

0.1 Coherence

From wiki:

Two wave sources are coherent if they have the same frequency ν and waveform and *constant* phase difference δ . We can always choose initial time such that one of the waves have a zero phase at $t = 0$. So if two wavefunctions $f_\nu(x, t)$ and $g_\nu(x, t)$ are coherent according to the above definition, and $f_\nu(x, t)$ has a zero phase at $t = 0$, we have that for all x and t

$$f_\nu(x, t) - g_\nu(x, t) = \delta,$$

since they have the same waveform and differ only by a constant phase.

0.2 Machine Learning in terms of statistical theory

iid - independent identically distributed. Usually used when referred to a set of realisations $\{x_i\}$ of some distribution $P(x)$.

The powerful thing is that we can use such x_i to get an approximation of integral over $P(x)$ as $1/N \sum f(x_i)$. The error goes as $1/\text{sqrt}(N)$.

0.2.1 What are we trying to do?

Learner's input:

- Domain set X , or instance space. Domain point $x \in X$ also called an instance.
- Label space: A set Y is called the label space, and individual points $y \in Y$ are labels.
- Training data: We need somewhere to start. So we have some training set $S = \{(x_i, y_i)\}$ that come from $X \times Y$. We should note that this set is a *realisation* from the unknown distribution $P(X, Y)$.

Learner's output:

- What we want from the learner is a *prediction rule* which is a function $h : X \rightarrow Y$. It serves to predict labels for data we haven't seen before.
- Several names, predictor, hypothesis, classifier...

Let's say that we have obtained a predictor in some way. We need a measure to see how good it is at prediction. The nice thing is that if we find out a measure we can also find the method of prediction, a maximisation of the measure, which in this case will mean a minimisation of the error.

We define a loss function $L(h(x), y)$ for the predictor $h(x)$. This can for a classification problem for example be the error rate, where you just count the number of times the predictor $h(x)$ fails to give the correct result on some set of testing data where we know the correct answer. We can define different loss functions. For regression we could use squared loss function

$$L(h(x), y) = (h(x) - y)^2. \quad (1)$$

To go further we define the *average* of the loss function, and we call this the *risk* of the predictor h . The average loss, or risk $R(h)$ is formally

$$R(h) = \int dx dy P(x, y) L(h(x), y),$$

which is just a standard definition of average (or expectation value). Using the product rule we rewrite this as

$$R(h) = \int dx P(x) dy P(y|x) L(h(x), y).$$

The optimal predictor is then the function that minimises this risk. But we don't know the true probability distribution, so we need to use the approximation "Markov density approx."

We have an empirical training set consisting of N pairs (x_i, y_i) . We get the *Empirical Risk Function*

$$R(h) \approx R_N(h) = \frac{1}{N} \sum_i L(h(x_i), y_i). \quad (2)$$

Now we arrive at the core of Machine Learning.

What we are doing is trying to find the predictor $h(x)_{\text{optimal}}$ that minimises the empirical risk function. This procedure is called Empirical Risk Minimisation (ERM), formally:

$$h(x)_{\text{optimal}} = \operatorname{argmin}_h \left[\frac{1}{N} \sum_i L(h(x_i), y_i) \right].$$

0.2.2 Bayes' theorem

Product rule for probability

$$P(x, y) = P(y)P(x|y) = P(x)P(y|x),$$

if x, y independent then $P(x, y) = P(x)P(y)$.

Bayes' Law (or Bayes' theorem) is a direct consequence of the product rule. We just divide the product rule by $P(x)$ or $P(y)$, to get (dividing by $P(y) \neq 0$)

$$\frac{P(x, y)}{P(y)} = P(x|y) = \frac{P(x)P(y|x)}{P(y)}.$$

The last equality is the usual statement of the theorem, here with events A, B instead of x, y :

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}.$$

To get to the form of the theorem in the lecture slides we need to use the law of total probability

$$P(B) = \int dA P(A)P(B|A).$$

Then we write the Bayes' theorem

$$P(A|B) = \frac{P(A)P(B|A)}{\int dA P(A)P(B|A)}.$$

With a discrete space we replace integrals by sums.