Machine Learning

Dedicated to Thomas Bayes

November 18, 2020

Part 1: The Basics

1 Machine Learning - Philosphy

- No free lunch theorem - We cannot learn without making assumptions. There are infinite models to particular problem.
- Inductive reasoning - allows for conclusion to be false although all premises are true.
- We create roughly 10^{19} bytes of data every other day, which means that we've created more data in the last two years than that for the rest of history.
- A model, e.g. $p(y|\theta)p(\theta)$ is a simplification/abstraction of the real world. For ML it is gerally a statistical description of the world that you can generate data from.
- "Learning is inference", and "Learning can only be achieved by making assumptions".
- ML is the science of making "handles" to incorporate assumptions, mathematical formulations of assumptions to update these assumptions from data to mimic "learning".

2 Probabilities

- *Uncertainty* is the "realisation" of an assumption and a *probability* is a quantification of uncertainty.
- **Joint Distribution** $p(X, Y) = p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$
- **Marginal Distribution** $p(X) = p(X = x_i) = \frac{\sum_j n_{ij}}{N} = \frac{c_i}{N}$
  - Marginalisation is an expectation over a conditional distribution. $p(X) = \int p(X|Y)p(Y)dX$.
- **Conditional Distribution** $p(X|Y) = \frac{n_{ij}}{c_i}$
- **Sum Rule**: Marginalising the Joint distribution: $p(X) = p(X = x_i) = \sum_j \frac{n_{ij}}{N} = \sum_j p(X = x_i, Y = y_j)$
- **Product Rule**: Combing Marginal and Conditional to form Joint distribution: $p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i}\frac{c_i}{N} = p(X|Y)p(X)$
- **BAYES RULE**: $\boxed{p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)p(X)}{\sum_X p(Y|X)p(X)}}$
- **Expectation**: Discrete: $\mathbb{E}[x] = \sum_i x_i p(x_i)$, Continuous: $\mathbb{E}[x] = \int xp(x)dx$, generally for a function $f(x)$, $\mathbb{E}[f(x)] = \int f(x)p(x)dx \approx \frac{1}{N}\sum_i f(x_i), x_i \sim p(x)$
- Variance: $var[f(x)] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$
- Covariance: $cov(x, y) = \mathbb{E}[(x - \mathbb{E}[x])(y - \mathbb{E}[y])]$

2.1 Notation

- Distribution over x_i is $p(X) = p(X = x_i)$, while the distibution evaluated at x_i is $p(x_i)$
- Sum Rule: $\boxed{p(X) = \sum_Y p(X, Y)}$ and Product Rule: $\boxed{p(X, Y) = p(X|Y)p(Y)}$ GENERALISE THESE
- Frequentist approach - probability is the frequency of a repeatable random event. Bayesian approach - probability is the quantification of a belief (of a random variable)

3 Distributions

3.1 The Gaussian Distribution

$$N(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\mathbf{\Sigma}|}} e^{(\mathbf{x}-\mathbf{mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{mu})}$$

3.2 Conjugacy

All posterior distributions are proportional to a likelihood distribution that has a prior distribution multiplying it. If the prior is a "conjugate" prior (a special kind of prior) then we know that the posterior will be of the same form as the prior. Note, it is always the "conjugate" prior over something ($p(\theta)$, where θ is the mean or variance or whatever). Like if we choose a conjugate prior over 'the mean' of a Gaussian likelihood to be Gaussian then the posterior will be Gaussian too.

3.3 Central Limit Theorem

- If you take the average of multiple large samples ($n > 25$) from any distribution, those averages will follow a Gaussian distribution.$https://www.youtube.com/watch?v = aIPvgiXyBMI$

-Part 2: Modelling

4 Linear Regression

- Machine Learning Procedure (for linear regression - although its essentially all the same):
  - 1) Define model $\mathbf{y} = \mathbf{wx} + \epsilon$
  - 2) Incorporate assumption, namely that noise is normally distributed and thereby derive likelihood. In other words, the likelihood is the model given the assumptions (and therefore a conditional distribution).
  - 3) Choose a conjugate prior (if possible) (Gaussian)
  - 4) Determine the Posterior - thus also Gaussian
- Prediction
  - Essentially you are trying to predict a new output $\mathbf{y}^*$ given a new input $\mathbf{x}^*$ - but not deterministically, you find the distribution that tells you the probability of seeing this output given the distribution of the weights. You perform a marginalisation/expectation. The best prediction will be the y^* value that has the highest probability given the highest probability of seeing a particular weighting.

do write formula if you have time

5 Dual Linear Regression

- 'Dual' here, like in Optimisation is simply used to mean 'rephrasing the problem' - in this case by using kernel function and mapping into higher/different spaces.
  - Steps: 1) Formulate posterior 2) Find stationary point of Posterior 3) rewrite $\mathbf{w}$ in terms of data 4) Perform kernel regression.
- This is the generalisation of non-parametric models like Gaussian processes.
- Model complexity depends on the data
- It encodes the relationship between variables using variables.
- One does not need to know potentially complex nonlinear functions of x, say $\phi(\mathbf{x})$, but instead the scalar inner-product $\phi(\mathbf{x})^T\phi(\mathbf{x})$. These are called **Kernel functions** and they are simply functions that take two inputs of any dimension and return a scalar value (so-called *induced space*). These kind of functions can be used as a way of measuring similarity between points - points that are close could return say a large value or say something tending to 1 and points that are far away from each other could return something small or tending to zero. This means:
  - Kernel's allow for 'implicit feature mapping' - think this simply means it allows us to map or keep features within an 'internal' domain.
  - We don't need to know the 'feature space', i.e. the space we would have mapped to.
    - I think what Carl is trying to clarify here is that when we try and transform/map a problem to a different domain so that we can perform a linear regression for example, we can instead use kernels to avoid getting overly complicated because they return something simple like a scalar.
  - Kernel's need to satisfy the triangle inequality
  - **The space can have infinite dimensionality**
  - 'The mapping can be non-linear but the problem remains linear'
  - Kernal Machines (kernel functions) allow us to:
    - Adapt model complexity to data.
    - Put non-vectorial data into vector space
    - keep the problem linear
    - Kernels are the covariance between two data points not data-dimensions.
    - Some typical kernel functions: $K(x,y) = (x^Ty + r)^n, K(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, K(x,y) = e^{-\alpha|x-y|}$
- **Kernel regression** can be is used to predict new points given data. The formula is $\mathbf{y}(\mathbf{x}^*) = \mathbf{t}^*? = k(\mathbf{x}^*, \mathbf{x})(\mathbf{K}(\mathbf{x}, \mathbf{x}) + \lambda\mathbf{I})^{-1}\mathbf{t}$, where $k(\mathbf{x}^*, \mathbf{x})$ is the kernel evaluated between the new input and all other old data points already present, $\mathbf{K}(\mathbf{x}, \mathbf{x})$ is the kernel between all old data points, λ is a parameter related to the noise and $\mathbf{y}(\mathbf{x}^*)$ is the output of the new data point $\mathbf{x}^*$

Kernel's seem mainly to be used to separate data ('The Kernel Trick'). Essentially you take something nonlinear and seemingly inseparable and beam it up into a higher dimension whereby the differences become clear and then you can return it neatly separated back to the original dimension.

6 Processes

6.1 Gaussian Processes L6

- Gaussian Processes are infinite generalisations of Gaussian distributions.
- Form $p(f|x,\theta) \sim \mathcal{GP}(\mu(x), k(x,x))$, where the mean is a function of the input only and the kernel covariance a function of two inputs, which encodes the behaviour of the function (the covariance between each of the points in the function).

- There is a prior over the space of functions. It contains **all** functions
- Predictions can be made by determining the conditional distribution $p(f^*|x^*, f)$ and the mean of this distribution is the same as that of the equation given by kernel regression.
- They allow us to make assumptions over the space of functions (Bayesian Optimisation)

6.2 Gaussian Processes and Unsupervised learning L7

- Increasing length scale in a GP, makes the function smoother.
- Unsupervised Learning
  - "Less supervised" learning → have to make stronger assumptions.
  - This means learning a **representation** not a relationship.
  - We are only given Output, Input is *latent*
  - In contrast to supervised learning which tries to find a conditional posterior distribution, unsupervised leaning intends to find an a priori distribution.
  - It is an ill-constrained problem. Need "another" representation.
  - Have to make assumptions and specify a distribution over the latent variables
  - Two things need to be determined: The weights of the mapping and the linear representation. Since the posterior is intractable, we marginalise out either the weight or latent variable and maximise the other - called **Type II maximum likelihood**
    - Supervised linear regression → $p(w|t) \propto p(t|w)p(W)$ → using conjugate prior this could result in finding posterior
    - Unsupervised linear regression → $p(w, X|t) \propto p(t|w, X)p(w)p(X)$ → posterior intractable, so have to marginalise and use type II max-likelihood
- Principle Component Analysis provides the same solution as the maximum likelihood solution, but solved by an eigenvalue problem instead. So its also the same as the linear regression solution.

6.3 Parametric vs Non-parametric models L9

- Parametric models are defined by **distributions** → they have a finite number of parameters
- Non-parametric Bayesian models are defined by **processes** (which is like an infinite collection of distributions and therefore has infinite parameters). It is a model on an infinite dimensional parameter space. The hierarchy is as follows:
  - Each evaluation of a process is a distribution and each evaluation of a distribution is a value, i.e. Process (∞ parameters) → Distribution (n parameters) → Value

6.4 Dirichlet Process L9

- Dirichlet are clever distributions (also conjugate prior to multinomial distr.) that allows you to place certain amounts of probability mass in various places in a certain space - which makes them useful for partitioning. It is a multivariate generalisation of the beta distribution.
- A Dirichlet process is the generalisation/ infinite extension of Dirichlet distribution. **This allows for a (potentially infinite) partitioning of components**.
- Dirichlet processes are priors over countably infinitely sets as apposed to a GP which is over uncountable infinite sets. I suppose its like a discrete version of a process.
- The Dirichlet Process is best formulated *constructively* by describing how we sample from a Dirichlet distribution. Construcive formulations include:

  - **Chinese restaurant process**: Samples are drawn from DP and shows partitioning on tables with probability $\frac{\alpha}{N-1+\alpha}$.
  - **Stick-breaking construction**: Sees samples from a DP process as breaking a stick recursively with the stick length drawn from a beta distribution.

6.5 Other Processes

7 Bayesian Optimisation L8

- The idea is to optimise a function/model that is unknown (black-box)
- 1) Formulate Prior (which is a Gaussian process) over unknown objective function (i.e. what you believe it could be) → 2) Query the objective function (test out a few inputs) → 3) Derive posterior given the outputs of the objective function → 4) Evaluate **acquisition function** which helps us determine where it is best to sample the unknown objective function from → 5) Repeat until you have to produce a result or are happy with the result. It's bit like playing detective, trying to find the minimum of an unknown function.

8 Generative Models

- **A model is equivalent to describing how the data has been generated.**
- The generative procedure of generative models are shown clearly using graphical models.
- A joint distribution in terms of conditional distributions define a generative model.

8.1 Topic Models L10

- **Hierarchical Models**: Building models by stacking assumptions together.
- Example of Generative model is given - Latent Dirichlet Allocation (LDA) which is used to generate text. It consists of two Dirichlet distributions (= multivariate beta dist.) for word and topic and two multinomial distributions for word and topic assignment. The posterior distribution is intractable. Review this lecture again.

8.2 Graphical Models L11

- **Lingo**: Node → Random variable, edge → stochastic/random relationship, plate → product, directed graph → Bayesian Network, undirected graph → Markov Random Field
- Directed Graphs: (Bayesian Network)
  - $p(\mathbf{x}) = \prod_i p(x_i|pa_i)$, where pa_i are all the nodes directed into the node x_i (parent nodes).
  - If a variable is circled then it needs to be in a conditional, e.g. $p(cicled|...)$ and will be part of the joint distribution to the left of the equation.
- **Explaining away** - When another variable is added such as to make other variables more specific - e.g. ϵ 'explains' away the noise from the data so that $\mathbf{w}$ can represent the signal entirely in linear regression.
- If two nodes are not connected directly but indirectly (tail-to-tail, head-to-tail) then these two nodes are called 'conditionally independent' as long as the arrows are not directed into the node connecting them (head-to-head) - in which case they are dependent. For head-to-head when not observed the nodes are independent and when they are observed they are dependent.
- Review Probability Calculations of p.36 onward L11 before exam
- Undirected Graph (Markov Blanket)

  - Its a subset of a directed graph, in other words a node that depends only on its parents and co-parents (parents of parents) are part of the Markov blanket.
  - Clique - a subset of nodes in a graph that all share an edge between pairs.
  - Undirected models are joint probabilities

- Causality: $p(x|y)$ does not mean y caused x!

————————————————————Part 3: Inference————————————————————

- The aim of inference is to reach the posterior distribution, i.e. what can we 'infer' given new observations?

- Analytically intractable - The evidence cannot be written as elementary functions, or a combination of infinite number of components.

- Computationally intractable - To expensive/ time consuming to compute.

9 Laplace Approximation L12

- This is essentially a Gaussian distribution approximating the posterior. A lot of posterior distributions will be one-model and symmetric so this is often a useful approximation. The fitted Gaussian is centred around the mode (place with most probability mass) of the posterior distribution.

- Aside: In the lecture, logistic regression is used as an example, this is a logistic function (sigmoid function) to perform regression on binary independent variables. Potentially review.

10 Sampling L13

- Inverse CDF transform sampling (change-of-variables): Take the distribution you want to sample from and calculate its integral, to find its CDF. Then invert that CDF, which now by definition can only take values between $[0, 1]$. Thus sample from a uniform distribution between $[0, 1]$ and plug them into the inverse CDF function, and as if by magic it will sample from the original distribution. Not helpful really if you don't know the original distribution though, but still pretty ingenious...

- Slight twist: We can also use sampling to approximate integrals or rather expectations of functions that are distributed by a pdf. Since the expected value (mean) of the unknown pdf is unknown, we take random samples from the unknown pdf and just average the values we get by the number of values seen - pretty basic idea. This can then generalised for any function: So

$$\mathbb{E}_{p(x)}(x) = \int xp(x)dx \approx \frac{1}{L}\sum_{i=1}^{L} x_i \rightarrow \mathbb{E}_{p(x)}(x) = \int f(x)p(x)dx \approx \frac{1}{L}\sum_{i=1}^{L} f(x_i)$$

  where $x_i \sim p(x)$

11 Variational Inference L14

- The idea is to find a distribution similar to the posterior using optimisation techniques.

- The evidence hinders us from reaches the posterior because it cannot be computed.

- We use Jensen's inequality to find a bound on the evidence, which is the KL divergence between the approximate posterior and the actual posterior, plus the evidence. This means that if the KL divergence becomes zero the inequality becomes an equality, and our approximate posterior matches the true posterior perfectly.

- Since we can't calculate the KL divergence (as we do not know the posterior) we can reformulate it as the sum of Shannon's entropy and an expectation of the joint distribution and the evidence.

- Now we rearrange for the evidence and we notice that in order to minimise the KL divergence we need to maximise the expectation and Shannon's entropy (so-called Evidence Lower Bound)
- Thus the optimisation problem for the variational inference method is simply to **maximise** the ELBO.
- The ELBO is non-convex and suffers from local minima.
- **Mean field variational Bayes** is a fully factorised form over the unknown variables that solves this optimisation problem by approximating the posterior (essentially means its trying to match/pair the marginals of the posterior distribution).

12 Neural Networks L15

- The power of neural networks lies in the fact that they can be used to approximate any kind of continuous functions.
- It does so, essentially, by changing weights in composite functions.
- For classification purposes they are also extremely useful because they can retain information while reducing it. For example if you feed the network tonnes of images of cats and dogs and it could classify them as either being a a cat or a dog. From what I understand, what its doing is finding the weights of a bunch of functions that are all composite together to approximate some unknown cats-and-dogs-classifying function.
- The underlying mathematics is as follows:
  - The 'Kernel of a function' is the set of inputs that produce the same output. As more compositions are added this set increases. $Kern(f_i) = \{(x, x')|f_i(x) = f_i(x')\}$
  - The 'Image of a function' is the set of all outputs given an input. (So if there is a threshold, say $[0, 1]$, then the output must lie between 0 and 1). This set decreases as more compositions are made, which means that the output becomes more and more restricted. $Image(f_i) = \{y \in Y|y = f(x_i), x \in X\}$

13 Reinforcement learning L16

- The idea is to reward certain preferred actions over less preferred actions, so that the machine 'learns' what you want to achieve. The actions are governed by a so-called 'policy' function (π) that tries to ensure that various actions are explored while simultaneously making the machine do more of the preferred actions (exploration vs exploitation). After an action has been completed the machine finds itself in a new state, the transition of which is governed by a different function (f) that incoorperates the dynamics (e.g. say the states were positions on a pendulum then the transition from one position to the next would be governed by $\ddot{\theta} = -g/l \sin\theta$ or whatever). So essentially, we model something and then we make 'decisions' based on that model such that we perform actions that maximise the reward.