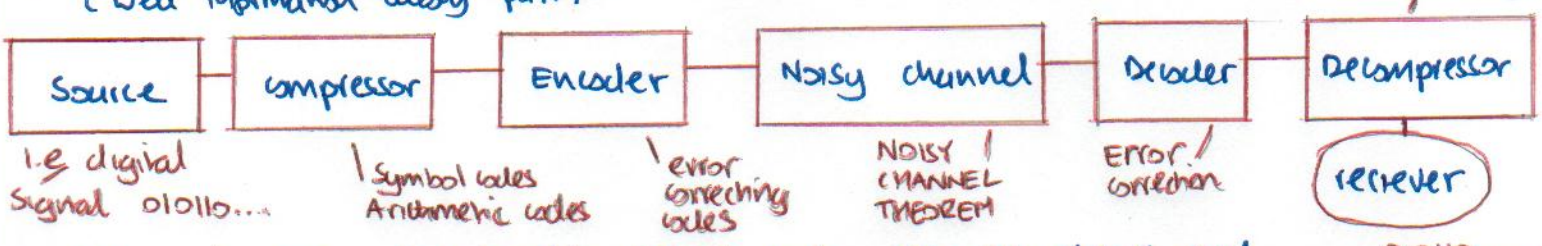
INFORMATION THEORY AND NEURAL NETWORKS

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point" (Claude Shannon, 1948).

1/ Pictorial synopsis of course: Source → reciever block diagram. (well information theory part).



2/ Information theory is intimately linked with probability theory and Bayesian inference.

Notation and definitions: An ensemble X is a triple (x, A_X, P_X) where the OUTCOME x is the value of a random variable which takes on one of a set of possible values from ALPHABET A_X with probability $p_i \in P_X$. Now consider joint ensembles of random variables x, y. (Definitions below can easily be extended to more variables). Start from joint probability distribution $P(x,y)$. (which must have property $\sum_{x,y} P(x,y) = 1$).

$$P(x) = \sum_y P(x,y) \qquad P(y) = \sum_x P(x,y) \qquad P(x,y) = P(x)\,P(y|x) = P(y)\,P(x|y)$$

$\Rightarrow$ BAYES THEOREM $P(y|x) = \frac{P(x|y)\,P(y)}{P(x)}$ (POSTERIOR = LIKELIHOOD × PRIOR / EVIDENCE)

$$h(x) = p(x) \log_2 \frac{1}{p(x)} \qquad H(X) = \sum_x p(x) \log_2 \frac{1}{p(x)} \qquad H(X,Y) = \sum_{x,y} P(x,y) \log_2 \frac{1}{P(x,y)}$$

SHANNON INFORMATION CONTENT — ENTROPY — JOINT ENTROPY

$$H(X|Y) = \sum_{x,y} P(x,y) \log_2 \frac{1}{P(x|y)} \qquad I(X;Y) = H(X) - H(X|Y)$$

CONDITIONAL ENTROPY — MUTUAL INFORMATION

$$D_{KL}(P\|Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}$$

RELATIVE ENTROPY OR KULLBACK-LEIBLER DIVERGENCE (between two probability distributions defined over the same alphabet A_X).

Theorems and inequalities: Jensen's inequality.

If $f(x)$ is a convex $\cup$ function and x a random variable:

$$\mathcal{E}[f(x)] \geq f(\mathcal{E}[x])$$

with equality when x = constant.

If $f(x)$ is a concave $\cap$ function and x a random variable:

$$\mathcal{E}[f(x)] \leq f(\mathcal{E}[x])$$

This can be used to prove Gibbs' inequality:

$$D_{KL}(P\|Q) \geq 0 \quad \text{with equality when } P = Q.$$

Also:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

with equality when $X = Y$.

$$I(X;Y) = I(Y;X) \geq 0$$

$$\mathcal{E}[x] = \sum_x x\,P(x) \qquad \text{EXPECTATION (mean } \mu)$$

$$V[x] = \sum_x x^2 P(x) - \mathcal{E}^2[x] \qquad \text{VARIANCE } (\sigma^2)$$

For random variables x, y:

$$\mathcal{E}[x+y] = \mathcal{E}[x] + \mathcal{E}[y]$$

$$V[x+y] = V[x] + V[y]$$

Scalars a, b.

But $\mathcal{E}[ax+by] = a\mathcal{E}[x] + b\mathcal{E}[y]$

$V[ax+by] = a^2 V[x] + b^2 V[y]$

Important probability distributions

BINOMIAL: $P(x|f,N) = \binom{N}{x} f^x (1-f)^{N-x}$ — "Probability of x occurrences of binary variable with probability f"

$\binom{N}{x} = \frac{N!}{(N-x)!\,x!}$ DISCRETE DISTRIBUTION $\mu = Nf$ $\sigma^2 = Nf(f-1)$

POISSON $P(x) = \frac{\mu^x}{x!} e^{-\mu}$ $\sigma^2 = \mu$ (should be $P(x|\mu)$?)

GAUSSIAN (NORMAL) $P(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Approximate Gaussian by Gaussian with $\mu = \sigma^2$

$\frac{e^{-\mu}\mu^x}{x!} \simeq \frac{1}{\sqrt{2\pi\mu}} e^{-\frac{(x-\mu)^2}{2\mu}}$ let $x = \mu \Rightarrow$ STIRLING'S APPROXIMATION

$\mu! \simeq \mu^\mu e^{-\mu}\sqrt{2\pi\mu}$ INFO ①

(STIRLING'S APPROXIMATION CONT....) $\Rightarrow \ln x! \simeq x\ln x - x + \frac{1}{2}\ln 2\pi x$ $(\mu = x)$.

$\therefore \ln\binom{N}{x} \simeq (N-x)\ln\frac{N}{N-x} + x\ln\frac{N}{x}$ * Define BINARY ENTROPY FUNCTION $H_2(x)$:

$H_2(x) = x\log_2\frac{1}{x} + (1-x)\log_2\frac{1}{1-x}$. Note $\frac{d}{dx}H_2(x) = \log_2\frac{1-x}{x}$. * valid for any base. (since all terms are logarithms)

$\therefore \log_2\binom{N}{x} \simeq NH_2(\frac{x}{N}) - \frac{1}{2}\log_2\left[2\pi N\cdot\frac{N-x}{N}\cdot\frac{x}{N}\right]$.

Central limit theorem: All distributions $P(x|M,\sigma) \to$ Gaussian as $|A_x| \to \infty$

$|A_x|$ = # values within ensemble.

Bayesian inference Two types of Bayesian inference problems

1. DIRECT HYPOTHESIS INFERENCE (this)
2. PARAMETERISED HYPOTHESIS INFERENCE (this)

1. Direct Hypothesis inference. A set of distinct hypothesis $\{H_i\}$ generate a set of data D. Relationship from H_i to D is given by the forward probability distribution $P(D|H_i)$. Apply Bayes' theorem to infer probability of H_i GIVEN observed data D.

$$P(H_i|D) = P(D|H_i)P(H_i)/P(D) \qquad P(D) = \sum_j P(D|H_j)P(H_j)$$

Examples:
* CRIMINAL CASES
* 3 DOORS PROBLEM.

2. PARAMETERISED Hypothesis Inference. A particular hypothesis H is parameterised by a set of variables x. i.e $H = H(x)$. Forward probability distribution is $P(D|H,x)$. Apply Bayes theorem to infer parameter x given observed data D, and hypothesis H. In this statement regard H as a generic structure varied by parameter x.

$$P(x|D,H) = \frac{P(D|H,x)P(x|H)}{P(D|H)} \qquad P(D|H) = \sum_x P(D|H,x)P(x|H)$$

Example
* Decay rate problem.

Note in many cases prior $P(x|H)$ will be uniform and thus can be absorbed into the normalisation 'Evidence' $P(D|H)$.

If $P(x|H)$ constant $\Rightarrow$ $$P(x|D,H) = \left[1 + \sum_{y\neq x}\frac{P(D|H,y)}{P(D|H,x)}\right]^{-1}$$

3/ Communication channels. (Noisy). A discrete memoryless channel Q is characterised by an input alphabet A_X, an output alphabet A_Y and a set of conditional probabilities $P(y|x)$ where $x \in A_X$, $y \in A_Y$.

Define $Q_{j|i} \equiv \theta_{j|i} = P(y=b_j|x=a_i)$. Write $\{x_i\}$ as a vector $\underline{x}$ and $P_X = \underline{P_x}$

$\Rightarrow \underline{P_y} = \underline{\underline{Q}}\,\underline{P_x}$ using $P(y) = \sum_x P(x,y) = \sum_x P(x)P(y|x)$. So $P_{y_j} = Q_{j|i}P_{x_i}$

Model channels

BINARY SYMMETRIC CHANNEL

$A_X = \{0,1\}$ $A_Y = \{0,1\}$

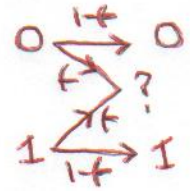
$P(y=0|x=0) = 1-f$ $P(y=0|x=1) = f$

$P(y=1|x=0) = f$ $P(y=1|x=1) = 1-f$

$$\underline{\underline{Q}}_{BS} = \begin{pmatrix} 1-f & f \\ f & 1-f \end{pmatrix}$$

$[\,0 \le f \le 1.$ for all channels.$\,]$

BINARY ERASURE CHANNEL



$A_X = \{0,1\}$ $A_Y = \{0,?,1\}$

$P(y=0|x=0) = 1-f$ $P(y=?|x=0) = f$ $P(y=1|x=0) = 0$

$P(y=0|x=1) = 0$ $P(y=?|x=1) = f$ $P(y=1|x=1) = 1-f$

$$\underline{\underline{Q}}_{BE} = \begin{pmatrix} 1-f & 0 \\ f & f \\ 0 & 1-f \end{pmatrix}$$

Z CHANNEL

$A_X = \{0,1\}$ $A_Y = \{0,1\}$

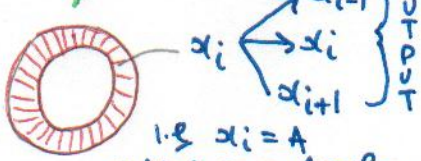
$P(y=0|x=0) = 1$ $P(y=1|x=0) = 0$ $P(y=0|x=1) = f$ $P(y=1|x=1) = 1-f$

$$\underline{\underline{Q}}_Z = \begin{pmatrix} 1 & f \\ 0 & 1-f \end{pmatrix}$$

NOISY TYPEWRITER

$A_X = A_Y =$ 27 letters $\{A, B, (\ldots, Z, -\}$

letters arranged in a circle.



i.e $x_i = A$ output = -, A or B.

$P(y=x_{i-1}|x_i) = \frac{1}{3}$

$P(y=x_i|x_i) = \frac{1}{3}$

$P(y=x_{i+1}|x_i) = \frac{1}{3}$

Define CAPACITY $C = \max_{P_X} I(X;Y)$ for a channel described above $P_Y = Q P_X$

The capacity of a channel is a good measure of the amount of error free information that can be transmitted over the channel per unit time.

Recipe for working out the capacity of a channel. ① Write $P_X = \{P_1, P_2, P_3 \dots P_{N-1}, 1-\sum_{i=1}^{N-1} P_i\}$ ($N = |P_X|$). ② Work out $I(X;Y)$. For mechanistic methods given $P_X = \{P_i\}$ and $P_Y = \{q_j\}$: (write $\{q_j\} = \{q_1, q_2, q_3 \dots q_M\}$; $M = |P_Y|$ } and $\|Q_{ji}\|$:

$I(X;Y) = H(X) - H(X|Y)$. $H(X) = \sum_x P(x) \log_2 \frac{1}{P(x)}$. $H(X|Y) = \sum_{x,y} P(x,y) \log_2 \frac{1}{P(x|y)}$

$P(x,y) = P(x) P(y|x) = P(y) P(x|y)$ $\therefore P(x|y) = P(y|x) \frac{P(x)}{P(y)}$ $\therefore H(X|Y) = \sum_{x,y} P(x) P(y|x) \log_2 \frac{P(y)}{P(y|x) P(x)}$

$\therefore I(X;Y) = \sum_x P(x) \log_2 \frac{1}{P(x)} - \sum_{x,y} P(x) P(y|x) \log_2 \frac{P(y)}{P(y|x) P(x)}$ *

OR (using P_i, q_j, Q_{ji}): $I(X;Y) = \sum_{i=1}^{N} P_i \log_2 \frac{1}{P_i} - \sum_{i=1}^{N} \sum_{j=1}^{M} P_i Q_{ji} \log_2 \frac{q_j}{Q_{ji} P_i}$

To find OPTIMAL INPUT DISTRIBUTION: $\{P_i^*\}$ Solve $\nabla I = 0$

where $\nabla = \left(\frac{\partial}{\partial P_1}, \frac{\partial}{\partial P_2} \dots \frac{\partial}{\partial P_{N-1}} \right)$ — Note last term is not independent of the others in order to absorb constraint $\sum_i P_i = 1$ into our minimisation analysis. (Easier than lagrange multipliers).

$\therefore I(X;Y)$ given $\{P_i^*\} =$ capacity C.

* Note (*) can be written in a more compact form

$I(X;Y) = \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)} = \sum_{x,y} P(x) P(y|x) \cdot \log_2 \frac{P(y|x)}{P(y)}$

$\Rightarrow I(X;Y) = \sum_{i,j} P_i Q_{ji} \log_2 \frac{Q_{ji}}{q_j}$

In continuous limit for continuous distributions

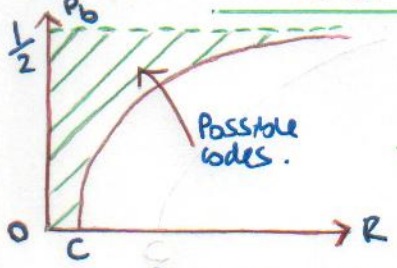
$I = \sum_{x,y} P(x) P(y|x) \log_2 \frac{P(y|x)}{P(y)} \to \int_{x,y} dx\,dy\, P(x) P(y|x) \log_2 \frac{P(y|x)}{P(y)}$

i.e for Gaussian channel: $P(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp[-(y-x)^2/2\sigma^2]$

NOISY CHANNEL CODING THEOREM.

For every discrete memoryless channel the capacity $C = \max_{P_X} I(X;Y)$ has the following property: For any $\epsilon > 0$ and $R < C$ for large enough N, there exists a code of length N and RATE $\geq R$ and a decoding algorithm s.t the maximal probability of bit error is $< \epsilon$.

If bit error probability P_b is acceptable $R(P_b) \leq \frac{C}{1 - H_2(P_b)}$



Definition of rate R: If # codewords $S = 2^K$, $K \in \mathbb{Z}$ $R \equiv K/N$ where N = length of the codewords.

(I suppose for non block codes i.e of varying length use 'expected length' L ? → see later for expected length)

— So can generate codes with zero bit error at non zero rate!

Unfortunately the N.C.C.T does not tell you how..... Derivation of N.C.C.T involves the notion of the typical set. If $\underline{x}$ is typical of $P(\underline{x})$ $\left| \frac{1}{N} \log_2 \frac{1}{P(\underline{x})} - H(\underline{x}) \right| < \beta$ where β is some (small) tolerance and N = codelength.

(similar definitions for $\underline{x}, \underline{y}$ jointly typical of $P(\underline{x}, \underline{y})$).

4/ Codes. Motivated by N.C.C.T we will look firstly at error correcting codes whose bit error probabilities and rates lie in shaded area of diagram above.

Two examples. (i) REPETITION CODES. (R_N). These repeat each bit of a binary signal during transmission over a noisy channel (N time). i.e for some $\underline{s} = 01010$ under R_3 transmitted signal $\underline{t} = 000111000111000$. Decoder reads in N bits at a time and takes a majority vote. i.e 001 ⇒ 0 for R_3

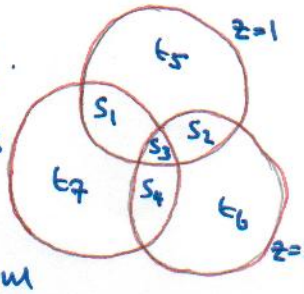
Repetition codes R_N have error probability $P_b = \sum_{n=(N+1)/2}^{N} \binom{N}{n} f^n (1-f)^{N-n}$ for a binary symmetric channel.

If information stored on one typical hard disc (1 Gbyte) is coded with R_N and no errors are expected during a ten year lifecycle $\Rightarrow P_b \sim 10^{-15} \Rightarrow N \sim 61$. So 61 hard drives are required to deliver adequate performance! (Use Stirling's approximation on leading order $\binom{N}{\frac{N+1}{2}}$ term above).

A better code is the HAMMING $(7,4)$ code.

If $\underline{s} = (s_1 s_2 s_3 s_4)$ transmit $\underline{t} = (s_1 s_2 s_3 s_4 t_5 t_6 t_7)$ $t_i, s_i \in \{0,1\}$.

$t_5 t_6 t_7$ are parity bits. They are set so $\sum_i s_i t_i$ in each circle on the right is even.



Define SYNDROME $\underline{z} = \{z_1 z_2 z_3\}$ where z_i is the sum of all bits in each circle modulo 2. If only one bit of $\underline{t}$ is flipped, computing syndrome and referring to circle diagram allows erroneous bit to be isolated. Hence Hamming $(7,4)$ code is error correcting up to 1 bit error.

SYNDROME z	000	001	010	011	100	101	110	111
UNFLIP THIS BIT	NONE	7	6	4	5	1	2	3

Symbol codes are designed to compress information before transmission over a noisy channel. In a symbol code each symbol has an associated binary string codeword. i.e. $A_X = \{a,b,c\}$, $P_X = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{2}\}$ $\Rightarrow C_X = \{00, 01, 1\}$ (by Huffman encoding scheme). Symbol code C for ensemble X with have EXPECTED LENGTH $L(C,X) = \sum_i p_i l_i$. $p_i \in P_X$ and l_i = length of codeword c_i.

A symbol code is UNIQUELY DECODABLE if each codeword has a distinct binary string and extended strings cannot be represented by more than one string of code symbols. i.e. if $a \Rightarrow 0$, $b \Rightarrow 00$ [00] could mean 'aa' or 'b'. All PREFIX CODES are uniquely decodable. A symbol code is called a prefix code if no codeword is a prefix of any other codeword. (Also known as 'instantaneous code' by virtue of 'decode as you read'). A COMPLETE CODE is uniquely decodable and satisfies the KRAFT INEQUALITY with equality. All uniquely decodable symbol codes satisfy the Kraft inequality.

KRAFT INEQUALITY $\sum_i 2^{-l_i} \leq 1$. Define $q_i \equiv \frac{2^{-l_i}}{z}$ where $z = \sum_i 2^{-l_i}$

Now from Gibbs inequality $D_{KL}(P||Q) \geq 0 \Rightarrow \sum_i p_i \log_2 \frac{p_i}{q_i} \geq 0 \Rightarrow \sum_i p_i \log_2 \frac{1}{q_i} \geq \sum_i p_i \log \frac{1}{p_i} = H(X)$

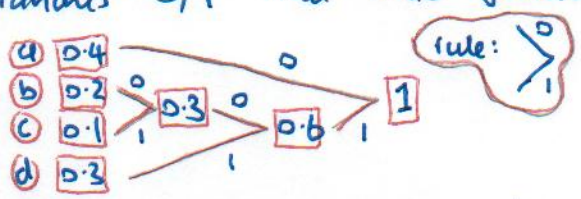
Now $\log_2 \frac{1}{q_i} = \log_2 z + l_i$. $\therefore \log_2 z \sum_i p_i + \sum_i p_i l_i \geq H(X)$. Now Kraft inequality $\Rightarrow z \leq 1 \Rightarrow \log_2 z \leq 0 \quad \therefore L(C,X) \geq H(X)$ i.e. can't compress further than the entropy of a source.

Note also $L(C,X) - H(X) = D_{KL}(P||Q)$ where $Q = \frac{\{2^{-l_i}\}}{\sum_i 2^{-l_i}}$. i.e. $D_{KL}(P||Q)$ is the difference between the actual codelength and the optimum.

Huffman coding – algorithm for producing optimal complete codes. (1) order codewords by probability. (2) combine two lowest probabilities in a tree diagram. (3) repeat until tree is complete. (4) Starting from top of tree label branches 0, 1 and thus generate the code.

Example: $A_X = \{a,b,c,d\}$ $P_X = \{0.4, 0.2, 0.1, 0.3\}$

$\Rightarrow C_X = \{0, 100, 101, 11\}$



Arithmetic codes combine a probabilistic model with an encoding algorithm that identifies each string with a sub interval of $[0,1]$ of size equal to the probability of that string under the model. The encoding is almost optimal in the sense that the compressed length of a string $\underline{x}$ closely matches the Shannon information content of $\underline{x}$ given the probabilistic model. Arithmetic codes fit with the philosophy that good compression requires data modelling, in the form of an adaptive Bayesian model.

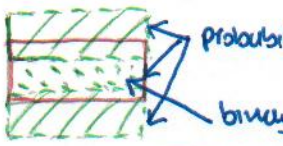
Note arithmetic codes (and symbol codes) are NOT error correcting so are used for compression prior to transmission over a noisy channel.

Demonstration of Arithmetic coding

let source string be $\underline{s} = [s_1 s_2 s_3 s_4 \ldots\ldots]$

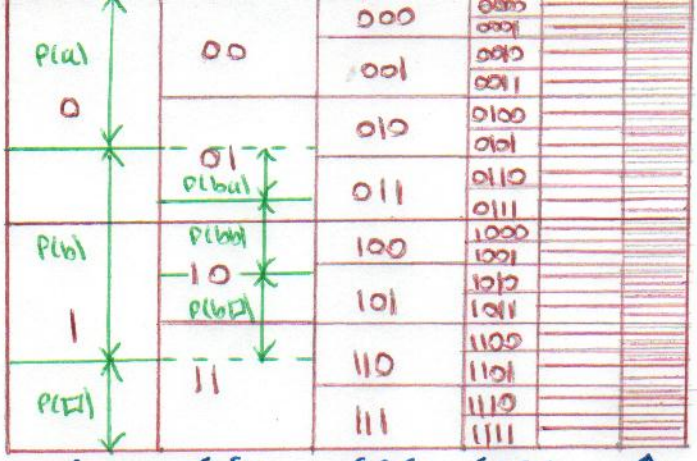
* superimpose probability model as intervals of $[0,1]$ on binary division map.
* recall of code as which is the longest interval within the binary division map. i.e 1... ⇒ b most likely, 10... ⇒ bb etc.

i.e



{ Encode, decode in same way *

in this case  is most likely



* i.e decoder needs model.

Simple (Bayesian) Arithmetic coding probability model.

consider alphabet $A_x = \{a, b, \square\}$ $\square$ ⇒ 'end of message'.
Assume characters are selected RANDOMLY with probabilities $\mathcal{P}_x = \{P_a, P_b, P_\square\}$.
excluding strings with $\square$ terminating them let:

BERNOULLI DISTRIBUTION $P(\underline{s}|P_a, F) = P_a^{F_a}(1-P_a)^{F_b}/N$ ← normalising constant

where strings so far (no $\square$ yet) has F_a 'a's and F_b 'b's.

From Bayes theorem: $P(P_a|\underline{s}, F) = \dfrac{P(\underline{s}|P_a, F)P(P_a)}{P(\underline{s}|F)} = P_a^{F_a}(1-P_a)^{F_b} P(P_a)/P(\underline{s}|F)$

lumping normalisation N into $P(\underline{s}|F)$.
If $P(P_a)$ is given by the distribution. (i.e constant).

$P(\underline{s}|F) = \int_0^1 dP_a\, P_a^{F_a}(1-P_a)^{F_b} P(P_a) = \dfrac{F_a! F_b!}{(F_a+F_b+1)!}$ ↑ Beta function

Now $P(a|\underline{s}, F) = \int_0^1 dP_a\, P(a|P_a) P(P_a|s, F)$. $P(a|P_a) = P_a$

$\therefore P(a|\underline{s}, F) = \int_0^1 dP_a\, P_a^{F_a+1}(1-P_a)^{F_b}/P(\underline{s}|F)$ using above analysis ⇒ $P(a|\underline{s}, F) = \dfrac{(F_a+1)! F_b!}{(F_a+F_b+2)!} \Big/ \dfrac{F_a! F_b!}{(F_a+F_b+1)!}$

⇒ $P(a|\underline{s}, F) = \dfrac{F_a+1}{F_a+F_b+2}$. Extending this result to longer alphabets $A_x = \{a_i, \square\}$

⇒ $P(a_i|\underline{s}, F) = \dfrac{F_{a_i}+\alpha}{\sum_i (F_{a_i}+\alpha)}$ where $\alpha = 1$. LAPLACE'S RULE.

If $\alpha \neq 1$ this corresponds to a DIRICHLET MODEL i.e a more general case.

Note probability over the characters is likely to be non uniform and prior $P(P_{a_i})$ may not be uniform either. Note Arithmetic codes can generate random numbers if a point is selected randomly between 0 and 1 and decoded (well binary encoded....)

5) Sampling from probability distributions. Often probability distributions are complex s.t. $Z_p = \int P^*(\underline{x})\, d^N\underline{x}$ (where $P^*(\underline{x})$ is unnormalised distribution - known) is hard to calculate. {Motivation for finding Z_p: Statistical physics - partition function etc....}

Laplace's method. * Assume $P^*(x)$ {1D example} has a peak at $x = x_0$. * Taylor expand $\ln P^*(x)|_{x=x_0}$. (Note first derivative = 0 since peak). $\ln P^*(x) \simeq \ln P^*(x_0) - \frac{c}{2}(x-x_0)^2 + \ldots$

where $c = -\dfrac{\partial^2}{\partial x^2} \ln P^*(x)|_{x=x_0}$. * Approximate $P^*(x)$ by unnormalised Gaussian $Q^*(x) = P^*(x_0) e^{-\frac{c}{2}(x-x_0)^2}$

$Z_Q = \int_{-\infty}^{\infty} Q^*(x)\, dx = P^*(x_0)\sqrt{\dfrac{2\pi}{c}}$. * let $Z_P \cong Z_Q$.

For $P^*(\underline{x})$ over k dimensional space:

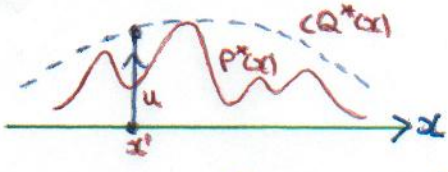
Define matrix $\underline{\underline{A}}$ s.t. $A_{ij} = -\dfrac{\partial^2}{\partial x_i \partial x_j} \ln P^*(\underline{x})|_{\underline{x}=\underline{x}_0}$; $Z_P \simeq Z_Q = P^*(\underline{x}_0) \dfrac{1}{\sqrt{\det \frac{1}{2\pi}\underline{\underline{A}}}} = P^*(\underline{x}_0)\sqrt{\dfrac{(2\pi)^k}{\det \underline{\underline{A}}}}$

[Proof: $\int d^k\underline{x} \exp\left[-\frac{1}{2}\underline{x}^T \underline{\underline{A}}\, \underline{x}\right] = \prod_{i=1}^{k} \int du_i \exp\left[-\frac{1}{2}\lambda_i u_i^2\right] = \prod_{i=1}^{k} \sqrt{\dfrac{2\pi}{\lambda_i}}$. $\prod_i \lambda_i = \det \underline{\underline{A}}$ ← known result.
- change basis $\underline{x}$ to one where $\underline{\underline{A}}$ is diagonalised - elements are eigenvalues λ_i].

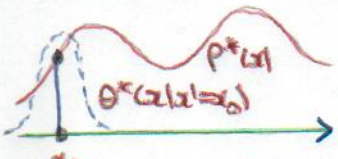
Monte Carlo Methods can show 'volume' of the typical set of an N dimensional distribution $\sim 2^{H(\underline{x})}$. So can become enormous. A similar approach to Laplace's method is IMPORTANCE SAMPLING where simpler distribution $Q(\underline{x}) = Q^*(\underline{x})/Z_Q$ is used to approximate $P(\underline{x})$

IMPORTANCE SAMPLING CONT.... Define estimator $\hat{\Phi}[\underline{x}] \approx \sum_i w_i \phi(\underline{x}) / \sum_i w_i$ where $w_i = \frac{P^*(\underline{x})}{Q^*(\underline{x})}$ and $\phi(\underline{x})$ is function whose expectation $\hat{\Phi}[\underline{x}]$ estimates. I.e mean if $\phi(x') = x'$, variance + mean² if $\phi(x') = x'^2$. TWO DIFFICULTIES: (i) long time to sample from typical set unless $Q \approx P$ (ii) weights of samples within typical set will vary by large factors because probabilities of points differ by $\sim e^{\sqrt{N}}$ (Analysis involving N dimensional Gaussians).

REJECTION SAMPLING. Find c s.t. $cQ^*(\underline{x}) > P^*(\underline{x})$ for all $\underline{x}$. * Randomly (uniformly) sample from $\{x\}$. * Evaluate $cQ^*(x)$ at sample point x'. * $u \sim$ uniform $(0, cQ^*(x'))$ * Evaluate $P^*(x')$. * If $u > P^*(x)$ REJECT and start again. * Otherwise accept x' as valid sample. GOOD if $P \approx Q$. If Q bad approx to P ⇒ c large ⇒ rejection frequency high – so slow. Not good for high dimensions.



METROPOLIS-HASTINGS * Sample first point x_0 randomly from $\{x\}$ uniformly with $P^*(x)$. * Evaluate $P^*(x_0)$. $Q^*(x|x_0)$ is defined (i.e a gaussian) and varies with x_0. (i.e mean of gaussian). * Sample next $x = x_1$ and compute: $a = \frac{P^*(x_1) Q^*(x_0|x_1)}{P^*(x_0) Q^*(x_1|x_0)}$. If $a \geq 1$ accept new point x_1.

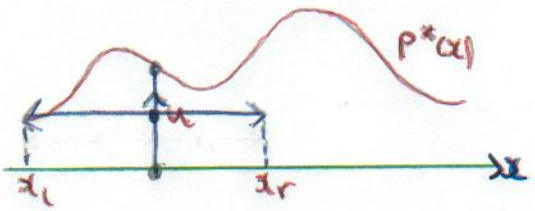


otherwise try again. Example of a MARKOV CHAIN where next step depends on the previous. * Random walk in simple space. I.e if typical dimension of sample space is $L = N\varepsilon$ where ε is the average step size and N is an integer: # steps in (an average) N^2. ∴ if τ is time interval between steps, total time to traverse L is $\sim N^2\tau = \frac{L^2}{\varepsilon^2}\tau$. Characteristic of all Monte Carlo methods listed here (from Newspeak) and $\frac{L^2}{\varepsilon^2}$ below.

GIBBS SAMPLING * Sample from (more tractable – assumed) CONDITIONAL DISTRIBUTION I.e $x_i^{(t+1)} \sim P(x_i^{(H)} | \{x_{j\neq i}^{(H)}\})$. [where x_i^{iH} is ith component of state $\underline{x}^{(t+1)}$].

Most (all?) Markov chain processes are ERGODIC I.e → to some invariant distribution as sample time → ∞. If not they may tend to a limit cycle between periodic irreducible sets.

SLICE SAMPLING * Sample point x' from $\{x\}$ (randomly). * Sample $u \sim$ Uniform $[0, P^*(x)]$. * Draw horizontal interval $[x_l, x_r]$ and uniformly sample from it. $x'' \sim$ uniform (x_l, x_r). * If $u > P^*(x'')$ modify x_l, x_r until $u \leq P^*(x'')$. Accept this x'' as a valid sample. PRO's – No rejections. CON's – random walk still, need to carefully adjust x_l, x_r s.t. DETAILED BALANCE is satisfied. I.e uniform distribution of (x,u) under $P^*(x)$ is invariant. (in what?)



VARIATIONAL METHODS Consider probability distributions of the form $P(\underline{x}|\beta,\underline{\Sigma}) = \frac{e^{-\beta E(\underline{x};\underline{\Sigma})}}{Z(\beta;\underline{\Sigma})}$ where $Z(\beta;\underline{\Sigma}) = \sum_{\underline{x}} e^{-\beta E(\underline{x};\underline{\Sigma})}$. As before Z is desirable but often intractable to calculate. As before approximate P by simpler distribution Q. Define $\beta\tilde{F}(\theta) = \sum_{\underline{x}} Q(\underline{x};\theta) \ln \frac{Q(\underline{x};\theta)}{e^{-\beta E(\underline{x};\underline{\Sigma})}} = D_{KL}(Q\|P) + \beta F$ where $\beta F \equiv -\ln Z$. VARIATIONAL FREE ENERGY

By Gibbs inequality $D_{KL}(Q\|P) \geq 0 \Rightarrow \tilde{F} - F \geq 0$ if $\beta > 0$. $\Rightarrow \tilde{F} \geq F$. So vary parameters $\{\theta\}$ (possibly more than one) to minimise $\tilde{F}$. Minimise $\tilde{F}$ and you will get a better estimate of F. Now $Z = e^{-\beta F}$ so can calculate Z this way.

ISING MODELS – Array of spins with couplings and applied field H that modify $E(\underline{x};\underline{\Sigma})$. Let $\underline{x}$ be state vector of a system with N spins where $x_n \in \{-1, +1\}$.

ISING MODELS CONT.... If NB spins m,n are neighbours with $(m,n) \in \mathcal{N}$: Energy of state $\underline{x}$ is:

$$E(\underline{x};\underline{J}) = -\left[\sum_{m,n\in\mathcal{N}} J x_m x_n + \sum_n H x_n\right]$$

J - coupling constant
H - " field
$J>0$ FERROMAGNETIC
$J<0$ ANTI "

can generalise: $J \to J_{mn}$, $H \to h_n$ {Spin glasses, Hopfield networks...}

ISING MODEL displays phase transitions as J, h, β are varied. {e.g. curie temp}. Note energy fluctuations have a peak. As an aside note relationship between heat capacity C and $\mathrm{Var}[E]$.

$C = \frac{\partial}{\partial T}\bar{E}$ where $\bar{E} = \frac{1}{Z}\sum_x e^{-\beta E(x)} E(x)$ {Boltzmann statistics

and $\beta = \frac{1}{k_B T}$. $Z = \sum_x e^{-\beta E(x)}$ $\therefore \frac{\partial \ln Z}{\partial \beta} = \frac{1}{Z}\sum_x (-E) e^{-\beta E} = -\bar{E}$. $\frac{\partial^2 \ln Z}{\partial \beta^2} = \mathrm{Var}(E)$

Now $\frac{\partial \bar{E}}{\partial T} = -\frac{\partial}{\partial T}\left(\frac{\partial \ln Z}{\partial \beta}\right) = -\frac{\partial^2 \ln Z}{\partial \beta^2}\frac{\partial \beta}{\partial T} = -\mathrm{Var}(E)\left(-\frac{1}{k_B T^2}\right) \Rightarrow C = \frac{\mathrm{Var}(E)}{k_B T^2}$.

Now variational free energy (defined above) $\tilde{F} = \frac{1}{\beta}\sum_x Q(x) \ln \frac{Q(x)}{e^{-\beta E}}$ ← natural by entropy

$\Rightarrow \beta\tilde{F} = \sum_x (Q \ln Q - Q \ln e^{-\beta E}) = -\sum_x Q \ln \frac{1}{Q} + \beta \sum_x Q E = \beta \langle E \rangle_Q - H_Q$.

Consider separable Q: $Q = \frac{1}{Z_Q}\exp\left[\sum_n a_n x_n\right] \Rightarrow H_Q = \sum_n H_2(q_n)$

Probability of spin $+1$

where $H_2(q_n) = q_n \ln \frac{1}{q_n} + (1-q_n)\ln \frac{1}{1-q_n}$ and $q_n = \frac{e^{a_n}}{e^{-a_n}+e^{a_n}} = \frac{1}{1+e^{-2a_n}}$

Using Σ convention $\langle E \rangle_Q = \left(+ J_{mn}\bar{x}_m\bar{x}_n + h_n \bar{x}_n\right)(-1)$

where $\bar{x}_n = \frac{e^{a_n} - e^{-a_n}}{e^{a_n}+e^{-a_n}} = \tanh(a_n) = 2q_n - 1$. So $\beta\tilde{F} = -\beta\left(J_{mn}\bar{x}_m\bar{x}_n + h_n\bar{x}_n\right) - \sum_n H_2(q_n)$

$\Rightarrow \frac{\partial \beta\tilde{F}}{\partial a_{n'}} = -\beta\left\{J_{mn}\left[\bar{x}_m\frac{\partial \bar{x}_n}{\partial a_{n'}} + \bar{x}_n\frac{\partial \bar{x}_m}{\partial a_{n'}}\right] + h_n\frac{\partial \bar{x}_n}{\partial a_{n'}}\right\} - \sum_n \frac{\partial H_2(q_n)}{\partial a_{n'}}$

{ m, m' are same thing — req'd to avoid Σ convention

Now $\frac{\partial \bar{x}_n}{\partial a_{n'}} = \frac{\partial \bar{x}_n}{\partial q_{n'}}\frac{\partial q_{n'}}{\partial a_{n'}} = 2\delta_{nn'}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)$ — NOT Σ over.. $\quad \frac{\partial H_2(q_n)}{\partial a_{n'}} = \frac{\partial H_2(q_n)}{\partial q_{n'}}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)$

$\therefore \frac{\partial \beta\tilde{F}}{\partial a_m} = -\beta\left\{J_{mn}\left[2\bar{x}_m\delta_{nn'}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right) + 2\bar{x}_n\delta_{mn'}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)\right] + 2h_n\delta_{nn'}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)\right\} - \sum_n \ln\left(\frac{1-q_n}{q_n}\right)\delta_{nn'}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)$

$= \ln\left(\frac{1-q_n}{q_n}\right)\delta_{nn'}\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)$

Now $\frac{1-q_n}{q_n} = \frac{1+e^{-2a_n}-1}{1+e^{-2a_n}}\cdot\left(1+e^{-2a_n}\right) = e^{-2a_n}$. $\therefore \ln\left(\frac{1-q_n}{q_n}\right) = -2a_n$.

{ This is cleaner if Σ convention is NOT used

$\therefore \frac{\partial \beta\tilde{F}}{\partial a_{n'}} = -2\beta\left(\frac{\partial q_{n'}}{\partial a_{n'}}\right)\left\{2J_{mn}\bar{x}_n + h_m\right\} + 2\frac{\partial q_{n'}}{\partial a_{n'}}a_m$

$\Rightarrow \frac{\partial \beta\tilde{F}}{\partial a_m} = 0$ when $a_m = \beta\sum_n\left(J_{mn}\bar{x}_n + h_m\right)$ where $\bar{x}_n = \tanh(a_n)$.

MEAN FIELD EQUATIONS FOR SPIN SYSTEM

6) NEURAL NETWORKS Unlike conventional CPU's and memory, organic brains are robust to error, associative (content addressable), distributed, massively ∥ ..... ye...

Idea of neural networks is to emulate this structure.

Single neuron

I inputs $\{x_i\}$

Activity $a = \sum_i w_i x_i$ Output $y(a) = f(a)$ $\{w_i\}$ weights.

$f(a) = a$

$f(a) = \frac{1}{1+e^{-a}}$ $[0,1]$

$f(a) = \tanh(a)$ $[-1,1]$

⋮

ARCHITECTURE: $\{w_i\}$; $f(a)$ ACTIVITY RULE: Dynamics of a with time.

LEARNING RULE: Dynamics of $\{w_i\}$.

Idea is to train neuron to give desired output t given input.
(Neuron → classifier). Simple learning rule: $\Delta w_i = \eta x_i \Delta y$ where $t = y + \Delta y$.

Capacity of single neuron with I inputs is $\sim 2I$ (2 bits / connection).
i.e. single neuron will almost certainly fail to memorize $> 2I$ states,
regardless of sophistication of learning rule.

Learning as inference: Good learning rule is to minimize $M(\underline{w})$
by changing $\underline{w} = \{w_i\}$. $M(\underline{w}) = G(\underline{w}) + \alpha E_W(\underline{w})$. $E_W(\underline{w}) = \frac{1}{2}\sum_i w_i^2$
(α constant)
$G(\underline{w}) = -\sum_n [t^{(n)} \log y^{(n)} + (1-t^{(n)}) \log(1-y^{(n)})]$ where $t^{(n)}$ is nth desired output.

* Interpret output $y(\underline{x};\underline{w})$ of neuron as (when $\underline{w}$ are specified) defining the probability that an input $\underline{x}$ belongs to a class $t=1$ rather than alternative $t=0$. Thus $y(\underline{x};\underline{w}) \equiv P(t=1|\underline{x},\underline{w})$. Then each value of $\underline{w}$ defines a different hypothesis about the probability of class 1 relative to class 0 as a function of $\underline{x}$.

So: $P(t=1|\underline{w},\underline{x}) = y$ $P(t=0|\underline{w},\underline{x}) = 1-y$ $\Rightarrow$ $P(t|\underline{w},\underline{x}) = y^t(1-y)^{1-t} = e^{t\log y + (1-t)\log(1-y)}$

Hence $P(t|\underline{w},\underline{x}) = e^{-G(\underline{w})}$ in general. $P(\underline{w}|\alpha) = \frac{1}{Z_W(\alpha)} e^{-\alpha E_W}$

(Data D; assume given inputs)

$\Rightarrow$ (Bayes theorem) $P(\underline{w}|D,\alpha) = \frac{P(D|\underline{w})P(\underline{w}|\alpha)}{P(D|\alpha)} = \frac{1}{Z_M} e^{-M(\underline{w})}$

So minimising $M(\underline{w})$ $\Rightarrow$ maximising $P(\underline{w}|D,\alpha)$.

Hopfield Network. - Example of a feedback network (as opposite to a feedforward one).

Learning rule is HEBB rule

$\frac{dw_{ij}}{dt} \sim \text{correlation}(x_i, x_j)$ (associative memory).

Capacity: $w_{ij} = x_i^{(n)} x_j^{(n)} + \sum_{n \neq n} x_i^{(n)} x_j^{(n)}$ (signal, noise)

$\Rightarrow a_i = \sum_{j \neq i} x_i^{(n)} x_j^{(n)} x_j^{(n)} + \sum_{j \neq i} \sum_{m \neq n} x_i^{(m)} x_j^{(m)} x_j^{(n)}$

$\Rightarrow E(a_i) = (I-1)x_i^{(n)}$

$\text{Var}(a_i) = (I-1)(N-1)$

Convention: w_{ij} denotes connection from neuron j to neuron i

Architecture: I neurons, all connected together through symmetric, bidirectional connections i.e. $w_{ij} = w_{ji}$.

Activity rule: $x(a) = \Theta(a) \equiv \begin{cases} 1 & a \geq 0 \\ -1 & a < 0 \end{cases}$

synchronous updates: All neurons compute their activation $a_i = \sum_j w_{ij} x_j$ then update their states simultaneously to $x_i = \Theta(a_i)$.

OR asynchronous updates: one neuron at a time computes its activation and updates its state. (Fixed or random sequence).

learning rule: intention MEMORIES are STABLE STATES of network's activity rule. Each memory is a binary pattern $x_i \in \{-1, 1\}$.

$w_{ij} = \eta \sum_n x_i^{(n)} x_j^{(n)}$ $\eta = \frac{1}{N}$ is often convenient (prevents largest possible weight going with N).

CONTINUOUS Hopfield network has:

$a_i = \sum_j w_{ij} x_j$ where $x_i = \tanh(a_i)$ or fix η and write $x_i = \tanh(\beta a_i)$
(β = gain). NOTE SIMILARITY TO ISING MODEL!!