Machine Learning in the Physical World

Introduction

Carl Henrik Ek

October 21, 2021

Abstract

This is the first lab-sheet in the new unit Machine Learning and the Physical world. We are a diverse set of people taking this unit and with that comes many different views on machine learning and statistics. The aim of this sheet is to try and communicate a foundation that is needed to digest the material that comes later on. For many of you this will be things that you have seen before. Do not be scared of the sheer size of the lab sheet we have been very verbose with derivations and explanations.

The intention of this unit is to look at machine learning when applied to data that comes from real underlying physical systems. The real-world is characterised by data that is both scarce and often uncertain. Furthermore our decisions are often costly and can have irreversible consequences. However, on the flip-side many of the systems we are interested to interact with we are not the first to study. This means that we often have a wealth of knowledge to build on. Each of these characteristics places different demands on machine learning. This unit is about methods of machine learning that are suitable for these scenarios.

This lab-sheet consists of two parts, in the first we introduce probabilistic modelling through a very simple example that you have most likely seen before namely figuring out if a coin is biased[1]. What we will use this example for is try to get the theory and the philosophy behind what we are trying to do into mathematically precise terms. We will then proceed and do exactly the same thing again but now with a slightly more interesting example namely that of fitting a line to a set of data-points. Your task is to see the similarity between both of these two examples, how they are actually exactly the same. The central concept comes directly from Laplace Demon Laplace, 1814 there exists infinitely many equally valid explanations of a finite sample set. Therefore as machine learners we do not deal in truth we deal in assumptions and beliefs. What we do is to use data to find support or rejections of our belief to reach an updated belief. In short, truth is not something that you learn, truth is something that you accept and in this unit we are interested in learning from data. We aim to make use of the knowledge that has already been discovered and from this create new knowledge by using data. Importantly we want to be able to quantify what we know and as you will see this you can do with probabilities by considering them as quantification of beliefs.

The second part of the lab-sheet consists of derivations of the Gaussian identities. Most of the models that we will use in this unit are in one way or another based on the Gaussian distribution. In the second part of the lab sheet we will work through and prove the identities that we are going to use. Now these things you will not have to be able to derive but if you are mathematically inclined and want to see what all the simple structures of Gaussian's comes its quite nice to go through this but trust me after this unit you will know these formulas by heart.

1 Bernoulli Trial

We have been given the task to observe a system which has a binary outcome, you can think of the example of modelling a coin toss. We will parametrise the system using a single parameter that describes how often

[1] Yes these coins do really exists `https://izbicki.me/blog/how-to-create-an-unfair-coin-and-prove-it-with-math.html`

we get each outcome, with the coin analogy how biased the coin is. We will refer to a single outcome of the system as x and if we run the system for N iterations we will refer to all the data $\mathbf{x}$. We will first create a function to generate the data by sampling from a distribution with known parameters, the machine learning task is then to *recover* the parameters of the distribution from only the data-points.

In order to phrase this as a learning problem we need to formulate the probabilistic objects that constitutes the model of how the data have been generated. We will formulate the *likelihood* function that will express our belief in specific types of data if we know the parametrisation of the system. Say that if I think that I have an unbiased coin then seeing 100 heads in a row would be a very unlikely outcome while seeing 50 heads and 50 tails would be highly likely. We will then parametrise the *prior* which will encode our belief in the parameter value of the system, i.e. how likely do we believe the coin to be biased before we see data. Having these two objects we have completely specified our model and can generate data. Now the *inference* procedure starts where we try to reach the *posterior* distribution. The posterior distribution is the distribution that encapsulate both our prior belief about the system with the evidence in the data.

1.1 Likelihood

The first thing we need to think of is what is the likelihood function. For a binary system it makes sense to use a `Bernoulli Distribution` as likelihood function,

$$p(x|\mu)\text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x}.$$

This distribution takes a single parameter μ which completely parametrises our likelihood. This means, we have a conditional distribution and the system is completely specified by μ. If we know μ we can generate output that is similar to what the actual system does, this means we can predict how the system behaves. Now we want to use examples of the systems predictions `training data` find μ. So far we have only set the likelihood for one data-point, what about when we see lots of them? Now we will make our first assumption, we will assume that each output of the system is independent. This means that we can factorise the distribution over several trials in a simple manner,

$$p(\mathbf{x}|\mu) = \prod_{i=1}^{N} \text{Bern}(x|\mu) = \prod_{i=1}^{N} \mu^x(1-\mu)^{1-x}.$$

Now we have the likelihood for the whole data-set $\mathbf{x}$. Think a bit about what this assumption implies. It means that,

1. each data-point is independent as our likelihood function is invariant to any permutation of the data.
2. it assumes that each of the data-points are generated by the same distribution. For the coin example this means that by tossing the coin you do not change the actual coin by the act of tossing it[2].

1.2 Prior

In order to say something about the system we need to have a prior belief about what we think the parameter μ is. Now our prior knowledge comes into play, what do I know about the system? If our system is the outcome of a coin toss then we have a lot of prior knowledge, most coins that I toss are unbiased so I have quite a good idea of what I think it should be. Another way of seeing this is that I would need to see a lot of coin tosses saying something else for me to believe that a coin is not biased. If it is not a coin toss but something that I have no experience in my prior might be different. Once we have specified the prior we can just do Bayes' rule and get the posterior,

$$p(\mu|\mathbf{x}) = \frac{p(\mathbf{x}|\mu)p(\mu)}{p(\mathbf{x})}.$$

[2]this is probably an OK assumption for metal coins while for chocolate coins this assumption is too simplistic

Now comes the first tricky part, what should the distribution be for the prior? If you choose your prior or likelihood wrong[3] then this computation might not even be analytically possible to perform. So this is when we will use *conjugacy*. First lets note that the posterior distribution is proportional to the likelihood times the prior (or the joint distribution),

$$p(\mu|\mathbf{x}) \propto p(\mathbf{x}|\mu)p(\mu).$$

The second thing we will think of is that it does make sense that if I have a prior of a specific form (say a Gaussian) then I would like the posterior to be of the same form (i.e. Gaussian). So this means that we should *choose* a prior such that when multiplied with the likelihood it leads to a distribution that is of the same form of the prior. So why is this useful? This is very useful because it means we do **not** have to compute the denominator in Baye's rule, the only thing we need to do is to multiply the prior with the likelihood and then **identify** the parameters that of the distribution, we can do this as we know its form.

So how do we know which distributions are conjugate to what? Well this is something that we leave to the mathematicians most of the time, we simply exploit their results. There is a list on Wikipedia of conjugate priors here URL. Its like a match making list for distributions. For the Bernoulli distribution the conjugate prior to its only parameter μ is the `Beta` distribution,

$$\text{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1},$$

where $\Gamma(\cdot)$ is the gamma function. The role of the Gamma function is to normalise this to make sure it becomes a distribution not just any function. Now we have choosen our prior we are ready to derive the posterior.

1.3 Posterior

To get to the posterior we are going to multiply the likelihood and the prior together,

$$\begin{aligned}
p(\mu|\mathbf{x}) &\propto p(\mathbf{x}|\mu)p(\mu) && (1)\\
&= \prod_{i=1}^{N} \text{Bern}(x|\mu)\text{Beta}(\mu|a,b) && (2)\\
&= \prod_{i=1}^{N} \mu_i^x(1-\mu)^{1-x_i}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} && (3)\\
&= \mu^{\sum_i x_i}(1-\mu)^{\sum_i(1-x_i)}\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} && (4)\\
&= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{\sum_i x_i}(1-\mu)^{\sum_i(1-x_i)}\mu^{a-1}(1-\mu)^{b-1} && (5)\\
&= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{\sum_i x_i+a-1}(1-\mu)^{\sum_i(1-x_i)+b-1}. && (6)
\end{aligned}$$

Now comes the trick with conjugacy, *we know the form of the posterior*. This means we can just identify the parameters of the posterior and in this case it is trivial,

$$p(\mu|\mathbf{x}) \propto \mu^{\underbrace{\sum_i x_i + a - 1}_{a_n - 1}}(1-\mu)^{\underbrace{\sum_i (1-x_i) + b - 1}_{b_n - 1}}.$$

Now what is left is to make sure that the expression is actually a probability distribution such that it integrates to one. This means that we need to solve the following,

$$1 = Z\int p(\mu|\mathbf{x})\mathrm{d}\mu = Z\int \mu^{a_n-1}(1-\mu)^{b_n-1}\mathrm{d}\mu.$$

[3] its a bit more complicated than this but you have to believe me on this right now but we will see more of this later in the unit

In this case this is trivial as we know the normaliser of the beta distribution which means that,

$$Z = \frac{\Gamma(a_n + b_n)}{\Gamma(a_n)\Gamma(b_n)}$$

This mean that my posterior is,

$$p(\mu|\mathbf{x}) = \text{Beta}(\mu|a_n, b_n) = \frac{\Gamma\left(\sum_i x_i + a + \sum_i (1 - x_i) + b\right)}{\Gamma\left(\sum_i x_i + a\right)\Gamma\left(\sum_i (1 - x_i) + b\right)} \mu^{\sum_i x_i + a - 1} (1 - \mu)^{\sum_i (1 - x_i) + b - 1}$$

So thats it, we have the posterior and now we can fix our parameters for the prior a and b and then compute the posterior and get a_n and b_n after seeing n data-points. So lets write code that simulates one of these experiments.

1.4 Implementation

Common practice if you want to test something is to generate data from your model with known parameters, throw away the parameters and then see if you can recover the parameter. What we first then want to do is to sample a large set of binary outcomes. We can do this by using the `Binomial`[4] in `numpy`. So we start of with setting μ to 0.2 and then generate 200 values from this distribution, i.e. running the system 200 iterations or tossing a coin 200 times. Then we define our prior by setting the parameters a and b. Now we can compute our posterior, we know its form, we both derived it above, but in most cases you just write it down, that is what you will do for linear regression. Now we can plot the posterior when we see more and more examples and see what will happen. Below is the image that it generated for me,

[4] `https://en.wikipedia.org/wiki/Binomial_distribution`

Code

```
import numpy as np
from scipy.stats import beta
import matplotlib.pyplot as plt

def posterior(a,b,X):
    a_n = a + X.sum()
    b_n = b + (X.shape[0]-X.sum())

    return beta.pdf(mu_test,a_n,b_n)

# parameters to generate data
mu = 0.2
N = 100

# generate some data
X = np.random.binomial(1,mu,N)
mu_test = np.linspace(0,1,100)

# now lets define our prior
a = 10
b = 10

# p(mu) = Beta(alpha,beta)
prior_mu = beta.pdf(mu_test,a,b)

# create figure
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(111)

# plot prior
ax.plot(mu_test,prior_mu,'g')
ax.fill_between(mu_test,prior_mu,color='green',alpha=0.3)

ax.set_xlabel('$\mu$')
ax.set_ylabel('$p(\mu|\mathbf{x})$')

# lets pick a random (uniform) point from the data
# and update our assumption with this
index = np.random.permutation(X.shape[0])
for i in range(0,X.shape[0]):
    y = posterior(a,b,X[:index[i]])
    plt.plot(mu_test,y,'r',alpha=0.3)

y = posterior(a,b,X)
plt.plot(mu_test,y,'b',linewidth=4.0)

# ignore this
plt.tight_layout()
plt.savefig(path, transparent=True)
return path
```

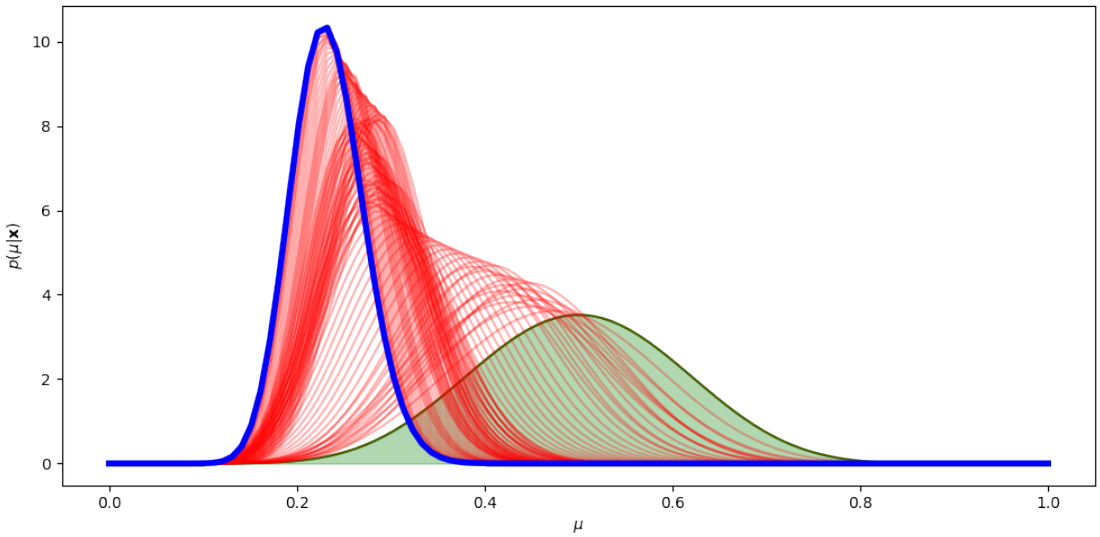

Figure 1: The green distribution is the prior distribution over μ and the red distributions are the updated belief when we see more and more data-points and the blue is the final posterior when we have seen all the points.

Now when we have built our model we can try out lots of different things, implement the tasks below and evaluate what they mean

Reflections

1. what happens if you choose a prior that is very confident at a place far from the true value?
2. what happens if you choose a prior that is very confident at the true value?
3. in the plot above we cannot see the order of the red lines, to do so create a plot where the `x-axis` is the number of data-points that you have used to compute the posterior and the `y-axis` is the distance of the posterior mean to the prior mean.
4. How much does the order of the data-points matter? Redo the plot above but with many different random permutations of the data. Now plot the `mean` and the `variance` of each iteration. What can you see?

1.5 Summary

The intention of this lab was to show the mechanism that allows us to learn from data, how we can take our prior beliefs and update them by learning from data. Even though a coin-toss might be a silly example[5] it importantly contains most of the elements that we will use to build models over the next couple of weeks. The prior is the object that contains our belief about the system, this is what we can get from domain experts, the likelihood describes how strong the evidence is for a specific model setting and the posterior gives us the updated belief once we have taken into account how well all models describe our data weighted by our beliefs.

[5] not if you are playing cricket where if you win the toss you **always** choose to bat

2 Linear Regression

Now we will take the methodology that we used in the bernoulli trial one step further and apply it to a more interesting problem. We will perform regression by fitting a function to a set of observed data. The key thing to see is that we are using exactly the same structure to learn, we formulate our beliefs and we integrate it with observed data to get an updated belief.

We observe a data-set $\mathcal{D} = \{\tilde{x}_i, y_i\}_{i=1}^N$ where we assume the following relationship between the variates,

$$y_i = f(\tilde{x}_i). \tag{7}$$

Our task is to infer the function $f(\cdot)$ from $\mathcal{D}$. To simplify things we are going to limit the hypothesis space to be only of linear functions. This means that we can write Eq 7 as,

$$y_i = w_1\tilde{x}_i + w_0 = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i = \begin{bmatrix} w_1 \\ w_0 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{x}_i \\ 1 \end{bmatrix} \tag{8}$$

where we have rewritten the input variate by appending a one so that we can write everything on matrix form. The task that we will perform in this lab is to infer the function parametrised by $\mathbf{w}$ by observing $\mathcal{D}$.

2.1 Model

Now we will make our first assumption, we will assume that the data we observed is not instantiations of the "true" underlying function but rather have been corrupted by *addative* noise. This means that we get the following model,

$$y_i = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \epsilon. \tag{9}$$

Now can we make an argument what form this noise will take? One assumption would be to say that the noise is independent of where in the input space we evaluate the function, this is called *homogenous* noise. Furthermore we could argue that we know the form of the noise, one idea would be to say that the noise is Gaussian,

$$\epsilon \sim \mathcal{N}(0, \beta^{-1}),$$

with precision β. Now this would mean is that if we could directly observe the noise we could formulate a likelihood as,

$$p(\epsilon) = \mathcal{N}(\epsilon|0, \beta^{-1}). \tag{10}$$

Now we can use Eq.9 and rewrite,

$$y_i = \mathbf{w}^{\mathrm{T}}x_i + \epsilon \tag{11}$$

$$y_i - \mathbf{w}^{\mathrm{T}}x_i = \epsilon \tag{12}$$

If we now combine this new expression of the noise with the assumption of the stochastic form in Eq.10 we get,

$$p(\epsilon) = \mathcal{N}(\epsilon|0, \beta^{-1}) \tag{13}$$

$$= \mathcal{N}(y_i - \mathbf{w}^{\mathrm{T}}x_i|0, \beta^{-1}) \tag{14}$$

$$= \mathcal{N}(y_i|\mathbf{w}^{\mathrm{T}}x_i, \beta^{-1}). \tag{15}$$

The last step can be done because we can "translate" a Gaussian distribution Figure 2.

What we have just done is to formulate a likelihood function. It describes how likely an observed output location is to have come from the a specific parametrisation of the function. Or more precisely this is the function that quantifies how much evidence a specific data point provides for a specific model.

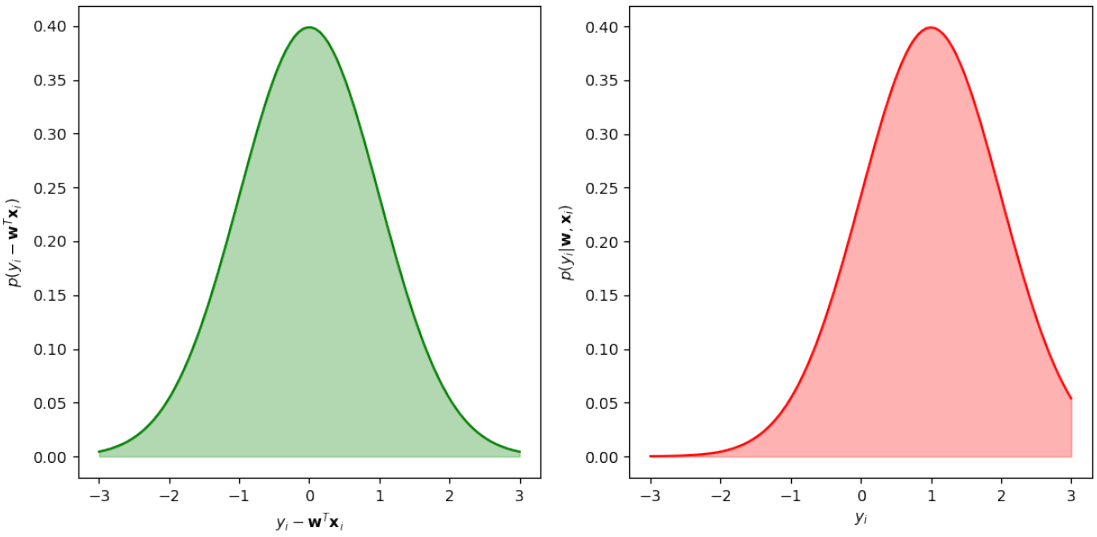

Figure 2: *Figure showing how we can re-parametrise a Gaussian by translation. In the first case we have a Gaussian with mean zero over the difference between the observed output y_i and the output of the function $\mathbf{w}^T\mathbf{x}_i$ which is the same as a Gaussian over the output y_i with mean $\mathbf{w}^T\mathbf{x}_i$ where in this case the latter equates to* 1.

Now the formulation is for a single data-point but if we assume that the noise is independent we can easily formulate the likelihood for a set of data,

$$p(\mathbf{y}|\mathbf{w}, \mathbf{X}) = \prod_{i=1}^{N} p(y_i|\mathbf{w}, \mathbf{x}_i), \tag{16}$$

where $\mathbf{X} = [x_1, \ldots, x_N]^{\mathrm{T}}$ and $\mathbf{y} = [y_1, \ldots, y_N]^{\mathrm{T}}$. Think back to the Bernoulli trial, its exactly the same thing. One important difference is how we motivated the construction of the likelihood, it came through making an assumption of the structure of the noise. Concretely we made three assumptions,

1. the noise is addative
2. the noise is Gaussian
3. the noise is independent of the input location.

These assumptions needs to be justified on a problem to problem basis. Say for example that we are measuring the vibrations in a car as a function of speed. It might be the case that the noise in the sensor measuring the vibrations also depends on the speed of the car then we should not use a homoscedastic noise model. Here to keep things simple we will only look at artificial data but later on in the module we will look at examples where we have prior knowledge.

So now we have our likelihood function and if we knew the parameters of the function $\mathbf{w}$ we would be able to generate data. However, we want to infer the these parameters from the data and to do so we need to formulate our beliefs over different parametrisations using a prior distribution.

2.2 Prior

In order to specify our prior distribution we will again use the concept of conjugacy. If we look at the likelihood in Eq.16 it is itself a Gaussian distribution[6]. We will also assume that the parameters of the noise

[6]check this for yourself by writing up the product of the individual terms

β is known. Now we can again look-up what the conjugate prior is for a Gaussian with known variance[7]] as it turns out this in itself is another Gaussian distribution. That Gaussians are conjugate to themselves is called *self-conjugacy*. Therefore to exploit conjugacy we will use a Gaussian prior for the parameters of the function such that,

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}_0, \mathbf{S}_0), \tag{17}$$

$$\mathbf{S}_0 = \lambda \mathbf{I}. \tag{18}$$

The structure of the prior covariance tells us that we assume that the two parameters w_1 and w_0 are independent with equal variance. Think about this assumption, does this make sense for a line? Well the great thing about distributions is that you can sample from it and generate the results that you can reason about. The code below will generate sample lines where $\mathbf{w}_0 = \mathbf{0}$ and $\mathbf{S}_0 = \mathbf{I}$. The resulting samples can be seen in Figure 3.

Code

```
import numpy as np
import matplotlib.pyplot as plt

def plot_line(ax, w):
    # input data
    X = np.zeros((2,2))
    X[0,0] = -5.0
    X[1,0] = 5.0
    X[:,1] = 1.0

    # because of the concatenation we have to flip the transpose
    y = w.dot(X.T)
    ax.plot(X[:,0], y)

# create prior distribution
tau = 1.0*np.eye(2)
w_0 = np.zeros((2,1))

# sample from prior
n_samples = 100

w_samp = np.random.multivariate_normal(w_0.flatten(), tau, size=n_samples)

# create plot
fig = plt.figure(figsize=(10,5))
ax = fig.add_subplot(111)

for i in range(0, w_samp.shape[0]):
    plot_line(ax, w_samp[i,:])

# save fig
plt.tight_layout()
plt.savefig(path, transparent=True)
return path
```

[7] https://en.wikipedia.org/wiki/Conjugate_prior

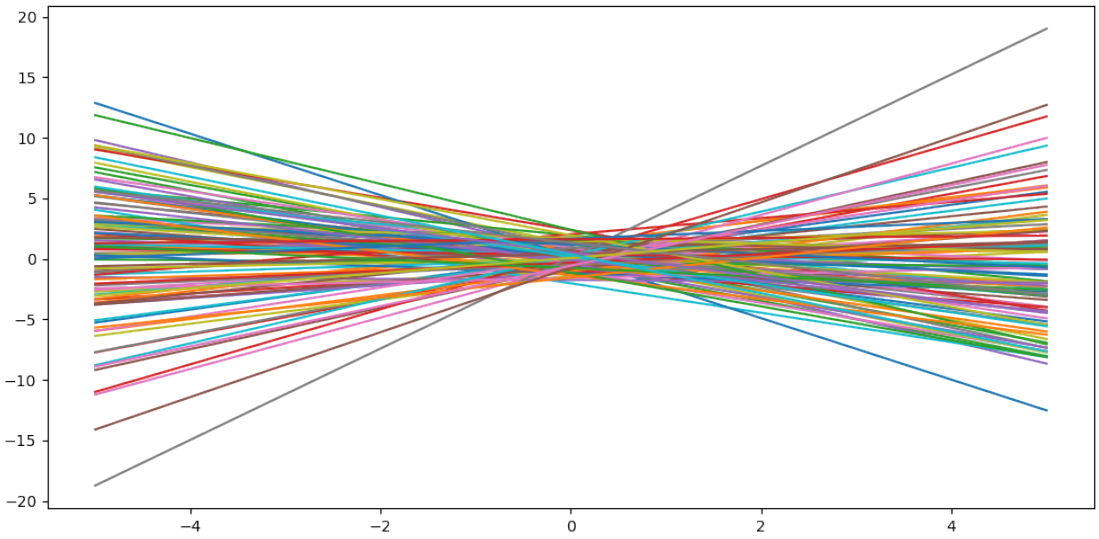

Figure 3: *Samples from the prior with the prior covariance set to identitiy and prior mean of 0.*

Reflections

1. what is the most likely line according to your prior belief?
2. what is the least likely line according to your prior belief?
3. is there any lines that have zero probability in this belief?

2.3 Posterior

Now we have encoded our prior belief and we have formulated our likelihood function and its time to formulate the posterior distribution. We will do exactly the same thing as we did in the Bernoulli trial but the math will be a bit more complicated.

To derive the posterior distribution we will use two concepts,

1. that the posterior is proportional to the likelihood times the prior
2. due to conjugacy we know that the posterior is a Gaussian

Proportionality means that,

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) \propto p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w}). \tag{19}$$

Conjugacy means that we know the form of the right-hand side,

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \mathcal{N}(\mathbf{w}|\mu(\mathbf{y}, \mathbf{X}), \sigma(\mathbf{y}, \mathbf{X})). \tag{20}$$

Now what remains to be done is to first multiply the left-hand side of Eq.19 and identify the unknown terms $\mu(\cdot, \cdot)$ and $\sigma(\cdot, \cdot)$ in Eq.20. Doing so will lead to the posterior distrubution,

$$p(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \mathcal{N}\left(\mathbf{w}|\left(\mathbf{S}_0^{-1} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\left(\mathbf{S}_0^{-1}\mathbf{w}_0 + \beta\mathbf{X}^{\mathrm{T}}\mathbf{y}\right), \left(\mathbf{S}_0^{-1} + \beta\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\right). \tag{21}$$

The full derivation of the conditional Gaussian comes in the second part of this lab-sheet Section 3 but for now lets just assume that this is actually the posterior.

Reflections

The expression above looks rather daunting at first but actually does make a lot of sense when you start looking at it. One way of making sense of the posterior is to look at some extreme scenarios, think about the following

1. what would happen if you assume a noise-free situation i.e. $\beta \to \infty$
2. what would happen if we assume a zero mean prior?
3. what happens if we do not observe any data?
4. when you observe more and more data which terms are going to dominate posterior?

Make sure that the expression makes sense and that you build an intuition.

2.4 Implementation

Once you have done the mathematical interogation of the posterior it is time to evaluate the model by generating some data and then aim to recover the parameters that generated this specific data. Again, just the same proceedure as we did in the Bernoulli trial. The first thing we need to do is to decide on some parameters, let us assume that the data have been generated as,

$$y_i = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \epsilon \tag{22}$$

$$\epsilon \sim \mathcal{N}(0, 0.3) \tag{23}$$

$$\mathbf{X} = \begin{bmatrix} -1 & 1 \\ -0.99 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix} \tag{24}$$

$$\mathbf{w} = \begin{bmatrix} -1.3 \\ 0.5 \end{bmatrix} \tag{25}$$

First visualise the prior distribution over $\mathbf{w}$. As this is a two dimensional distribution we have to show this as an image. One simple way is to create a function that samples evaluates the distribution on a grid and then creates a contour plot,

Code

```
"""
Create a contour plot of a two-dimensional normal distribution

Parameters
----------
ax : axis handle to plot
mu : mean vector 2x1
Sigma : covariance matrix 2x2

"""
from scipy.stats import multivariate_normal

def plotdistribution(ax,mu,Sigma):
    x = np.linspace(-1.5,1.5,100)
    x1p, x2p = np.meshgrid(x,x)
    pos = np.vstack((x1p.flatten(), x2p.flatten())).T

    pdf = multivariate_normal(mu.flatten(), Sigma)
    Z = pdf.pdf(pos)
    Z = Z.reshape(100,100)

    ax.contour(x1p,x2p,Z, 5, colors='r', lw=5, alpha=0.7)
    ax.set_xlabel('w_0')
    ax.set_ylabel('w_1')

    return
```

Now we will do an iterative proceedure where we pick a random point from the data-set, compute and visualise the posterior and visualise the sample functions from the same distribution. So first combine the code to plot sample with the visualisation of the distribution and then generate a loop similar to this,

Code

```
index = np.random.permutation(X.shape[0])
   for i in range(0, index.shape[0]):
       X_i = X[index,:]
       y_i = y[index]

       # compute posterior
       # visualise posterior
       # visualise samples from posterior with the data
       # print out the mean of the posterior
```

You can iterate through this and add a pause statement in your loop or you can skip the loop completely and just run the code above by setting `i` as a variable and testing for interesting values. If this works as it should you should be able to regenerate plots similar to the ones shown in Figure 4. Observe what the mean of the posterior is, in an ideal setting we should eventually recover the parameters that generated the data. Play with the parameters of the model, the noise variance, the prior and get an intuitive feeling for how everything fits together.

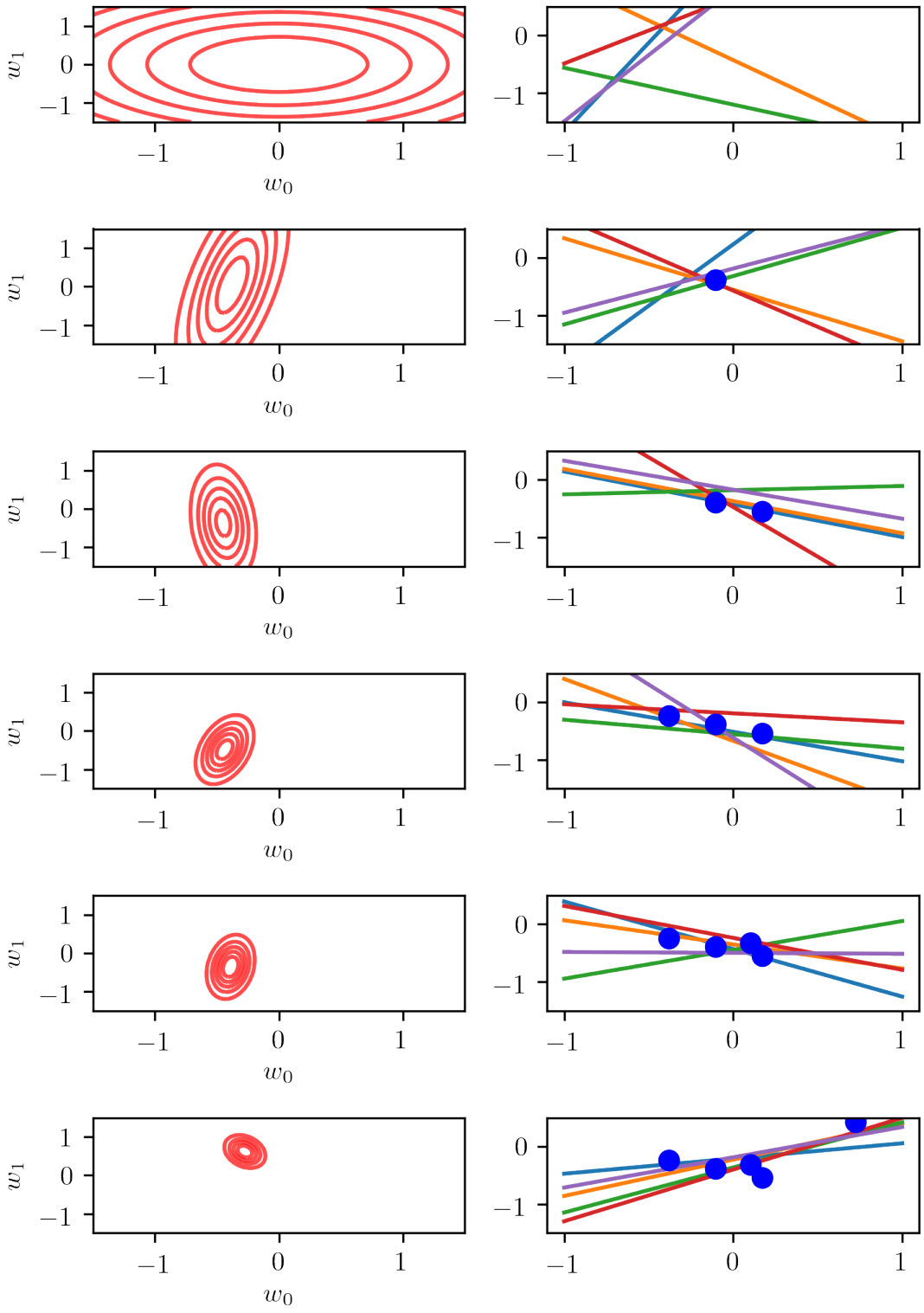

Figure 4: *The right pane shows the posterior distribution after seeing the data in the left pane where also samples from the posterior is shown. The top row of the plot we have not observed any data and the posterior reverts back to the prior.*

Reflections

1. our prior is spherical, what assumption does this encode? Does this make sense for a line?
2. with a few data-points the posterior starts quickly to look non-spherical, what does this mean? Does this make sense?
3. with many data-points the posterior becomes spherical again? Why is this? Look at Eq.21 can you see why this is the case for this data?

Take the equation for the posterior and write up each of the elements of the covariance matrix,

$$\mathbf{S}_0^{-1} + \beta \mathbf{X}^{\mathrm{T}} \mathbf{X}$$

The first term stays the same so won't change when the data increases so the changes we see comes from the second term. The second term will have the sum of the squares of each dimension of the input on the diagonal. One of the dimensions is always constant 1 which means that the off diagonal terms will be the sum of the locations, as it is constant 1 times the location. Now, this means that the off diagonal terms can be negative, while the diagonal always stays positive, therefore with increasing data, and centered data, the covariance becomes more and more spherical.

2.5 Predictive Posterior

So far we have a way of learning the parameters of the function, but the parameters is just a means to an end, what we really want is to perform predictions. This means that given a new input location $\mathbf{x}_*$ we want to have a distribution over what we believe the output location to be. Of course this distribution should take into account the training data we have used to learn the weights of the function. The way to get to this point is to *marginalise out* the parameters of the function $\mathbf{w}$. In other words generate all possible lines and weigh them with how much we believe in each of them based on what we have learned[8]. This can be done as follows,

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \int p(y_*|\mathbf{x}_*, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{y}) \mathrm{d}\mathbf{w}, \tag{26}$$

where we integrate the likelihood for a new point with the posterior distribution over the parameters. Being that both these are Gaussian we can compute this integral again in closed form which leads to the following distribution,

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(\mathbf{y}|\mathbf{m}_N^{\mathrm{T}} \mathbf{x}_*, \beta^{-1} + \mathbf{x}_*^{\mathrm{T}} \mathbf{S}_N \mathbf{x}_*\right), \tag{27}$$

where $\mathbf{m}_N$ and $\mathbf{S}_N$ is the posterior mean and variance for $\mathbf{w}$ having seen N points from the training data. As they have this nice dependency just write a new function that wraps the posterior over the weights something like,

Code

```
def predictiveposterior(m0, S0, beta, x_star, X, y):
    mN, SN = posterior(m0, S0, beta, X, y)

    m_star = mN.T.dot(x_star)
    S_star = 1.0/beta + x_star.T.dot(SN).dot(x_star)

    return m_star, S_star
```

[8] it might seem obvious but this is a really powerful statement

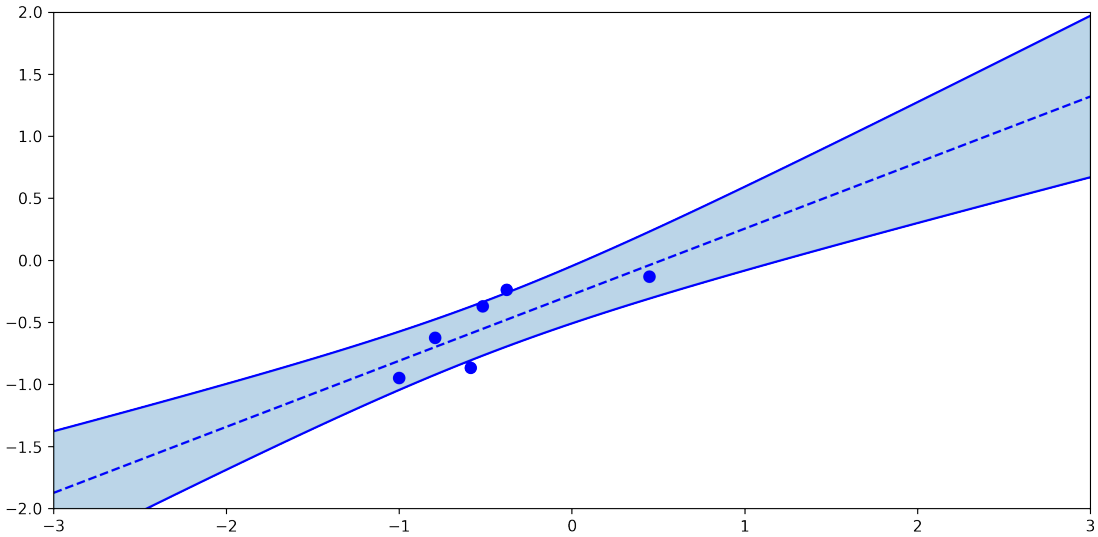

Figure 5: *The above figure shows the predictive posterior given by marginalising the posterior over the parameters that is shown in last row of Figure 4. Notice the growth of the uncertainty away from $x = 0$. Think about it as balancing a ruler on your finger, the further away from the pivot point you get the bigger the movement of the ruler.*

2.6 EXTRA Non-linear basis functions

If you are interested you can extend the linear case and work with a non-linear basis function instead. The code should be exactly the same you just first need to map the input data to a different space Z and then perform the linear regression there,

$$\Phi : \tilde{\mathcal{X}} \rightarrow \mathcal{Z} \tag{28}$$

$$y_i = \mathbf{w}^{\mathrm{T}} \Phi(\mathbf{x}_i) + \epsilon \tag{29}$$

A simple example of a mapping is to use an exponential function as,

$$\phi_d(\tilde{x}_i) = e^{-(\tilde{x}_i - b_d)^{\mathrm{T}}(\tilde{x}_i - b_d)},$$

where b_d is the "center" of the basis function. You can distribute these linearly along the input space, say that if you pick 10 basis functions you will do linear regression in a 10 dimensional space but map the result back to the original problem space where the 10 dimensional line will look like a non-linear function. It is a bit mind boggling to get your head around this one so experiment till you understand the concept. We will look at this in more detail in the next lab where we will take this to the extreme and look at a scenario where we have infinitely many basis functions.

2.7 Summary

Now you have reached the end of this lab and hopefully you manged to generate the plots and have understood the connection between the assumptions, the mathematical inference proceedure and the results. Even though it might feel like simple examples that you might have seen before conceptually this is everything but trivial. We have made a pathway from Laplace philosophical argumentation to a concrete mathematical framework. We have the tools to encode our knowledge using distributions and we have a means of quantifying our knowledge after having seen data. This is what we need to apply machine learning to physical systems. When going deeper into machine learning it is easy to get lost in specifics when doing so try to come back to the simple concepts that we worked out in this lab-sheet.

3 Gaussian Identities

The Gaussian distribution is often introduced in what can be argued the first example of machine learning namely the discovery of the planet Cerces Serio et al., 2002. Due to its desirable mathematical properties and the central limit theorem we often encounter the Gaussian distribution in machine learning. Therefore it is a good idea to have done or at least seen the derivations of the Gaussian identities. In most text-books these are never derived properly and I think there is a certain beauty to these derivations which allows you to appreciate how special the humble Gaussian distribution really is.

It is very likely that there is errors in the following derivation, if you spot any, please let me know so that I can update the document.

3.1 The Gaussian

Lets first introduce the Gaussian distribution over a variable $\mathbf{x} \in \mathbb{R}^D$,

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} e^{\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{30}$$

where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\Sigma}$ is the covariance matrix. The mean takes the same dimensionality as the $\mathbf{x}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^D$. The characteristics of the Gaussian comes from the expression in the exponential,

$$(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}). \tag{31}$$

Let us first look at the special case where $\boldsymbol{\Sigma}$ is a diagonal matrix,

3.1.1 Diagonal Covariance $\boldsymbol{\Sigma}$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \Sigma_{11} & 0 & \dots & 0 \\ 0 & \Sigma_{22} & \vdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \Sigma_{DD} \end{bmatrix} \tag{32}$$

The motivation for looking at the diagonal is because the covariance matrix appears as an inverse in the exponential. The inverse of a matrix is sometimes challenging to interpret except for in the diagonal case when the inverse is reached trivially,

$$\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} \frac{1}{\Sigma_{11}} & 0 & \dots & 0 \\ 0 & \frac{1}{\Sigma_{22}} & \vdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\Sigma_{DD}} \end{bmatrix}, \tag{33}$$

by simply inverting each diagonal element.

Now when we know the inverse of the covariance lets go back and expand the exponent in the Gaussian,

$$(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = [x_1 - \mu_1, \dots, x_D - \mu_D] \begin{bmatrix} \frac{1}{\Sigma_{11}} & 0 & \dots & 0 \\ 0 & \frac{1}{\Sigma_{22}} & \vdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \frac{1}{\Sigma_{DD}} \end{bmatrix} \begin{bmatrix} x_1 - \mu_1 \\ \vdots \\ x_D - \mu_D \end{bmatrix} \tag{34}$$

$$= (x_1 - \mu_1)\frac{1}{\Sigma_{11}}(x_1 - \mu_1) + \ldots + (x_D - \mu_D)\frac{1}{\Sigma_{DD}}(x_D - \mu_D) \tag{35}$$

$$= \sum_{i=1}^{D} \frac{1}{\Sigma_{ii}}(x_i - \mu_i)^2 \tag{36}$$

Now lets try to interpret what this actually means. First we can see that $\mathbf{x}$ only appears in a quadratic term. This means that we know that the each $(x_i - \mu_i)^2$ term will be positive. So we have a positive term multiplied by another positive term[9] which means that we have a sum of D positive terms in the exponent. Due to the minus sign in front of the exponent in Eq. 3.1 this means that the maximum value we will be able to get is when $x_i = \mu_i$ i.e. at the mean Figure 6. In effect the role of $\frac{1}{\Sigma_{ii}}$ is to scale the value of $(x_i - \mu_i)^2$, lets consider the following scenarios,

Σ_{ii} **is large** this means that the factor $\frac{1}{\Sigma_{ii}}$ is small. This means that a large deviation from the mean have a small effect. We can interpret this as we are *uncertain* about the exact value of dimension i.

Σ_{ii} **is small** now a small deviation from the mean will have a large effect on the exponent, we can interpret this as we are certain about the value of this dimension.

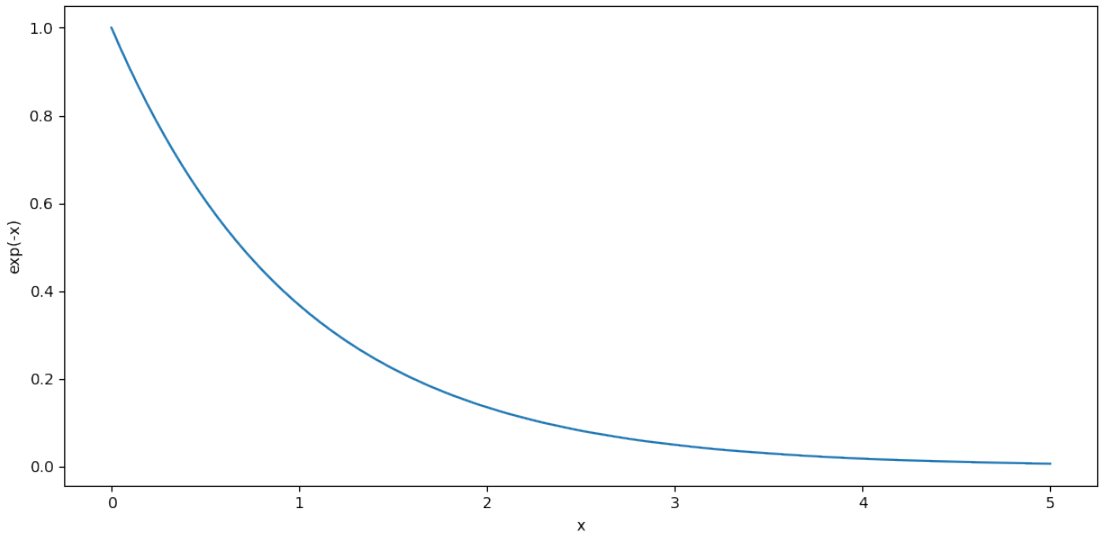

Figure 6: *The above figure shows the exponential fall-off.*

Now lets proceed to look at the case with a general covariance structure.

3.1.2 General Covariance

The covariance matrix specifies the covariance between each dimension, therefore the matrix is guaranteed to be square. This means that we can easily decompose and write it using its eigenvalue decomposition,

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}, \tag{37}$$

where $\mathbf{\Lambda}$ is a diagonal matrix and $\mathbf{U}$ is an orthonormal matrix. This decomposition allows us write the inverse covariance in a simple manner,

$$\mathbf{\Sigma}^{-1} = \left(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}\right)^{-1} = \mathbf{U}^{-\mathrm{T}}\mathbf{\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{\mathrm{T}}, \tag{38}$$

where we have used the fact that $\mathbf{U}$ is an orthonormal matrix i.e. $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$. Now we can write down the exponent of the Gaussian in the same way as for the diagonal case (but in a slightly less verbose manner),

$$(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = (\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}). \tag{39}$$

[9]The definition of variance is $\mathbb{E}\left[(x - \mathbb{E}[x])^2\right]$

Now we will split the diagonal matrix and write it as two factors as $\mathbf{\Lambda} = \mathbf{\Lambda}^{\frac{1}{2}}\mathbf{\Lambda}^{\frac{1}{2}}$,

$$(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}) \tag{40}$$

$$= (\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}) \tag{41}$$

$$= (\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu}))^{\mathrm{T}}(\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu})) \tag{42}$$

where we have used the rule of transposes $(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$.

The new quadratic form we have derived allows for some interesting interpretations. Looking at the term that involves $\mathbf{x}$ we can see that compared to the diagonal case we pre-multiply it as,

$$\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu}), \tag{43}$$

this is just a general linear mapping and you can think of it as a rotation of the basis to represent $\mathbf{x}$. Thinking of this as a mapping allows for further interpretation.

1. **View 1**

$$(\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu}))^{\mathrm{T}}\mathbf{I}(\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu})) \tag{44}$$

   We can easily add a identity matrix in the place where we initially had the co-variance matrix. This allows us to see the mapping $\mathbf{U}^{\mathrm{T}}\mathbf{\Lambda}^{-\frac{1}{2}}$ as the mapping that projects the data to a space where the co-variance is identity, i.e. where each dimension is independent and have the equal variance, this is known as a *spherical* co-variance.

2. **View 2**

$$(\mathbf{U}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}))^{\mathrm{T}}\mathbf{\Lambda}^{-1}(\mathbf{U}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu})) \tag{45}$$

   In this interpretation we keep the diagonal part of the eigendecomposition in the place of the co-variance matrix. This means that we can think of the mapping $\mathbf{U}^{\mathrm{T}}$ as the mapping that projects the data to a space where each dimension is independent but of different variance.

3.1.3 Independent Multivariate Gaussians

Quite often we will work with independent multi-variate Gaussian variables. Say that we have a set of data $\mathbf{X} \in \mathbb{R}^{N\times D}$ so N data-points that are each D dimensional. Assuming they are independent Gaussian distributions we can write the joint probability as follows,

$$p(\mathbf{X}|\boldsymbol{\mu},\mathbf{\Sigma}) = \prod_{i=1}^{N}\frac{1}{(2\pi)^{\frac{1}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}}e^{\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x}_i-\boldsymbol{\mu})} \tag{46}$$

$$= \frac{1}{(2\pi)^{\frac{N}{2}}|\mathbf{\Sigma}|^{\frac{N}{2}}}e^{\frac{1}{2}\mathrm{tr}((\mathbf{X}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu}))}, \tag{47}$$

where we have simply moved the product up into the exponent. Importantly the product $(\mathbf{X}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})$ will now generate a matrix but it is only the diagonal elements of this N matrix that corresponds to the previous expression whereby we take the `trace` operator of this matrix.

3.1.4 Precision Matrix

Often we will use and refer to the precision and the precision matrix of a Gaussian. The precision matrix is simply the inverse co-variance matrix. As you saw from the derivation above we usally have expressions where the co-variance appears as an inverse so it is sometimes easier to think of precision rather than variance.

$$\mathbf{\Lambda} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \dots & \Lambda_{1D} \\ \Lambda_{21} & \Lambda_{22} & \dots & \Lambda_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{D1} & \Lambda_{D2} & \dots & \Lambda_{DD} \end{bmatrix} = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \dots & \Sigma_{1D} \\ \Sigma_{21} & \Sigma_{22} & \dots & \Sigma_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{D1} & \Sigma_{D2} & \dots & \Sigma_{DD} \end{bmatrix}^{-1}. \tag{48}$$

A large precision therefore means that we have a small variance, i.e. we are certain of the value for this dimension.

Now when we are a bit more familiar with the characteristics of the Gaussian it is time to derive the identities of the distribution so that we can use it in our learning framework.

3.2 Gaussian Marginal

Let us begin with the Gaussian marginal distribution, to keep things reasonably compact I will derive everything for the two dimensional case but everthing translates to more dimensions. What we want to achieve is to get from a joint Gaussian distribution $p(x_1, x_2)$ to the distribution $p(x_1)$. To simplify notation we will write our two-dimensional Gaussian using a precision matrix rather than a co-variance,

$$p(x_1, x_2) = \mathcal{N}\left(\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}, \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}\right) \tag{49}$$

Our task is now to integrate out x_2 from the above and reach the marginal over x_1 as,

$$p(x_1) = \int p(x_1, x_2) \mathrm{d}x_2. \tag{50}$$

The first thing we will do is to expand the exponent of the joint distribution,

$$E = -\frac{1}{2}(x_1 - \mu_1)^{\mathrm{T}}\Lambda_{11}(x_1 - \mu_1) - \frac{1}{2}(x_1 - \mu_1)^{\mathrm{T}}\Lambda_{12}(x_2 - \mu_2) \tag{51}$$

$$- \frac{1}{2}(x_2 - \mu_2)^{\mathrm{T}}\Lambda_{21}(x_1 - \mu_1) - \frac{1}{2}(x_2 - \mu_2)^{\mathrm{T}}\Lambda_{22}(x_2 - \mu_2) \tag{52}$$

$$\tag{53}$$

Being that the marginal distribution should only be a distribution over x_1 what we will now try to do is to isolate out the terms involving x_2 from the expression above. This will be a long-windy proceedure, but its just simple algebra even though it looks like a lot.

$$E = -\frac{1}{2}(x_1^{\mathrm{T}}\Lambda_{11}x_1 + \mu_1^{\mathrm{T}}\Lambda_{11}\mu_1 - 2x_1^{\mathrm{T}}\Lambda_{11}\mu_1 \tag{54}$$

$$+ x_1^{\mathrm{T}}\Lambda_{12}x_2 - x_1^{\mathrm{T}}\Lambda_{12}\mu_2 - \mu_1^{\mathrm{T}}\Lambda_{12}x_2 + \mu_1^{\mathrm{T}}\Lambda_{12}\mu_2 \tag{55}$$

$$+ x_2^{\mathrm{T}}\Lambda_{21}x_1 - x_2^{\mathrm{T}}\Lambda_{21}\mu_1 - \mu_2^{\mathrm{T}}\Lambda_{21}x_1 + \mu_2^{\mathrm{T}}\mu_1 \tag{56}$$

$$+ x_2^{\mathrm{T}}\Lambda_{22}x_2 - \mu_2^{\mathrm{T}}\Lambda_{22}\mu_2 - 2x_2^{\mathrm{T}}\Lambda_{22}\mu_2) \tag{57}$$

$$\tag{58}$$

Now we can simplify the above expression using the fact that the covariance matrix is symmetric such that $\Lambda_{12} = \Lambda_{21}^{\mathrm{T}}$.

$$E = -\frac{1}{2}(x_2^{\mathrm{T}}\Lambda_{21}x_1 - x_2^{\mathrm{T}}\Lambda_{21}\mu_1 + x_2^{\mathrm{T}}\Lambda_{21}x_1 - x_2^{\mathrm{T}}\Lambda_{21}\mu_1 + x_2^{\mathrm{T}}\Lambda_{22}x_2 - 2x_2^{\mathrm{T}}\Lambda_{22}\mu_2 \tag{59}$$

$$+ x_1^{\mathrm{T}}\Lambda_{11}x1 + \mu_1^{\mathrm{T}}\Lambda_{11}\mu_1 + \mu_2^{\mathrm{T}}\Lambda_{22}\mu_2 - 2x_1^{\mathrm{T}}\Lambda_{11}\mu_1 - 2x_1^{\mathrm{T}}\Lambda_{12}\mu_2 + 2\mu_1^{\mathrm{T}}\Lambda_{12}\mu_2 \tag{60}$$

$$= -\frac{1}{2}(x_2^{\mathrm{T}}\Lambda_{22}x_2 + 2x_2^{\mathrm{T}}\Lambda_{21}x_1 - 2x_2^{\mathrm{T}}\Lambda_{21}\mu_1 - 2x_2^{\mathrm{T}}\Lambda_{22}\mu_2 \tag{61}$$

$$+ x_1^{\mathrm{T}}\Lambda_{11}x_1 + \mu_1^{\mathrm{T}}\Lambda_{11}\mu_1 + \mu_2^{\mathrm{T}}\Lambda_{22}\mu_2 - 2x_1^{\mathrm{T}}\Lambda_{11}\mu_1 - 2x_1^{\mathrm{T}}\Lambda_{12}\mu_2 + 2\mu_1^{\mathrm{T}}\Lambda_{12}\mu_2) \tag{62}$$

$$E = -\frac{1}{2}\left((x_2^{\mathrm{T}}\Lambda_{22}x_2 - 2x_2^{\mathrm{T}}\Lambda_{22}(\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1)) \right. \tag{63}$$

$$\left. - 2x_1^{\mathrm{T}}\Lambda_{12}\mu_2 + 2\mu_1^{\mathrm{T}}\Lambda_{12}\mu_2 + \mu_2^{\mathrm{T}}\Lambda_{22}\mu_2 + x_1^{\mathrm{T}}\Lambda_{11}x_1 - 2x_1^{\mathrm{T}}\Lambda_{11}\mu_1 + \mu_1^{\mathrm{T}}\Lambda_{11}\mu_1\right) \tag{64}$$

$$= -\frac{1}{2}\left(x_2 - (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1))\right)^{\mathrm{T}}\Lambda_{22}\left(x_2 - (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1))\right) \tag{65}$$

$$+ \underbrace{\frac{1}{2}\left(x_1^{\mathrm{T}}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}x_1 - 2x_1^{\mathrm{T}}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{12}\mu_1 + \mu_1^{\mathrm{T}}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\mu_1\right)}_{A} \tag{66}$$

$$- \underbrace{\frac{1}{2}\left(x_1^{\mathrm{T}}\Lambda_{11}x_1 - 2x_1^{\mathrm{T}}\Lambda_{11}\mu_1 + \mu_1\Sigma_{11}\mu_1\right)}_{B} \tag{67}$$

From the expansion that we have derived we can see that we have three terms in the exponent. Importantly the last two terms do not include x_2 so we will now deal with them one by one. Our aim is to re-write them as quadratic expressions.

$$A = \frac{1}{2}\left(x_1^{\mathrm{T}}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}x_1 - 2x_1^{\mathrm{T}}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{12}\mu_1 + \mu_1^{\mathrm{T}}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}\mu_1\right) \tag{68}$$

$$= \frac{1}{2}\left((x_1-\mu_1)^{\mathrm{T}}(\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21})(x_1-\mu_1)\right), \tag{69}$$

where we have again used the fact that a co-variance matrix is symmetric such that $\Lambda_{12} = \Lambda_{21}^{\mathrm{T}}$.

$$B = \frac{1}{2}\left(x_1^{\mathrm{T}}\Lambda_{11}x_1 - 2x_1^{\mathrm{T}}\Lambda_{11}\mu_1 + \mu_1\Sigma_{11}\mu_1\right) = \frac{1}{2}\left((x_1-\mu_1)^{\mathrm{T}}\Lambda_{11}(x_1-\mu_1)\right) \tag{70}$$

Importantly the two quadratic expressions we have written are both in terms of $x_1 - \mu_1$ so we can now but together the two expressions into one,

$$A - B = \frac{1}{2}\left((x_1-\mu_1)^{\mathrm{T}}(\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21} - \Lambda_{11})(x_1-\mu_1)\right). \tag{71}$$

Now lets take a step back and look at what we are aiming for. We have re-written the exponent as two separate terms, one term including x_2 and one which only includes x_1,

$$p(x_1, x_2) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}e^{E_1}e^{E_2}, \tag{72}$$

where,

$$E_1 = -\frac{1}{2}\left(x_2 - (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1))\right)^{\mathrm{T}}\Lambda_{22}\left(x_2 - (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1))\right) \tag{73}$$

$$E_2 = -\frac{1}{2}\left((x_1-\mu_1)^{\mathrm{T}}(\Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21})(x_1-\mu_1).\right. \tag{74}$$

If we now go back to the formulation of the marginalisation we want to do we can exploit this structure.

$$p(x_1) = \int p(x_1, x_2)\mathrm{d}x_2 = \int \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}e^{E_1}e^{E_2}\mathrm{d}x_2 \tag{75}$$

$$= \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}e^{E_2}\int e^{E_1}\mathrm{d}x_2 \tag{76}$$

The above is true because the way we have re-written the exponent such that x_2 only appears in E_1.

We will now proceed to integrate out x_2 from the first term in the exponent. Rather than doing this brute-force we can actually be a bit clever. If we look at E_1 we can see that it is also a quadratic form over x_2 just as the normal Gaussian,

$$E_1 = -\frac{1}{2}\left(x_2 - (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1))\right)^{\mathrm{T}}\Lambda_{22}\left(x_2 - (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1))\right), \tag{77}$$

where we can think of $(\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1 - \mu_1))$ as the mean and Λ_{22} as the precision matrix. As we know that a Gaussian integrates to 1 and that the term in front of the exponential does not contain x_2 the following relationship needs to hold,

$$\int \frac{1}{(2\pi)^{\frac{D_2}{2}}|\Lambda_{22}^{-1}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_2-\tilde{\mu}_2)^{\mathrm{T}}\Lambda_{22}(x_2-\tilde{\mu}_2)} \mathrm{d}x_2 = 1 \tag{78}$$

$$\int e^{-\frac{1}{2}(x_2-\tilde{\mu}_2)^{\mathrm{T}}\Lambda_{22}(x_2-\tilde{\mu}_2)} \mathrm{d}x_2 = (2\pi)^{\frac{D_2}{2}}|\Lambda_{22}^{-1}|^{\frac{1}{2}}, \tag{79}$$

where $\tilde{\mu_2} = (\mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(x_1 - \mu_1))$ and D_2 is the dimensionality of x_2. This means that we can re-write the expression as,

$$p(x_1) = (2\pi)^{\frac{D_2}{2}}|\Lambda_{22}^{-1}|^{\frac{1}{2}} \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} e^{E_2} \tag{80}$$

$$= \frac{1}{(2\pi)^{\frac{D-D2}{2}}|\Lambda_{22}^{-1}|^{-\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} e^{E_2}. \tag{81}$$

Now we want to re-write the expression that involves the determinant that will be the normaliser of our distribution. To do so we will use a set of rules,

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = |A||D - CA^{-1}B| \tag{82}$$

$$\Rightarrow |\Sigma| = |\Sigma_{11}||\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}|\Sigma_{12}| \tag{83}$$

Now we need the final piece of the puzzle and that is we need to re-write the precision matrix Λ_{22} in terms of a co-variance term. In order to do so we will have to use what is called a Schur complement. If you haven't seen this, or it was a long time ago I will show what they are in Section 3.4.1.

The Schur complement of Λ_{22} is,

$$\Lambda_{22}^{-1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \tag{84}$$

This means that we can simplify the terms that involves the derminants as follows,

$$|\Lambda_{22}^{-1}|^{-\frac{1}{2}}|\Sigma|^{\frac{1}{2}} = |\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}|^{-\frac{1}{2}}|\Sigma_{11}|^{\frac{1}{2}}|\Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}|^{\frac{1}{2}} \tag{85}$$

$$= |\Sigma_{11}|^{\frac{1}{2}}. \tag{86}$$

Now we can write down the full expression of the marginal distribution as follows,

$$p(x_1) = \frac{1}{(2\pi)^{\frac{D_1}{2}}|\Sigma_{11}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_1-\mu_1)^{\mathrm{T}}(\Lambda_{11}-\Lambda_{12}\Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}(x_1-\mu_1)} \tag{87}$$

where we have used the fact that $D = D_1 + D_2$. The final step is now to re-write the expression in precision matrices in terms of a co-variance matrix. Again we will use the Schur complement to do this,

$$\Sigma_{11}^{-1} = \Lambda_{11} - \Lambda_{12}\Lambda_{22}^{-1}\Lambda_{21}, \tag{88}$$

which leads to the final expression,

$$p(x_1) = \frac{1}{(2\pi)^{\frac{D_1}{2}}|\Sigma_{11}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_1-\mu_1)^{\mathrm{T}}\Sigma_{11}^{-1}(x_1-\mu_1)}. \tag{89}$$

The above is the marginal distribution of a Gaussian. As you can see it is actually really simple to reach it, even though the proof was rather long, you simply pick the submatrix from the mean vector and the covariance matrix that corresponds to the variables that you want and this leads to the marginal distribution.

Now when we have shown the marginal it is time to move further to look at what the conditional distribution of the Gaussian is.

3.3 Conditional Gaussian

Now when we have computed the marginal Gaussian distribution we will use this result to compute the conditional Gaussian distribution. To do so we will start of with the product rule,

$$p(x_1, x_2) = p(x_1|x_2)p(x_2). \tag{90}$$

As we have already proved what $p(x_2)$ is and because we define the joint distribution we already know two out of three components above. Let us start by writing up the joint distribution,

$$p(x_1, x_2) = \mathcal{N}\left(\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right) \tag{91}$$

$$\propto e^{-\frac{1}{2}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}} \tag{92}$$

our task is now to factor out the marginal,

$$p(x_2) = \frac{1}{(2\pi)^{\frac{D_2}{2}}|\Sigma_{22}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_2-\mu_2)^{\mathrm{T}}\Sigma_{22}^{-1}(x_2-\mu_2)} \tag{93}$$

$$\propto e^{-\frac{1}{2}(x_2-\mu_2)^{\mathrm{T}}\Sigma_{22}^{-1}(x_2-\mu_2)}. \tag{94}$$

We will now use Schur complements to achieve this by expression the inverse of the full co-variance matrix decomposed in such a way that Σ_{22}^{-1} gets isolated. First lets look at the exponent of the joint distribution,

$$E = -\frac{1}{2}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}^{-1}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \tag{95}$$

$$= -\frac{1}{2}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} I & 0 \\ \Sigma_{22}^{-1}\Sigma_{21} & I \end{bmatrix}\begin{bmatrix} (\Sigma/\Sigma_{22})^{-1} & 0 \\ 0 & \Sigma_{22}^{-1} \end{bmatrix}\begin{bmatrix} I & -\Sigma_{12}\Sigma_{22}^{-1} \\ 0 & I \end{bmatrix}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \tag{96}$$

$$= -\frac{1}{2}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix}^{\mathrm{T}}\begin{bmatrix} (\Sigma/\Sigma_{22})^{-1} & -(\Sigma/\Sigma_{22})^{-1}\Sigma_{12}\Sigma_{22}^{-1} \\ -\Sigma_{21}\Sigma_{22}^{-1}(\Sigma/\Sigma_{22})^{-1} & \Sigma_{22}^{-1} \end{bmatrix}^{-1}\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \tag{97}$$

$$= -\frac{1}{2}(x - (\mu_1 + \Sigma_{21}\Sigma_{22}^{-1}(x_2 - \mu_2)))^{\mathrm{T}}(\Sigma/\Sigma_{22})^{-1}(x - (\mu_1 + \Sigma_{21}\Sigma_{22}^{-1}(x_2 - \mu_2))) \tag{98}$$

$$\underbrace{-\frac{1}{2}(x_2 - \mu_2)^{\mathrm{T}}\Sigma_{22}^{-1}(x_2 - \mu_2)}_{E_2}. \tag{99}$$

The last term in the expression above E_2 is exactly the exponent of the marginal distribution of x_2. Due to the product rule this means that we now know that the remaining term needs to be the exponent of the conditional Gaussian distribution. Therefore we only need to identify the parameters of a Gaussian to write down the posterior,

$$p(x_1|x_2) \propto e^{-\frac{1}{2}(x-\underbrace{(\mu_1 + \Sigma_{21}\Sigma_{22}^{-1}(x_2 - \mu_2))}_{\text{mean}})^{\mathrm{T}}(\underbrace{\Sigma/\Sigma_{22}}_{\text{covariance}})^{-1}(x-(\mu_1+\Sigma_{21}\Sigma_{22}^{-1}(x_2-\mu_2)))}, \tag{100}$$

from which we get the conditional distribution as,

$$p(x_1|x_2) = \mathcal{N}(x_1|\mu_1 + \Sigma_{21}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}), \tag{101}$$

where we have written out the Schur complement.

3.4 Appendix

3.4.1 Schur Complement

Inverting matrices can be a very tedious thing and is sadly something that we have to do alot when we work with Gaussians. There is one tool though that will help us immensely and that is Schur complements. I will

here derive what a Schur complement is as we will use it several times when we prove the Identities. First lets motivate the complement. If we have a block diagonal matrix the inverse of the matrix can be computed by taking the inverse of each block in turn,

$$\begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}^{-1} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix}. \tag{102}$$

Now lets say that we have a general matrix M,

$$M = \begin{bmatrix} E & F \\ G & H \end{bmatrix} \tag{103}$$

that we want to find the inverse of. Importantly we want to exploit the block structure of the inverse we have written above. The trick that we will do is to re-write our matrix M as a decomposition that allows us to diagonalise it into blocks. We will now try to come up with a decomposition that will "clear out" off-diagonal blocks in turn for us.

$$\begin{bmatrix} I & A \\ 0 & I \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} E + AG & F + AH \\ G & H \end{bmatrix} \tag{104}$$

Now the sub-matrix A in the expression above is for me to choose and as I want to make the product block-diagonal I will choose it such that it "clears out" the sub-matrix $F - AH$ therefore we choose it to be,

$$\begin{aligned} F + AH &= 0 && (105)\\ \Rightarrow AH &= -F && (106)\\ \Rightarrow AHH^{-1} &= -FH^{-1} && (107)\\ \Rightarrow A &= -FH^{-1}. && (108) \end{aligned}$$

Now lets compute the resulting matrix after we pre-multiply[10],

$$\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} E + FH^{-1}G & 0 \\ G & H \end{bmatrix} \tag{109}$$

So we have managed to clear out the sub-matrix that is above the diagonal, next we will apply the same idea to clear out the sub-matrix below the diagonal G. In order to do this we will rather post-multiply the matrix,

$$\begin{bmatrix} E + FH^{-1}G & 0 \\ G & H \end{bmatrix} \begin{bmatrix} I & 0 \\ B & I \end{bmatrix} = \begin{bmatrix} E - FH^{-1}G & 0 \\ G - HB & H \end{bmatrix}. \tag{110}$$

Now we want to choose our matrix B such that the sub-matrix $G - HB$ disapears,

$$\begin{aligned} G - HB &= 0 && (111)\\ \Rightarrow HB &= -G && (112)\\ \Rightarrow H^{-1}HB &= -H^{-1}G && (113)\\ \Rightarrow B &= -H^{-1}G. && (114) \end{aligned}$$

If we post-multiply with our new matrix we get the following results,

$$\begin{bmatrix} E + FH^{-1}G & 0 \\ G & H \end{bmatrix} \begin{bmatrix} I & 0 \\ B & I \end{bmatrix} = \begin{bmatrix} E - FH^{-1}G & 0 \\ 0 & H \end{bmatrix}, \tag{115}$$

[10]Remember that matrix multiplication is not commutative

and we have reached a block-diagonal matrix. Now let us write down the full decomposition,

$$\underbrace{\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix}}_{A}\underbrace{\begin{bmatrix} E & F \\ G & H \end{bmatrix}}_{M}\underbrace{\begin{bmatrix} I & 0 \\ -H^{-1}G & I \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} E - FH^{-1}G & 0 \\ 0 & H \end{bmatrix}}_{C}. \tag{116}$$

This means that we have a decomposition $AMB = C$ and now we are interested in computing the inverse of M in a manner so that we exploit the block-diagonal structure.

$$(AMB)^{-1} = C^{-1} \tag{117}$$
$$B^{-1}M^{-1}A^{-1} = C^{-1} \tag{118}$$
$$\underbrace{BB^{-1}}_{=I}M^{-1}A^{-1} = BC^{-1} \tag{119}$$
$$M^{-1}\underbrace{A^{-1}A}_{=I} = BC^{-1}A \tag{120}$$
$$M^{-1} = BC^{-1}A \tag{121}$$
$$\tag{122}$$

The above means that we can write the inverse of matrix A as a product of two matrices A and B and the inverse of a block-diagonal matrix C this leads to following expression of the inverse,

$$M^{-1} = \begin{bmatrix} I & 0 \\ -HG^{-1} & I \end{bmatrix}\begin{bmatrix} (E - FH^{-1}G)^{-1} & 0 \\ 0 & H^{-1} \end{bmatrix}\begin{bmatrix} I & -FH^{-1} \\ 0 & I \end{bmatrix} \tag{123}$$
$$= \begin{bmatrix} (E - FH^{-1}G)^{-1} & -(E - FH^{-1}G)^{-1}FH^{-1} \\ -H^{-1}G(E - FH^{-1}G)^{-1} & H^{-1} + H^{-1}G(E - FH^{-1}G)^{-1}FH^{-1} \end{bmatrix} \tag{124}$$

This means we have now expressed the inverse of the matrix M as the inverse of a block-diagonal matrix. In the case above we have blocked out the submatrix H, which sits alone in the expression of C we will therefore refer to $E - FH^{-1}G$ as the Schur complement of M with respect to H and its often indicated by M/H. If this seems unclear look at how it is used when we compute the conditional gaussian distribution and hopefully it will all come together.

References

Laplace, Pierre Simon (1814). *A philosophical essay on probabilities.*
Serio, G. Foderà et al. (2002). *Giuseppe Piazzi and the Discovery of Ceres.*