Information Theory, Pattern Recognition and Neural Networks
Sketch of expected answers

PART III PHYSICS EXAM 2001

1

(a) An essay on hash codes and/or Hopfield networks is expected here. Examples of topics to mention include:

Hopfield network: Definition of the dynamics. Existence of a Lyapunov function, so the dynamics are known to be stable. The Hebb rule for storing patterns in a Hopfield network. Robustness of Hopfield network to parameter damage. Hopfield network can store multiple memories in a given piece of hardware. Capacity of memory is $0.14N$, *i.e.*, about 0.3 bits per connection. Signal-to-noise method for estimating capacity. Spurious memories. Pairs of nearby patterns are not so easy to memorize. Derivation of Hebbian learning rule from Boltzmann machine learning.

Hash codes: provide a (non-robust) way of recovering a memory given part of its content. Properties of random hash codes. Examples of simple hash functions. Scaling of the required size of hash table with number of memories, to avoid collisions. Speed-up offered by hash functions compared with, say, alphabetical storage.

(b) Description of Metropolis method and Gibbs sampling.

Gibbs sampling requires sampling from conditional densities. (Not necessarily possible.)

Metropolis requires only evaluation of the target density at a given point. But depends on a sensible choice of proposal density. If choice is bad then method will go nowhere. Gibbs can be viewed as a special case of Metropolis. Gibbs has no parameters (nice).

Both methods suffer from random walk effects (at least, if Metropolis means a standard method with a simple proposal density). Time per independent sample scales as $(L/\epsilon)^2$. [But neither method has catastrophic failure in high dimensions.] Both methods are superior to rejection sampling because they do get around eventually, whereas rejection sampling in high dimensions might never produce a single point.

Random walk behaviour can be reduced by Hybrid Monte Carlo (Metropolis with gradient information); and by overrelaxation (Gibbs sampling variant).

Problem of setting step size can be evaded by using Slice sampling.

With all these methods, difficult to detect convergence. Exact sampling method offers an answer to this question in some cases.

2

(a) $I(X;Y) = H(X) - H(X|Y)$.

$I(X;Y) = H_2(p_0) - qH_2(p_0)$.

Maximize over p_0, get $C = 1 - q$.

(b) The (2,1) code is $\{\mathtt{01}, \mathtt{10}\}$. With probability q, the `1` is lost, giving the output `00`, which is equivalent to the "?" output of the Binary Erasure Channel. With probability $(1-q)$ there is no error; the two input words and the same two output words are identified with the `0` and `1` of the BEC. The equivalent BEC has erasure probability q. Now, this shows the capacity of the Z channel is at least half that of the BEC.

(c) The (7,4) Hamming code can detect and correct at most one flip; an error occurs if there are two or more.

[Either] The probability of block error is dominated by the probability of two flips, which is about $\binom{7}{2} f^2$.

[Or] The probability of error is

$$\sum_{r \geq 2} \binom{N}{r} f^r (1-f)^r$$

which is about 0.002.

(d) Bayes' theorem

$$\begin{aligned}
\log \frac{P(s=1|\mathbf{r})}{P(s=2|\mathbf{r})} &= \log \frac{P(\mathbf{r}|s=1)P(s=1)}{P(\mathbf{r}|s=2)P(s=2)} \\
&= \log \left(\frac{1-f}{f}\right)^{2r_1 - 1} + \log \left(\frac{1-f}{f}\right)^{-(2r_3-1)} + \log \frac{P(s=1)}{P(s=2)} \\
&= w_1 r_1 + w_3 r_3 + w_0,
\end{aligned}$$

where

$$\begin{aligned}
w_1 &= 2 \log \left(\frac{1-f}{f}\right) \\
w_3 &= -2 \log \left(\frac{1-f}{f}\right) \\
w_0 &= \log \frac{P(s=1)}{P(s=2)} \\
(w_2 &= 0)
\end{aligned}$$

which we can rearrange to give

$$P(s=1|\mathbf{r}) = \frac{1}{1 + \exp\left(-w_0 - \sum_{n=1}^{3} w_n r_n\right)}.$$

This can be viewed as a neuron with two or three inputs, one from r_1 with a positive weight, and one from r_3 with a negative weight, and a bias. (Picture here of blob with three lines.)

3

(a) Using the Huffman algorithm we arrive at this symbol code

a_i	p_i	$\log_2 \frac{1}{p_i}$	l_i	$c(a_i)$
111	1e-06	19.9	5	00000
110	9.9e-05	13.3	5	00001
101	9.9e-05	13.3	5	00010
011	9.9e-05	13.3	5	00011
001	0.0098	6.7	3	001
010	0.0098	6.7	3	010
100	0.0098	6.7	3	011
000	0.97	0.0	1	1

The expected length is 1.06, (but 1 would be accurate enough!) and the entropy of $\mathbf{x}$ is 0.24. The ratio length / entropy is 4.4. (Answer 4 to 1 decimal place.)

Arithmetic code $N = 1000$: expected length is N times the entropy, i.e. 80 bits.

Variance is found from variance of the number of 1s, which is Npq, and the factor is $\log_2[(1-f)/f]$, so the standard deviation is $3.14 * \log_2[(1-f)/f] = 21$.

Final answer should be 80 ± 21 bits.

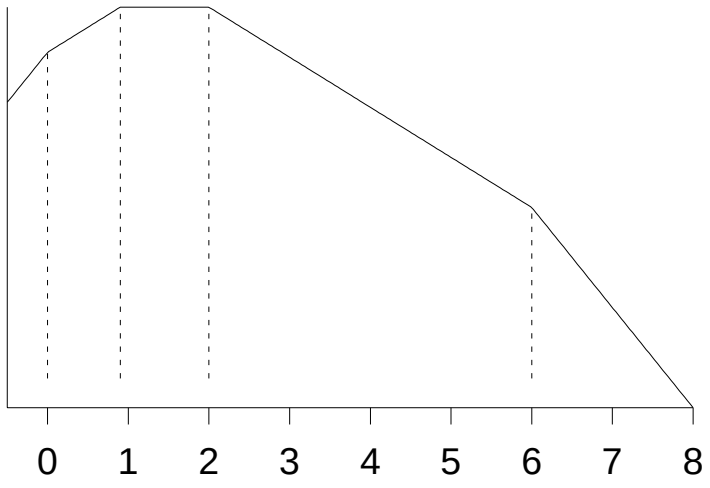
(b) Bayes theorem:

$$P(\mu|\{x_n\}) = \frac{P(\mu)\prod_n P(x_n|\mu)}{P(\{x_n\})}$$

The likelihood function contains a complete summary of what the experiment tells us about μ. Expression for the log likelihood,

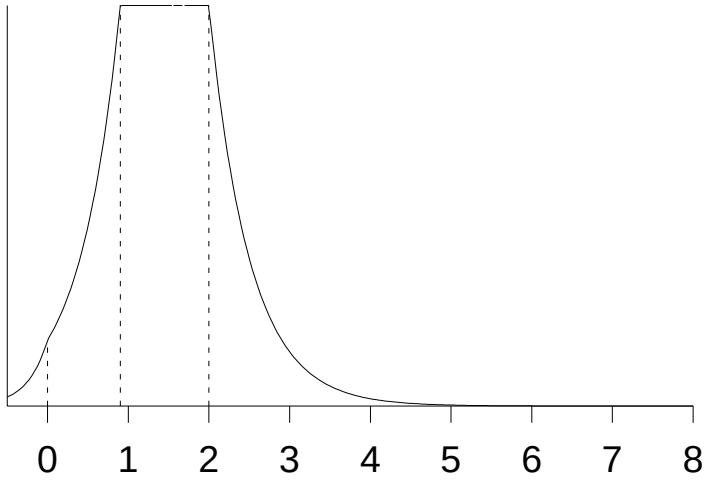
$$L(\mu) = -\sum_n |x_n - \mu|.$$

Sketch of likelihood function on a log scale.



Note gradient changes by 2 as you pass each data point. Gradients are 4, 2, 0, -2, -4.

Sketch of likelihood function on a linear scale.



Exponential functions have lengthscales 1/4, 1/2, 1/2, 1/4.

The most probable values of μ are 0.9–2, and the posterior probability falls by a factor of e^2 once we reach -0.1 and 3, so a range of plausible 0 values for μ is $(-0.1, 3)$.