Chapter 1 : Introduction

Training, Testing set, Hold-out or Validation set

The ability to categorize correctly new examples that differ from those used for training is known as generalization.

Finally, the technique of **reinforcement learning (Sutton and Barto, 1998)** is concerned with the problem of finding suitable actions to take in a given situation in order to **maximize a reward**. Here the learning algorithm is not given examples of optimal outputs, in contrast to supervised learning, but must instead discover them by a process of trial and error.

A general feature of re-inforcement learning is the trade-off between ***exploration***, in which the system tries out new kinds of actions to see how effective they are, and ***exploitation***, in which the system makes use of actions that are known to yield a high reward.

Error functions

Model selection for deciding order for polynomial curve fitting

Overfitting :

Say, in poly curve fitting problem, we try to fit a 9-order polynomial to describe sine curve, weights for higher power generally keep on increasing to fine-tune output to best possible sine. Higher flexibility due to higher order might mean, model is simply learning noise as well. Leads to overfit.

We see that, for a given model complexity, the over-fitting problem become less severe as the size of the data set increases. One rough heuristic that is sometimes advocated is that the number of data points should be no less than some multiple (say 5 or 10) of the number of adaptive parameters in the model.

Also, there is something rather unsatisfying about having to limit the number of parameters in a model according to the size of the available training set. It would seem more reasonable to choose the complexity of the model according to the complexity of the problem being solved.

By adopting a ***Bayesian*** approach, the over-fitting problem can be avoided. We shall see that there is **no difficulty** from a Bayesian perspective in employing models for which the number of parameters greatly exceeds the number of data points. Indeed, in a Bayesian model the effective number of parameters adapts automatically to the size of the data set.

Shrinkage :

One technique that is often used to control the over-fitting phenomenon in such cases is that of **regularization**, which involves adding a penalty term to the error function

Note that often the coefficient β_0 is omitted from the regularizer penalty term (sum of squares of betas in Ridge) because its inclusion causes the results to depend on the choice of origin for

the target variable *(Hastie et al., 2001)*, or it may be included but with its own regularization coefficient.

In the context of neural networks, this approach is known as **weight decay**

Probability Theory :

Causes of uncertainty : Noise and Limited data size.

Joint probability

Sum Rule :
$p(X) = \sum_Y p(X, Y)$

Product Rule :
$p(X, Y) = p(Y|X)p(X)$

Joint probability of X and Y = Conditional probability of Y given X * Marginal probability of X

Given, joint probability is symmetrical, we avail *Baye's theorem*

Prior vs Posterior :

Consider 2 boxes with different number of apples and oranges. If we had been asked which box had been chosen before being told the identity of the selected item of fruit, then the most complete information we have available is provided by the probability $p(B)$. We call this the **prior probability** because it is the probability available before we observe the identity of the fruit. Once we are told that the fruit is an orange, we can then use Bayes' theorem to compute the probability $p(B|F)$, which we shall call the **posterior** probability because it is the probability obtained after we have observed F.

Probability density :

Given we want to understand probability distribution in a continuous setting. If the probability of a real-valued variable x falling in the interval $(x, x + \delta x)$ is given by $p(x)\delta x$ for $\delta x \to 0$, then $p(x)$ is called the probability density over x.

Under a nonlinear change of variable, a probability density transforms differently from a simple function, due to the **Jacobian factor**. For instance, if we consider a change of variables $x = g(y)$, then a function $f(x)$ becomes $f(y) = f(g(y))$. Now consider a probability density $p_x(x)$ that corresponds to a density $p_y(y)$ with respect to the new variable y, where the suffices denote the fact that $p_x(x)$ and $p_y(y)$ are different densities. Observations falling in the range $(x, x + \delta x)$ will, for small values of δx, be transformed into the range $(y, y + \delta y)$ where $p_x(x)\delta x$ almost equals $p_y(y)\delta y$, and hence

$p_y(y) = p_x(g(y)) * |g'(y)|$

One consequence of this property is that the concept of the maximum of a probability density is dependent on the choice of variable.

Jacobian factor Reference : https://stats.stackexchange.com/questions/236840/different-probability-density-transformations-due-to-jacobian-factor

Note that if x is a discrete variable, then p(x) is sometimes called a probability mass function because it can be regarded as a set of 'probability masses' concentrated at the allowed values of x

Expectations and covariances :

The average value of some function $f(x)$ under a probability distribution $p(x)$ is called the expectation of $f(x)$ and will be denoted by $E[f]$. Probability essentially acts as weights.

Conditional Expectation for a function (given say y) essentially considers conditional probabilities of x given y as weights.

Variance :

$var[f] = E(f(x) - E[f(x)])^2)$
Expanding we have :
$var[f] = E[f(x)^2] - E[f(x)]^2$

Similarly we have covariance as :
$cov[x, y] = E_{x,y}[xy] - E[x]E[y]$

Bayesian probabilities :

Rationale : Bayes theorem allows us to evaluate the uncertainty in w (weights in curve fitting example) after we have observed D (observed data) in the form of the posterior probability $p(w|D)$. Bayes theorem becomes :

$$posterior = (likelihood * prior)/normalizationconstant$$

In both the Bayesian and frequentist paradigms, the likelihood function $p(D|w)$ plays a central role. However, the manner in which it is used is fundamentally different in the two approaches. In a frequentist setting, w is considered to be a **fixed parameter**, whose value is determined by some form of 'estimator', and error bars on this estimate are obtained by considering the distribution of possible data sets D.

By contrast, from the Bayesian viewpoint there is only a single data set D (namely the one that is actually observed), and the uncertainty in the parameters is expressed through a probability distribution over w.

Say, coin tossed thrice gives heads repeatedly. Frequentist argues probability of heads is 1 based on data, whereas Bayesian updates their priors (starting with 0.5) and generate posteriors.

New evidence does not completely determine your beliefs, it should update prior beliefs

Frequentist vs Bayesian :

Philosophical Q - What is probability ?

Freq. : Related to frequency of repeated events
Bayesian : Related to our own uncertainty of events

So,
Freq. derive variations in data whereas Bayesians analyse variations in belief

Freq. say model is fixed and we observe data around them, Bayesians argue data is fixed and model can vary around it

Say u have 50 values for flux of star brightness and you need to find true value, frequentist applies maximum likelihood - likelihood (under independence) get tigher and tighter near the central value upon repeated observations

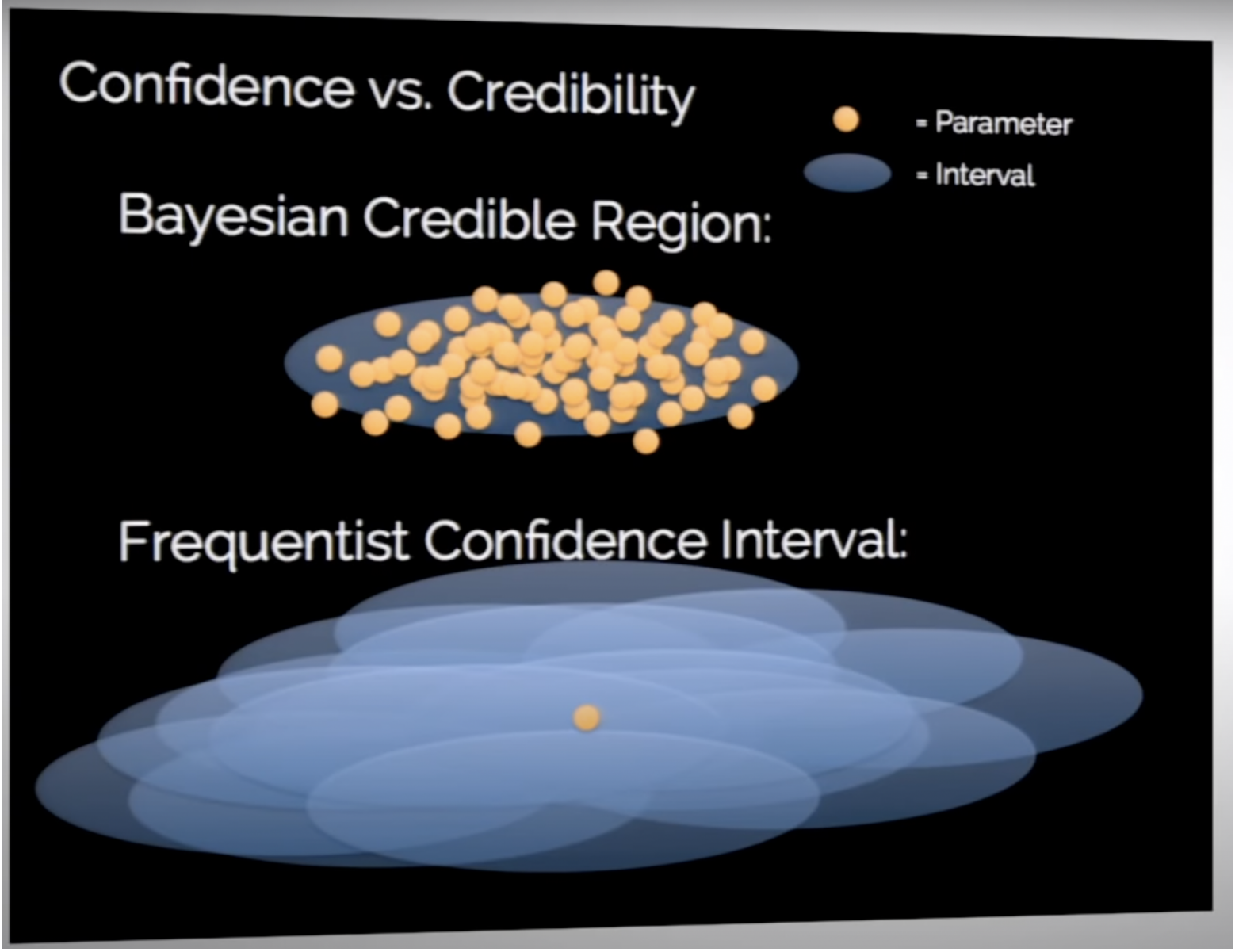
MLE gives mean

Bayesian says I want $P(F_{true}|Data) = \frac{P(Data|F_{true}) * P(F_{true})}{P(Data)}$

Similar results for simple cases. Problems when :

- Nuisance parameters : Bayes Billiard Game
- Uncertainty : 95% of cases confidence interval contains true x versus 95% probability that true x is in region
- Known priors from different source

Freq. Fixed true value, varying CI/model ; Bayesian Fixed CI/model varying true value



https://www.youtube.com/watch?v=KhAUfqhLakw

Gaussian

Sampling methods, such as Markov chain Monte Carlo

Reciprocal of the variance, written as $\beta = 1/\sigma^2$, is called the precision

Univariate Gaussian :

$$\mathcal{N}\left(x|\mu,\sigma^2\right) = \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

We are also interested in the Gaussian distribution defined over a D-dimensional vector x of continuous variables, which is given by

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

where the D-dimensional vector μ is called the mean, the D × D matrix $\sum$ is called the covariance, and $|\sum|$ denotes the determinant of $\sum$.

Data points that are drawn independently from the same distribution are said to be independent and identically distributed, which is often abbreviated to i.i.d. Say we're given some points and have to curve fit a normal distribution of parameters μ and σ^2. For this we define the likelihood function as joint probability of the dataset and given we assume datapoints drawn from idd, we have :

$$p(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N}\mathcal{N}\left(x_n|\mu,\sigma^2\right).$$

For practical and computational reasons, we try maximizing the log likelihood which gets written in the form :

$$\ln p\left(\mathbf{x}|\mu,\sigma^2\right) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi).$$

Maximizing with respect to μ we get μ_{ML} equal to the *sample mean* and maximizing with respect to σ^2 we get *sample variance around*μ_{ML}
Even though we are performing joint maximization wrt both, for Gaussian the solution decouples and we can first evaluate mean then variance.

Limitations of MLE : MLE approach systematically *underestimates* the variance of the distribution. This is an example of a phenomenon called *bias* and is related to the problem of over-fitting encountered in the context of polynomial curve fitting.
Underestimation by $N-1/N$

The issue of bias in maximum likelihood lies at the root of the over-fitting problem that we encountered earlier in the context of polynomial curve fitting.

https://dawenl.github.io/files/mle_biased.pdf