Machine Learning in the Physical World

Evidence

Carl Henrik Ek

October 14, 2020

Abstract

Last week we looked at models, we have gone from simple models to figure out if a coin is biased or not to much more complicated infinite dimensional objects to place distributions over the space of functions. We followed the same procedure in each of these tasks, formulate a likelihood and a prior and try to get to the posterior. In this worksheet we are not going to introduce a new model but instead look at the one part of the probabilistic framework that we have so far ignored, the *evidence* or *marginal likelihood*. Hopefully after doing this lab you will see that this object is not just an annoying object that we try to do our best to avoid working with, no its actually the most important of all the probabilistic objects that we have in our arsenal. So lets get aquainted with the evidence.

In this lab we will look at the role the evidence or the marginal likelihood plays in machine learning. We will follow the excellent paper Murray et al., 2005. The evidence is the probability distribution that is left when we have integrated out everything. Say that we have observed a set of data $\mathcal{D}$ and we have built up a model of this parameterised by a set of parameter θ the evidence is,

$$p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)\mathrm{d}\theta. \tag{1}$$

This means that the evidence is the distribution over the data space that is created if we average *all" of the possible hypothesis that we have relative to how probable we think that they are. So lets try and get an intuition for this. Lets say that we have a modelling scenario where we have a Gaussian model, and we do not know the mean nor the variance, now to make it simple we have an hypothesis space that only includes three different possible settings of the parameters. This would mean the evidence is an average over these three Gaussians as,

$$p(\mathcal{D}) = \sum_{\boldsymbol{\theta}} p(\mathcal{D}|\theta)p(\theta). \tag{2}$$

In the code below I have written up an example of this and the result can be seen in Figure 1.

Code

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

x = np.linspace(-6,6,200)
pdf1 = norm.pdf(x,0,1)
pdf2 = norm.pdf(x,1,3)
pdf3 = norm.pdf(x,-2.5,0.5)

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

ax.plot(x,pdf1,color='r',alpha=0.5)
ax.fill_between(x,pdf1,color='r',alpha=0.3)
ax.plot(x,pdf2,color='g',alpha=0.5)
ax.fill_between(x,pdf2,color='g',alpha=0.3)
ax.plot(x,pdf3,color='b',alpha=0.5)
ax.fill_between(x,pdf3,color='b',alpha=0.3)

pdf4 = 0.3*pdf1 + 0.2*pdf2 + 0.5*pdf3
ax.plot(x, pdf4, color='k', alpha=0.8, linewidth=3.0, linestyle='--')
ax.fill_between(x, pdf4, color='k', alpha=0.5)

# REMOVE THIS
plt.tight_layout()
plt.savefig(path, transparent=True)
return path
```

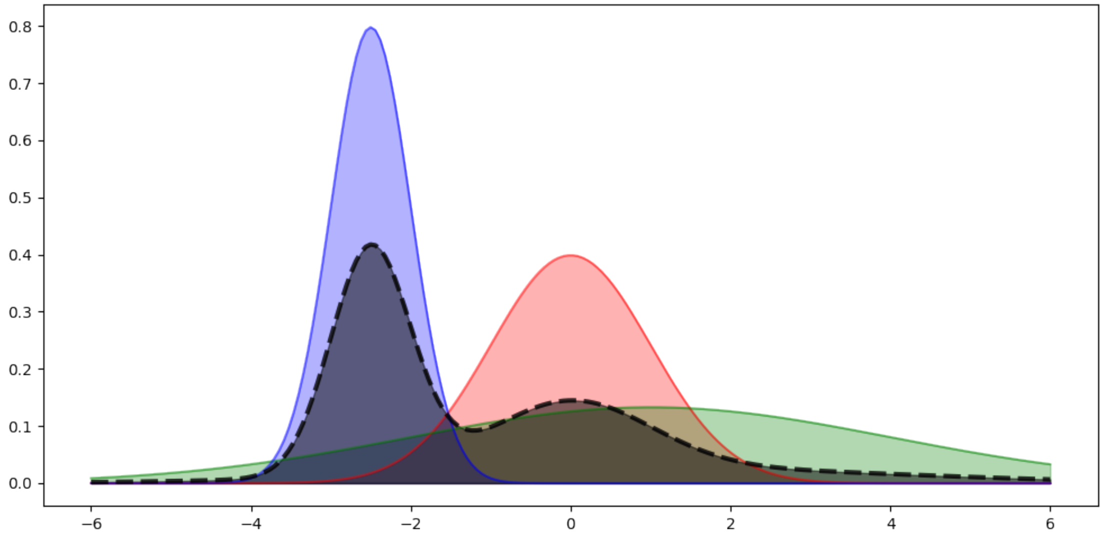

Figure 1: The above figure shows three different parameter settings of a model, *red*, *green* and *blue*. We now marginalise out the parameter according to our belief $p(\theta = \text{red}) = 0.3$, $p(\theta = \text{green}) = 0.2$ and $p(\theta = \text{blue}) = 0.5$ which leads to the evidence in black.

So we should think of the evidence how a model and our beliefs places probability mass over the space where we can later observe data. So now if we would observe some data $\mathbf{Y}$ we can evaluate this under the evidence and say, *what is the evidence that this model is the "right" one?*. Now this becomes very interesting when we have several models. So lets pick another model where instead of Gaussian distribution we have a model using a Laplace distribution giving rise to the evidence plot in Figure 2. So clearly this model and our beliefs in this model places distribution slightly differently across the space.

Code

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

x = np.linspace(-6,6,200)
pdf1 = laplace.pdf(x,0,1)
pdf2 = laplace.pdf(x,-1,1)
pdf3 = laplace.pdf(x,-2.5,0.5)

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

ax.plot(x,pdf1,color='r',alpha=0.5)
ax.fill_between(x,pdf1,color='r',alpha=0.3)
ax.plot(x,pdf2,color='g',alpha=0.5)
ax.fill_between(x,pdf2,color='g',alpha=0.3)
ax.plot(x,pdf3,color='b',alpha=0.5)
ax.fill_between(x,pdf3,color='b',alpha=0.3)

pdf4 = 0.3*pdf1 + 0.2*pdf2 + 0.5*pdf3
ax.plot(x, pdf4, color='k', alpha=0.8, linewidth=3.0, linestyle='--')
ax.fill_between(x, pdf4, color='k', alpha=0.5)

# REMOVE THIS
plt.tight_layout()
plt.savefig(path, transparent=True)
return path
```

So how is this useful, well, so far we have not seen any data, lets say that we now are observing some data $\mathbf{Y}$ which is all centered around -1 as. If we now evaluate the evidence for this data under the two different models we will see that the data is more probably under the model using the Laplace distribution compared to the Gaussian distribution. Simply because the latter model places more of its probability mass just there. This gives us the following relationship,

$$p_{\text{Gaussian}}(\mathcal{D} = \mathbf{Y}) < p_{\text{Laplace}}(\mathcal{D} = \mathbf{Y}), \tag{3}$$

therefore if we would have to choose a model to use to represent this data we would say *There is more or higher evidence for the Laplace model compared to the Gaussian model, we would therefore choose the Laplace model.* This is a really powerful statement as it allows us to test different hypothesis about *model* not just *parameters* of models using this evidence.

This line of reasoning have lead to one of the more famous plots in machine learning, which I like to call the MacKay plot after David Mackay. David was a very influential person in the machine learning community and I would argue that he has a lot to do with the prominent position the UK has played in the development of this field. Slightly outside this topic he also wrote an excellent book on global warming called Without

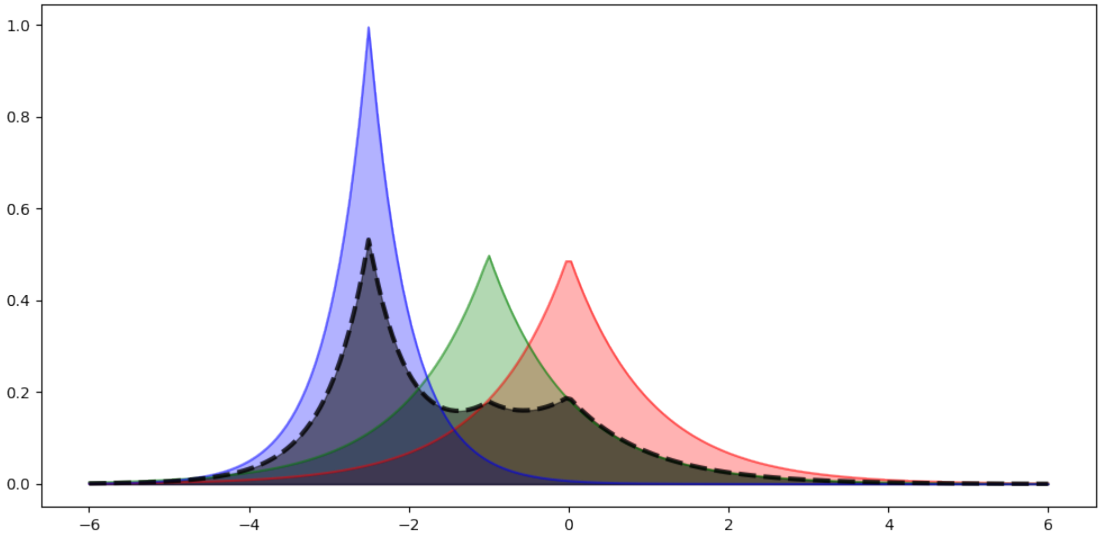

Figure 2: The evidence computed for a slightly different model where we have Laplace distributions that we do not know the parameters of.

Hot Air which is a fantastic read about the challenges of energy. But back on track, David put this Figure 3 of the evidence in his PhD thesis.

The idea of the plot is that if you sort the data-space such that the simpler the data is the further left it is. Now this would mean that the "simpler" the model is then it will place probability mass further left. Now if we are now to choose model between the three above we can use the argument of Occams Razor which simply states that you should *pick the simplest model possible that explains your data.* Now as probability distributions have to sum to 1 the model that has the highest probability when evaluated at the data will thefore have to have placed less probability mass elsewhere and therefore contain less explanations. Therefore picking the model with the maximum evidence would therefore be choosing the model according to Occams Razor. In Figure 3 model $\mathcal{M}_1$ is too simple and $\mathcal{M}_3$ is too complicated and explains too many things while $\mathcal{M}_2$ is just right and this is exactly what the evidence would encode as $p_{\mathcal{M}_2}(\mathcal{D} = \mathcal{D}_0)$ would be higher than both $p_{\mathcal{M}_1}(\mathcal{D} = \mathcal{D}_0)$ and $p_{\mathcal{M}_3}(\mathcal{D} = \mathcal{D}_0)$. Think about this, does it make sense?

Hopefully you think that the argument above makes sense and see that this can be quite a powerful technique. I think its a really strong argumentations and a nice codification of Occam's Razor. Say that you meet someone that believes that the earth is flat, most likely they will present you with evidence that they think supports their theory and most likely[1] the examples that they will give and the explanations that they give will fit. But still, you don't believe them, you think that their arguments are so convoluted, so complicated so really you just go about your way believing the earth is a sphere and then move on. So really, you decided that their model didn't fit because the model that they choose was too *complicated* so you choose another model that was *simpler* but importantly both provided an equivivalent explanation of the data. This means you have acted according to Occams Razor. However, if this is the case, why does anyone believe that the earth is flat[2] simply because the notion of simple is not universal. We all have a different idea of what is simple, and because of this the implementation of Occam's Razor is subjective. You first have to define what simple is in order to use the concept, again, *there is no free-lunch.* What we will now do for the rest of the worksheet is to implement a scenario where we can actually test these concepts and make our understanding of this a bit more clear. We will do so by repeating the experiments of Murray et al., 2005. This paper is a really good read, what I especially like about it is that it doesn't provide an answer, it just raises more

[1] otherwise they will be a bit silly

[2] or that Bristol Rovers is a good football team

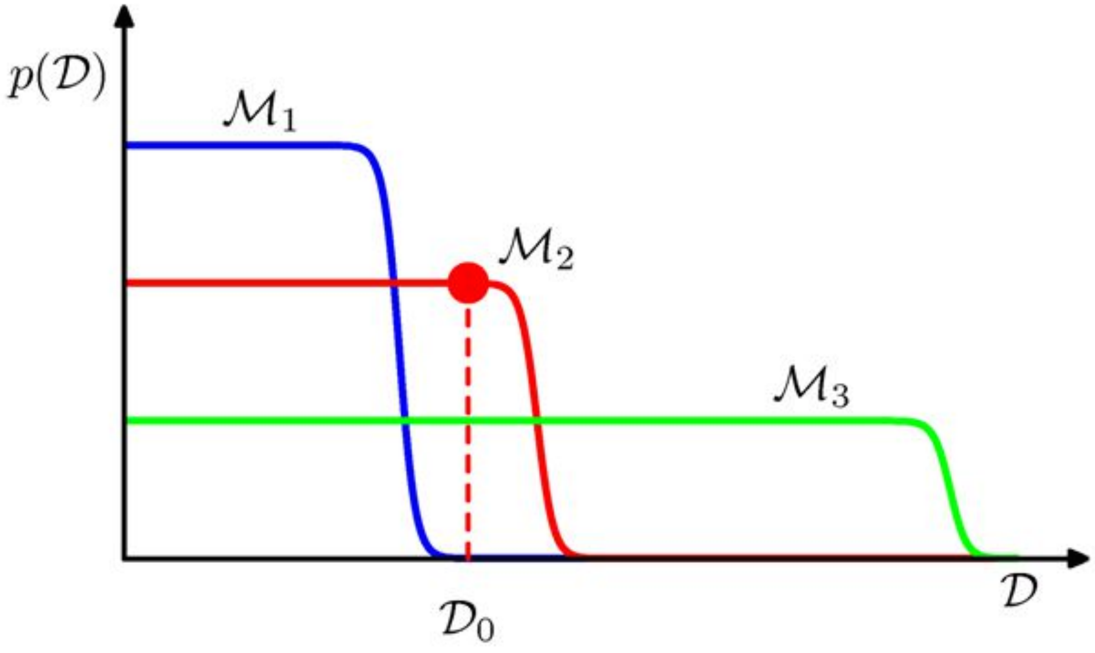

Figure 3: This is the famous Mackay plot, the idea is that if you compute the evidence under three different models, the green, red and blue. The data that you actually observe is $\mathcal{D}_0$ which is evaluated under the three different distributions.

questions.

1 A Note on the Evidence and Bayesian Occam's Razor

The reason that the module initially have focused on conjugate models is because we wanted to avoid computing Baye's Rule to reach the posterior distribution. Conjugacy allowed us to avoid computing the denominator and simply multiply prior and likelhood, then identify the terms to be able to normalise the posterior. For most models this is not possible and we are required to actually perform the full computation and solve an often intractable integral to reach the posterior distribution. Now the object that we want to look at is the evidence so avoiding computation of it is rather pointless. Because of this we are going to choose a discrete data domain whos cardinality is so small so that we can actually evaluate the evidence for all possible data-sets. We will start off by first creating the data-set then we will move on and create a set of models that we can compute the evidence under.

1.1 Data

Consider a very simple data domain $\mathcal{D} = \{y^i\}_{i=1}^9$ where $y^i \in \{-1, 1\}$. This data is structured according to a grid whos locations can be parametrised by $\mathcal{X} = \{\mathbf{x}^i\}_{i=1}^9$ where $\mathbf{x}^i = (\{-1, 0, +1\}, \{-1, 0, +1\})$. This means that our data domain $\mathcal{D}$ contains $2^9 = 512$ different elements which is small enough for us to reason about but still complicated enough that it requires a sensible model.

We will now generate all possible data-sets so that we can evaluate the evidence. The code below will generate a list that you can iterate through with all the possible data points.

Code

```
import itertools as it

def generate_data(N=3):
    D2 = np.array(list(it.product([-1,1],repeat=N*N)))
    D = [];
    for i in range(0,len(D2)):
        d = D2[i]
        D.append(d.reshape([N,N]))
        x = [];
    for i in range(-(N-2),N-1):
        for j in range(-(N-2),N-1):
            x.append(np.array([float(i),float(j)]))
    return (D,x)
```

1.2 Models

Given the data defined above we wish to create a model, i.e. something that will explain the statistical variations that are possible in $\mathcal{D}$. The simplest model that (I) can think of is something that simply takes all its probability mass and places it uniformly over the whole data space,

$$p(\mathcal{D}|M_0, \boldsymbol{\theta}_0) = \frac{1}{512}. \tag{4}$$

The first model Eq. 4 does not take any parameters at all which means it has no flexibility and uses no information about $\mathcal{D}$ except for its cardinality. We can use what we know about the data in order to specify something slightly more representative. If we assume that each y^i are independent we can factorise the model into 9 separate models,

$$p(\mathcal{D}|M_1, \boldsymbol{\theta}_1) = \prod_{n=1}^{9} p(y^i|M_1, \boldsymbol{\theta}_1), \tag{5}$$

where θ_i^j means the j:th element of the parameter vector for the t:th model. Each model can be expressed using an exponential function which relates the vaule y^i to its location $\mathbf{x}^i$,

$$p(\mathcal{D}|M_1, \boldsymbol{\theta}_1) = \prod_{n=1}^{9} \frac{1}{1 + e^{-y^n \theta_1^1 x_1^n}}, \tag{6}$$

We can continue to add more parameters and create further models,

$$p(\mathcal{D}|M_2, \boldsymbol{\theta}_2) = \prod_{n=1}^{9} \frac{1}{1 + e^{-y^n(\theta_2^1 x_1^n + \theta_2^2 x_2^n)}} \tag{7}$$

$$p(\mathcal{D}|M_3, \boldsymbol{\theta}_3) = \prod_{n=1}^{9} \frac{1}{1 + e^{-y^n(\theta_3^1 x_1^n + \theta_3^2 x_2^n + \theta_3^3)}}, \tag{8}$$

Now we can implement the three different models such that we can return the probability for a specific data-point.

Code

```
def model0(theta,x,y):
    return 1.0/(pow(2.,len(x)))

def model1(theta,x,y):
    model = 1.0
    for i in range(len(x)):
        model *= 1.0/(1+exp(-y[i]*theta[0]*x[i][0]))
    return model

def model2(theta,x,y):
    model = 1.0
    for i in range(len(x)):
        model *= 1.0/(1+exp(-y[i]*(theta[0]*x[i][0]+theta[1]*x[i][1])))
    return model

def model3(theta,x,y):
    model = 1.0
    for i in range(len(x)):
        model *= 1.0/(1+exp(-y[i]*(theta[0]*x[i][0]+theta[1]*x[i][1]+theta[2])))
    return model
```

Now we have both the models and the data and its time to move on to the actual computation of the evidence.

1.3 Evidence

The evidence of a model M_i is the distribution $p(\mathcal{D}|M_i)$. This distribution tells us how and where the model spreads its probability mass. Occam's razor can be interpreted in terms of the evidence such as we should choose a model which places most of its mass where we will see data and as little as possible elsewhere. In the previous section we have defined a small simple data domain $\mathcal{D}$ and we will now evaluate where the different models defined above places their probability mass.

In order to "reach" the evidence of a model we need to first remove the dependency of the variable $\boldsymbol{\theta}$. This can be done by marginalising out the parameters from the model,

$$p(\mathcal{D}|M_i) = \int_{\forall\boldsymbol{\theta}} p(\mathcal{D}|M_i,\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{9}$$

The marginalisation above requires one more object that we haven't seen before $p(\boldsymbol{\theta}|M_i)$. This is the prior over the parameters of the model. Being Bayesian implies that you need to take uncertainty into account in all steps of your calculations this is true for the data but also true for the parameters. As we do not really know much at all about the parameters we would like to be very uncertain and allow for a large range of possible values of $\boldsymbol{\theta}$. One prior would be to choose a simple Gaussian with zero mean and a very large variance,

$$\begin{aligned} p(\boldsymbol{\theta}|M_i) &= \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \qquad (10) \\ \boldsymbol{\mu} &= \mathbf{0} \\ \boldsymbol{\Sigma} &= \sigma^2\boldsymbol{I} \\ \sigma^2 &= 10^3 \end{aligned}$$

Now when we have defined the prior $p(\boldsymbol{\theta})$ we just need to perform the marginalisation in Eq. 9 to be able to evaluate the evidence. However, this integration is rather tricky to do analytically which means that we will here use an approximate integral using a naive Monte Carlo approach,

$$p(\mathcal{D}|M_i) \approx \frac{1}{S}\sum_{s=1}^{S} p(\mathcal{D}|M_i, \boldsymbol{\theta}^s), \tag{11}$$

$$\boldsymbol{\theta}^s \sim p(\boldsymbol{\theta}|M_i) \tag{12}$$

where s indexes the samples from the prior of the parameters. Do not worry too much about this proceedure right now, hopefully there was enough insight into sampling during the lecture but as I said then, approximative inference is not the topic of this course. Right now, just see this as a black-box proceedure that we are doing.

The code that you need in order to generate the evidence for the different models is

Code

```
# generate the parameter vector for the samples
def generate_parameters(N,d,mu,sigma):
    return sigma*np.random.randn(N,d)+mu

# generate the evidence
def compute_evidence(y,x,theta,model):
    evidence = 0.0
    for i in range(len(theta)):
        evidence += model(theta[i],x,y)
    return evidence/len(theta)
```

Now we have our models, we have our data and we have an approach to reach the evidence for each model. It is now time to run some experiments and see where this leads to. If you have set things up correctly you should be able to run something similar to,

Code

```
N = 3;
nr_samples = pow(10,2)
sigma = pow(10,1.5)
mu = 0
[D, x] = generate_data(N)
[theta, theta_prior] = generate_parameters(nr_samples,3,mu,sigma)

evidence = np.zeros([4,len(D)])

for i in range(len(D)):
    evidence[0,i] = compute_evidence(D[i].ravel(),x,theta,model0)
    evidence[1,i] = compute_evidence(D[i].ravel(),x,theta,model1)
    evidence[2,i] = compute_evidence(D[i].ravel(),x,theta,model2)
    evidence[3,i] = compute_evidence(D[i].ravel(),x,theta,model3)
```

What we now have is the evidence compute under each of the different models. We can now look at how the distribute probability mass over $\mathcal{D}$. The big question is how to sort the data-set, what order should the data have? One thing that you can try is to sort the data according to one model and plot all model using that one. You can get the indices that sorts an array by using `np.argsort()`. So try something like this,

Code

```
index = np.argsort(evidence[1,:])
ax.plot(evidence[0,index], 'r')
ax.plot(evidence[1,index], 'g')
ax.plot(evidence[2,index], 'b')
ax.plot(evidence[3,index], 'k')
```

When you have got everything running, go back to the arguments we did to motivate the evidence. Does it make sense, do you feel that this is supported in the experiments?

2 Summary

Hopefully you have reached the end of this lab and quite possibly you are a bit confused at this point. What am I actually supposed to have learnt from this? Lets go and think about it from the start of the course, we argued that it is impossible to make any type of learning without making assumptions or having beliefs. Now we have just taken this to its extreme and made an argument that this is also true for Occam's Razor, this is truly a subjective argument as it relies on the concept of simple. The second argument is looking at the plots seeing how the different models distribute their probablity mass, some models represent certain types of data well and seeing that when building models it is alway a choice, if you are good at something you always pay the price for being bad at something else, the important thing is therefore to choose the model which is good at the relevant thing, the thing that you are interested in.

References

Murray, Iain and Zoubin Ghahramani (Aug. 2005). *A note on the evidence and Bayesian Occam's razor*. Tech. rep. GCNU-TR 2005-003.