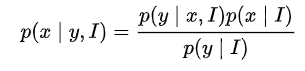
# 1. Read First( The reason we use MCMC)

<https://towardsdatascience.com/bayesian-inference-problem-mcmc-and-variational-inference-25a8aa9bce29>

* We cannot compute the normalization constant. So we have the likelihood and prior and use MCMC to approximate the posterior distribution because we don’t have the normalization constant at the denominator due to high dimensionality.
* Look at the example Bayesian method for hackers, look at chapter “Getting our Priorities Straight” i.e. the stock prick example. They model Prior distributions, they calculate the likelihood x Prior. Since we cannot calculate the normalizing constant, then we compute the posterior distribution through MCMC.
* Usually a posterior is calculated iteratively, first we assume our prior distribution, then we calculate our posterior after observing some data which becomes our new prior distribution, then after observing data our posterior is update with the previous step posterior behaving as our prior.

**Bayesian Updating using KL-Divergence:** <https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence#Bayesian_updating>

* **KL-Divergence** can be used as a measure of **information gain** in **moving** from **a prior distribution** to a **posterior distribution**.

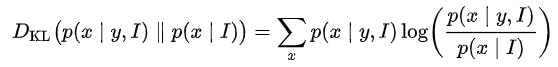
****

Same value x

Same value x

* Notice in the above equation, on the RHS of the equation

we have the prior distribution p(x) so we are moving towards the posterior distribution p(x|y) on the LHS of the equation.

****

Posterior Distribution

Prior Distribution

* So basically we are **computing our Posterior Distribution** and we are saying the we are measuring the change from our Prior Distribution to Posterior Distribution using KL - Divergence.
* So why KL – Divergence is good, is because we can incorporate prior knowledge in our prior distribution. When we are calculating our posterior distribution, we are not directly using a prior distribution to update the posterior distribution, instead we want the change from the prior distribution to the posterior distribution to be small using the KL-Divergence, so basically we can conclude the we are using the prior distribution to update the posterior distribution(Neural Network) exactly because the posterior distribution will be close to prior distribution(of expert knowledge)
* This is exactly what Alpha Star used for imitation learning according to Oriol Vinyals with Lex Fridman interview. They had human replay which they used supervised learning to get a human like policy. (Usually we would make this expert knowledge, but they wanted human like behavior in interview.) So then this policy was used as a prior distribution to update the actual Alpha Star posterior distribution (Reinforcement Deep Network Policy), by minimizing the KL-Divergence between the Prior Distribution(Human Replays Policy Used) and the Posterior Distribution. (Reinforcement Learning Algorithm)
* **My own Example for Trading(Full Thorough Understanding):**
  + I have a Prior Distribution over an expert traders actions fitted using either Neural Networks, Gaussian Processes or some distribution.
  + Next I have a Reinforcement Learning algorithm to find a Posterior Distribution by observing new data, by measuring KL-Divergence between my Prior and Posterior distribution after observing new data for my Reinforcement Learning Algorithm.

This is my Prior Distribution (Expert Knowledge)

* + To update my posterior distribution, I

will be minimizing:

# 2. Gaussian Processes

* Multivariate Gaussain Distributions over **functions**.
* The **Prior**  is a kernel e.g. exponentiated quadratic prior.
* When we sample from the prior, it’s without observing any data points.
* And the sample drawn is a **Function**.
* The posterior distribution is used to predict **y** given input data **x**. So a distribution over **y** is outputted with a **mean** and **covariance**.
* Find out about a function that has multiple variables i.e. f(x,y,z,p) to draw from a gaussain.
* In link below example, (x1, y1) are features i.e. coordinates on the x, y plane. Where x1 is the input and y1 is the output.
* Then 5 functions are sampled from the posterior distribution.
* Next your we make prediction on our test points from the posterior distribution. So our functions from the posterior distribution is plotted with our test points to see how well our model generalizes.
* Neil Lawrence, if you have more than one output in the function, use a mixture of Gaussian processes to decide which ones the output may come from. Deciding on the number of mixtures, is difficult…will have to do cross validation.
* Basically the **mean** and **covariance** **parameters** are **learnt** from the **training data**. The posterior distribution will have fixed parameters(mean, covariance) which will be used to make the prediction. These **parameters** are **fixed** just like a **neural network**.
* In GPFlow an implementation using tensorflow, **MCMC (Markov Chain Monte Carlo)** and **Variational Inference** is used to fit the model i.e. compute the **posterior distribution**.
* A fully Bayesian approach in: <https://blog.dominodatalab.com/fitting-gaussian-process-models-python/>
  + Specify **likelihood** and **priors** for kernel parameters.
  + Rather than **optimize** to fit the Gaussian Process, we **fit** the **Gaussian Process** by using **sample method**. This uses **Hamiltonian Monte Carlo(HMC)** which is an **efficient form** of **MCMC**. So basically we computing the **posterior distribution** using **HMC**.
  + Models being fit to **very large datasets** using **MCMC is slow**, so it is better to replace the true **posterior distribution** with a **simpler approximation** such as **Variational Inference**.
  + In **Variational Inference** we can use **optimization** to **parameterize** the **approximation** so that it is as **close** as **possible** to the **target distribution**.

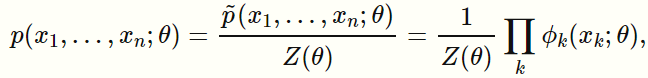
Link: <https://peterroelants.github.io/posts/gaussian-process-tutorial/>

# 3. Sampling Methods (Every Sampling Method)

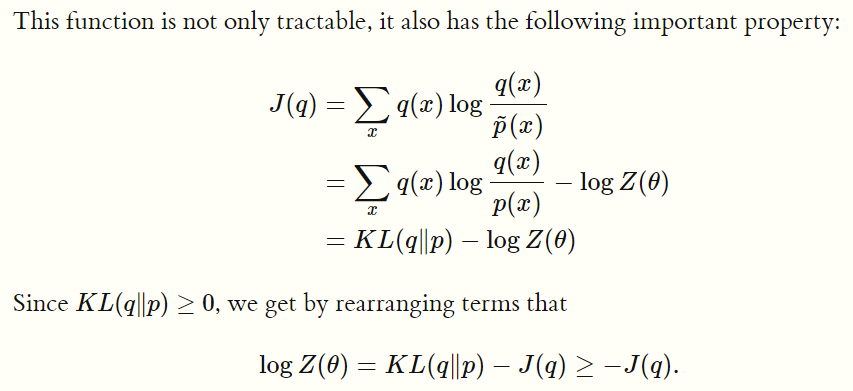
<https://ermongroup.github.io/cs228-notes/inference/sampling/>

# 4. Explanation for KL – Divergence above and how it is using Variational Inference

* First Define Bayes Equation: Notes are from link below
* <https://ermongroup.github.io/cs228-notes/inference/variational/>



* is our **normalization constant**. It is exactly **p(x)** from the denominator in **p(y|x)** in bayes equation.
* is actually our numerator **p(x|y)p(y)** in **p(y|x)** in bayes equation.

****

* Notice the **first line** in the equation is just the divergence between the **numerator**  and **prior**
* The **second line** converts the into modelling the posterior distribution directly because we now have  **instead of (** which is the numerator in main equation). So we don’t have to think about the **numerator p(x|y)p(y)** and how we model this because we are using the posterior directly. This gives us the normalizing constant in this equation
* All we need is our known distribution **q(y) or q(x) in the equation above** which we can think of as our prior distribution **p(y)** in **p(x|y)p(y)** in bayes theorem i.e. **p(y|x)**
* The **third line:** , includes our normalizing constant
* **Now** that we have this equation we can begin to form the evidence lower bound which I now understand fully.
* We first arrange this equation to make the normalizing constant the subject of the formula:
* We know that because which is **POSITIVE** which means that the **LHS** is greater than the **RHS** of the equation. So we created the **RHS:** which is the **Evidence Lower Bound (ELB)**
* **Combining** all we yield
* Lets use my own logic which is exactly the same as the explanation. If we decrease the term , we get
* So the **RHS: Evidence Lower Bound i.e.** becomes close to and equal to the **LHS** of the equation: , so we say that by **minimizing** then we are **maximizing** **Evidence Lower Bound i.e. ,** RHS which is the Evidence Lower bound.
* **Important part** , log z equates to the evidence lower bound, remember that log z is our **normalizing constant**, so by minimizing the divergence we are maximizing the Evidence Lower Bound i.e. which is approximating our **NORMALIZING CONSTANT** which equates to  **WOW !!! So we basically compute our posterior distribution by using KL – Divergence**
* Same as for bayes equation p(y|x) = p(x|y)p(y)/p(x), so

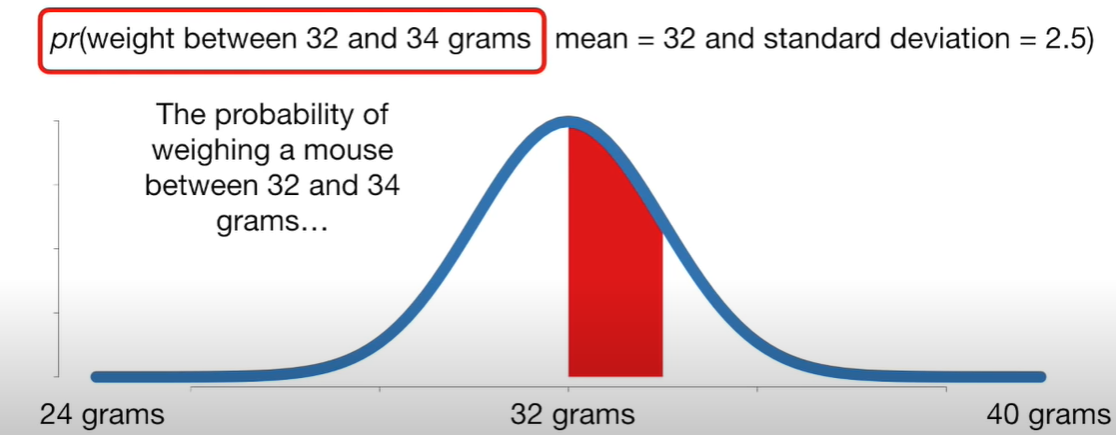
* Monte Carlo methods will always converge to a global optimum, Variational mehtods will never converge to a global optimum because the normalizing constant approximated is rather than .
* Variational Mehods are better because we can measure how good our solution is. We only have a finite amount of time to approximate. Using Monte Carlo Methods takes tooooo long and we don’t know how good of a solution we have. If we had an infinite amount of time, the Monte Carlo will converge to a global optimum and would be the best solution.
* KL – Divergence can be applied to different distributions, at first I thought that distributions have to be the same dimension, but a Multinomial Distribution can be compared to a gaussian distribution or dirichelet process, gaussian process, or softmax distribution. All we do is, we have to customly write our loss funcion if we are using different dimension distributions. So we take the distribution functions that we are comparing and substitute it into this formula getting a super complicated formula after substitution:

# 5. Mean Field Inference

* All this is doing is using many prior distributions for the above q(x), and picking the best q(x) that gives us the lowest error for our KL – Divergence and gives us the maximum Evidence Lower Bound.

# 6. Likelihood vs Probability

* Probability is the area under the curve.
* Assuming that we are given our data X and our parameters; **mean:** , **standard deviation:** ,
* The probability is the area under our curve of data X given the parameters of distribution. **.**
* The diagram below illustrates the probability which is the area under the curve.
* The **Y-axis** is the **likelihood**. If we are given the **probability density function**, then this gives us the area under the curve, but the **pdf** is not the y – axis.
* The second diagram below illustrates the likelihood which is the y – axis, i.e. **.** Given that we observed our data(x), what is our likelihood of our parameters (y – axis), we already know our parameters, but we are getting the likelihood of the parameters after observing some data.

****

