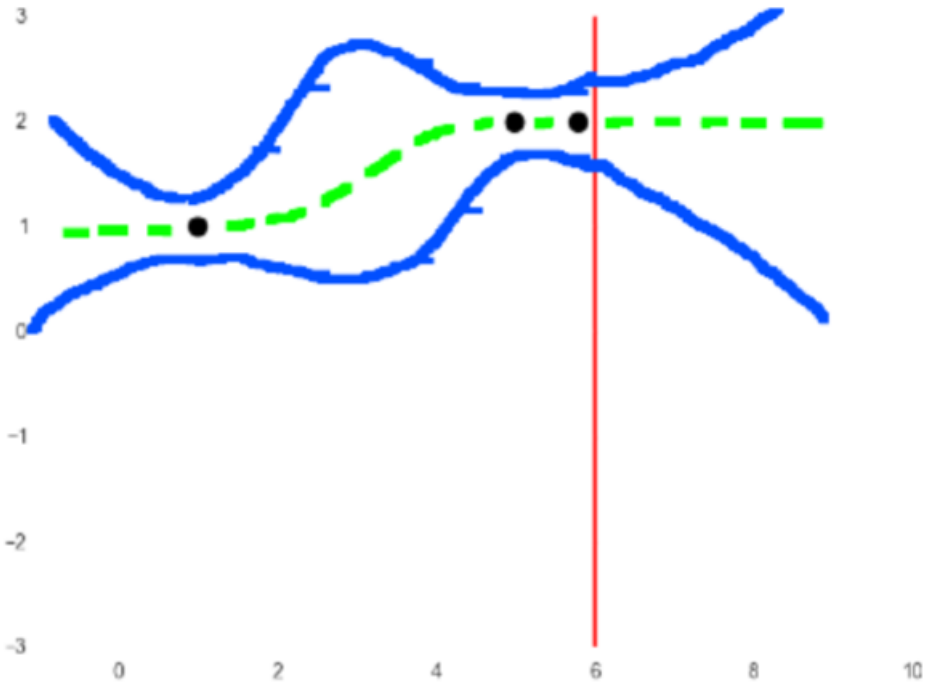


Figure 2: Effect of our prior on our posterior. The doted line indicates the mean, while the solid blue line the variance. Notice how, when we examine data points that are further away from our three training data, the variance increases.

Q7

In general:

$$p(\boldsymbol{T}, \boldsymbol{X}, f, \boldsymbol{\theta}) = p(\boldsymbol{T}|\boldsymbol{X}, f, \boldsymbol{\theta})p(\boldsymbol{X}, f, \boldsymbol{\theta}) = p(\boldsymbol{T}|\boldsymbol{X}, f, \boldsymbol{\theta})p(f|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{X}, \boldsymbol{\theta})$$

However, $\boldsymbol{T}$ is conditionally independent of $\boldsymbol{X}, \boldsymbol{\theta}$, given f (f already uses $\boldsymbol{X}, \boldsymbol{\theta}$ in its definition), thus $p(\boldsymbol{T}|\boldsymbol{X}, f, \boldsymbol{\theta}) = p(\boldsymbol{T}|f)$. Furthermore, in this scenario, our training features $\boldsymbol{X}$ are known and fixed (unlike representation learning where they are unknown and we impose a distribution on them). Similarly, we consider a point estimate of our hyperparameters $\boldsymbol{\theta}$ (a frequentist approach). It is possible to adopt a more bayesian approach, and consider a distribution over $\boldsymbol{\theta}$, however we choose not to.

As such,

$$p(\boldsymbol{T}, \boldsymbol{X}, f, \boldsymbol{\theta}) = p(\boldsymbol{T}|f)p(f|\boldsymbol{X}, \boldsymbol{\theta})$$

Keep in mind, that in the above expression, we should also take into account the noise variance σ^2, even though we do not explicitly condition on it.

Graphical model:

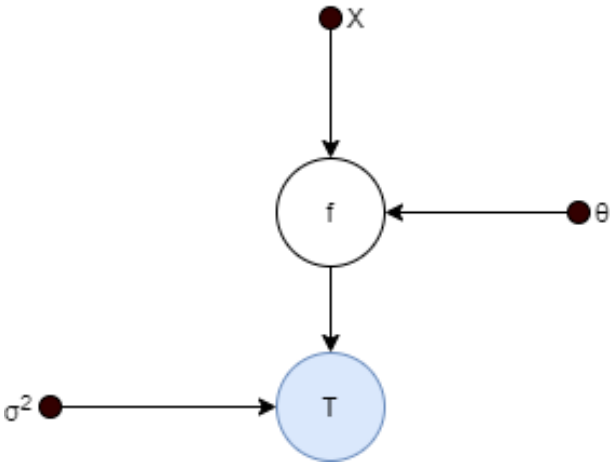


Figure 3: Simple graphical model of the above expression. Edges showcase causal like dependencies. Points indicate fixed values (no distribution), while normal vertices represent random variables (that follow a distribution). $\boldsymbol{T}$ is highlighted in blue because it is the observed variable.

Q8

Q8.1

$$p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{\theta}) = \int p(\boldsymbol{T}, f|\boldsymbol{X}, \boldsymbol{\theta})df = \int p(f|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{T}|f, \boldsymbol{X}, \boldsymbol{\theta})df = \int p(f|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{T}|f)df$$

We thus see how $p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{\theta})$ connects our prior $p(f|\boldsymbol{X}, \theta)$ to the data $p(\boldsymbol{T}|f)$.

Q8.2

The uncertainty of our functions f, filters through the marginalization (by integrating out f) via the term $p(f|\boldsymbol{X}, \boldsymbol{\theta})$, giving higher weights to more likely values of the function f. Thus, the uncertainty in f gets propagated into $p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{\theta})$.

Q8.3

It is pivotal to realize, that even though we marginalized the values of our function f, we still need to take into account that we assumed a certain value (conditionalizing on them) on the kernel hyperparameters

$\boldsymbol{\theta}$. For example, from bishop's coursebook:

$$k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \theta_0 e^{-\frac{\theta_1}{2}||\boldsymbol{x_i}-\boldsymbol{x_j}||^2} + \theta_2 + \theta_3 \boldsymbol{x_i}^T \boldsymbol{x_j}$$

Notice how different values of $\boldsymbol{\theta}$ form different kernel functions. That is, our kernel k represents a **family of kernels**, rather than a single one. In order to work with a specific kernel from this family, we need to condition on its hyperparameters.

Q9

During this practical exercise, we will do a hands on approach on linear regression. More specifically, we will attempt to approach the real regression line

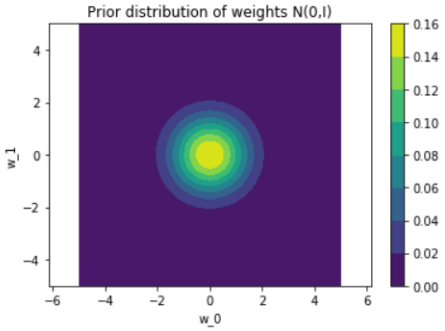
$$y = w_1 x + w_0 = 0,5x - 1.5$$

To do so, we generated data, incorporating random noise $\epsilon \sim N(0, 0.2)$, such that our data points satisfy:

$$y_i = 0.5x_i - 1.5 + \epsilon, \; x \in [-1, 1]$$

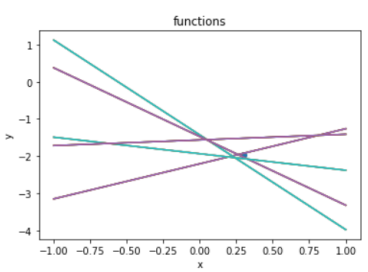
Our goal is to estimate the posterior distribution of the weights w_0, w_1, based on a number of data points (x_i, y_i), and examine the properties of this approach.

We begin by imposing a prior on our weights $p(\boldsymbol{w}) \sim N(\boldsymbol{0}, I)$:

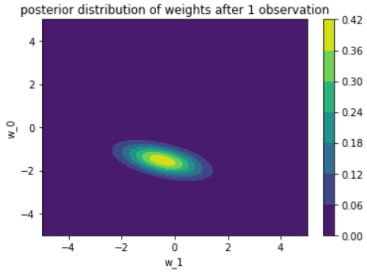


We did not center our Gaussian distribution around the real underlying weights $(-1.5, 0.5)$, since in practice we have no knowledge of them. Instead, we opted into choosing the simplest go-to Gaussian distribution.

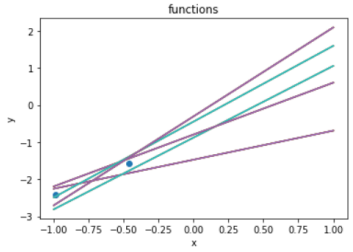
Afterwards, we use the data points from our sample to compute the posterior. We computed the posterior utilizing one, two, five, seven and 100 data points. In each case, we plotted the posterior distribution (contour) and we sampled 5 $\boldsymbol{w}$ from the posterior, plotting the corresponding lines:



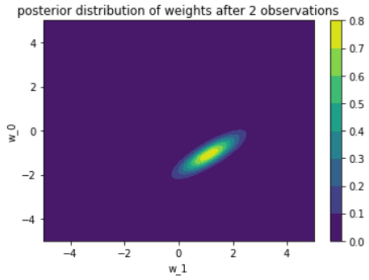
(a) Functions



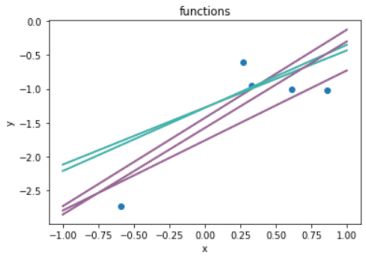
(b) Posterior



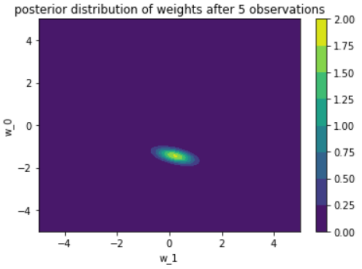
(a) Functions



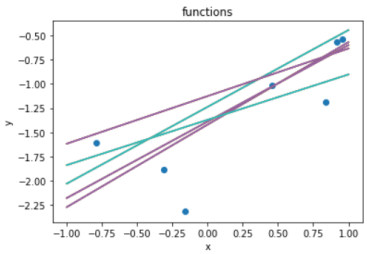
(b) Posterior



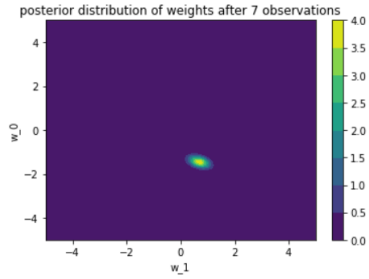
(a) Functions



(b) Posterior



(a) Functions

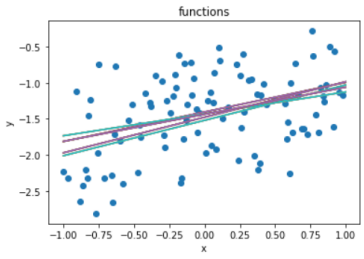


(b) Posterior

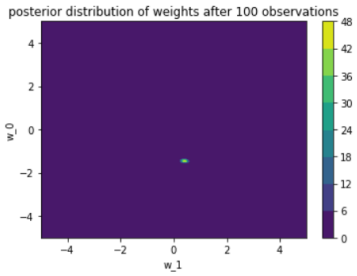
Notice two interesting properties:

- As we consider more and more data points, the distribution of the posterior becomes more centered around the real values $(-1.5, 0.5)$, while the variance decreases.
- This variance shrinkage is also evident in the models (left graphs), where we notice that the linear models vary less as more data points are being considered. For instance, when we consider 100 data points, we can clearly see how our lines are very similar, while our posterior is extremely centered.

This effect on the models and the prior makes perfect sense, since the extra data points provide us more information and helps us pinpoint the underlying weights. In the beginning, we have few data points and as such our estimate is till "vague" (relatively large variance). As we accumulate more knowledge, we are able to make our estimate more precise, narrowing the variance in our models and our posterior, while also approaching the real underlying weights that generated the data points.

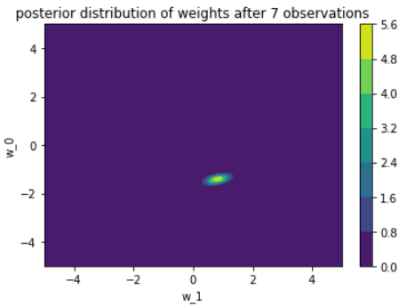


(a) Functions

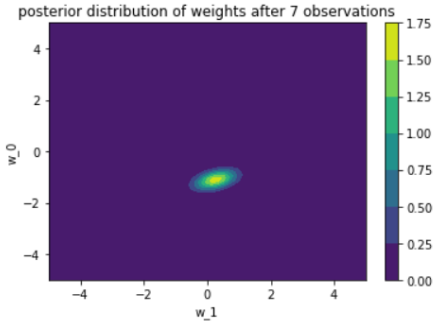


(b) Posterior

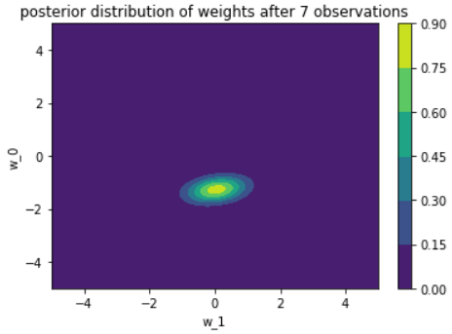
Lastly, we tinkered with the noise variance, testing the robustness of this approach for different variances, namely $\sigma^2 = 0.1, 0, 4, 0.8$:



(a) 0.1 noise variance



(b) 0.4 noise variance



(c) 0.8 noise variance

Increasing the noise variance makes the sample more noisy and as such, the underlying linear function that generated the data becomes harder to learn. This "difficulty" is translated into the posterior which has a higher variance (notice how the posterior's uncertainty-variance increases as we go from figure (a) to (c)).

Q10

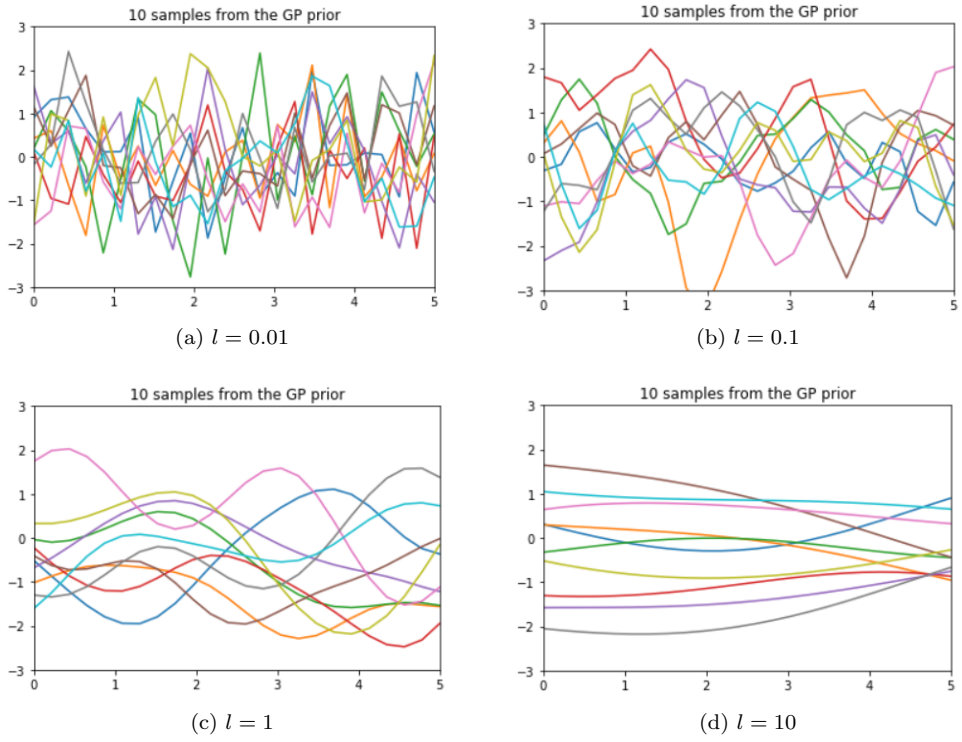
Q10.1

Our GP prior can be expressed as $p(f|\boldsymbol{X}, \boldsymbol{\theta}) = N(\mathbf{0}, k(\boldsymbol{X}, \boldsymbol{X}))$, that is, it follows a normal distribution with zero mean and a covariance matrix:

$$\Sigma(i,j) = k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \sigma_f^2 e^{-\frac{(\boldsymbol{x_i}-\boldsymbol{x_j})^T(\boldsymbol{x_i}-\boldsymbol{x_j})}{l^2}}$$

Q10.2

We experimented with four different values of l, namely $l \in \{0.01, 0.1, 1, 10\}$ and examined the qualitative differences in the distribution. Our intuition says, beforehand, that increasing l, will result in smoother curves (for our functions f), since increasing l augments the similarity between points. That is, if $l_1 < l_2$ then $k(\boldsymbol{x_i}, \boldsymbol{x_j}, l_1) < k(\boldsymbol{x_i}, \boldsymbol{x_j}, l_2)$. This means that each data point is more related with all of the other data points, limiting its degrees of freedom. This results in a smoother curve, since the f values are more related to each other. To test this claim-intuition, we will draw 10 samples from the prior and visualize them:



(a) $l = 0.01$ (b) $l = 0.1$

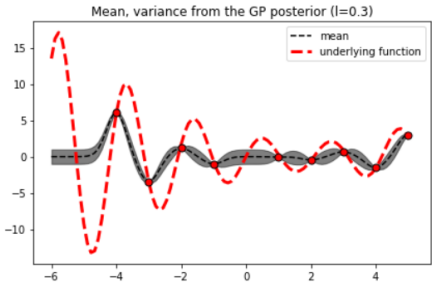
(c) $l = 1$ (d) $l = 10$

Figure 4: Visualization for four different values of l. We assumed $\sigma_f = 1$ in every case. As expected, the more we increase l, the smoother the functions f.

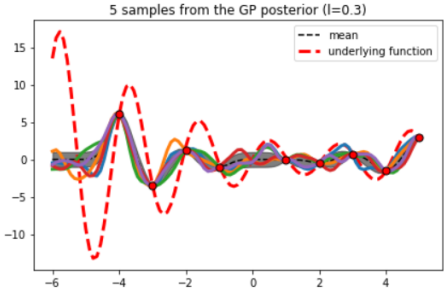
Q11

The posterior distribution reflects our updated belief, after we have observed the new data points. If we have not yet observed any data points, then this belief is the same as our prior belief, meaning that the posterior is essentially the same with our prior.

After generating the data, we drew the following posterior distribution:



(a) Posterior distribution



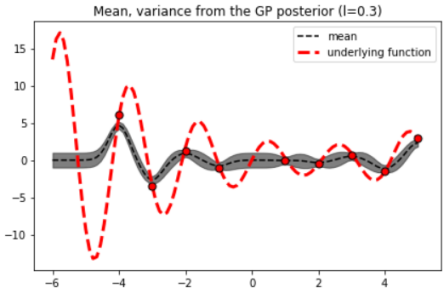
(b) Sampling from Posterior distribution

Figure 5: As showed in the legend, the red dotted line corresponds to the underlying function $f(x) = (2 + (0.5x - 1)^2)\sin(3x)$, while the black dotted line to the mean of the posterior. The shadowed margin illustrates one standard deviation.

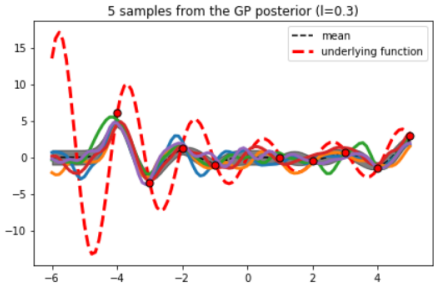
We notice how the posterior distribution is different from our prior:

- The mean of the distribution is no longer zero, but rather $\mu(x_{new}) = k(x_{new}, \boldsymbol{X})^T K(\boldsymbol{X}, \boldsymbol{X})^{-1}\boldsymbol{y}$
- The variance heavily depends on the training set, where we observe that the further away our x_{new} is from our training sample $\boldsymbol{X}$, the higher the variance of $f(x_{new})$.
- Since our GP made no noise assumption, the variance at the training data $\boldsymbol{X}$ is 0. That is, all $f^* \sim$ posterior have the same values at the training data.

The first two results are reasonable and are what we would expect and want from such a process. The third one, however, is problematic, as it is in general erroneous to assume that our data were indeed generated with zero noise. In order to deal with this, we can alter our kernel, by adding a diagonal covariance matrix (recall that adding two kernels results in a function that is also a kernel). This diagonal covariance adds "diagonal" noise, making our model more robust. The results of adding an isotropic diagonal matrix $0.3 * I$ is shown in the following graph:



(a) Noisy Posterior distribution



(b) Sampling from noisy Posterior distribution

Figure 6: Notice how both our samples and our mean do not necessarily pass through our data point (x_i, y_i). This is because the model now takes into account the added noise

Q12

If we examine the marginalization:

$$p(\boldsymbol{T}|\boldsymbol{W}) = \int p(\boldsymbol{T}|\boldsymbol{W}, \boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}$$

we see that $p(\boldsymbol{X})$ acts as a weight. That is, we give a higher weight to observations $\boldsymbol{X}$ that give a larger value to $p(\boldsymbol{X})$. In that sense, we have a *preference* for observations that are close to the peak

of $p(\boldsymbol{X}) = N(\mathbf{0}, I)$. Hence, we prefer $\boldsymbol{X}$ that lie near $\mathbf{0}$. Furthermore, we do not exhibit any sort of preference towards a certain dimension, as showcased from the unitary I matrix (the spread is the same in all directions/dimensions).

Q13

We wish to compute

$$p(\boldsymbol{Y}|\boldsymbol{W}) = \int p(\boldsymbol{Y}|\boldsymbol{W}, \boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X}$$

We know however that both the likelihood and the prior follow Gaussian distributions. Thus, $p(\boldsymbol{Y}|\boldsymbol{W})$ is fully described via its mean and covariance matrix. Recall that for every observation in $\boldsymbol{Y} = [\boldsymbol{y_1}, ..., \boldsymbol{y_N}]$, we have:

$$\boldsymbol{y_i} = \boldsymbol{W}\boldsymbol{x_i} + \boldsymbol{\mu} + \boldsymbol{\epsilon}$$

$\boldsymbol{\epsilon} \sim N(0, \sigma^2 I)$. Then, when we condition on W, we have:

$$\mu = E[\boldsymbol{y}|\boldsymbol{W}] = \boldsymbol{W}E[\boldsymbol{x}] + \boldsymbol{\mu} + E[\boldsymbol{\epsilon}] = \boldsymbol{\mu}$$

$$\Sigma = Var[\boldsymbol{y}|\boldsymbol{W}] = \boldsymbol{W}Var[\boldsymbol{x}]\boldsymbol{W}^T + Var[\boldsymbol{\mu}] + Var[\boldsymbol{\epsilon}] = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I$$

Note: we used the fact that $Var[\boldsymbol{W}\boldsymbol{x}] = E[(\boldsymbol{W}\boldsymbol{x} - \mathbf{0})(\boldsymbol{W}\boldsymbol{x} - \mathbf{0})^T] = E[\boldsymbol{W}\boldsymbol{x}\boldsymbol{x}^T\boldsymbol{W}^T] = \boldsymbol{W}E[\boldsymbol{x}\boldsymbol{x}^T]\boldsymbol{W}^T = \boldsymbol{W}Var[x]\boldsymbol{W}^T$. Thus each observation $\boldsymbol{y} \sim N(0, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I)$ and :

$$P(\boldsymbol{Y}|\boldsymbol{W}) = \prod_{i=1}^{N} N(\boldsymbol{y_i}|\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I) = \prod_{i=1}^{N} N(\boldsymbol{y_i}|\boldsymbol{\mu}, \Sigma) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{y_i}-\boldsymbol{\mu})} =$$

$$\frac{1}{(2\pi)^{\frac{Nd}{2}}|\Sigma|^{\frac{N}{2}}} e^{\sum_{i=1}^{N} -\frac{1}{2}(y_i-\boldsymbol{\mu})^T\Sigma^{-1}(y_i-\boldsymbol{\mu})}$$

Q14

Q14.1

Though the three approaches share a common goal, they differ fundamentally:

- The Maximum Likelihood approach takes into account our feature values $\boldsymbol{X}$ and picks the parameters $\boldsymbol{W}$ which optimally describes the data ($\boldsymbol{Y}$), **without taking into account** our prior beliefs on $\mathbf{W}$. It is a special case of the maximum a posteriori approach, where we assume a non-informative uniform prior $p(\boldsymbol{W})$. In the log space, this search boils down to:

$$\boldsymbol{W_{ML}} = argmin_{\boldsymbol{W}}(-\frac{1}{2}\sum_{i=1}^{N} ||\boldsymbol{t}_i - \boldsymbol{W}\boldsymbol{x}_i||^2)$$

- The Maximum a posteriori approach is more sophisticated, since it also considers our prior belief. As such, it lets us incorporate prior knowledge on our parameters, to help us aid our learning. In the log space:

$$\boldsymbol{W_{MAP}} = argmin_{\boldsymbol{W}}(-\frac{1}{2\sigma^2}\sum_{i=1}^{N} ||\boldsymbol{t}_i - \boldsymbol{W}\boldsymbol{x}_i||^2 + \log p(\boldsymbol{W}))$$

  If we assume $p(\boldsymbol{w}) = MN(\mathbf{0}, I, \tau^2 I))$, then $p(\boldsymbol{W}) = \prod_{i=1}^{d} p(\boldsymbol{W_i}) = \prod_{i=1}^{d} N(\mathbf{0}, \tau^2 I)$. Thus:

$$\boldsymbol{W_{MAP}} = argmin_{\boldsymbol{W}}(-\frac{1}{2\sigma^2}\sum_{i=1}^{N} ||\boldsymbol{t}_i - \boldsymbol{W}\boldsymbol{x}_i||^2 + \sum_{i=1}^{d} \log N(\mathbf{0}, \tau^2 I))$$

- The type two maximum likelihood approach is similar to type I, only that it seeks to maximize the **marginalized likelihood** (likelihood after we marginalize out $\boldsymbol{X}$). As we will show in the following question (Q15), this ends up (for our problem):

$$\boldsymbol{W} = argmin_{\boldsymbol{W}}(\frac{Nd}{2}\log 2\pi + \frac{N}{2}\log|\Sigma| + \sum_{i=1}^{N}\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu))$$

ML and MAP converge to the same parameter $\boldsymbol{W}$. After all, ML is MAP considering a non informative prior. Given enough data points, both of them will converge to the same value. However, the speed in which they will converge depends on how accurate our prior is. The closer our prior is to the true posterior distribution (e.g. according to KL-divergence), the faster our convergence.

If this intuitive reasoning does not suffice, recall the simple, one dimensional output y case, where after applying maximum a posteriori on linear regression we got ridge regression:

$$\boldsymbol{w}_{MAP} = argmin \underbrace{\frac{1}{2{\sigma_y}^2}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^T\boldsymbol{x_i})}_{\text{Maximum Likelihood}} + \frac{1}{2{\sigma_w}^2}||\boldsymbol{w}||_2^2$$

Notice, how, as we observe more and more data, the common term (sum of squares) becomes larger (sum of N positive numbers). As such, the penalty becomes less relevant and thus $\boldsymbol{w_{MAP}} \rightarrow \boldsymbol{w_{ML}}$.

Another difference has to do with how the model can process new information. When using MAP, we can simply iteratively compute the posterior for data point n (the new information), by using the posterior we had up until $n-1$ as our prior. Maximum Likelihood (type one and two), instead, must do all calculations from the beginning since it can not take into account previously accumulated knowledge.

Q14.2

We need to show that:

$$argmax_{\boldsymbol{W}}\frac{p[\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W}]p[\boldsymbol{W}]}{\int p[\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W}]*p[\boldsymbol{W}]d\boldsymbol{W}} = argmax_{\boldsymbol{W}}p[\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{W}]p[\boldsymbol{W}]$$

Note how

$$\int p[\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{Y}]*p[\boldsymbol{W}]d\boldsymbol{W} = \int p[\boldsymbol{Y},\boldsymbol{W}|\boldsymbol{X}]d\boldsymbol{W} = p[\boldsymbol{Y}|\boldsymbol{X}]$$

As expected, marginalizing $\boldsymbol{W}$, gives us $p[\boldsymbol{Y}|\boldsymbol{X}]$, which **no longer depends** on $\boldsymbol{W}$. Thus, we can exclude it from the argmax operator, since it does not depend on $\boldsymbol{W}$. In general:

$$argmax_x(c\cdot f(x)) = argmax_x(f(x))$$

for constant c (in regards to x).

Q14.3

$$\int p(\boldsymbol{Y}|\boldsymbol{W},\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X} = p(\boldsymbol{Y}|\boldsymbol{W})$$

Thus, when maximizing type II error:

$$\hat{\boldsymbol{W}} = argmax_{\boldsymbol{W}}p(\boldsymbol{Y}|\boldsymbol{W})$$

This approach is sensible, since it follows the same philosophy of the maximum likelihood, only that it first marginalizes out $\boldsymbol{X}$. It therefore seeks out the parameters $\boldsymbol{W}$ that are more likely to produce our target outputs $\boldsymbol{Y}$, regardless of our training features $\boldsymbol{X}$ (we have marginalized them). This approach is especially reasonable in representation learning, where we do not know the $\boldsymbol{X}$ from which our target $\boldsymbol{Y}$ derived (rather, we are aware of their distribution $p(\boldsymbol{X})$).

Q15

We can avoid using matrix norm distribution (which we are less familiar), by expressing

$$P(\boldsymbol{Y}|\boldsymbol{W}) = \prod_{i=1}^{N} N(\boldsymbol{y_i}|\boldsymbol{\mu}, \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I) = \prod_{i=1}^{N} N(\boldsymbol{y_i}|\boldsymbol{\mu}, \Sigma) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|\Sigma|}} e^{-\frac{1}{2}(\boldsymbol{y_i}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{y_i}-\boldsymbol{\mu})} =$$

$$\frac{1}{(2\pi)^{\frac{Nd}{2}}|\Sigma|^{\frac{N}{2}}} e^{\sum_{i=1}^{N} -\frac{1}{2}(y_i-\boldsymbol{\mu})^T\Sigma^{-1}(y_i-\boldsymbol{\mu})}$$

Thus

$$L(\boldsymbol{W}) = -\log p(\boldsymbol{Y}|\boldsymbol{W}) = \frac{Nd}{2}\log 2\pi + \frac{N}{2}\log|\Sigma| + \sum_{i=1}^{N}\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)$$

However, $\sum_{i=1}^{N}\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu) \in R$, hence

$$\sum_{i=1}^{N}\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu) = tr(\sum_{i=1}^{N}\frac{1}{2}(y_i-\mu)^T\Sigma^{-1}(y_i-\mu)) = \frac{N}{2} * tr(\Sigma^{-1}\frac{\sum_{i=1}^{N}(y_i-\mu)(y_i-\mu)^T}{N})$$

Let $S = \frac{\sum_{i=1}^{N}(y_i-\mu)(y_i-\mu)^T}{N}$. Then:

$$L(\boldsymbol{W}) = \frac{Nd}{2}\log 2\pi + \frac{N}{2}\log|\Sigma| + \frac{N}{2}tr(\Sigma^{-1}S)$$

Utilizing the matrix cookbook ([7]), we have:

$$K = \log|\Sigma| \Rightarrow dK = \frac{1}{|\Sigma|}|\Sigma|tr(\Sigma^{-1}d\Sigma) = tr(\Sigma^{-1}d\Sigma)$$

$$dtr(\Sigma^{-1}S) = tr(d\Sigma^{-1}S)$$

$$d\Sigma^{-1} = -\Sigma^{-1}d\Sigma\Sigma^{-1}$$

$$d\Sigma = d\boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{W}d\boldsymbol{W}^T$$

Thus:

$$dL = \frac{N}{2}(tr(\Sigma^{-1}d\Sigma) - tr(\Sigma^{-1}d\Sigma\Sigma^{-1}S)) = \frac{N}{2}(tr(\Sigma^{-1}(d\boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{W}d\boldsymbol{W}^T)) - tr(\Sigma^{-1}(d\boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{W}d\boldsymbol{W}^T)\Sigma^{-1}S))$$

However:

$$tr(\Sigma^{-1}d\boldsymbol{W}\boldsymbol{W}^T) = tr((\Sigma^{-1}d\boldsymbol{W}\boldsymbol{W}^T)^T) = tr(\boldsymbol{W}d\boldsymbol{W}^T\Sigma^{-T}) = tr(\boldsymbol{W}d\boldsymbol{W}^T\Sigma^{-1}) = tr(\Sigma^{-1}\boldsymbol{W}d\boldsymbol{W}^T)$$

Then:

$$dL = N(tr(\Sigma^{-1}d\boldsymbol{W}\boldsymbol{W}^T) - tr(\Sigma^{-1}d\boldsymbol{W}\boldsymbol{W}^T\Sigma^{-1}S)) = N(tr((\Sigma^{-1}\boldsymbol{W}^T - \Sigma^{-1}S\Sigma^{-1}\boldsymbol{W}^T)d\boldsymbol{W})$$

To summarize, we have:

$$dL = N(tr((\Sigma^{-1}\boldsymbol{W}^T - \Sigma^{-1}S\Sigma^{-1}\boldsymbol{W}^T)d\boldsymbol{W})) = N(tr((\Sigma^{-1}\boldsymbol{W}^T - \Sigma^{-1}S\Sigma^{-1}\boldsymbol{W})^T d\boldsymbol{W}))$$

$$dy = tr(A^T d\boldsymbol{X}) \Rightarrow \frac{dy}{d\boldsymbol{X}} = A$$

$$\frac{dL}{d\boldsymbol{W}} = N(\Sigma^{-1}\boldsymbol{W} - \Sigma^{-1}S\Sigma^{-1}\boldsymbol{W})$$

Finally (agrees with [3]):

$$\frac{dL(\boldsymbol{W})}{d\boldsymbol{W}} = -N(\Sigma^{-1}(\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y_i}-\boldsymbol{\mu})(\boldsymbol{y_i}-\boldsymbol{\mu})^T)\Sigma^{-1}\boldsymbol{W} - \Sigma^{-1}\boldsymbol{W})$$

$$= -N((\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I)^{-1}(\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y_i}-\boldsymbol{\mu})(\boldsymbol{y_i}-\boldsymbol{\mu})^T)(\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I)^{-1}\boldsymbol{W} - (\boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I)^{-1}\boldsymbol{W})$$

We are thus searching for the stationary points:

$$S\Sigma^{-1}\boldsymbol{W} = \boldsymbol{W},\ S = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y_i}-\boldsymbol{\mu})(\boldsymbol{y_i}-\boldsymbol{\mu})^T,\ \Sigma = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2 I$$

Q16

In order to solve this problem we essentially must perform factor analysis, and more specifically Probabilistic PCA. First, we generated the data, utilizing the expressions given in the exercise, that is:

$$\boldsymbol{Y} = f_{lin}(f_{nonlin}(\boldsymbol{x}))$$

$$\boldsymbol{x} = [0, ..., 4\pi]$$

$$|\boldsymbol{x}| = N$$

$$f_{nonlin}(\boldsymbol{x}) = [\sin x_i - x_i \cos x_i, \cos x_i + x_i \sin x_i]$$

$$f_{lin}(\boldsymbol{x'}) = \boldsymbol{x'}\boldsymbol{A}^T$$

$$\boldsymbol{A} \in R^{10x2}$$

$$a_{i,j} \sim N(0, 1)$$

In order to perform PPCA, we first need to find the optimal $\boldsymbol{W_{ML}}$, as defined in the previous question. To do so, we can simply perform gradient descent on the objective function $L(\boldsymbol{W})$.

After obtaining $\boldsymbol{W_{ML}}$, we compute:

$$\boldsymbol{\mu}_{ML} = \bar{\boldsymbol{y}}$$

$$\sigma^2_{ML} = \frac{1}{D-M}\sum_{i=M+1}^{D} \lambda_i \ , \lambda \text{ eigenvalues of S}$$

In our case, since we know that our underlying data was generated by a **linear model** of dimension 2, we have $M = 2$ (D is simply the dimensionality of our observable data, which in this case is 10). For more detailed information, please refer to Bishop and Tipping ([3]).

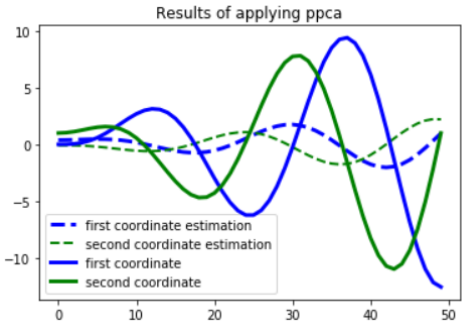
After computing all of these parameters, then **one of the possible solutions** (I will come back to this) can be derived from the distribution ([3]):

$$p(\boldsymbol{x}|\boldsymbol{y}) = N((\boldsymbol{W}^T\boldsymbol{W} + \sigma^2 I)^{-1}\boldsymbol{W}^T(\boldsymbol{y} - \boldsymbol{\mu}), \sigma^2(\boldsymbol{W}^T\boldsymbol{W} + \sigma^2 I)^{-1})$$

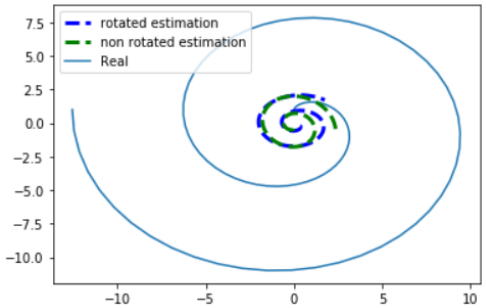
More specifically, $\boldsymbol{x} = argmax(p(\boldsymbol{x}|\boldsymbol{y})) = (\boldsymbol{W_{ML}^T}\boldsymbol{W_{ML}} + \sigma^2_{ML} I)^{-1}\boldsymbol{W_{ML}^T}(\boldsymbol{y} - \boldsymbol{\mu}_{ML})$.

Note that this will give us the $\boldsymbol{x'}$ and not the one dimensional underlying $\boldsymbol{x}$, since the calculus we performed above was based on the assumption of a linear underlying mapping (inner product of our latent variable x and weights).

After performing PPCA, we obtained:



(a) PPCA Estimation of latent variables x'



(b) Estimation illustrated as (x'_1, x'_2)

Figure 7: Results are not as accurate as one might hoped, however they do capture the general periodic principle. This discrepancy is most likely due to the loss of information that occurs when project our initial ten dimensional data ($\boldsymbol{y_i} \in R^{10}$) to two ($\boldsymbol{x'_i} \in R^2$).

The results are not exactly what we expected. Even though they manage to capture the periodic nature of our underlying $\boldsymbol{X'}$, the scale seem to differ substantially.

Lets us come back to what we expressed earlier, namely that **the solution to this problem is not unique**. We know, in fact, that our problem is invariant to the rotation of $\boldsymbol{W}_{ML}$. Indeed:

$$\hat{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{R}$$

$$\hat{\boldsymbol{W}}\hat{\boldsymbol{W}}^T = \boldsymbol{W}\boldsymbol{R}(\boldsymbol{W}\boldsymbol{R})^T = \boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^{\boldsymbol{T}}\boldsymbol{W}^{\boldsymbol{T}} = \boldsymbol{W}I\boldsymbol{W}^T = \boldsymbol{W}\boldsymbol{W}^T$$

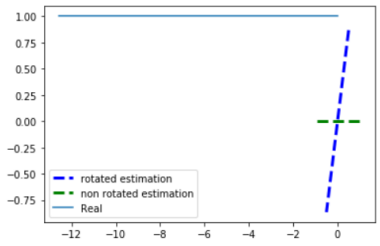
Recall how the objective function in question 15 is a function of only $\boldsymbol{W}\boldsymbol{W}^T$. Thus, any rotation of $\boldsymbol{W}_{\boldsymbol{ML}}$ assigns the same value to the objective function and is thus a solution. Due to this invariance, Gradient descent gives different $\boldsymbol{W}_{ML}$ solutions, depending on its (random) initialization.

We tested this theoretical result, by applying the following linear transformation:

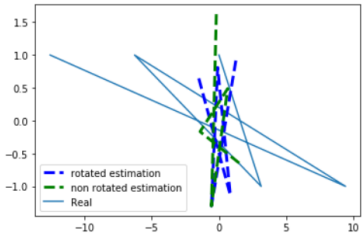
$$\begin{bmatrix} \cos\frac{\pi}{3} & -\sin\frac{\pi}{3} \\ \sin\frac{\pi}{3} & \cos\frac{\pi}{3} \end{bmatrix}$$

verifying our theoretical results.

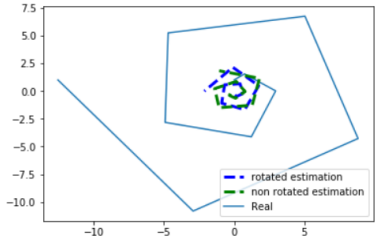
The solution seems to not depend on the number of data N, given that they are sufficiently large. The number of necessary inputs depends on the complexity of the underlying mapping. For instance, it is impossible to capture the underlying spiral nature by using only a few data points (for instance 3 points). After a certain point, however, the result is stable. By stable we mean that increasing N only makes the line smoother, without changing any of its qualitative properties (periodic, spiral).



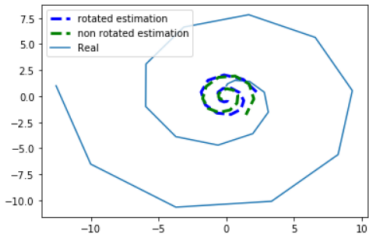
(a) $N = 2$



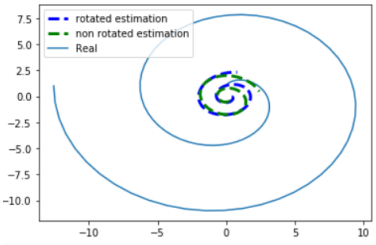
(b) $N = 5$



(c) $N = 10$



(d) $N = 20$



(e) $N = 100$

Figure 8: Results of PPCA as function of the size of the observed data - samples N

Unfortunately, there is no general rule of thumb for the amount of data N that is sufficient to uncover the underlying representation X ([6]).

Q17

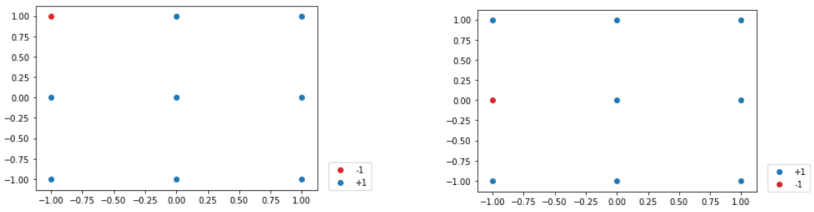
Model M_0 is the simplest model, since it has no model parameters ($\boldsymbol{\theta_0}$ does not appear in the formula). It furthermore does not depend on the data at all, since all data sets are equally likely. It is therefore a "good" model due to its simplicity. It is equivalent in imposing a non-informative prior on our weights $\boldsymbol{w}$. It is also able to explain every data set (assigning the same probability of $\frac{1}{512}$). However, it is also a "bad" model, since due to its simplicity, it fails to take into account any knowledge we may have on the structure of the space of data sets. Other more complicated models that take this into account will likely outperform this simple model. It is inflexible, since it fails to fluctuate its probability, depending on the data set D.

An interesting observation is that there is another notion of complexity that, in our case, contradicts my previous statement. Namely, even though it is simple in regards to the number of parameters, it nevertheless is capable of describing (assigning a non negligible probability) all of the data sets in $\mathcal{D}$. In my opinion, the key take away from this study is that even though there might be, in certain problem instances, contradicting notions of complexity, the Bayesian framework mitigates this issue, since it does not worry itself with model selection based on "Occam's Razor". Marginalization solves this discrepancy, since we need not base our selection on imposing a certain complexity ordering on our models.

From here on out, if not stated otherwise, when I will be referring to complexity I will be taking into account the number of independent parameters that describe this model and its **Capacity**.

Q18

Model M_0 is less flexible than M_1, since it assigns the same probability at every data set D. Furthermore, notice how M_1, takes into account only the first coordinate of the variable x^i. This means that it assigns equal probability to data sets D_a and D_b if D_b can derive from D_a, by doing simple permutations **per column** (if wee see our grid as a 3×3 matrix).



(a) First Data set (b) Second Data set

Figure 9: The two data samples are assigned the same probability, according to model M_1. Notice how D_b can be produced by D_a by doing permutations in the first column, namely swapping (-1,1) with (-1,0).

It also does not take into account the values t that correspond to grids in the middle ($x_1 = 0$). As such, it assigns equal probability to data sets that differ only along $x_1 = 0$. These observations will manifest themselves in the following evidence graphs, taking the form of plateaus. Lastly, since M_1 fails to describe data sets $D \in \mathcal{D}$ that can not be interpreted using a linear decision boundary (in $\boldsymbol{x}$ space), we expect its probability distribution to be less spread over $\mathcal{D}$ than M_0.

Q19

One can see the models M_0, M_1, M_2, M_3 as gradually more complex variations of the other (parameter wise). M_0 is the simplest model. It is therefore the least flexible and most restrictive. M_3 is traditional logistic regression, M_2 is logistic regression without a bias term, M_1 is like M_2 that only takes into account the first coordinate (logistic regression without bias, utilizing only the first coordinate of x^i).

The more parameters a model has, the more flexible it is (it varies more depending on your data set) and less restrictive. We can understand "restrictiveness" in two manors:

- The more independent parameters a model has, the more functions it can learn (increased model Capacity). Every model learned from M_2 can be learned by M_3 by setting the bias term $\theta_3^3 = 0$. Similarly, M_1 is M_2 with $\theta_2^2 = 0$. M_0 can derive from M_1, if $\theta_1^1 = 0$.
- Another way to conceptualize restrictiveness, is by deriving M_2 from M_3, after **imposing** a prior (restriction) on the bias, namely $\theta_3^3 = 0$ (similarly the rest).

Model M_1 is better suited for data sets that have the same value t^i per column, since it is a simple model that manages to explain such a relationship (it does not confuse it with any other data set, since it is unique under per column permutations). For example:

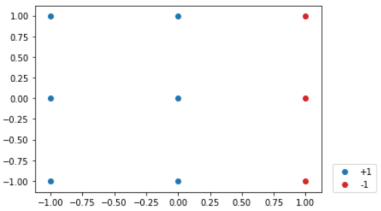


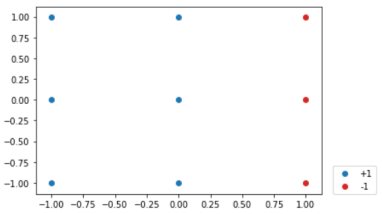
Figure 10: Notice how M_1 does not change under per column permutations. As such, M_1 is better suited in explaining such a data set, since it does confuse it with any other data set.

Model M_3 is well suited for models that show an overwhelming amount of $t = 1$ or $t = -1$, since such an "odd" data set can be explained using the bias term that M_3 has (which shifts the line from the origin).

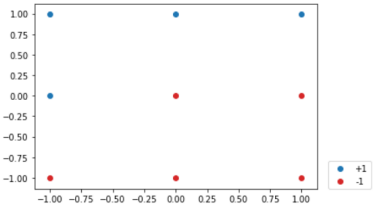
Model M_2 is something in between M_1 and M_3. It is therefore able to explain linear decision boundaries which rely both on x_1 and x_2, but yet have a generally even amount of (linearly separable) $t = 1$ and $t = -1$.

Model M_0 is able to better describe data sets that can not be sufficiently explained using a linear decision boundary.

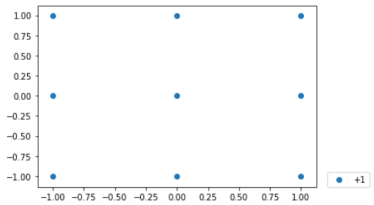
In regards with uncertainty, if we measure it as the confidence a model has in describing a give data set, then our models are less uncertain for data sets to which they assign a high probability. If by uncertainty we mean the variance of the evidence distribution, then the simpler models have lower variance, since they distribute their probability mass over fewer data sets. In this sense, M_0 showcases the highest uncertainty, while M_1 the least.



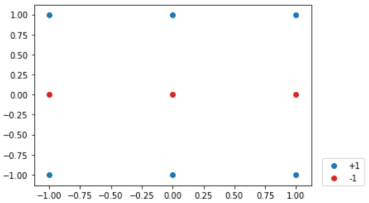
(a) Notice how M_1 does not change under per column permutations. As such, M_1 is better suited in explaining such a data set, since it does confuse it with any other data set. It is a simple model that captures such a dependency (that does not rely on x_2).



(b) Notice how we have a generally even amount of targets $t = 1$ and $t = -1$, while also having the grid observations $t^{(n)}$ be linearly separable by a line that passes through the origin.



(c) Notice how uneven the distribution of $t^{(n)}$ is. Such a data set can be well explained by a linear decision boundary that contains a bias term, to sway it away from the origin.



(d) The distribution of $t^{(n)}$ can not be sufficiently described simply using a linear decision boundary. Thus, M_0 outperforms other models.

Figure 11: Examples of data sets in which each model excels in.

Q20

Marginalization is a process in which we eliminate the presence of a parameter in a distribution, by averaging it out (weighted sum/integral utilizing its distribution as weight). This is a useful technique, since we do not need to explicitly assume a specific value for our parameter (equivalently, impose a Dirac distribution on it), but rather we consider all of its possible values, as described by its distribution. This makes the whole process more robust, as we take into account information about the whole distribution, and not only of a specific value. In some sense, it is the "heart and soul" of the bayesian mindset.

In this context, we could marginalize out θ, in order to end up with a marginalized distribution $P(D|M_i)$. This would give us the marginalized likelihood of the evidence, given a certain model.

$$P(D|M_i) = \int P(D|M_i, \boldsymbol{\theta}) P(\boldsymbol{\theta}|M_i) d\boldsymbol{\theta}$$

Q21

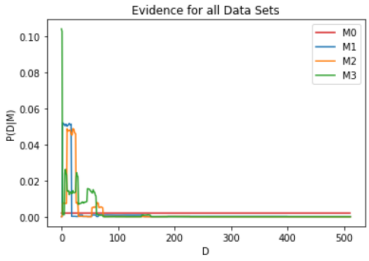
By closely inspecting the mean and the covariance matrix, we can deduce the following:

- Since the covariance matrix is diagonal, the parameters θ_i are not correlated (and thus independent, since they follow a Gaussian distribution). This tells us that our parameterization is meaningful, since the parameters are uncorrelated, meaning that they introduce more information to our models. On the contrary, if $Cov(\boldsymbol{\theta_i}, \boldsymbol{\theta_j}) = Var(\boldsymbol{\theta_i}) = \sigma^2$, then our paraters would be fully correlated $\theta_1 = \theta_2 = \theta_3$, meaning that we would effectively only utilize a single parameter.
- The mean (expected) value of each such θ_i is zero.
- The high variance $\sigma^2 = 1000$ implies that our models are "sensitive" in the $\boldsymbol{x}$ space. That is, a

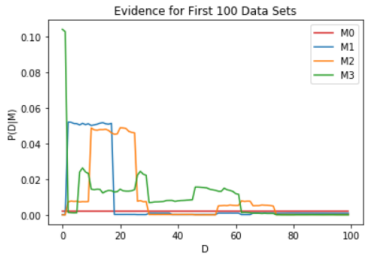
small shift in $\boldsymbol{x}$ (e.g. one coordinate changes from 0 to 1), is multiplied with a large corresponding weight (since it has large variance), which results in a large shift in probability. We can therefore deduce that this creates **sharp linear boundaries** in the $\boldsymbol{x}$ space. On the contrary, if we assumed a small variance σ^2, then our weights would be centered around their mean $\mathbf{0}$, making all our models similar to M_0 (recall how M_0 derives from the other models, when we take $\boldsymbol{\theta} = \mathbf{0}$).

Q22

I first ordered the data sets, implementing the heuristic described in the original paper's appendix ([5]). Afterwards, I drew the respective graphs (over all data sets, as well as focusing on the first 100):



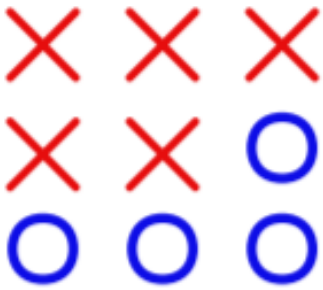
(a) Evidence probability over all data sets



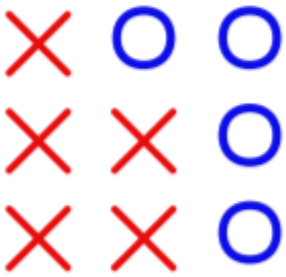
(b) Evidence probability over first 100 data sets

Figure 12: Evidence probability $p(D|M)$ of Data Sets for each model.

An interesting quality showcased in these graphs are the relatively flat regions-plateaus in the models M_1 and M_2 (they are not completely flat-horizontal due to errors induced by our Monte Carlo simulation). These regions correspond to data sets that get assigned the same probability, as we discussed in question 19. For model M_1, as we argued before, this refers to data sets exhibit a notion of per column permutation invariance (see Q19). For M_2, however, this corresponds to data sets that are invariant in respect to rotation, for example:



(a) Data set (a)



(b) Data set (b)

Figure 13: Data sets (a) and (b) have rotational symmetry, that is (b) derives from (a) if we rotate it 90 degrees counter clock wise

One can show that (via calculations) M_2 exhibits this notion of symmetry ([5]) and that it assigns the same probability to the above data sets.

The evidence graphs confirm our previous discussion, regarding the spread of the evidence distribution $P(D|M_i)$, when we examine M_1, M_2, M_3. We see how M_3, as the most complex model (with the most parameters), spreads its distribution over most data sets of the data set space (see figure 12 (b)). This agrees with what we observed in the graph illustrated in the assignment, where we see the more complex model spreading its distribution over more data sets:

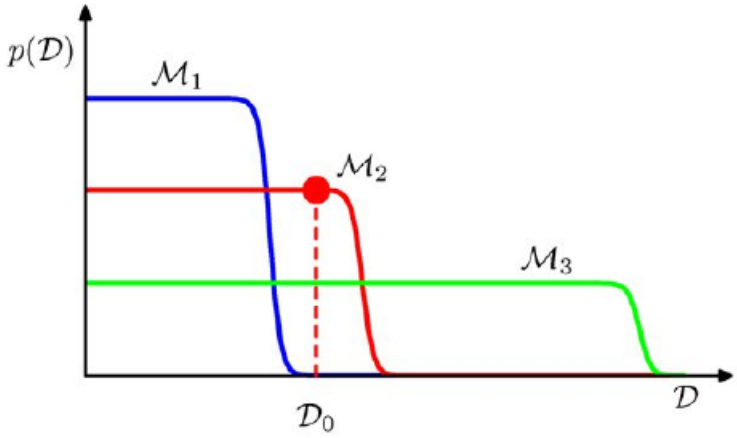


Figure 14: Bayesian interpretation of Occam's razor. More complex models are able to explain more data sets than simpler ones, at the expense of getting outperformed by simpler models to some data ([2])

However, even though M_0 is the simplest model in regards to capacity and number of parameters, it manages to spread its evidence over more data sets, making it the most complex model under this notion of complexity. As we stated earlier, this constitutes a key argument **for** adopting the Bayesian framework.

In the more practical aspect, I used Monte Carlo approximation, utilizing 10^5 samples. I also considered $p(\boldsymbol{\theta}) = N(\mathbf{0}, \sigma^2 I)$ with $\sigma^2 = 100$. These considerations were based on two reasons:

1. In order to compare my results in a more meaningful way with the ones of the original paper ([5]). I thus preferred to utilize the same prior with what was used in the paper.

2. Based on the experiments of the original paper. Namely, they used 10^8 Monte Carlo samples for a normal prior with $\sigma^2 = 100$. The more uncertain a distribution is, the more samples it requires in order to better approximate the integral using a Monte Carlo approximation. As such, approximating the integral under a prior with variance $\sigma^2 = 1000$ would require a sampling of comparable (if not greater) size. However, I opted into assuming a less uncertain distribution (as in the paper), in order to avoid the computational burden.

Afterwards, the following sums were computed, for which we know from theory:

$$\sum_{i=1}^{512} P(D_i|M_j) = 1$$

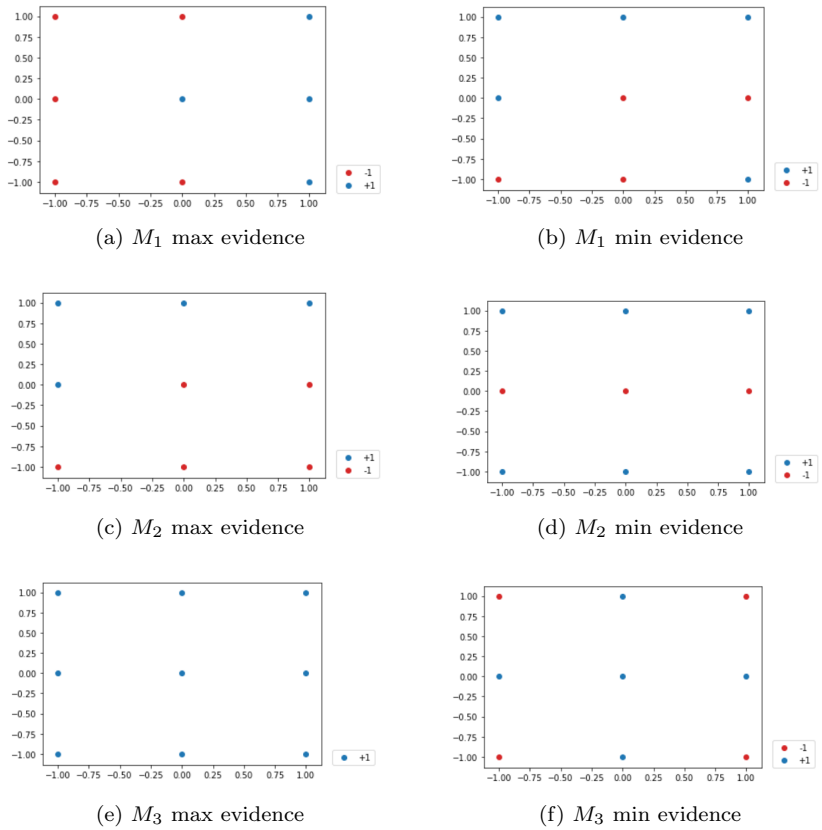
since we have considered every Data Set in our Data domain (it constitutes a probability mass function over $\mathcal{D}$). More formally:

$$1 = P(\mathcal{D}|M_j) = \sum_{i=1}^{512} P(D_i|M_j)$$

The experiments validate our beliefs, within reasonable error (since I used Monte Carlo approximation):

$$\sum_{i=1}^{512} P(D_i|M_0) = 1,\ \sum_{i=1}^{512} P(D_i|M_1) = 0.9976852,\ \sum_{i=1}^{512} P(D_i|M_2) = 0.99721955,\ \sum_{i=1}^{512} P(D_i|M_3) = 1.00487017$$

Q23



(a) M_1 max evidence (b) M_1 min evidence

(c) M_2 max evidence (d) M_2 min evidence

(e) M_3 max evidence (f) M_3 min evidence

Figure 15: Evidence probability $p(D|M)$ of Data Sets for each model.

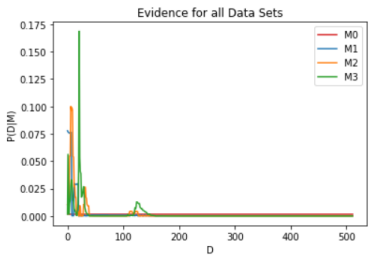
The data seem to verify our prior theoretical beliefs, as explained in question 19 (within some margin of error induced by our Monte Carlo Approximation). For instance, we see how well suited our logistic regression model M_3 is in explaining the "odd" occurrence of 1 appearing throughout the entire grid. Such a Data Set could only be deemed probable by a model with a bias term. Similarly, notice how well (high probability) M_1 assigns to a model that respects its "column invariance" (recall it only looks at the first coordinate). On the contrary, it fails to assign a meaningful probability to a Data Set that does not respect this property (15 (b)). More generally, we see how all models M_1, M_2, M_3 fail to explain data sets that do not seem to be sufficiently explained by a linear decision boundary.

Q24

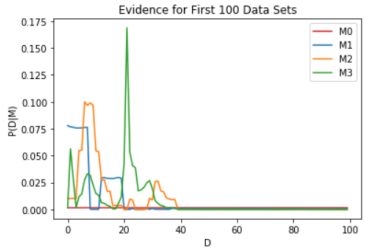
I first altered the mean of our prior, changing it to $[5, 5, 5]$. This slightly altered the distributions. In fact, we can expect the extent of this distribution change to depend on the ratio $\frac{\Delta\mu}{\sigma}$. This stems from the fact that $P(D|M_i)$ are continuous functions of our priors. As such, the similarity of our priors filter into that of our evidence $P(D|M_i)$. There are many ways to measure similarity and distances between distributions (e.g. KL divergence). However, an intuitive way of looking at it is by computing the overlap. Clearly, since our priors are normal distributions, one can measure the degree of overlap by examining the means of the distributions and their variance. High variance leads to "spread out" priors which overlap more when their means are relatively close. That is, for a fixed $\Delta\mu$, increasing the variance leads to an increase in similarity of the distribution as they overlap more. Conversely, if we keep the variance stable and pull the centers of the distributions apart from each other (increase $\Delta\mu$), then we will decrease the overlap. Thus, we expect that the similarity of the priors, and as such, the extent to which our $P(D|M)$ differ, depends on the ratio $\frac{\Delta\mu}{\sigma}$. For instance, shifting our mean by 5 (as I did

in this scenario), while having $\sigma^2 = 1$ is a lot more meaningful than if $\sigma^2 = 10^4$ (for a more rigorous mathematical argumentation, see Bhattacharyya distance [4]).

The evidence of this new experiment is as follows:



(a) Evidence probability over all data sets



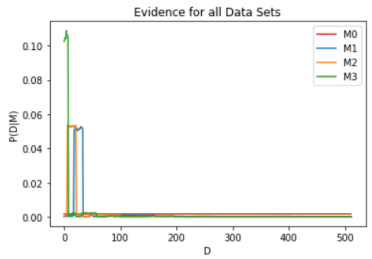
(b) Evidence probability over first 100 data sets

Figure 16: Evidence probability $p(D|M)$ of Data Sets for each model, for a prior of mean $\mu = [5, 5, 5]$.

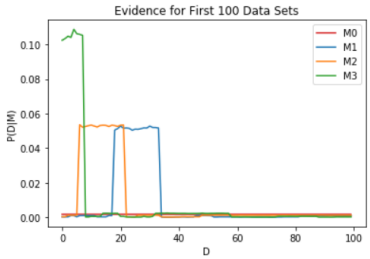
An interesting observation is that the resulting distributions contain more plateaus. This is because our parameters no longer have symmetry around $\mathbf{0}$. Due to this previously existing symmetry, a plateau not only contained data sets that were invariant in regards to some notion o symmetry that M_i exhibited, but also the flipped data set D' ($t_{D'}^{(n)} = -t_D^{(n)}$). Since this no longer holds, the plateaus are thinner. Notice now, model M_1 (blue line) no longer attributes most of its probability mass on a single plateau. The extent of this effect is again dependent on $\frac{\Delta\mu}{\sigma}$, since one can make the argument that for a large σ, shifting our mean by only 5 is not enough to substantially break this symmetry.

The distribution also changes if we keep θ constant ($\boldsymbol{\theta} = \mathbf{0}$) and tinker with σ^2, as explained earlier when addressing the sharp linear boundaries formed in the x space. Taking a prior with a small σ^2 (small θ uncertainty), makes the more complicated models M_1, M_2, M_3 resemble M_0 (since $\boldsymbol{\theta}$ is more closely centered around 0).

Finally, adding dependencies between $\boldsymbol{\theta_i}$ will result in simpler models, since we are effectively reducing the capacity of each model, imposing restrictions on our parameters. For example, if we assume that our $\boldsymbol{\theta_i}$ are completely correlated (Pearson correlation of 1), such that $\theta_0 = \theta_1 = \theta_2$, then clearly we can see that we have restricted the distributions of our $P(D|M_i)$. In fact, in this scenario, M_1, M_2, M_3 can be described by a single parameter.



(a) Evidence probability over all data sets



(b) Evidence probability over first 100 data sets

Figure 17: Evidence probability $p(D|M)$ of Data Sets for each model, for a prior of mean $\mu = [0, 0, 0]$ and correlation matrix $\sigma^2\mathbf{1}$. That is $Cov(\boldsymbol{\theta_i}, \boldsymbol{\theta_j}) = Var(\boldsymbol{\theta_i}) = \sigma^2$. In this sense, we have that $\theta_1 = \theta_2 = \theta_3$

As expected, the models are less complex, that is: more restricted and less flexible. Notice how they assign most of their probability density on fewer data sets. M_1 remains unaltered, since it only contains one parameter.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. (2006).

[2] Pawel Herman. DD2434 Machine Learning, Advanced Course, Assignment 1. (2019)

[3] Michael E. Tipping, Christopher M. Bishop. *Probabilistic Principal Component Analysis*. September 27, 1999. http://www.robots.ox.ac.uk/~cvrg/hilary2006/ppca.pdf

[4] Wikipedia, *Bhattacharyya distance*. https://en.wikipedia.org/wiki/Bhattacharyya_distance

[5] I. Murray and Z. Ghahramani. *A note on evidence and Bayesian Occam's razor*. Technical report (2005). http://mlg.eng.cam.ac.uk/zoubin/papers/05occam/occam.pdf.

[6] Mundfrom, D.J., Shaw, D.G., Ke, T.L. .*Minimum Sample Size Recommendations for Conducting Factor Analyses*. (2005)

[7] Kaare Brandt Petersen, Michael Syskind Pedersen. *The Matrix Coocbook*. (2012)