Tutorial: Information Theory and Statistics

Bin Yu
Statistics Department, EECS Department
UC Berkeley
www.stat.berkeley.edu/users/binyu

ICMLA 2008, San Diego
Entropy: physics origin

Clausius, Boltzman, Gibbs
Rudolf Julius Emmanuel Clausius

Born: 2 Jan 1822 in Koslin, Prussia (now Koszalin, Poland)
Died: 24 Aug 1888 in Bonn, Germany

Clausius was a theoretical physicist who played an important role in establishing theoretical physics as a discipline. His most famous paper was read to the Berlin Academy on 18 February 1850 and published in Annalen der Physik in the same year. This paper marks the foundation of the modern thermodynamics. In his paper of 1865 Clausius stated the First and Second laws of thermodynamics in the following form.

1. The energy of the universe is constant.
2. The entropy of the universe tends to a maximum.
1850: Idea of entropy by Clausius, but not the term. However, he had
\[
\bar{d}Q = dU + \bar{d}W,
\]
where \(Q\) is the heat, \(U\) is the intrinsic energy, and \(W\) is the external work.

1865: Entropy formula first appeared. Clausius refined first and second laws of Thermodynamics using entropy \(S\)
\[
dS = \bar{d}Q/T,
\]
where \(Q\) is the heat or internal energy and \(T\) the temperature.
Ludwig Boltzmann

Born: 20 Feb 1844 in Vienna, Austria  
Died: 5 Oct 1906 in Duino (near Trieste), Austria (now Italy)

A theoretical physicist at Vienna (and Graz), Boltzmann’s fame is based on his invention of statistical mechanics. This he did independently of Willard Gibbs. Their theories connected the properties and behaviour of atoms and molecules with the large scale properties and behaviour of the substances of which they were the building blocks.
1877: Boltzmann quantifies entropy of an equilibrium thermodynamic system as

\[ S = K \log W, \]

where \( S \) - entropy, \( K \) - Boltzmann constant, \( W \) - number of microstates in the system. It has been said that Planck was the first who wrote this down.
Born: 11 Feb 1839 in New Haven, Connecticut, USA

Died: 28 April 1903 in New Haven, Connecticut, USA

A Europe-trained mathematical physicist at Yale College. His work on statistical mechanics provided a mathematical framework for quantum theory and for Maxwell’s theories. His last publication, Elementary Principles in Statistical Mechanics, beautifully lays a firm foundation for statistical mechanics.
1870’s: Gibbs gives a general entropy expression for a thermodynamic system:

\[ S = - \sum_j p_j \log p_j, \]

where \( p_j \) is the probability that the system is at microstate \( j \).

If \( p_j = 1/W \), then Gibbs’ definition agrees with Boltzman’s.
Claude Elwood Shannon’s entropy in communication theory

Born in Gaylord, Michigan, on April 30, 1916

Shannon is considered as the founding father of electronic communications age. His work on technical and engineering problems within the communications industry laid the groundwork for both the computer industry and telecommunications.
Shannon (1948): A Mathematical Theory of Communication

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

In Shannon’s theory, a message is a random draw from a probability distribution on messages and entropy gives the data compression (source coding) limit.
Shannon’s information theory

Shannon’s entropy measures ”information” content in a message, but this ”information” is not the meaningful information. It is simply the uncertainty in the message just as Boltzmann-Gibbs entropy measures the disorder in a thermodynamic system.

Shannon’s theory concerns with point-to-point communications as in telephony and gives limits on coding. It consists of:

1. Source coding: limits on data storage (transmission over noiseless channels) (our text files are stored in binary format or through a binary code on our computers)

2. Channel coding: limits on data transmission over noisy channels.
What is a code?

Given a discrete *alphabet* \( \mathcal{X} \) of message symbols, a binary *code* is a mapping from *symbols* in \( \mathcal{X} \) to a set of *codewords* of binary strings.

\( \mathcal{X} \) could be the Roman letters, numbers, and other writing symbols.

Example: \( \mathcal{X} = \{a, b, c\} \). Here is one binary code:

\[
\begin{align*}
  a & \rightarrow 00 \\
  b & \rightarrow 01 \\
  c & \rightarrow 10
\end{align*}
\]

\( aabacbcbaa \rightarrow 0000100100110010000 \)

\( bcccbabbea \rightarrow 01101010010001101000 \).

Each requiring 20 bits (binary digits by Tukey).
What is decoding?

Decoding involves splitting the encoded string into segments or sub-strings of 0’1 and 1’s, and then performing a table lookup to see which symbol is associated with each segment.

Another code:

\[
\begin{align*}
a & \rightarrow 0 \\
b & \rightarrow 10 \\
c & \rightarrow 11
\end{align*}
\]  

\(aabacbcbaa \rightarrow 0010011101110000\)

\(bccebabcca \rightarrow 10111111001011110.\)

This is a prefix code; that is, no codeword is the prefix of another.

This property means that encoded messages are uniquely decodable, even if we don’t include separation commas between codewords.
Examples of real codes: ASCII and Unicode

a. **ASCII**- the American Standard Code for Information Interchange:

7 bit binary code for the alphabet set of Roman letters, digits, mathematical symbols, and punctuations.

Used for storing and transmitting English documents.

Each symbol is mapped into a digit between $0 - 127 = 2^7 - 1$ and then this number can be coded with 7 bits.

b. **Unicode** is a 16-bit code that assigns a unique number to every character or symbol in use, from Bengali to Braille. $2^{16} = 65,000$ codewords. They are fixed length codes.
Examples of real codes: Morse code

Original Morse code: \(\lambda\) contains dots, dashes, and pauses.

Modern version by A. Vail: \(\lambda\) contains Roman letters, numbers and punctuation.

Codewords: strings from dot, dash, short gap, medium gap, long gap.

Design strategy: frequently used letters are assigned short code words for compression. For example, ”e” is a single dot, ”t” is a single dash. Two-symbol codewords are given to ”a”, ”i”, ”m” and ”n”. They got their frequencies from printer’s type box.
Examples of real codes: Braille

L. Braille (1829): 6-bit code allows 63 possible codes (flat or raised dot).

26 are used to encode Roman letters, the rest to encode common words (and, for, of, the, with) and common two letter combinations (ch, gh, sh, th, wh, ed, er, ou, ow).

Direct use of frequent structures leads to better compression.
Lab Problem I: Decoding Braille

Using the code book given, decode the Braille text into English.

```
<table>
<thead>
<tr>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Text to be decoded:

```
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . |
```
Codes and Probability Distributions

Given a binary code on \( \mathcal{X} \), the length function \( L \) maps symbols in \( \mathcal{X} \) to the length of their codeword in bits. Using the code in (0.2), we have \( L(a) = 1 \) and \( L(b) = L(c) = 2 \).

In general, there is a correspondence between the length function of a prefix code and the quantity \(- \log_2 Q\) for a probability distribution \( Q \) defined on \( \mathcal{X} \).

Kraft’s inequality:

For any binary prefix code, the code length function \( L \) must satisfy the inequality

\[
\sum_{x \in \mathcal{X}} 2^{-L(x)} \leq 1. \tag{0.3}
\]

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a prefix binary code with these code lengths.

Proof: Use the one-one mapping of a binary prefix code and a binary tree with codewords only on the end-nodes.
Prefix codes and probability distributions are equivalent

Using the Kraft inequality, we can take any length function $L$ and construct a distribution as follows

$$Q(x) = \frac{2^{-L(x)}}{\sum_{x \in \mathcal{X}} 2^{-L(x)}} \quad \text{for any } a \in \mathcal{X}. \quad (0.4)$$

Conversely, for any distribution $Q$ on $\mathcal{X}$ and any $x \in \mathcal{X}$, we can find a prefix code with length function

$$L(x) = \lceil -\log Q(x) \rceil,$$

the smallest integer greater than or equal to $-\log Q(a)$, because

$$L(x) \geq \log Q(x), \text{ hence } -L(x) \leq \log Q(x),$$

and it follows that

$$\sum 2^{-L(x)} \leq \sum 2^{\log Q(x)} = \sum Q(x) = 1.$$
Making frequency counting precise (Shannon, 1948)

Suppose our messages are constructed by randomly selecting elements of $\mathcal{X}$ according to a distribution $P$. Then, the expected length of a code is given by

$$EL = \sum_{x \in \mathcal{X}} P(x) L(x).$$  \hspace{1cm} (0.5)

Entropy serving as a lower bound:

Suppose the elements of $\mathcal{X}$ are generated according to a probability distribution $P$. For any prefix code on $\mathcal{X}$ with length function $L(\cdot)$, the expected code length $L$ is bounded below

$$L \geq - \sum_{x \in \mathcal{X}} P(x) \log_2 P(x) = H(P)$$  \hspace{1cm} (0.6)

where equality holds if and only if $L(x) = - \log_2 P(x)$. 
Proof

By Kraft’s inequality, \( C_L = \sum_x 2^{-L(x)} \leq 1 \).

Then
\[
Q(x) = 2^{-L(x)}/C_L.
\]
is a probability distribution.

Since \( E_P L = E_P[-\log(Q(X)) - \log C_L] \geq E_P[-\log(Q(X))], \)
\[
EP_L - H(P) \geq -E_P \log \frac{Q(X)}{P(X)} \geq* - \log E_P \frac{Q(X)}{P(X)} = - \log 1 = 0.
\]

\( \geq* \) holds because of Jensen’s inequality applied to the convex function \(-\log:\)
for \( Y = Q(X)/P(X) \)
\[
(E - \log Y) \geq - \log EY, \quad \text{which is the same as} \quad -(E \log Y) \geq - \log EY.
\]
The inequality \( E_P \log \frac{Q(X)}{P(X)} \geq 0 \) is called the \textit{Information Inequality}. 
Formal Definition of Entropy

Given a probability function $P$ defined on a discrete alphabet $\mathcal{X}$, we define the entropy $H(P)$ to be

$$H(P) = -\sum_{x \in \mathcal{X}} P(x) \log P(x) = -E_P \log P(X).$$ (0.7)

The logarithm in this expression is usually in base 2, and the units of entropy are referred to as *bits* (coined by Tukey).
Coding algorithms when $P$ is known

Given $P$ on $\mathcal{X} = \{x_1, \ldots, x_k\}$, let code length function be

$L^*(x) = \lceil -\log P(x) \rceil$.

$$\sum_{x \in \mathcal{X}} 2^{-\lceil -\log P(x) \rceil} \leq \sum_{x \in \mathcal{X}} 2^{\log P(x)} = \sum_{x \in \mathcal{X}} P(x) = 1.$$  

(0.8)

Therefore, by Kraft’s inequality $L^*$ corresponds to prefix code. Since the ceiling operator introduces an error of at most one bit,

$$H(P) \leq E L^* \leq H(P) + 1$$

(0.9)

from Information inequality.
Shannon code with known $P$

Suppose $P(x_1) \geq P(x_2) \geq \cdots \geq P(x_k)$, and let

$$F_i = \sum_{j=1}^{i-1} P(x_j),$$

which is the CDF function at $x_{i-1}$.

Shannon code:

$$x_i \rightarrow F_i \text{ rounded to } \lceil -\log P(x_i) \rceil \text{ bits.}$$

Obviously this code has code length $L^*(x_i) = \lceil -\log P(x_i) \rceil$. 
Shannon code: example

Let \( P \) be a distribution on \( \{a, b, c\} \) with probability \( 11/20, 1/4, 1/5 \). Its entropy \( H(P) = 1.439 \) bits. Then

\[
F_1 = 0, \quad F_2 = 11/20, \quad F_3 = 4/5,
\]

\[
\lceil -\log P_1 \rceil = \lceil 0.86 \rceil = 1; \quad \lceil -\log P_2 \rceil = \lceil 2 \rceil = 2; \quad \lceil -\log P_3 \rceil = \lceil 2.3 \rceil = 3
\]

Mapping \( F_i \)'s to the right numbers of bits:

\[
C(F_1) = 0, \quad C(F_2) = 10, \quad C(F_3) = 110,
\]

because

\[
F_1 = 0, \quad F_2 = 11/20 = 1/2 + 1/20; \quad F_3 = 4/5 = 1/2 + 1/4 + 1/20.
\]

Expected code length \( L = 11/20 + 2 \times 1/4 + 3 \times 1/5 = 8/5 = 1.6 \) bits.

which is within 1 bit of the entropy 1.439 bits.
Huffman code with known $P$

Shannon’s code achieves entropy when it is applied to $n$-blocks of messages. The optimal coding question remained open for a finite alphabet. Huffman (1952) solved this problem by designing the now so-called Huffman code, which constructs in a bottom-up and greedy fashion.

”Growing” the coding tree from the leave nodes:

for the $P$ with $P(a) = 11/20$, and $P(b) = 1/4$, and $P(c) = 1/5$

- find the two elements $b$ and $c$ with the smallest probabilities and connect them with leaves 0 and 1 to form the intermediate node $bc$ and assign it probability the sum of $P(b)$ and $P(c)$ which is $9/20$.

- iterate the process until no element is left.
This gives the Huffman code:

\[ a \rightarrow 0, \quad b \rightarrow 10, \quad c \rightarrow 11, \]

with an expected code length

\[ 1 \times \frac{11}{20} + 2 \times \frac{1}{4} + 2 \times \frac{1}{5} = 1.45, \]

which is better than Shannon code’s 1.6 bits, and much closer to the entropy rate 1.439 bits.

Huffman code is OPTIMAL because it gives the prefix code with the smallest expected code length.
Shannon code reaches entropy limit for iid data almost surely

Asymptotic Equal Partition (AEP)

Suppose $X_1, \ldots, X_n$ iid, then average code length for the Shannon code on $n$-tuples goes to the entropy almost surely.

\[
\frac{1}{n} L^*(X_1, \ldots, X_n) \rightarrow H(P).
\]

Proof:

\[
L^*(X_1, \ldots, X_n)/n = -\log P(X^n)/n + O(1/n).
\]

LLN applied to $-\log P(X^n)/n = \sum [-\log P(X_i)]/n$.

$n$-block codes achieve entropy rate!

Shannon-McMillian-Breiman theorem: AEP for stationary and ergodic processes.
Proof of AEP: uniform distribution in n-tuple space

In the space of n-tuple sequences \( \{1, \ldots, k\}^n \),

\[
P(X^n) \approx 2^{-nH(P)}
\]

by LLN for "most" sequences.

This gives an approximate uniform distribution and its entropy is the log of the number of possibilities, that is,

\[
\log 1/2^{-nH(P)} = nH(P).
\]

So Boltzman's definition of entropy in the uniform case can be seen the general one in this sense.
Arithmetic code with known $P$

Both Shannon and Huffman codes require sorting, expensive for large alphabets (large blocks) and not so easy to update when the block size changes.

Shannon-Fano-Elias is a modification on Shannon code:

Let

$$
\bar{F}(x_i) = \sum_{j=1}^{i-1} P(x_j) + \frac{1}{2} P(x_i).
$$

Map:

$$
x_i \rightarrow \bar{F}(x_i) \text{ rounded to } \lceil -\log P(x_i) \rceil + 1 \text{ bits}.
$$

This code is easily updated when more symbols come along. It acquired a new name *Arithmetic Code*, when applied to blocks.
Given a random variable $X$ with a probability distribution $P$ defined on a discrete alphabet $\mathcal{X}$, we define the entropy $H(X)$ or $H(P)$ to be

$$H(X) = H(P) = - \sum_{x \in \mathcal{X}} P(x) \log P(x) = -E_P \log P(X). \quad (0.10)$$

The logarithm in this expression is usually in base 2, and the units of entropy are referred to as *bits* (coined by Tukey).
Examples of entropy

If $X$ is Bernoulli($p$), then

$$H(p) = -p \log p - (1 - p) \log(1 - p), \quad \text{while } V(x) = p(1 - p).$$

Both are maximized at $p = 1/2$. And $H(p)$ is much steeper at $p = 0, 1$ than at the $p = 1/2$, although

$$H''(p) = -\frac{1}{p(1 - p)} = -4 \quad \text{at } p = 1/2.$$
Examples of entropy

Poisson ($\mu$):

$$H_e(X) = \mu - \mu \log_e \mu + E \log_e X!$$

Geometric (p):

$$H(X) = H(p)/p$$

Multinomial ($n; p_1, \ldots, p_k$):

$$n \sum_{j=1}^{k} -p_j \log p_j$$
Axioms for the entropy function (Shannon, 1948)

Continuity  \( H(p_1, \ldots, p_k) \) is a continuous function of the vector \( p \). This makes sense because we would not want small changes in \( p \) to yield large differences in information.

Monotonicity  For \( p_j = 1/k \), the entropy \( H(1/k, \ldots, 1/k) \) should be an increasing function of \( k \). When dealing with choices between equally likely events, there is more choice or uncertainty when there are more possible events.

Conditioning  \[ H_k(p_1, \ldots, p_k) = H_{k-1}(p_1 + p_2, p_2, \ldots, p_k) + (p_1 + p_2)H_2\left(\frac{p_1}{p_1+p_2}, \frac{p_2}{p_1+p_2}\right) \]

If a nonnegative function satisfying the above, it has to be the entropy function (up to a constant).

Joint Entropy and Conditional Entropy

For a pair of random variables \((X, Y)\) with a joint distribution \(P(x, y)\), we can define the joint entropy

\[
H(X, Y) = \sum_x \sum_y P(x, y) \left[- \log P(x, y)\right].
\]

The conditional entropy of \(H(X|Y)\) is the average entropy of \(X\) given \(Y\):

\[
H(X|Y) = E_Y H(X|Y = y) = \sum_y H(X|Y = y) P(Y = y).
\]
Similarly we can define the joint entropy for any $X_1, \ldots, X_n$

$$H(X_1, \ldots, X_n) = \sum_{x_1, \ldots, x_n} P(x_1, \ldots, x_n) [\log P(x_1, \ldots, x_n)].$$

It follows that if $X_1, \ldots, X_n$ are iid, then

$$H(X_1, \ldots, X_n) = nH(X_1),$$

with the special case that for a binomial random variable $X$ with a success probability $p$:

$$H(X) = nH(p).$$
Properties of Entropy

$P$ is a distribution on $\{x_1, \ldots, x_k\}$.

1. $H(P) \geq 0$

2. $H_b(P) = (\log_b a)H_a(P)$

3. $H(X|Y) \leq H(X)$ with equality only if $X$ and $Y$ are independent, where

   $$H(X|Y) = \sum_y P(Y = y)H(X|Y = y).$$

4. $H(X_1, \ldots, X_n) \leq \sum_{i=1}^n H(X_i)$ with equality only if the $X_i$ are independent.

5. $H(P) \leq \log k$ with equality only if $P$ is uniform on $\mathcal{X}$.

6. $H(P)$ is concave in $P$. 
A sketchy proof:

1. Obvious because $- \log P(x) \geq 0$.

2. Obvious because of the base change formula for $\log$ function.

3. Intuitively obvious because conditioning reduces uncertainty.

$$H(X) - H(X|Y) = I(X,Y) = E \log(P(X,Y)/P(X)P(Y)) \geq 0$$

by the information inequality

4. Use (3) and chain rule:

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}, ..., X_1).$$

5. Because $H(P)$ is maximized when $P(x) = 1/K$. That is, the solution of $\max H(P)$ subject to $\sum_x P(x) = 1$ is $P(x) = 1/K$ – uniform is the maxent distribution with no contraints.
6. It follows from the concavity of \( \log \).
Entropy rate for stationary sequences

For a stationary sequence $X_1, \ldots, X_n, \ldots$, we can define its entropy rate as

$$H(X^\infty) = H(X_1, \ldots, X_n, \ldots) = \lim_{n \to \infty} \frac{H(X_1, \ldots, X_n)}{n},$$

which also equals to

$$\lim_{n \to \infty} \frac{H(X_n|X_{n-1}, \ldots, X_1)}{n}.$$

Proof: It follows from two facts:

1. $H(X_n|X_{n-1}, \ldots, X_1)$ is a non-negative decreasing sequence by property (3) and stationarity. so it has a limit.

2. The chain rule and Cesaro mean:

$$\frac{H(X_1, \ldots, X_n)}{n} = \frac{1}{n} \sum_{i} H(X_i|X_{i-1}, \ldots, X_1).$$
Entropy rate of a Markov sequence

For a stationary and ergodic Markov chain with stationary distribution $\pi_i$ and transition matrix $p_{ij}$,

The entropy rate is

$$\sum_i \pi_i \sum_j [-p_{ij} \log p_{ij}].$$

For a two-state Markov chain with a transition Matrix

$$\begin{pmatrix} 1 - p_1 & p_1 \\ p_2 & 1 - p_2 \end{pmatrix}$$

and stationary distribution

$$\pi_1 = \frac{p_2}{p_1 + p_2}; \pi_2 = \frac{p_1}{p_1 + p_2}.$$
The entropy of $X_n$ is

$$H(X_n) = H(\pi_1) = H\left(\frac{p_2}{p_1 + p_2}\right).$$

However, the entropy rate of the sequence is LOWER due to the dependence and it is

$$H(X_2|X_1) = \frac{p_2}{p_1 + p_2} H(p_1) + \frac{p_1}{p_1 + p_2} H(p_2)$$

For low flip rates $p_1 = 0.01$, $p_2 = 0.02$, the marginal entropy is

$$H(\pi_1) = H(2/3) = 0.92\text{ (bits)},$$

while the entropy rate of the sequence is

$$\frac{2}{3} \times H(0.01) + \frac{1}{3} \times H(0.02) = 2 \times 0.08/3 + 0.14/3 = 0.1\text{ (bits)}.$$
A Version of Second Law of Thermodynamics

For a stationary Markov sequence $X_1, \ldots, X_n, \ldots$, the conditional entropy $H(X_n|X_1)$ is non-decreasing, while the marginal entropy $H(X_n)$ is fixed.

Proof:

$$H(X_n|X_1) \geq H(X_n|X_1, X_2) \quad (property \ (3))$$
$$= H(X_n|X_2) \quad (Markov \ property)$$
$$= H(X_{n-1}|X_1) \quad (Stationarity)$$
Motifs: chromosome regions with specific biological structural significance or function. Usually short: 6-20 base pairs. Examples: splice sites, transcription factor binding sites, translation initiation sites, enhancers, silencers.

The table below is a weight matrix learned from 15,155 mamalian donor sites (exon and intron junctions) from the SpliceDB database. Entries are frequencies of bases at each position.

<table>
<thead>
<tr>
<th>Base</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>33</td>
<td>61</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>53</td>
<td>71</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>C</td>
<td>37</td>
<td>13</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>8</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>G</td>
<td>18</td>
<td>12</td>
<td>80</td>
<td>100</td>
<td>0</td>
<td>42</td>
<td>12</td>
<td>81</td>
<td>22</td>
</tr>
<tr>
<td>T</td>
<td>12</td>
<td>14</td>
<td>7</td>
<td>0</td>
<td>100</td>
<td>2</td>
<td>9</td>
<td>6</td>
<td>46</td>
</tr>
</tbody>
</table>
A graphical method to display patterns in a set of aligned sequences:

- Height of stack at each position is the "information" content from the frequencies:

  \[
  \text{max. entropy} - \text{estimated entropy} = 2 - \text{estimated entropy}
  \]

- Letters (A, T, G, C) are arranged in decreasing order of frequency whose heights are proportional to the frequencies.

The entropy estimate is the plug-in estimate.

Thanks to Xiaoyue Zhao and Terry Speed for providing the data.
Given an iid sequence $X_1, \ldots, X_n$ with probabilities $p_1, \ldots, p_k$ on \{1, ..., $k$\}, $N_j = \sum_i I(X_i = j)$ for $j = 1, \ldots, k$ are multinomial, and MLE of $p$’s are

$$\hat{p}_j = N_j / n.$$

Then the plug-in MLE of $H(X)$ is

$$\hat{H} = H(\hat{p}_1, \ldots, \hat{p}_k) = -\sum_{j=1}^{k} \frac{N_j}{n} \log \frac{N_j}{n}.$$
Miller (1954) showed that

\[ H - \hat{H} = \sum_j \frac{N_j}{n} \log \frac{N_j}{n p_j} + \sum_j \left\{ \frac{N_j}{n} - p_j \right\} \log p_j \]

Bias

\[ E(H - \hat{H}) = E\left( \sum_j \frac{N_j}{n} \log \frac{N_j}{n p_j} \right), \]

because the second term has expectation zero.

\[ 2 \sum_j N_j \log \frac{N_j}{n p_j} \] has an approximate \( \chi^2_{k-1} \) distribution; hence

\[ E(H - \hat{H}) \approx \frac{(k - 1)}{2n} + O(1/n^2). \]

The \( 1/n^2 \) term is actually

\[ \left( \sum \frac{1}{p_j} - 1 \right)/(12n^2). \]
Limiting Distributions of $\hat{H}$

From Miller’s expansion, we can easily see that when $X$ is NOT uniform,

$$\sqrt{n}(\hat{H} - H) \rightarrow N(0, \sigma_{\hat{H}}^2),$$

where $\sigma_{\hat{H}}^2 = -\sum_{j \neq j'} p_j p_{j'} \log p_j \log p_{j'} + \sum_j p_j (1 - p_j)(\log p_j)^2$.

When $X$ is uniform, a faster convergence rate holds:

$$n(\hat{H} - H) \rightarrow \frac{1}{2} \chi^2_{k-1}.$$
Strong et al’s method of entropy estimation in neuroscience

For a window size $T$, take non-overlapping windows and estimate the joint probabilities of $T$-tuples and plug in these empirical joint probabilities to get entropy estimate $\hat{H}_T$ for window size $T$. Stationarity is implicitly assumed.

For a sequence of size $n$ with enough mixing, one could generalize Miller’s result to show that the bias of Strong et al’s estimate is of order

$$O(2^T/n),$$

which follows that when $T = O(\log n)$ the bias is of order $O(1)$. 
Natural vs synthetic stimuli to song birds

Spike trains (0-1 sequence) are recorded on a single neuron in the auditory pathway of a song bird while sound stimuli are played to the bird.

Sequence 1 ("natural" file):
Natural stimulus: bird song.

Sequence 2 ("synthetic" file):
Synthetic stimulus: man-made songs matching some power-spectrum characteristics of a bird song.

Use the Strong et al method to estimate the entropy rates of these two sequences and what hypothesis do you want to put forward based on these estimates?

In R, type "source("strong.r")" and "r.x=entropyrate(x,maxwordlen=L)" gives the entropy rates for T=1,..., L estimated from x.

Data provided by F. Theunissen Lab, UC Berkeley
If $X$ is a continuous random variable with probability density function $f(x)$, then

$$H(X) = - \int f(x) \log f(x) \, dx = E(- \log f(X))$$

If one discretize $X$ with precision $\delta > 0$ into a discrete variable $X_\delta$,

$$H(X_\delta) \approx H(X) - \log \delta.$$ 

Hence differential entropy could be negative.

For Gaussian $N(\mu, \sigma^2)$,

$$H(X) = \frac{1}{2} \log(2e\pi\sigma^2).$$

For exponential $(\lambda)$

$$H(X) = \log[e/\lambda].$$
Do

Downward bias of MLE plug in of differential entropy

Suppose $f(x^n, \theta)$ is a parametric n-tuple density function of a stationary and ergodic sequence, $\hat{\theta}$ is the MLE of $\theta$.

Then the MLE plug-in estimate of differential entropy rate

$$\hat{H}_n(f) = -\log f(x^n, \hat{\theta})/n,$$

underestimates $H(f_n) = E[-\log f(X^n, \theta)/n]$ under regularity conditions and the expected bias is

$$\frac{1}{2}d/n,$$

where $d$ is the dimension of $\theta$, because

$$-\log f(x^n, \theta) + \log f(x^n, \hat{\theta}) \approx \frac{1}{2} \chi^2_d.$$
In his famous 1957 paper ("information theory and statistical mechanics"), Ed. T. Jaynes wrote:

Information theory provides a constructive criterion for setting up probability distributions on the basis of partial knowledge, and leads to a type of statistical inference which is called the maximum entropy estimate. It is least biased estimate possible on the given information; i.e., it is maximally noncommittal with regard to missing information.

That is to say, when characterizing some unknown events with a statistical model, we should always choose the one that has Maximum Entropy.

Applications: computer vision, spatial physics, natural language processing.
Examples of Maxent distributions

Example 1: Gaussian

If $X$ is continuous and has known first and second moments $\alpha_i$ for $i = 1, 2$ and $\alpha_2 - \alpha_1^2 > 0$, then the maxent distribution is $N(\mu, \sigma^2)$ with

$$\mu = \alpha_1, \sigma^2 = \alpha_2 - \alpha_1^2.$$ 

Example 2: Exponential

If $X$ is positive and continuous and has a known first moment $\alpha_1$, then $X$ is exponential with mean $\alpha_1$. 

Example 3: Uniform on a finite set $\{1, \ldots, k\}$ is the maxent distribution with no moment constraints.
Example 4 (Boltzmann) The maxent distribution on a finite set \( \{1, \ldots, k\} \) with a first moment constraint \( \alpha_1 > 0 \) is

\[
p_j = \frac{e^{\lambda j}}{\sum_{j=1}^{k} e^{\lambda j}}.
\]

That is, the most probable "macrostate" (probability distribution) is \((p_1, \ldots, p_k)\) as above, provided that

\[
\sum_{j} j p_j = \alpha_1
\]

is fixed.
Maxent distributions are in the exponential family

Maxent problem:
Maximize the entropy of $f$ over all probability density functions such that

- $f(x) \geq 0$
- $\int f(x) \, dx = 1$
- $\int f(x) T_i(x) \, dx = \alpha_i$ for $i = 1, \ldots, m$

The maxent solution takes the form

$$f(x) = e^{\lambda_0 - 1 + \sum_{i=1}^{m} \lambda_i T_i(x)},$$

where the $\lambda$’s are chosen to satisfy the constraints.
A derivation through calculus

Let

\[ J(f) = -\int f \log f + \lambda_0 \int f + \sum_i \lambda_i \int f T_i. \]

Differentiate with respect to \( f(x) \):

\[ \frac{\partial J}{\partial f(x)} = -\log f(x) - 1 + \lambda_0 + \sum_i \lambda_i T_i(x). \]

Setting this to zero, we get

\[ f(x) = e^{\lambda_0 - 1 + \sum_{i=1}^m \lambda_i T_i(x)}, \]

where the \( \lambda \)'s are chosen to satisfy the constraints.
Back to coding: what if we don’t know P and use a wrong code?

We consider again the creation of codes, this time with an eye toward their performance. Suppose we create a code based on the frequency of words found in the New York Times in 2004, and then use it to encode stories from this year.

Codes based on words

- We collected all the articles from the issues appearing April 20, 2004 and March 9, 2005

- In the two issues of the paper, we collected over 17,213 words (including numbers and abbreviations), with 7,457 appearing only once
Codes based on words

• Let $P(x)$ and $Q(x)$ denote the frequency of word $x$ from the 2004 and 2005 issues, respectively

• The entropies are given by

$$\sum_{x} Q(x) \log \frac{1}{Q(x)} = 12.77 \quad \text{and} \quad \sum_{x} P(x) \log \frac{1}{P(x)} = 12.94$$

• The most frequent words appearing in these texts do not carry content; they are pronouns, articles and prepositions

• We might expect that many of these non-content words appear in roughly the same proportions in 2004 as in 2005
<table>
<thead>
<tr>
<th>word</th>
<th>n.04</th>
<th>freq.04</th>
<th>bits.04</th>
<th>n.05</th>
<th>freq.05</th>
<th>bits.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>6375</td>
<td>0.0626</td>
<td>4</td>
<td>5783</td>
<td>0.0622</td>
<td>5</td>
</tr>
<tr>
<td>to</td>
<td>2777</td>
<td>0.0273</td>
<td>6</td>
<td>2543</td>
<td>0.0274</td>
<td>6</td>
</tr>
<tr>
<td>of</td>
<td>2708</td>
<td>0.0266</td>
<td>6</td>
<td>2365</td>
<td>0.0254</td>
<td>6</td>
</tr>
<tr>
<td>a</td>
<td>2557</td>
<td>0.0251</td>
<td>6</td>
<td>2497</td>
<td>0.0269</td>
<td>6</td>
</tr>
<tr>
<td>and</td>
<td>2338</td>
<td>0.0230</td>
<td>6</td>
<td>2137</td>
<td>0.0230</td>
<td>6</td>
</tr>
<tr>
<td>in</td>
<td>2248</td>
<td>0.0221</td>
<td>6</td>
<td>2107</td>
<td>0.0227</td>
<td>6</td>
</tr>
<tr>
<td>that</td>
<td>1377</td>
<td>0.0135</td>
<td>7</td>
<td>1315</td>
<td>0.0142</td>
<td>7</td>
</tr>
<tr>
<td>said</td>
<td>972</td>
<td>0.0096</td>
<td>7</td>
<td>1027</td>
<td>0.0111</td>
<td>7</td>
</tr>
<tr>
<td>for</td>
<td>952</td>
<td>0.0094</td>
<td>7</td>
<td>893</td>
<td>0.0096</td>
<td>7</td>
</tr>
<tr>
<td>he</td>
<td>901</td>
<td>0.0089</td>
<td>7</td>
<td>741</td>
<td>0.0090</td>
<td>7</td>
</tr>
</tbody>
</table>
The wrong codebook

- While many of the non-content words seem to have similar distributions between 2004 and 2005, what about the people and places that make up the news?
- The who, what, where, when and why of the daily news certainly changes from year to year
Old news: Words that gained popularity

<table>
<thead>
<tr>
<th>word</th>
<th>n.04</th>
<th>freq.04</th>
<th>bits.04</th>
<th>n.05</th>
<th>freq.05</th>
<th>bits.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>lebanon</td>
<td>1</td>
<td>1.96e-05</td>
<td>16</td>
<td>49</td>
<td>5.38e-04</td>
<td>11</td>
</tr>
<tr>
<td>lebanese</td>
<td>1</td>
<td>1.96e-05</td>
<td>16</td>
<td>47</td>
<td>5.16e-04</td>
<td>11</td>
</tr>
<tr>
<td>arts</td>
<td>0</td>
<td>9.82e-06</td>
<td>17</td>
<td>34</td>
<td>3.76e-04</td>
<td>12</td>
</tr>
<tr>
<td>bolton</td>
<td>0</td>
<td>9.82e-06</td>
<td>17</td>
<td>28</td>
<td>3.12e-04</td>
<td>12</td>
</tr>
<tr>
<td>Hezbollah</td>
<td>1</td>
<td>1.96e-05</td>
<td>16</td>
<td>28</td>
<td>3.12e-04</td>
<td>12</td>
</tr>
<tr>
<td>march</td>
<td>30</td>
<td>3.04e-04</td>
<td>12</td>
<td>103</td>
<td>1.12e-03</td>
<td>10</td>
</tr>
<tr>
<td>prison</td>
<td>10</td>
<td>1.08e-04</td>
<td>14</td>
<td>27</td>
<td>3.01e-04</td>
<td>12</td>
</tr>
<tr>
<td>syria</td>
<td>9</td>
<td>9.82e-05</td>
<td>14</td>
<td>30</td>
<td>3.33e-04</td>
<td>12</td>
</tr>
</tbody>
</table>
Old news: Words that dropped in popularity

<table>
<thead>
<tr>
<th>word</th>
<th>n.04</th>
<th>freq.04</th>
<th>bits.04</th>
<th>n.05</th>
<th>freq.05</th>
<th>bits.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>saatchi</td>
<td>41</td>
<td>4.12e-04</td>
<td>12</td>
<td>0</td>
<td>1.08e-05</td>
<td>17</td>
</tr>
<tr>
<td>dvd</td>
<td>32</td>
<td>3.24e-04</td>
<td>12</td>
<td>0</td>
<td>1.08e-05</td>
<td>17</td>
</tr>
<tr>
<td>cantalupo</td>
<td>32</td>
<td>3.24e-04</td>
<td>12</td>
<td>0</td>
<td>1.08e-05</td>
<td>17</td>
</tr>
<tr>
<td>april</td>
<td>111</td>
<td>1.10e-03</td>
<td>10</td>
<td>15</td>
<td>1.72e-04</td>
<td>13</td>
</tr>
<tr>
<td>bonds</td>
<td>57</td>
<td>5.69e-04</td>
<td>11</td>
<td>8</td>
<td>9.68e-05</td>
<td>14</td>
</tr>
<tr>
<td>kerry</td>
<td>43</td>
<td>4.32e-04</td>
<td>12</td>
<td>3</td>
<td>4.30e-05</td>
<td>15</td>
</tr>
<tr>
<td>tax</td>
<td>32</td>
<td>3.24e-04</td>
<td>12</td>
<td>8</td>
<td>9.68e-05</td>
<td>14</td>
</tr>
<tr>
<td>campaign</td>
<td>58</td>
<td>5.79e-04</td>
<td>11</td>
<td>26</td>
<td>2.90e-04</td>
<td>12</td>
</tr>
</tbody>
</table>
Quantifying the difference

- Let $Q(x)$ and $P(x)$ denote the probabilities of word $x$ in 2004 and 2005, respectively

- We can then write the difference in code lengths as

$$\log \frac{1}{Q(x)} - \log \frac{1}{P(x)} = \log \frac{P(x)}{Q(x)}$$

- If the code from 2004 gives a shorter codeword, this quantity is negative; when the code from 2004 assigns a longer codeword, it is positive
Quantifying the difference

- Averaging over the distribution of words from the 2005 paper, the expected difference in code lengths is given by

\[ \sum_{x} P(x) \log \frac{P(x)}{Q(x)} \]

- Can this quantity ever be negative? More to the point, by using 2004s can we ever achieve (on average) a shorter encoding of the paper in 2005?
Quantifying the difference

- In fact, we know from the Information Inequality that the difference in average code lengths must be non-negative

- That is, we know that

\[
\sum_x P(x) \log \frac{1}{Q(x)} - \sum_x P(x) \log \frac{1}{P(x)} = \sum_x P(x) \log \frac{P(x)}{Q(x)} \geq 0
\]
Coding loss - redundancy

- If we use our code from 2004, we can encode the 2005 paper with an average code length of

$$\sum_x P(x) \log \frac{1}{Q(x)} = 13.29$$

- Instead, if we build a code using the frequencies from 2005, we have an average code length equal to the entropy, or

$$\sum_x P(x) \log \frac{1}{P(x)} = 12.94$$

- The difference is $13.29 - 12.94 = 0.35$
Which code?

- What is not mentioned here is the effort required to decode a message; word-based models require large codebooks
- In the next section of this workshop, we compare different codes or “models” for a data set and explicitly balance the compression achievable with a given code against the cost of representing its codebook
- This will yield the principle of minimal description length; but before we get ahead of ourselves...
Kullback-Leibler Divergence

The Kullback-Leibler divergence between two probability distributions \( P \) and \( Q \) is defined to be

\[
D(P \| Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}
\]

where \( 0 \log 0/q = 0 \) and \( p \log p/0 = \infty \)

In many respects it acts as a measure of dissimilarity or “distance” between distributions
KL as redundancy in coding

KL divergence as a loss: redundancy in coding

- Returning to our coding example, suppose we create code lengths

\[ L(x) = \left\lceil \log \frac{1}{Q(x)} \right\rceil \]

for some distribution \( Q \)

- Assuming the true distribution of the source is \( P \), we can easily show that expected code length satisfies

\[ H(P) + D(P \| Q) \leq EL < H(P) + D(P \| Q) + 1 \]
In their original paper, Kullback and Leibler cast their divergence measure as a tool for distinguishing between statistical populations.

In fact, they refer to the quantity

$$\log \frac{P(x)}{Q(x)}$$

as “the information in $x$ for discrimination between” the distributions $P$ and $Q$ (we recognize this as the logarithm of likelihood ratio, or Bayes factor assuming a uniform prior on the two distributions).

Their divergence is then the mean information for discrimination per observation from $P$. 

**KL as a ”distance”**
Example: the gamma family

- Below we have plotted the densities for five gamma distributions.
- The scale parameter for each is just 1, and the shape parameters are 1, 2, 3, 4 and 5 (tracking the modes from left to right).
Example: the gamma family

- We present the pairwise KL divergences in the form of a (dis)similarity matrix.
- Within each row, the largest distances occur in the first column; as expected, the exponential distribution looks quite distinct.
- Notice the asymmetry in the table, it is also the most extreme for entries involving the first column/row.
shape parameter

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.54</td>
<td>1.78</td>
<td>3.42</td>
<td>5.34</td>
</tr>
<tr>
<td>2</td>
<td>0.42</td>
<td>0.00</td>
<td>0.27</td>
<td>0.95</td>
<td>1.91</td>
</tr>
<tr>
<td>3</td>
<td>1.15</td>
<td>0.23</td>
<td>0.00</td>
<td>0.18</td>
<td>0.64</td>
</tr>
<tr>
<td>4</td>
<td>1.98</td>
<td>0.72</td>
<td>0.16</td>
<td>0.00</td>
<td>0.13</td>
</tr>
<tr>
<td>5</td>
<td>2.85</td>
<td>1.34</td>
<td>0.53</td>
<td>0.12</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Properties of KL divergence

- Scaling: KL is scale-invariant

- In these examples, we are essentially computing the distance between the distributions for a single observation. If we instead consider n independent observations from $P$ and $Q$, $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, we have

$$D(P^n \| Q^n) = \sum_{x \in \mathcal{X}^n} P^n(x) \log \frac{P^n(x)}{Q^n(x)} = nD(P \| Q)$$
KL divergence

KL divergence as a “distance”

- In many language applications, documents are treated naively as bags of words

- The frequency of words becomes a clue to their content; documents that use words in roughly in the same proportion are said to be similar

- What problems might we encounter applying KL divergence directly to this kind of data?
Problem 1: Asymmetry

- The symmetry issue is often addressed by defining a new measure that is the sum

\[ D(P\|Q) + D(Q\|P) \]

- This is often called the J-divergence and was originally studied by Jeffreys in the context of identifying invariant prior distributions

- Jeffreys wrote the J-divergence in the form

\[ D(P\|Q) + D(Q\|P) = \sum_x [P(x) - Q(x)] \log \frac{P(x)}{Q(x)} \]
Problem 2: For some distributions, the supports are different.

- In this case, the KL divergence is infinite; while this may be sensible in some applications, it is a bit extreme here.

- To soften the blow, Lin (1991) proposed the Jensen-Shannon divergence

\[
JS(P, Q) = \frac{1}{2} D \left( P \| \frac{P + Q}{2} \right) + \frac{1}{2} D \left( Q \| \frac{P + Q}{2} \right)
\]

- What properties does this divergence have? Actually

\[
JS(P, Q) = \frac{1}{2} H \left( \frac{P + Q}{2} \right) - \frac{1}{2} H(P) - \frac{1}{2} H(Q)
\]
Example: News stories

- Let's consider stories taken from a single day of the New York Times, January 2, 2004
- We applied the JS divergence to pairs of stories to compute a (symmetric) dissimilarity matrix
- A simple visualization device was used to check the output for clusters, categorizing stories solely on their section of the paper

Example: News

- The plot shows the result of simple multidimensional scaling applied to the JS dissimilarity matrix
- Notice the isolation of certain categories like business
MDS: Multidimensional scaling, widely used in Psychology, Social Science and Economy. In general, it maps the documents to points in the 2-dim Euclidean space to minimize some stress measure such as

$$\sum_{i,j}(d_{ij} - JS(document_i||document_j))^2,$$

where $d_{ij}$ is the reproduced distance in $R^d$ of the mapped points where $d$ is often 2.

In R, there is a function cmdscale (classical multidimensional scaling) the algorithm is based on


which uses the principle coordinate system for the MDS mapping.
KL divergence in statistics: Fisher information

- Given a parameter vector \( \theta \), consider the family of distributions \( P_\theta \) and let \( P_{\theta^*} \) denote the "true" data generating distribution

- The KL divergence (viewed as a function of \( \theta \)) behaves like a quadratic

\[
D(P_{\theta^*} \parallel P_\theta) \approx \frac{1}{2}(\theta - \theta^*)^t I(\theta^*)(\theta - \theta^*)
\]

where the Fisher information

\[
I(\theta) = E_\theta \left[ -\frac{\partial^2}{\partial_i \partial_j} \log P_\theta(X) \right]
\]
Fisher information is similar at a high level to Shannon’s information. They both provide limits on how well we can do:

- Shannon information: how much we can compress a sequence of symbols;
- Fisher information: how well we can estimate a parameter based on a sequence of observations (data).

For n iid observations, $I_n(\theta) = nI(\theta)$. 
Cramer-Rao: if $T$ is an unbiased estimator of $\theta$,

$$V_{\theta}(T(X_1, \ldots, X_n)) \geq \frac{1}{I_n(\theta)}.$$  

It is sometimes called information inequality in the stats literature.

Compare with Shannon’s source coding theorem.

Asymptotically, the MLE achieves the Fisher information in the limit.
KL divergence and its relation to proper distances

- We have seen that KL divergence represents a loss in the context of coding and that it can be used, at least informally, as a measure of dissimilarity between distributions.

- We will now consider the relationship between KL and the L1 and L2 norms, ultimately uncovering a geometric interpretation that will tie this material to the maximum entropy methods from this morning.
KL divergence and the L1 norm

- The L1 distance between $P$ and $Q$ is defined to be

$$\|P - Q\|_1 = \sum_x |P(x) - Q(x)|$$

- The KL divergence provides us with an upper bound on this distance; namely

$$\frac{1}{2 \ln 2} \|P - Q\|_1^2 \leq D(P \| Q)$$

- This is known as Pinsker’s inequality
From now on, I will use $K$ if the logs in the KL divergence is based $e$: 

$$D(P||Q) = K(P, Q)/\ln 2.$$

We use the density version of the Kullback-divergence.

$$K(f, g) = \int f(x) \ln(f(x)/g(x)) dx.$$
Bretagnolle-Huber Inequalities:

- \( K(f, g) \geq -\ln(1 - \frac{1}{4} L_1^2(f, g)) \)
- \( K(f, g) \geq -\ln(2 - L_1(f, g)) \)

In terms of \( K \), Pinsker’s inequality becomes

\[
K(f, g) \geq \frac{1}{2} L_1^2(f, g).
\]
For the two distributions used in Lin (1996),

\[ p_1 = (t, 1 - t), \quad p_2 = (1 - t, t) \]

For the first panel, black - K, red- \( L_2 \), green - L2, blue- \( Hellinger^2 \)

For the second panel, black- K, red - BH’s first bound, green - BH’s second bound, blue - Pinsker bound.
Last lecture, we talked about:

- Connections of $K$ with other proper distances
- Basic maximum likelihood theory (Fisher information, Cramer-Rao inequality, score vector, asymptotic normality of MLE)
- Connection of $K$ with Fisher information
Today: proofs

- Proof of BH inequalities (HW: Pinsker’s inequality)
- Proofs: equivalent Fisher information expressions; Cramer-Rao inequality. Asymptotic approximation of MLE in terms of score function and Fisher information, normality of MLE.
Cramer-Rao: if the true distribution belongs to the parametric model, then for any unbiased estimator \( \hat{\theta}_n \)

\[
E(\hat{\theta}_n - \theta)(\hat{\theta}_n - \theta)' \geq J_n(\theta)^{-1},
\]

where \( J_n(\theta) \) is the Fisher information matrix,

\[
J_n(\theta) = E_\theta \left( \frac{\partial}{\partial \theta} \ln g_\theta(X_1, \ldots, X_n) \right) \left( \frac{\partial}{\partial \theta} \ln g_\theta(X_1, \ldots, X_n) \right)'.
\]
If $X_1, \ldots, X_n$ are iid, then

$$J_n = nJ_1,$$

a property similar to Shannon’s entropy. Also similar to Shannon’s entropy, Fisher information characterize the limit of possible for parameter estimation if the true distribution is contained in the parametric family.
There is an equivalent expression for the Fisher information. We now introduce some notation. Let the log-likelihood function be

\[ l_n(\theta) = \ln g_\theta(X_1, \ldots, X_n), \]

and the score vector

\[ U_n = (U^1, \ldots, U^d)^T = (\partial l_n / \partial \theta_1, \ldots, \partial l_n / \partial \theta_d)^T, \]

with

\[ E_\theta U_n = (0, \ldots, 0)^T. \]
\[ J_n = \left[ -E_\theta \frac{\partial^2 l_n}{\partial \theta_i \partial \theta_j} \right]_{d \times d}. \]

The proof is straightforward from the following facts

\[ E_\theta \frac{\partial l_n}{\partial \theta_i} = 0. \]

\[ \frac{\partial l_n}{\partial \theta_i} = \frac{\partial g_\theta}{\partial \theta_i} / g_\theta. \]

\[ \frac{\partial^2 l_n}{\partial \theta_i \partial \theta_j} = \frac{\partial^2 g_\theta(X^n)}{\partial \theta_i \partial \theta_j} / g_\theta - \frac{\partial g_\theta(X^n)}{\partial \theta_i} \frac{\partial g_\theta(X^n)}{\partial \theta_j} / g_\theta^2. \]
Expanding $0 = \frac{\partial l_n}{\partial \theta_i} (\hat{\theta})$ around the true parameter $\theta$, we get

$$\sqrt{n}(\hat{\theta} - \theta) \approx [J_n(\theta)]^{-1} \times \sqrt{n}U_n.$$  

The asymptotic normality of the maximum likelihood estimator follows:

$$\sqrt{n}(\hat{\theta} - \theta) \rightarrow_d N(0, J_1^{-1}),$$

provided that the observations are iid and the parametric family is "nice".
If we measure the distance between the distribution by the KL divergence $K$, the following equation confirms the pivotal role that the Fisher information plays in parameter estimation:

$$\lim_{\Delta \theta \to 0} \frac{1}{(\Delta \theta)^2} K(g_\theta \| g_{\theta + \Delta \theta}) = \frac{1}{2} J_1(\theta),$$

(0.13)
Mutual information:

For a pair of random variables with joint distribution $P(x, y)$ and marginal distributions $P(x)$ and $P(y)$, the mutual information

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} P(x, y) \log \frac{P(x, y)}{P(x)P(y)}, = K(P(x, y), P(x)P(y)).$$

(0.14)

$I(X, Y) = 0$ if and only if $X$ and $Y$ are independent.
I measures the information $X$ and $Y$ contain about each other or the dependence between them.

The definition of $I(X, Y)$ can easily be extended to the continuous case by replacing the summation by an integration.

$$I(X, Y) = H(Y) - H(Y|X); \text{ or } H(Y) = I(X, Y) + H(Y|X)$$

Or symmetrically,

$$I(X, Y) = H(X) + H(Y) - H(X, Y).$$
Examples: Bivariate Gaussian

Assume \((X, Y)\) are normal with mean zero and variance 1 and correlaction coefficient \(\rho\). It is straightforward to calculate that

\[
I(X, Y) = -\frac{1}{2} \ln(1 - \rho^2).
\]

\(I(X,Y)\) doesn’t differentiate between positive and negative correlations.

New York Times stories:

\(X\) and \(Y\) consecutive words,

\[
H(X) = 7.1, \quad H(Y) = 7.3, \quad H(X, Y) = 10.0
\]

\[
H(Y|X) = H(X, Y) - H(X) = 2.9, \quad I(X, Y) = 4.4.
\]
Chain rule for mutual information based on the chain rule for entropy and the entropy difference expression of mutual information:

\[ I(X_1, \ldots, X_n; Y) = \sum_{i=1}^{n} I(X_i, Y | X_{i-1}, \ldots, X_1). \]
Data Processing Inequality:

If $X \rightarrow Y \rightarrow Z$ are Markov, then

$$I(X, Y) \geq I(X, Z). \quad (0.15)$$

Proof: Using the chain rule on mutual information,

$$I(X; Z, Y) = I(X, Z) + I(X, Y|Z) \geq 0.$$ 

Similarly,

$$I(X; Y, Z) = I(X, Y) + I(X, Z|Y) = I(X, Y)$$

since $X$ and $Z$ are independent given $Y$. Therefore

$$I(X, Y) \geq I(X, Z).$$

It follows that: for any function $g$, $I(X, Y) \geq I(X, g(Y))$. 
Factorization Theorem for sufficient statistics:

$T$ is a sufficient statistic for $f_\theta$ if and only if

$$f_\theta(x_1, \ldots, x_n) = g_\theta(T(x_1, \ldots, x_n))h(x_1, \ldots, x_n | T(x_1, \ldots, x_n))$$

(0.16)

where $g$ and $h$ are two nonnegative functions and $h(x_1, \ldots, x_n | T(x_1, \ldots, x_n))$ does not depend on $\theta$. 
Examples of sufficient statistics:

- If $X_1, \ldots, X_n$ are i.i.d. Gaussian with mean $\theta$ and variance $1$, $T(X) = \bar{X}$.

- $X_1, \ldots, X_n$ i.i.d. Bernoulli($p$). $T(X) = \sum_{i=1}^{n} X_i$.

- $X_1, \ldots, X_n$ i.i.d. $N(\theta, \sigma^2)$. $T(X) = (\sum X_i, \sum X_i^2)$, or equivalently $(\bar{x}, s^2)$.

- $X_1, \ldots, X_n$ i.i.d. Uniform($\theta, \theta + 1$). $T(X) = (\min(X_1, \ldots, X_n), \max(X_1, \ldots, X_n))$. 

Mutual Information and Sufficiency:

$T$ is a sufficient statistic for $f_\theta(X)$ iff

$$I(\Theta, X) = I(\Theta, T(X)).$$

Since $T$ is a mapping from $X$, we have

$$X \rightarrow T(X),$$

which implies, by the data processing inequality, that

$$I(\Theta, T(X)) \leq I(\Theta, X). \quad (0.17)$$

On the other hand, $T$ is a sufficient statistic,

$$\Theta \rightarrow T(x) \rightarrow x,$$

so

$$I(\Theta, T(X)) \geq I(\Theta, X). \quad (0.18)$$

Combining two inequalities gives $I(\Theta, T(X)) = I(\Theta, X)$, so the mutual
information of the parameter and the data is the same as mutual information of the parameter and the sufficient statistic.
The converse:
Fano’s inequality:

Suppose $X$ unknown, $Y$ is our data. When $H(X|Y)$ is large, we can not hope to guess or estimate well $X$ from $Y$.

Fano’s inequality quantifies this observation, and it has two important applications to be covered in the later lectures:

proving the converse to Shannon’s channel coding theorem and

giving lower bounds in minimax density estimation.
Fano’s inequality:

Suppose $X \to P(Y|X) \to Y \to \hat{X}$, that is, $\hat{X}$ is an estimate of $X$ based on $Y$. Let $P_e = P(X \neq \hat{X})$ be the probability of error. Then

$$H(P_e) + P_e H(X) \geq H(X|Y)$$  (0.19)

where $H(P_e) = -P_e \log P_e - (1 - P_e) \log(1 - P_e)$. Note that if $P_e = 0$, then $H(X|Y) = 0$. 
In our document example, $H(X) = 7.1$, $H(X|Y) = 2.7$, Fano’s inequality gives:

$$1 + 7.1P_e \geq 2.7 \text{ or } P_e \geq 1.7/7.1 = 0.239,$$

which means that one cannot guess the next word from the previous word with an accuracy better than 23.9%, or the error rate is at least 23.9%, no matter how sophisticated your method is.

Note that this lower bound depends on only two summary quantities of the joint distribution:

$$H(X), H(X|Y).$$
For a symmetric binary channel with cross-error probability $\epsilon < 1/2$, let $X$ be an input variable with distribution $1/2, 1/2$,

$$H(Y|X) = h(\epsilon) = H(X|Y), \quad H(Y) = H(X) = 1.$$  

$$H(P_e) + P_e \geq H(\epsilon),$$  

The maximum likelihood (and the Bayesian maximum posterior) decoder is $\hat{X} = Y$, and it has an error $\epsilon$.

When $\epsilon$ is small, the LHS of the inequality is dominated by $H(P_e)$ which matches the RHS. In this sense, the inequality is "almost" tight.
Classical nonparamatric function estimation:

- histogram
- kernel method
- orthogonal basis functions
- polynomials
- nearest neighbor
- smoothing splines
- wavelet-based
- ...

Computation is often an after-thought ...
Machine learning nonparametric function estimation:

- boosting

- support vector machines or kernel machines

- ...

Computation is a major design criterion...
Minimax framework for nonparametric estimation:

a well-defined framework to seek optimality, but the classes of functions are not as natural as the parametric classes.

Pioneers:

Hasiminski Ibragimov, 78, 81 (Russia)

Bretagnolle and Huber, 78, Assouad, 83 (France)

Le Cam, 73, Stone, 83 (Berkeley)
Non-parametric minimax density estimation

Given a class of functions $\mathcal{F}$ and a loss function $L(f, \hat{f})$ on $f \in \mathcal{F}$ and an estimator $\hat{f}$ of $f$, we would like to minimize the worst case expected loss:

$$\max_{f \in \mathcal{F}} E_f L(f, \hat{f}).$$

An estimator $\hat{f}^*$ based on $n$ samples of data said to be minimax rate optimal if

$$\min_{\hat{f}} \max_{f \in \mathcal{F}} E_f L(f, \hat{f})$$

has the same order as

$$\max_{f \in \mathcal{F}} E_f L(f, \hat{f}^*),$$

as $n \to \infty$. 

116
For smooth density function classes, the minimax risk usually converges to zero at a rate $n^{-\alpha}$ for some $0 < \alpha < 1/2$, which is slower than the usual parametric rate of $n^{-1/2}$. To be more precise, if the logarithm of the density has $p$ continuous derivatives the rate is $\alpha = 2p/(2p + 1)$. 
How are the minimax optimal rates obtained traditionally?

Lower bound: build an subclass indexed by a hypercube with expanding dimensions by having local perturbations of a uniform density function within the constraints of the class. Depending on the perturbation is up or down, we get the hypercube subclass.

Upper bound: one tries different known density estimators such as histogram or kernel estimators to match the rates in the lower bounds.

There was no systematic way of finding minimax optimal estimators, until

Yang and Barron (1999) unified the calculations in the lower and upper bounds by introducing the use of KL divergence and associated inequalities.
Set-up:

Recall:

\[ H(f, g) = \left( \int (\sqrt{f} - \sqrt{g})^2 \, dx \right)^{1/2}. \]

\[ K(f \parallel g) = \int f \log \frac{f}{g} \, dx. \]

Given a smooth density class \( \mathcal{M} \),

Assumption: \( K(f, g) \asymp H^2(f, g) \) when \( f \) and \( g \) are "close", i.e., there exist constants \( A \) and \( A' \) s.t.

\[ AH^2(f, g) \leq D(f \parallel g) \leq A'H^2(f, g) \text{ when } H^2(f, g) \leq \epsilon. \]
Idea:

approximate the class by an \( \epsilon \)-net and turn the problem into a discrete one.

This net approximate well and at the same time its members are not close to each other.

Upper bound: mixture of product densities in the approximating net.

the denser the net, the better the approximation, but the worse the estimation cost.

Lower bound: Fano’s inequality in terms of Mutual information.

The denser the net, the harder for us to tell them apart, hence the larger the detection error (Fano’s). But we need also the net members to be far apart (packing number) so that the loss function \( D \) or \( H^2 \) can be bounded below by the detection error.
Definition 1 (Minimal covering number). of a set $\mathcal{M}$, $N_d(\epsilon_n)$, is the minimum number of balls of radius (in terms of $\sqrt{D}$) $\epsilon_n$ needed to cover $\mathcal{M}$. That is, given an $\epsilon_n$-cover $T_D(\epsilon_n) = \{f_1, \ldots, f_K\}$ with cardinality $K = |T_D(\epsilon_n)|$ such that for any $f \in \mathcal{M}$, there is an $f_i \in T_D(\epsilon_n)$ with the property that $D(f, f_i) \leq \epsilon_n^2$, we have $N_D(\epsilon_n) \leq K$. $V_D(\epsilon_n) = \log(N_D(\epsilon_n))$ is called the metric entropy.

Definition 2 (Maximum packing number). of a set $\mathcal{M}$ is the maximum number of balls of radius (in terms of $H$) $\epsilon_n$ that can be packed inside $\mathcal{M}$ so that their centers are in $\mathcal{M}$ and the balls do not overlap.
Mixture of Product Measures as an Estimator:

Suppose $G_D(\epsilon_n)$ is a minimal cover of $\mathcal{M}$ at radius $\epsilon_n$, let $p$ be the mixture of the product measures of the centers in the cover:

$$
p(x^n) = \frac{1}{N_D(\epsilon_n)} \sum_{i}^{N_D(\epsilon_n)} f_i(x^n).
$$

Then for any $f \in \mathcal{M}$,

$$
D(f(x^n) || p(x^n) \leq V_D(\epsilon_n) + n\epsilon_n^2.
$$
Moreover, the mixture density $p(x^n)$ is the Bayes estimator of $f(x^n) \in G(\epsilon_n)$ with respect to the uniform prior on $G(\epsilon_n)$ and risk or loss function $D$. That is,

$$p(x^n) = \arg\min_{Q \circ Q_n} \frac{1}{N_D} \sum_{i=1}^{N_D} D(f_i(x^n) || Q(x^n)).$$

The same conclusion still holds when the prior is not uniform and $G$ is any arbitrary density set.
Upper bound:

Using the notations in the previous lemma, construct a marginal density using the mixture density on the n-tuples:

$$
\hat{f}(x) = \frac{1}{n} \sum_{i=0}^{n-1} p(x_{i+1} = x | x^i) = \frac{1}{n} \sum_{i=0}^{n-1} \hat{f}_i,
$$

where \( \hat{f}_i = p(x_{i+1} = x | x^i) \). Then for any \( f \in \mathcal{M} \),

$$
Ef(x^n)D(f || \hat{f}) \leq \frac{1}{n} V(\epsilon_n) + \epsilon_n^2.
$$
Lower bound:

Let \( \hat{f} \) be an estimate based on an i.i.d. sample \( X_1, \ldots, X_n \) from \( f \in \mathcal{M} \). Then

\[
\min_{\hat{f}} \max_{f \in \mathcal{M}} E_f H^2(f, \hat{f}) \asymp \min_{\hat{f}} \max_{f \in \mathcal{M}} E_f D(f \| \hat{f}) \geq \epsilon_n^2 / 8
\]
Two main results for the lower bound:

1. Given a maximal packing net $G'(\epsilon_n)$, for any estimator $\hat{f}$,

$$\max_{f \in \mathcal{M}} E_f H^2(f, \hat{f}) \geq (\epsilon_n/2)^2 \max_{f \in G'(\epsilon_n)} P_f\{f \neq \tilde{f}\}$$

where $\tilde{f} = \arg \min_{f \in G(\epsilon_n)} H^2(\hat{f}, f)$.

2. Use Fano’s inequality, we can prove

$$\max_{f \in G(\epsilon_n)} P_f\{f \neq \tilde{f}\} \geq 1/2.$$
Example: Lipschitz Class

For the Lipschitz class defined earlier, it can be shown that

\[ V(\epsilon) = \frac{1}{\epsilon}. \]

It follows that the optimal rate is \( n^{-2/3} \) after verifying that \( H^2 \) and \( D \) are equivalent for all members of the class.
Channel Capacity

Before Shannon’s 1948 work, the common belief was that errors had to occur with a positive chance for any noisy channel. Shannon (1948) proved it possible to transmit without errors in the limit when the message size gets large.

However, he had to

1. allow an arbitrarily small but non zero probability of error;

2. use the channel many times in succession (blocks) to the law of large numbers kicks in;

3. employ a random code book by looking at the average probability of error over randomly generated block books to prove the existence of at least one good code book.
Definition 3 (Discrete Channel). A discrete channel is characterized by, for any given input $x \in$ a finite alphabet $\mathcal{X}$, a conditional distribution $p(\cdot | x)$ on a finite alphabet $\mathcal{Y}$. The channel is said to be memoryless if

$$p(x_1, \ldots, x_n; y_1, \ldots, y_n) = \prod_{i=1}^{n} p(y_i | x_i)p(x_i).$$

Definition 4 (Channel Capacity). The information channel capacity of a discrete memoryless channel is

$$C = \max_{p(x)} I(X; Y),$$

where the maximum is taken over all possible input distribution $p$. 

129
Example 0.1 (Binary symmetric channel with cross-over probability $\epsilon$).

$$C = 1 - H(\epsilon).$$

*Maximum capacity is 1 when $\epsilon = 0$ or the channel is noiseless.*
Example 0.2 (Erasure channel erasure probability $\epsilon$). Here the output alphabet $\mathcal{Y} = \{0, e, 1\}$ where $e$ for erasure.

$$C = 1 - \epsilon.$$  

Maximum capacity is 1 when $\epsilon = 0$ or the channel has no erasure.
Theorem 1 (Properties of C). 1. \( C \geq 0 \).

2. \( C \leq \log |\mathcal{X}| \).

3. \( C \leq \log |\mathcal{Y}| \).

4. \( C \) is continuous in \( p(x) \).

5. \( C \) is concave in \( p(x) \).
The most important continuous channel is the Guassian additive noise channel and it is an accurate description of physical channels such as in  deep-space communication.

**Example 0.3 (Gaussian Channel with a Power Constraint \( P \)).** Given a continuous random input variable \( X \), the output

\[
Y = X + Z,
\]

where \( Z \) is the channel noise which is independent of \( X \) and has a Gaussian distribution with mean 0 and variance \( N \). Then

\[
C = \max_{EX^2 \leq P} I(X, Y) = \frac{1}{2} \log(1 + P/N).
\]

Hence the lower the channel noise and the higher the power, the larger the channel capacity.
Definition 5 ((M, n) Code). An \((M, n)\) code for a channel consists of the following

1. An index set \(\{1, \ldots, M\}\);

2. An encoding function \(X^n: \{1, 2, \ldots, M\} \to \mathcal{X}^n\), yielding codewords \(X^n(1), \ldots, X^n(M)\) which form the codebook.

3. A decoding function

\[ g: \mathcal{Y}^n \to \{1, \ldots, M\}. \]
For a given \((M, n)\) code, we use \(\lambda^{(n)}\) and \(P_e^{(n)}\) to denote the maximum probability of error and the average probability of error respectively. That is,

\[
\lambda^{(n)} = \max_{i \in \{1, \ldots, M\}} \lambda_i
\]

where

\[
\lambda_i = P(g(Y^n) \neq i \mid X^n = X^n(i)).
\]

\[
P_e^{(n)} = \frac{1}{M} \sum_i \lambda_i.
\]

We often denote an \((M, n)\) code as a \((2^nR, n)\) code with \(R = \log M/n\) (bits per transmission) being the rate of the code.
Theorem (Shannon’s channel coding theorem):

For a discrete memoryless channel, all rates below capacity $C$ are achievable. Specifically, for every rate $R < C$, there exists a sequence of $(2^{nR}, n)$ codes with maximum probability of error $\lambda^{(n)}$ going to zero as $n \to \infty$. Conversely, any sequence of $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ must have $R \leq C$. 
Method of Types:

If we assume our source is iid on a finite alphabet $\mathcal{X} = \{1, \ldots, m\}$, for an observed sequence $X_1, \ldots, X_n$, the counts of them in the $m$ categories form a multinomial distribution and we can send the estimated parameters in this model first, or equivalently, the counts $N_1, \ldots, N_m$ or frequencies $P_x = (N_1, \ldots, N_m)/n$ in the $m$ categories among the $X$’s, which is the empirical distribution of $X_1, \ldots, X_n$. 
Type $P_x$:

The sequence $X_1, \ldots, X_n$ is said to have a type $P_x$ if its empirical distribution is $P_x$.

Type Class $Q$:

$$T(Q) = \{x = (x_1, \ldots, x_n) \in \mathcal{X}^n : P_x = Q\}.$$
Let \( \mathcal{P}_n \) be the collection of type classes.

Theorem: The number of type classes

\[
|\mathcal{P}_n| \leq (n + 1)^m.
\]

For \( m = 2 \) or the binary case,

\[
||\mathcal{P}_n|| = (n + 1).
\]
Theorem (Size of a type class $T(P)$):

$$\frac{1}{(n+1)^m}2^{nH(P)} \leq |T(P)| \leq 2^nH(P).$$

Apparently, in the binary case, we know the number of type classes is $n + 1$ rather than the bound $(n + 1)^2$ used in the general case proof, so we can sharpen the lower bound to

$$\frac{1}{n+1}2^{nH(k/n)}.$$
Theorem (Probability of a type class $T(Q)$):

For any type $P$ and any distribution $Q$, the probability of the type class $T(P)$ under $Q^n$ is $2^{-nD(P||Q)}$ to first order in the exponent. More precisely,

$$\frac{1}{(n+1)^m}2^{-nD(P||Q)} \leq |Q^n(T(P))| \leq 2^{-nD(P||Q)}.$$
Example 0.4 (Universal Source Coding). For a given sequence $x^n$, it takes less than $m \log(n + 1)$ bits to send over the type class information and if we use a uniform code over this type class (all the sequences have the same probability in a type class under an iid assumption on the source distribution), then it takes $\log |T(P_{x^n})| \leq nH(P_{x^n})$ bits to transmit the membership of the observed $x^n$ in the type class. Hence the average code length of resulted two-part code $L(x^n)/n$ tends to $H(P)$ almost surely as $n \to \infty$ because $H(P_{x^n}) \to H(P)$ almost surely.
Large Deviations:

**Definition 6 (Exponent).** A real number $\alpha$ is said to be the exponent for a sequence $a_n$, $n \geq 1$, where $a_n \geq 0$ and $a_n \to 0$ if

$$\lim_{n \to \infty} \left( -\frac{1}{n} \log a_n \right) = -\alpha.$$
CLT provides good approximations to probabilities of events which are "moderate deviations" from the typical.

For rare events or large deviations from the typical, CLT fails to provide good approximations.

Rare events arise in many fields from reliability to finance (prob. of large portfolio losses).
Theorem (Sanov’s theorem) Let $X_1, \ldots, X_n$ be iid $Q$. Let $E$ be a set of probability distributions on a finite alphabet $\mathcal{X}$. Then

$$Q^n(E) = Q^n(E \cap \mathcal{P}_n) \leq (n + 1)^m 2^{-nD(P^*||Q)},$$

where

$$P^* = \arg\min_{P \in E} D(P||Q),$$

is the distribution in $E$ that is closest to $P$ in KL divergence. If, in addition, the set $E$ is the closure of its interior, then

$$\frac{1}{n} P^n(E) \rightarrow -D(P^*||Q).$$
KL divergence and Euclidean distance

- While not a true metric, KL divergence has many of the same geometric properties as squared Euclidean distance.

- For example, given a (closed, convex) set of probability distributions $\mathcal{P}$, we can find a unique $P^* \in \mathcal{P}$ satisfying

$$D(P^* \parallel Q) = \min_{P \in \mathcal{P}} D(P \parallel Q)$$

which we call the I-projection of $Q$ onto $\mathcal{P}$.

- Through I-projections we uncover the “information geometry” behind KL divergence.
I-projections

- While many facts about KL divergence were considered by Kullback and his co-authors, fundamental geometric concepts and their ties to information theory were explored by Csiszar

I-projection of $Q$ onto a set of distributions $\mathcal{P}$.

$$D(P^* \| Q) = \min_{P \in \mathcal{P}} D(P \| Q)$$

If $\mathcal{P}$ is convex and $Q(x) > 0$ for all $x \in \mathcal{X}$, then $D(P \| Q)$ is continuous and strictly convex in $P$ so that $P^*$ exists and is unique.
• First, we have an analog of the cosine inequality in Euclidean geometry

\[ D(P \parallel Q) \geq D(P \parallel P^*) + D(P^* \parallel Q) \]

where

\[ D(P^* \parallel Q) = \min_{P \in \mathcal{P}} D(P \parallel Q) \]
I-projection in linear spaces

- Let \( \mathcal{L} \) denote a set of probability distributions that satisfy a set of linear constraints
  \[
  \mathcal{L} = \left\{ P : \sum_x P(x) f_i(x) = \alpha_i, \ i = 1, \ldots, K \right\}
  \]

- We have seen constraints of this form earlier in the context of maximum entropy estimation and large deviations.
I-projection in linear spaces

- Let $P^*$ denote the I-projection of $Q$ onto $\mathcal{L}$; then for all $P \in \mathcal{L}$ we have the following equality

$$D(P \| Q) = D(P \| P^*) + D(P^* \| Q)$$

- KL divergence is then seen to have geometric properties similar squared Euclidean distance
I-projection and Maximum Entropy

• Let $Q_0$ denote the uniform distribution on $\mathcal{X}$; with a little algebra, we can write the entropy of any distribution as follows

$$H(P) = - \sum_x P(x) \log P(x)$$

$$= - \log \frac{1}{|\mathcal{X}|} - \sum_x \left[ P(x) \log P(x) - \log \frac{1}{|\mathcal{X}|} \right]$$

$$= - \frac{1}{|\mathcal{X}|} \log \frac{1}{|\mathcal{X}|} - \sum_x \left[ P(x) \log \frac{P(x)}{\frac{1}{|\mathcal{X}|}} \right]$$

$$= H(Q_0) - D(P \parallel Q_0)$$
I-projection and Maximum Entropy

• Therefore, finding the distribution that has maximal entropy over the set $\mathcal{L}$ is the same as finding the I-projection of the uniform distribution into $\mathcal{L}$

• We can in fact push this connection farther and relate Maximum Entropy and Maximum Likelihood through I-projections
I-projection and exponential families

- As we saw earlier, sets of linear constraints can be linked to special exponential families

- Let \( Q \) be some probability distribution and define

\[
\mathcal{E} = \left\{ P : P_{\lambda}(x) = c(\lambda_1, \ldots, \lambda_K) Q(x) e^{\sum_i \lambda_i f_i(x)} \right\}
\]

- where \( c(\lambda_1, \ldots, \lambda_K) \) is a normalizing constant

\[
c(\lambda_1, \ldots, \lambda_K) = \frac{1}{\sum_x Q(x) e^{\sum_i \lambda_i f_i(x)}}
\]
I-projection and exponential families

- The I-projection of $Q$ onto $\mathcal{L}$ is in $\mathcal{E}$ or is in the exponential family. That is, there is a $\lambda^* = (\lambda_1^*, \ldots, \lambda_K^*)$ such that $P_{\lambda^*} \in \mathcal{L}$ and $P_{\lambda^*}$ is the I-projection of $Q$ onto $\mathcal{L}$; that is,

$$P_{\lambda^*} = \arg\min_{P_\lambda \in \mathcal{L}} D(P_\lambda \parallel Q)$$
I-projection and Maximum Likelihood

- In many cases, the constraints we are dealing with involve matching empirical and model-based averages of some collection of features

- Let $X_1, \ldots, X_n$ denote a collection of training samples and let $\tilde{P}_n(x)$ be their empirical distribution

- For “features” $f_1(x), \ldots, f_K(x)$, let $\mathcal{L}$ consist of probability distributions $P(x)$ that satisfy the constraints

$$\sum_x P(x)f_i(x) = \sum_x \tilde{P}_n(x)f_i(x)$$
I-projection and Maximum Likelihood

- Corresponding to the linear space $\mathcal{L}$ we again define the exponential family

$$\mathcal{E} = \left\{ P : P_\lambda(x) = c(\lambda_1, \ldots, \lambda_K)Q(x)e^{\sum_i \lambda_if_i(x)} \right\}$$

- We then have the following equivalent characterizations I-Projection (Maximum Entropy with $Q$ not uniform)

$$P^* = \arg\min_{P \in \mathcal{L}} D(P\|Q)$$

Maximum Likelihood

$$P^* = \arg\min_{P_\lambda \in \mathcal{E}} D(\tilde{P}_n\|P_\lambda)$$
I-projection and Maximum Likelihood

- With a little algebra, the we can give the last expression a more familiar form

$$D(\tilde{P}_n \| P_\lambda) = \sum_x \tilde{P}_n(x) \log \frac{\tilde{P}_n(x)}{P_\lambda(x)} = C - \frac{1}{n} \sum_i P_\lambda(X_i)$$

which we then optimize over $\lambda = (\lambda_1, \ldots, \lambda_K)$
I-projections

- So-called feature constraints combined with the principle of Maximum Entropy give rise to an exponential family

- When the constraints match theoretical and empirical averages of features, the Maximum Entropy solution is equivalent to Maximum Likelihood in the exponential family
Computing I-projections

- One of the earliest examples of this kind of model involves essentially $I \times J$ tables where the constraints consist of matching marginal distributions.

- The Iterative Proportional Fitting (IPF) procedure alternates between matching row and column sums.
Computing I-projections

- If we let $p_i^*$ denote the constrained row sums and $p_j^{**}$ the column sums, then IPF looks like

$$p_{ij}^k = \begin{cases} p_{ij}^{k-1} \frac{p_i^*}{p_i^{*k-1}} & k \text{ odd} \\ p_{ij}^{k-1} \frac{p_j^{**}}{p_j^{**k-1}} & k \text{ even} \end{cases}$$

- It can be shown that this represents a series of alternating I-projections, where odd and steps projects the previous solution into the linear space with correct row sum and column sums, respectively (starting from $Q$)
Computing I-projections

- The first application of IPF seems to have come from Kruithof in 1937 who applied it to estimating telephone traffic. \( p_{ij} \) is the number of calls from exchange \( i \) to exchange \( j \) on a given day, \( p_i^* \) and \( p_j^{**} \) are the number of outgoing and incoming calls, and \( q_{ij} \) is the previous day’s traffic.

- Recently, applications have emerged involving traffic across computer networks that are very similar in spirit; Cao et al (2000), Liang and Yu (2003), Liang et al (2005) apply it to so-called network tomography problems.
IPF is also called Standard Iterative Scaling (cf. Csiszar and Shields, 2004).

Let \( \{ B_1, \ldots, B_k \} \) be a partition of \( \mathcal{X} \), and the \( B \)-clumping linear family

\[
\mathcal{L} = \{ P : P(B_i) = \alpha_i, \ i = 1, \ldots, k \}
\]

B’s could be the rows (or columns) in a contingency table.

Then the I-projection of \( Q \) to \( \mathcal{L} \) takes the scaling form:

\[
P^*(x) = c_i Q(x), \ x \in B_i,
\]

where

\[
c_i = \frac{\alpha_i}{Q(B_i)}.
\]

That is, \( P^* \) reweights \( A \) to match the clumping masses in \( \mathcal{L} \).
Alternating minimization:

Given $m$ linear families $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m$, let $P_0 = Q$, $P_i = I$ projection of $P_{i-1}$ to $\mathcal{L}_i$,... For $n > m$, let $\mathcal{L}_n = \mathcal{L}_i$ where $i = n (mod m)$. That is $\mathcal{L}_1, \ldots, \mathcal{L}_m$ are repeated cyclically.

Theorem:

If $\bigcap_{i=1}^{m} \mathcal{L}_i = \mathcal{L} \neq \emptyset$,

$$P_n \to P^*,$$

where $P^*$ is the unique I-projection of $Q$ to $\mathcal{L}$. 
Generalized Iterative Scaling:

WLOG, assume

$$\mathcal{L} = \{P : \sum_{x \in \mathcal{X}} P(x) f_i(x) = \alpha_i, 1 \leq i \leq m\},$$

where

$$f_i(x) \geq 0, \sum_i f_i(x) = 1, x \in \mathcal{X}.$$ 

Let $P^*$ be the I-projection of $Q$ to $\mathcal{L}$, then

$$b_n(x) \to Q(x),$$

where $b_n$ are obtained through GIS.
GIS:

\[
b_0(x) = Q(x); \]

\[
b_{n+1}(x) = b_n(x) \prod_{i=1}^{m} \left( \frac{\alpha_i}{\beta_{n,i}} \right) f_i(x), \]

\[
\beta_{n,i} = \sum_{x \in \mathcal{X}} b_n(x) f_i(x). \]

- In IPF, the \( f_i \) are either 1 or 0, and the ratios \( \frac{\alpha_i}{\beta_{n,i}} \) are the same scalings.
- \( b_n \) are not necessarily probability distributions, but their limit is.
Rissanen’s Minimum Description Length Principle

Choose the model that gives the shortest description of data.

- MDL’s intellectual root is K-complexity because it seeks the shortest description/program of data. However,

- In MDL, the description is made equivalent to a probabilistic model, bypassing the non-computability of K-complexity, and

- MDL lays its foundation on Shannon’s information theory. That is, the description is not the length of a binary program, but the length of a binary code using the equivalence of a code with a probability distribution via Kraft’s inequality.
MDL is a general principle for Statistical Inference without assuming there is a true data generating model or distribution.

A central question in MDL research:

Which description to use based on a model?

or why a particular description form could be used in MDL to select a model?
1. Parametric estimation

Each model corresponds to a particular distribution in a parametric family.

By Shannon’s source coding theorem, the best code based on a particular distribution is

\[- \log f_\theta(x^n)\]

So this is the most efficient description based on one distribution and it qualifies to be used in MDL for ”model” selection or parameter estimation.

Suppose further that the parameter space is compact, then we can use a fixed precision to quantize the parameter space and with a uniform code on the parameters, using model ”$\theta$” to code the data string $x^n$ leads to a code length

\[- \log f_\theta(x^n) + \text{a fixed cost for sending the discretized parameter value.}\]

Hence, MDL=MLE in this case.
The problem of model selection

Example 1:

\[ y_i = \text{a quantitative pheno-type value (average no of bristles on fruit flies)} \]

\[ x_i = \text{indicator vector of genetic markers and their interactions (39 dim)} \]

Scientific question:

which markers or interactions of markers are responsible for the bristles?
One formulation to help answer the scientific question:

model selection in regression.

Let submodel class be denoted by $\mathcal{M}_\gamma$

where $\gamma$ is a binary vector in $\{0, 1\}^{39}$ indicating which predictors are in $1'$s and which are out $0'$s.

Goal: choose one which is the "best".
Another example:

Stimulus-response function estimation in neuroscience.

\[ y_t = \text{average neuron firing rate at time interval } "t" \]

\[ x_t = \text{vector of "relevant" stimulus features at time interval } "t". \]

Scientific question: which features drive the neuron or make the neuron fire?

One formulation: model selection regression.
For the model selection problem, MLE doesn’t work because it chooses the largest model in a nested situation like in regression.

In the early 70’s, many revisions of MLE were proposed, and they are precursors to MDL on model selection:

Mallows’ Cp, 70’s (regression, prediction justification)

Akaike’s AIC, 73, 74 (KL divergence justification)

Schwartz’s BIC, 78 (approximation to Bayesian model selection)

They are all penalizing the maximized log likelihood by some complexity measure of the submodel class.
In model selection, the problem is to choose a submodel CLASS (not a single distribution), but

which description length to use in MDL?

We need a code length most representative of the whole model class. That is, we need a universal version of the Shannon coding theorem.

What is the ultimate limit of coding over a class of distributions without knowing which distribution generated the data string?
Rissinen (1996) provided such a theorem.

Given a parametric family or model class

$$\mathcal{M} = \{ f_\theta(x^n) : \theta \in \Theta \subset \mathbb{R}^k \},$$

let $E_\theta \{ \cdot \}$ denote the expectation with respect to a random variable (data string) $X^n$ having density $f_\theta$.

For any density (or prefix code) $q(x^n)$, the Kullback-Leibler divergence between $f_\theta$ and $q$ is given by

$$R_n(f_\theta, q) = E_\theta \log \frac{f_\theta(X^n)}{q(X^n)}$$

$$= E_\theta \left\{ - \log q(X^n) - \left[ - \log f_\theta(X^n) \right] \right\}.$$  \hfill (0.20)

$R_n(f_\theta, q)$ represents the expected extra nats needed to encode the data string $X^n$ using $q$ rather than the optimal scheme based on $f_\theta$. 

174
Defining a valid description length for a data string based on models from the class $\mathcal{M}$ reduces to finding a density $q$ that achieves the “smallest” redundancy possible for all members in $\mathcal{M}$. To make this concrete, we present Rissanen’s lower bound on redundancy in a well-defined global sense over the entire class $\mathcal{M}$. 
Assume that a $\sqrt{n}$-rate estimator $\hat{\theta}(x^n)$ for $\theta$ exists and the distribution of $\hat{\theta}(X^n)$ has uniformly summable tail probabilities:

$$P_{\theta}\left\{ \sqrt{n}\|\hat{\theta}(X^n) - \theta\| \geq \log n \right\} \leq \delta_n, \quad \text{for all } \theta \text{ and } \sum_n \delta_n < \infty,$$

where $\|\theta\|$ denotes some norm in $^k$. Then, for any density $q$, Rissanen (1986a) finds that

$$\liminf_{n \to \infty} \frac{E_{\theta} \log[f_\theta(X^n)/q(X^n)]}{(k/2) \log n} \geq 1,$$

for all $\theta \in \Theta$, except on a set of $\theta$ with a Lebesgue measure zero. This exceptional set depends on $q$ and $k$. 

176
Minimax Lower Bound:

The bound (0.21) holds for almost every value of $\theta \in \Theta$, hence the term pointwise. We now turn to a minimax version of this result. cf. Barron et al. (1998) for minimax approach in MDL and nonparametric estimation.

First, we define the minimax redundancy to be

$$ R_n^+ = \min_q \sup_{\theta \in \Theta} R_n(f_\theta, q). $$ (0.22)

This expression has a simple interpretation as the minimum over all coding schemes for $X^n$ of the worst case redundancy over all parameter values $\theta$. 

177
Consider a prior distribution \( w(\theta) \) on the parameter space \( \Theta \) and define the Bayes redundancy associated with a density \( q \) relative to \( w \) as

\[
R^*_n(q, w) = \int_{\Theta} R_n(f_\theta, q) w(d\theta).
\]  
(0.23)

The minimal Bayes redundancy for a given \( w \) is given by

\[
R_n(w) = \min_q R^*_n(q, w),
\]  
(0.24)

which is achieved by the mixture distribution

\[
m^w(x^n) = \int_{\Theta} f_\theta(x^n) w(d\theta).
\]  
(0.25)
Let $\Theta$ denote the random variable induced by the prior $w$, then the last expression above is also the mutual information $I_w(\Theta; X^n)$ between $\Theta$ and the random variable $X^n = X_1, \ldots, X_n$.

$$R_n(w) = I_w(\Theta; X^n).$$  \hfill (0.26)

Let $R_{n}^{-}$ denote the worst case minimal Bayes redundancy among all priors $w$:

$$R_{n}^{-} = \sup_w R_n(w) = C(\Theta; X^n).$$  \hfill (0.27)

$R_{n}^{-}$ is thus referred to as the *channel capacity*. 
Assuming $\theta$ is to be sampled from a distribution $w(\theta)$, the channel capacity represents the maximal message rate that the noisy channel allows. The capacity-achieving distribution “spaces” the input values of $\theta$, countering the channel noise and aiding message recovery (see Cover and Thomas, 1991).

Now, observe that the channel capacity $C(\Theta; X^n)$, bounds the minimax redundancy $R_n^+$ (0.22) from below:

$$R_n^+ = \min_q \sup_{\theta} R_n(f_\theta, q)$$

$$\geq \sup_w \min_q \int_\Theta R_n(f_\theta, q) w(d\theta)$$

$$= \sup_w \min q R_n^*(q, w) \quad (0.28)$$

$$= \sup_w R_n(w) \quad (0.29)$$

$$\equiv C(\Theta; X^n),$$

where the equalities (0.28) and (0.29) are simply the definitions of the Bayes redundancy (0.23) and the minimal Bayes redundancy (0.27), respectively.
Haussler (1997) demonstrates that in fact the minimax redundancy (0.22) is equal to the channel capacity:

\[ R_n^+ = C(\Theta; X^n) = R_n^- . \] (0.30)

According to this result, if we can calculate the capacity of the channel defined by the pair \( w \) and \( f_{\theta} \), then we can get the minimax redundancy immediately. This statement was first proved by Gallager (1976), although the minimax result of this type for general loss functions was known prior to this point (cf. Le Cam, 1986). See also Davisson (1973), Davisson and Leon-Garcia (1980) and Csiszár (1990).
To be useful, this equivalence requires us to compute the channel capacity for a pair \( w \) and \( f_\theta \). Unfortunately, this can be a daunting calculation. When both the prior and density function are smooth, however, a familiar expansion can be employed to derive a reasonable approximation. Let \( I(\theta) \) denote the Fisher information matrix defined by

\[
I_{i,j}(\theta) = E \left[ \frac{\partial}{\partial \theta_i} \log f(X | \theta) \frac{\partial}{\partial \theta_j} \log f(X | \theta) \right] \quad \text{for all } i, j = 1, \ldots, k.
\]
Assume the observation sequence \( X^n = X_1, \ldots, X_n \) are iid (or memoryless in the parlance of information theory) from some distribution \( f_\theta \) in the class \( \mathcal{M} \). Under regularity conditions on the prior \( w \) and the model class \( \mathcal{M} \), Clarke and Barron (1990) derived the following expansion in the general \( k \)-dimensional case (see Ibragimov and Has’minsky, 1973, for the 1-dimensional case). Let \( K \) be a compact subset in the interior of \( \Theta \). Then, given a positive, continuous prior density \( w \) supported on \( K \), the expected redundancy (0.21) evaluated at the mixture distribution \( m^w \) (0.25) can be expanded as

\[
R_n(f_\theta, m^w) = \frac{k}{2} \log \frac{n}{2\pi e} + \log \frac{\sqrt{\det I(\theta)}}{w(\theta)} + o(1),
\]

where the \( o(1) \) term is uniformly small on compact subsets interior to \( K \).
Averaging with respect to $w$ yields an expansion for the minimal Bayes redundancy, or mutual information, (0.26)

$$R_n(w) = I_w(\Theta; X^n)$$

$$= \frac{k}{2} \log \frac{n}{2\pi e} + \int_K w(\theta) \log \frac{\sqrt{\det I(\theta)}}{w(\theta)} d\theta + o(1).$$

The middle term is maximized by Jeffreys’ prior (when this prior is well-defined):

$$w^*(\theta) = \frac{\sqrt{\det I(\theta)}}{\int_K \sqrt{\det I(\theta)} d\theta},$$
Hence the minimax redundancy satisfies

\[ R_n^+ = \min_q \sup_{\theta \in \Theta} R_n(f_{\theta}, q) = \frac{k}{2} \log \frac{n}{2\pi e} + \log \int_K \sqrt{\det I(\theta)} \, d\theta + o(1). \tag{0.31} \]

Recalling the equivalence (0.30) and the channel capacity interpretation of the worst case minimal Bayes redundancy, Jeffreys’ prior is now seen to be the \textit{capacity-achieving} distribution for the channel defined by the pair \( w \) and \( f_{\theta}(x^n) \).

Intuitively, sampling a message \( \theta \) according to Jeffreys’ prior will result in channel inputs that are well separated in the sense that the probability of correctly reconstructing the message from \( X^n \) is high.
The leading term in (0.31) is the same $\frac{k}{2} \log n$ as in Rissanen’s pointwise lower bound (0.21). Any code that achieves this leading term (to first order) on expected redundancy over a model class qualifies as a code to be used as the description length in the MDL selection for a model (Barron et al., 1998, address qualifying coding schemes based on the constant term). Such codes fairly represent all the members in the model class (in the minimax sense) without the knowledge of exactly which distribution generated our data string.
To gain perspective, we now contrast the analysis of the Kullback-Leibler divergence $R_n(f_\theta, q)$ defined in (0.21) that is carried out for the derivation of $AIC$ with the analysis presented above. For $AIC$, we replace the distribution $q$ with $f_{\hat{\theta}_n}$, where $\hat{\theta}_n$ is the maximum likelihood estimator of $\theta$.\(^a\) Under standard assumptions, the estimate $\hat{\theta}_n$ converges to $\theta$ in such a way that $R_n(f_\theta, f_{\hat{\theta}_n})$ has a negative $\frac{1}{2} \chi^2_k$ limiting distribution. Therefore, the Kullback-Liebler divergence $R_n(f_\theta, f_{\hat{\theta}_n})$ has a limiting mean of $-\frac{k}{2}$.

\(^a\)Note that $f_{\hat{\theta}_n}$ is an estimator of the joint density of $x^n$, but is not a joint distribution. Therefore, it cannot be used to generate a code.
The minimax calculation in (0.31) is focussed on a $q$ which is a joint density of $x^n$ and determined by the set $\Theta$.

It is shown in Rissanen (1996) that the minimax redundancy is achieved asymptotically by the joint density (when it exists) corresponding to the normalized maximum likelihood (NML) code.

That is, $f_{\hat{\theta}_n}(x^n)/C_n$ where $C_n$ is the normalization constant required to make $f_{\hat{\theta}_n}(x^n)$ into a joint density or a code. The $-\frac{k}{2}$ term from the unnormalized maximum likelihood estimator as in the $AIC$ case appears as $\frac{k}{2} \log \frac{1}{e}$ and the rest of the terms in (0.31) give the asymptotic expansion of $C_n$ (cf. Barron et al, 1998).

Hence, $MDL$ criteria that achieve minimax redundancy can be viewed as more conservative criteria that $AIC$ from the perspective of Kullback-Leibler divergence.
The general case:

For more general parameter spaces, Merhav and Feder (1995) prove that the capacity of the induced channel is a lower bound on the redundancy that holds simultaneously for all sources in the class except for a subset of points whose probability, under the capacity-achieving probability measure, vanishes as $n$ tends to infinity. Because of the relationship between channel capacity and minimax redundancy, this means that the minimax redundancy is a lower bound on the redundancy for “most” choices of the parameter $\theta$, hence generalizing Risssanen’s lower bound.
For the case when the source is memoryless, that is, when the observations are conditionally independent given the true parameter $\theta$, and have a common distribution $f_\theta$, $\theta \in \Theta$, Haussler and Opper (1997) obtain upper and lower bounds on the mutual information in terms of the relative entropy and Hellinger distance. Using these bounds and the relation between the minimax redundancy and channel capacity, asymptotic values for minimax redundancy can be obtained for abstract parameter spaces.
Forms of valid description length (achieving the lower bound)

1. Two-stage Description Length

We first choose a member of the class $\mathcal{M}$ and then use this distribution to encode $x^n$. Because we are dealing with a parametric family, this selection is made via an estimator $\hat{\theta}_n$ after which a prefix code is built from $f_{\hat{\theta}_n}$.

Ultimately, the code length associated with this scheme takes the form of a penalized likelihood, the penalty being the cost to encode the estimated parameter values $\hat{\theta}_n$. 

Stage 1:

*The description length* \( L(\hat{\theta}_n) \) *for the estimated member* \( \hat{\theta}_n \) *of the model class.*

In the first stage of this coding scheme, we communicate an estimate \( \hat{\theta}_n \) (obtained by, say, ML or some Bayes procedure). This can be done by first discretizing a compact parameter space with precision \( \delta_m = 1/\sqrt{n} \) (\( m \) for the model) for each member of \( \theta \), and then transmitting \( \hat{\theta}_n \) with a uniform encoder.
The intuitive argument is that $1/\sqrt{n}$ represents the magnitude of the estimation error in $\hat{\theta}_n$ and hence there is no need to encode the estimator with greater precision. In general, our uniform encoder should reflect the convergence rate of the estimator we choose for this stage. Assuming the standard parametric rate $1/\sqrt{n}$, we will pay a total of $-k \log \frac{1}{\sqrt{n}} = \frac{k}{2} \log n$ nats to communicate an estimated parameter $\hat{\theta}_n$ of dimension $k$. 
Although the uniform encoder is a convenient choice, we can take any continuous distribution $w$ on the parameter space and build a code for $\hat{\theta}_n$ by again discretizing with the same precision $\delta_m = 1/\sqrt{n}$:

$$L(\hat{\theta}_n) = -\log w([\hat{\theta}_n]_{\delta_m}) + \frac{k}{2} \log n,$$

where $[\hat{\theta}_n]_{\delta_m}$ is $\hat{\theta}_n$ truncated to precision $\delta_m$. In the MDL paradigm, the distribution $w$ is introduced as an ingredient in the coding scheme and not as a Bayesian prior.
However, if we have reason to believe that a particular prior $w$ reflects the likely distribution of the parameter values, choosing $w$ for description purposes is certainly consistent with Shannon’s Source Coding Theorem. It is clear that both recipes lead to description lengths with the same first order term

$$L(\hat{\theta}_n) \approx \frac{k}{2} \log n,$$

where $k$ is the Euclidean dimension of the parameter space.
Stage 2: *The description length of data based on the transmitted distribution.*

In the second stage of this scheme, we encode the actual data string
\[ x^n = (x_1, \ldots, x_n) \]
using the distribution indexed by \([\hat{\theta}_n]_{\delta_m}\). For continuous data, we can discretize with a precision \(\delta_d\) (e.g. to be machine precision). The description length for coding \(x^n\) is then
\[
- \log f(x_1, \ldots, x_n|[\hat{\theta}_n]_{\delta_m}) - n \log \delta_d.
\]
When the likelihood surface is smooth as in regular parametric families, the difference
\[
\log f(x_1, \ldots, x_n | [\hat{\theta}_n]_{\delta_m}) - \log f(x_1, \ldots, x_n | \hat{\theta}_n)
\]
is of a smaller order of magnitude than the model description length \( \frac{k}{2} \log n \). In addition, the quantity \( n \log \delta_d \) is constant for all the models in \( \mathcal{M} \). Hence we often take
\[
- \log f(x_1, \ldots, x_n | \hat{\theta}_n),
\]
the negative of the maximized log-likelihood for the MLE \( \hat{\theta}_n \), as the simplified description length for a data string \( x^n \) based on \( f(\cdot | \hat{\theta}_n) \).
Combining the code or description lengths from the two stages of this coding scheme, we find that for regular parametric families of dimension \(k\), the (simplified) two-stage MDL criterion takes the form of \(BIC\)

\[
- \log f(x_1, \ldots, x_n | \hat{\theta}_n) + \frac{k}{2} \log n.
\]  

(0.32)

Because

\[
- \log f(x_1, \ldots, x_n | \theta_n) - (- \log f(x_1, \ldots, x_n | \theta)) \approx -\frac{1}{2} \chi^2_k,
\]

we can easily see that the two-stage code length is VALID (achieving the lower bound).
Again, the first term represents the number of nats needed to encode the date sequence $x^n$ given an estimate $\hat{\theta}_n$, while the second term represents the number of nats required to encode the $k$ components of $\hat{\theta}_n$ to precision $1/\sqrt{n}$.

It is worth noting that the simplified two-stage description length is valid even if one starts with a $1/\sqrt{n}$-consistent estimator other than the MLE, even though traditionally only MLE has been used. This is because only the rate of a $1/\sqrt{n}$-estimator is reflected in the $\log n$ term. In more complicated situations such as the clustering analysis presented in Section 4, more than two stages of coding might be required.
Example:

Selecting between \( \mathcal{M}_0 \) and \( \mathcal{M}_1 \).

Encoding \( x^n = (x_1, \ldots, x_n) \) based on \( \mathcal{M}_0 = \{N(0, 1)\} \):

\[
L_0(x^n) = \frac{1}{2} \sum_{t=1}^{n} x_t^2 + \frac{n}{2} \log(2\pi).
\]

Next, consider encoding \( x^n \) via a two-stage scheme based on the class

\[
\mathcal{M}_1 = \{N(\theta, 1) : \theta \neq 0\}
\]

If we estimate \( \theta \) by the MLE \( \hat{\theta}_n = \bar{x}_n \), the two-stage description length (0.32) takes the form

\[
L_1(x^n) = \frac{1}{2} \sum_{t=1}^{n} (x_t - \bar{x}_n)^2 + \frac{n}{2} \log(2\pi) + \frac{1}{2} \log n. \tag{0.33}
\]
Therefore, following the MDL principle, we choose $\mathcal{M}_0$ over $\mathcal{M}_1$ based on the data string $x^n$, if

$$|\bar{x}_n| < \sqrt{\log(n)/n}.$$  

In this case, the MDL criterion takes the form of a likelihood ratio test whose significance level shrinks to zero as $n$ tends to infinity.
It is easy to see that AIC is equivalent to: choosing $\mathcal{M}_0$ iff

$$|\bar{x}_n| < \sqrt{2/n}.$$ 

A model selection criterion is called *model selection consistency* if the model selection criterion selects the true model with probability tending to 1 when the sample size increases.
AIC is not consistent unless the largest model is the true model.

BIC is consistent.
2. Mixture MDL and Stochastic Information Complexity

\[ m(x^n) = \int f_\theta(x^n)w(\theta)d\theta. \] (0.34)

Again, we introduce \( w \) not as a prior in the Bayesian sense, but rather as a device for creating a distribution for the data based on the model class \( \mathcal{M} \). Given a precision \( \delta_d \), we obtain the description length

\[ - \log m(x^n) = - \log \int f(x_1, \ldots, x_n | \theta)w(\theta)d\theta + n \log \delta_d. \]

Ignoring the constant term, we arrive at

\[ - \log \int f(x_1, \ldots, x_n | \theta)w(\theta)d\theta. \] (0.35)
This integral has a closed form expression when $f(\cdot | \theta)$ is an exponential family and $w$ is a conjugate prior, as was the case in the 1-dim Gaussian example below. When choosing between two models, the mixture form of MDL is equivalent to a Bayes factor (Kass and Raftery, 1995) based on the same priors. A popular method for calculating Bayes factors involves the use of Markov chain Monte Carlo (George and McCulloch, 1997), which can therefore be applied to obtain the description length of mixture codes.

From Clarke and Barron (1990), we can see that when $w$ is smooth, the mixture form achieves the lower bound so it is also valid to use in MDL.
Example continued: If we put a Gaussian prior $w = N(0, \tau)$ on the mean parameter $\theta$ in $\mathcal{M}_1$ (note that $\tau$ is the variance), we find

$$-\log m(x^n) = \frac{n}{2} \log(2\pi) + \frac{1}{2} \log(I_n + \tau J_n) + \frac{1}{2} x'_n (I_n + \tau J_n)^{-1} x_n$$

(0.36)

where $I_n$ is the $n \times n$ identity matrix, and $J_n$ is the $n \times n$ matrix of 1’s.

Simplifying the above expression, we arrive at

$$\frac{1}{2} \sum_t x_t^2 - \frac{1}{2} \frac{n}{1 + 1/(n\tau)} \bar{x}_n^2 + \frac{n}{2} \log(2\pi) + \frac{1}{2} \log(1 + n\tau)$$

(0.37)
Comparing this to the description length for the two-stage encoder (0.33), we find a difference in the penalty

\[ \frac{1}{2} \log(1 + n\tau) \]  \hspace{1cm} (0.38)

which (to first order) is asymptotically the same as that associated with $BIC$, $\frac{1}{2} \log n$. Depending on the value of the prior variance $\tau$, the quantity (0.38) represents either a heavier ($\tau > 1$) or a lighter ($\tau < 1$) penalty. In Figure 1 we present a graphical comparison for two values of $\tau$. 
Figure 1: Comparing the penalties imposed by $BIC$ and the mixture form of MDL for $\tau = 0.5$ and $\tau = 2$. The sample size $n$ ranges from 1 to 50.
An analytical approximation to the mixture $m(\cdot)$ in (0.34) is obtained by Laplace’s expansion when $w$ is smooth (Rissanen, 1989). Essentially, we arrive at a two-stage description length which we will call the Stochastic Information Complexity:

$$SIC(x^n) = -\log f(x^n|\hat{\theta}_n) + \frac{1}{2} \log(\hat{\Sigma}_n),$$

(0.39)

where $\hat{\theta}_n$ is the MLE and $\hat{\Sigma}_n$ is the Hessian matrix of $-\log f(x^n|\theta)$ evaluated at $\hat{\theta}_n$. For iid observations from a regular parametric family and as $n \to \infty$,

$$\frac{1}{2} \log(\hat{\Sigma}_n) = \frac{1}{2} \log(nI(\hat{\theta}_n))(1 + o(1)) = \frac{k}{2} \log n(1 + o(1)).$$

(0.40)
Here, $I(\cdot)$ is the Fisher information matrix of a single observation. The middle term in this chain of equalities,

$$
\frac{1}{2} \log(nI(\hat{\theta})),
$$

(0.41)
can be interpreted as the number of nats needed to encode the $k$ estimated parameter values if we discretize the $j$th parameter component with a precision $SE(\hat{\theta}_j) = 1/\sqrt{nI_{jj}(\theta)}$ (provided the estimated parameters are either independent or the discretization is done after transforming the parameter space so that the information matrix under the new parameterization is diagonal). It is obviously sensible to take into account the full estimation error when discretizing, and not just the rate. The final equality in (0.40) tells us that in the limit, $SIC$ is approximately $BIC$ or two-stage MDL.
For finite sample sizes, however, $SIC$’s penalty term is usually not as severe as $BIC$’s, and hence in some situations, $SIC$ outperforms $BIC$. Rissanen (1989, pp. 151, Table 6) illustrates this difference by demonstrating that $SIC$ outperforms two-stage MDL when selecting the order in an AR model with $n = 50$. In Section 4, we will present many more such comparisons in the context of ordinary linear regression.
3. Predictive Description Length

Any joint distribution \( q(\cdot) \) of \( x^n = (x_1, \ldots, x_n) \) can be written in its predictive form

\[
q(x^n) = \prod_{t=1}^{n} q(x_t|x_1, \ldots, x_{t-1}).
\]

Conversely, given a model class \( \mathcal{M} \), it is a simple matter to obtain a joint distribution for \( x^n \) given a series of predictive distributions. In many statistical models, each of the conditionals \( f_\theta(x_j|x_1, \ldots, x_{j-1}) \) share the same parameter \( \theta \).\(^a\) For iid data generated from a parametric family \( \mathcal{M} \), this is clearly the case. Other applications where this property holds include time series, regression and generalized linear models.

\(^a\)Typically, \( f(x_1) = f_0(x_1) \) will not depend on \( \theta \), however.
Suppose that for each $t$, we form an estimate $\hat{\theta}_{t-1}$ from the first $(t - 1)$ elements of $x^n$. Then, the expression

$$q(x_1, \ldots, x_n) = \prod_t f_{\hat{\theta}_{t-1}}(x_t | x_1, \ldots, x_{t-1})$$

(0.42)

represents a joint distribution based on the model class $\mathcal{M}$ that is free of unknown parameters. The cost of encoding a data string $x^n$ using (0.42) is

$$- \log q(x_1, \ldots, x_n) = - \sum_t \log f_{\hat{\theta}_{t-1}}(x_t | x_1, \ldots, x_{t-1}).$$

(0.43)

The MDL model selection criterion based on this form of description is called \textit{PMDL} for its use of the predictive distribution (0.42) and PMDL is especially useful for time series models (cf. Hannan and Rissanen, 1982; Hannan, McDougall and Poskitt, 1989; Huang, 1990).
By design, predictive MDL is well suited for time series analysis, where there is a natural ordering of the data; on-line estimation problems in signal processing; and on-line data transmission applications like the binary string example discussed Section 2. At a practical level, under this framework both sender and receiver start with a pre-determined encoder $f_0$ to transmit the first data point $x_1$. This accounts for the leading term in the summation (0.43). At time $t$, because the previous $(t - 1)$ points are known at each end of the channel, the distribution $f_{\hat{\theta}_{t-1}}(x_t|x_1, \ldots, x_{t-1})$ is also known. This is the $t$th term in the summation (0.43). By using the predictive distributions to sequentially update the code, both the encoder and decoder are in effect learning about the true parameter value, and hence can do a better job of coding the data string (provided that one member of the model class actually generated the data).
Example continued If we take the initial density $f_0$ as $N(0, 1)$ and set

$$\hat{\theta}_{t-1} = \bar{x}_{t-1} = \sum_{i=1}^{t-1} x_i / (t - 1)$$

(with $\bar{x}_0 = 0$) based on $M_1$, then

$$- \log q(x^n) = - \sum_{t=1}^{n} \log f_{\hat{\theta}_{t-1}} (x_t | x^{t-1})$$

$$= n \frac{1}{2} \log(2\pi) + \frac{1}{2} \sum_{t=1}^{n} (x_t - \bar{x}_{t-1})^2. \quad (0.44)$$
The reasoning we followed in deriving $PM_{DL}$ is identical to the prequential approach to statistics advocated by Dawid (1984, 1991). The form (0.43) appeared in the literature on Gaussian regression and time series analysis as the *predictive least squares criterion* long before the development of MDL, and early work on $PM_{DL}$ focused mainly on these two applications. The interested reader is referred to Rissanen (1986b), Hemerly and Davis (1989), Hannan and Kavalieris (1984), Hannan, McDougall and Poskitt (1989), Hannan and Rissanen (1982), Gerencsér (1994), Wei (1992), and Speed and Yu (1994). The recent results of Qian, Gabor and Gupta (1996) extend the horizon of this form of MDL to generalized linear models.
Model selection consistency property of PMDL in the example
Although in special cases such as multinomial the ordering disappears when a Bayes estimator is used for the prediction, in general $PMDL$ depends on a sensible ordering of the data. It is not clear how useful it will be in, say, multivariate regression problems. To get around this problem, Rissanen (1986b) suggests repeatedly permuting the data before applying $PMDL$, and then averaging the predictive code lengths. In Section 4, we avoid these complications and only discuss $PMDL$ in the context of time series data.
Other Forms of Description Length:

The MDL principle offers one the opportunity to develop many other forms of description length, in addition to the three discussed above. For example, weighted averages or mixtures of the three common forms will give rise to new description lengths that all achieve the pointwise and minimax lower bounds on redundancy, and hence can all be used for model selection. Further investigation is required to determine how to choose these weights in different modeling contexts.
Normalized Maximum Likelihood (NML) Form

Rissanen (1996) developed an MDL criterion based on the normalized maximum likelihood coding scheme of Shtarkov (1987) (cf. Barron et al., 1998). For a flavor of how it was derived, we apply \textit{NML} (for normalized maximized likelihood) to the binary, DJIA up-and-down indicators introduced in Section 2.

Example: \textbf{Normalized Maximized Likelihood Coding in the iid Bernoulli case}

For $x_1, \ldots, x_n$ iid Bernouli data with probability $p$, the maximum likelihood estimator of $p$ is

$$\hat{p}_n = \sum_i x_i / n.$$  \hspace{1cm} (1)

The maximized likelihood is

$$h(x_1, \ldots, x_n) = \hat{p}^{\sum_i x_i} (1 - \hat{p})^{n - \sum_i x_i},$$

which is a positive function on the binary $n$-tuples.
Let $C_n = \sum_{x_1,\ldots,x_n} h(x_1, \ldots, x_n)$ be the normalizing constant, then

$$q(x_1, \ldots, x_n) = h(x_1, \ldots, x_n)/C_n.$$ 

In the continuous case, one needs to make sure $h$ is integratable:

$$h(x^n) = f(x^n | \hat{\theta}_n),$$

where $\hat{\theta}_n$ is the maximum likelihood estimator based on $x_1, \ldots, x_n$.

In the $f = \mathcal{N}(\theta, 1)$ case as in many continuous cases,

$$h(x_1, \ldots, x_n) \propto \exp\left(-(\sum_i (x_i - \bar{x}_n)^2 / 2)\right)$$

is not integratable, unfortunately.

$n = 2$,

$$h(x_1, x_2) \propto \exp^{-(x_1-x_2)^2},$$
not integratable.
Linear Regression Models

response variable: $y$ potential predictors $x_1, \ldots, x_M$.

Associate with each predictor $x_m$ a binary variable, $\gamma_m$, and consider models given by

$$y = \sum_{\gamma_m=1} \beta_m x_m + \epsilon,$$  \hspace{1cm} (0.45)

where $\epsilon$ has a Gaussian distribution with mean zero and unknown variance $\sigma^2$.

The vector $\gamma = (\gamma_1, \ldots, \gamma_M) \in \{0, 1\}^M$ will be used as a simple index for the $2^M$ possible models given by (0.45). Let $\beta_\gamma$ and $X_\gamma$ denote the vector of coefficients and the design matrix associated with those variables $x_m$ for which $\gamma_m = 1$. 

223
We apply MDL to the problem of linear regression model selection, or equivalently, the problem of identifying one or more vectors $\gamma$ that yield the “best” or “nearly best” models for $y$ in equation (0.45). In many cases, not all of the $2^M$ possibilities make sense, and hence our search might be confined to only a subset of index vectors $\gamma$.

The concept of “best,” or more precisely the measure by which we compare the performance of different selection criteria, is open to debate.

Two criteria:

consistency and prediction performance.

In general, we should apply any selection procedure with some care, examining the structure of several good-fitting models rather than restricting our attention to a single “best.” This point tends to be lost in simulation studies that necessitate blunt optimization of the criterion being examined.
Our review of regression problems draws from number of sources on MDL (see Rissanen, 1987 and 1989; Speed and Yu, 1993; and Barron et al, 1998) as well as the literature on Bayesian variable selection (see Smith and Spiegelhalter, 1980; O’Hagan, 1994; Kass and Raftery, 1995; and George and McCulloch, 1997). Because the need for selection in this context arises frequently in applications, we will derive several MDL criteria in detail.
Following the general recipe given in the previous sections, the MDL criteria we derive for regression can all be written as a sum of two code lengths

\[ L(y|X_\gamma, \gamma) + L(\gamma). \]  \hfill (0.46)

\[ P(\gamma) = \left( \frac{1}{2} \right)^k \left( \frac{1}{2} \right)^{M-k} = \left( \frac{1}{2} \right)^M. \]  \hfill (0.47)

Therefore, the cost \( L(\gamma) = -\log P(\gamma) \) is constant. When we have reason to believe that smaller or larger models are preferable, a different Bernoulli model (with a smaller or larger value of \( p \)) can be used to encode \( \gamma \). This approach has been taken in the context of Bayesian model selection and is discussed at the end of this section.
For the most part, we will work with maximum likelihood estimates for both the regression coefficients $\beta$ (also known as ordinary least squares or OLS estimates) and the noise variance $\sigma^2$,

$$\hat{\beta} = (X'X)^{-1}X'y \quad \text{and} \quad \hat{\sigma}^2 = \|y - X\hat{\beta}\|^2/n.$$  \hspace{1cm} (0.48)

Finally, we take $RSS$ to represent the residual sum of squares corresponding to this choice of $\hat{\beta}$.  

227
Two-stage MDL:

\[ \frac{1}{2\sigma^2} RSS + \frac{k}{2} \log n \] (0.49)

when \( \sigma^2 \) is known, and

\[ \frac{n}{2} \log RSS + \frac{k}{2} \log n \] (0.50)

when it is unknown.
In both cases, the penalty applied to the dimension $k$ depends on the sample size $n$. Related criteria like Mallows’ $C_p$ (Mallows, 1973) and Akaike’s $AIC$ (Akaike, 1974) differ only in the size of this penalty:

$$C_p = \frac{1}{2\sigma^2} RSS + k \quad \text{and} \quad AIC = \frac{n}{2} \log RSS + k,$$

(0.51)

where we have again ignored terms that do not depend on our choice of model.\(^a\)

While keeping the general form of these criteria, various authors have suggested other multipliers in front of $k$ that can offer improved performance in special cases: see Sugiura (1978) and Hurvich and Tsai (1989) for a corrected version of $AIC$ for small samples; Hurvich, Simonoff and Tsai (1998) for $AIC$ in nonparametric regression; and Mallows (1995) for an interpretation of $C_p$ when a different value of the penalty on model size is desired. Shortly, we will present an application in which a multiple of the $BIC$ penalty is proposed as the “correct” cost for a particular class of problems arising in genetics.

\(^a\)The form of $C_p$ given above applies when $\sigma^2$ is known. If not, Mallows (1973) suggests using an unbiased estimate $\hat{\sigma}^2$. 
Mixture MDL:

We form a mixture distribution for $y$ (conditional on our choice of model and the values of the predictors $X$) by introducing a density function $w(\beta, \sigma^2)$,

$$m(y|X) = \int f(y|X, \beta, \tau) w(\beta, \tau) \, d\beta \, d\tau. \quad (0.52)$$

To obtain a closed-form expression for $m(y|X)$, Rissanen (1989) takes $w$ to be a member of the natural conjugate family of priors for the normal linear regression model (0.45); namely the so-called normal-inverse-gamma distributions (see the appendix). Under this density, the noise variance $\sigma^2$ is assigned an inverse-gamma distribution with shape parameter $a$. Then, conditional on $\sigma^2$, the coefficients $\beta$ have a normal distribution with mean zero and variance-covariance matrix $\frac{\sigma^2}{c} \Sigma$, where $\Sigma$ is a known, positive definite matrix.
In his original derivation, Rissanen (1989) selects $\Sigma$ to be the $k \times k$ identity matrix. Sidestepping this decision for the moment, the mixture code length for $y$ computed from (0.35) is given by

$$\begin{align*}
- \log m(y|X) \\
= - \log m(y|X, a, c) \\
= - \frac{1}{2} \log |c\Sigma^{-1}| + \frac{1}{2} \log |c\Sigma^{-1} + X^t X| - \frac{1}{2} \log a + \frac{n + 1}{2} \log (a + \ldots) 
\end{align*}$$

(0.53)

where

$$R_c = R_c = y^t y - y^t X (c\Sigma^{-1} + X^t X)^{-1} X^t y.$$
In expression (0.54), we have made explicit the dependence of the mixture code length on the values of two hyperparameters in the density $w$: $a$, the shape parameter of the inverse-gamma distribution for $\sigma^2$, and $c$, the (inverse) scale factor for $\beta$.

Rissanen (1989) addresses the issue of hyperparameters by picking $a$ and $c$ to minimize the quantity (0.54) model by model. It is not difficult to see that $
abla = R_c/n$, while for most values of $\Sigma$, $\hat{c}$ must be found numerically. An algorithm for doing this is given in the appendix. Treating $a$ and $c$ in this way, however, we lose the interpretation of $-\log m(y|X, \hat{a}, \hat{c})$ as a description length.
To remain faithful to the coding framework, the optimized hyperparameter values $\hat{a}$ and $\hat{c}$ must also be transmitted as overhead. Explicitly accounting for these extra factors gives us the mixture code length

$$- \log m(y|X, \hat{a}, \hat{c}) + L(\hat{a}) + L(\hat{c}).$$

(0.55)

Because $\hat{a}$ and $\hat{c}$ are determined by maximizing the (mixture or marginal) log-likelihood (0.54), they can be seen to estimate $a$ and $c$ at the standard parametric rate of $1/\sqrt{n}$. Therefore, we take a two-stage approach to coding $\hat{a}$ and $\hat{c}$ and assign each a cost of $\frac{1}{2} \log n$ bits. Rissanen (1989) argues that no matter how we account for the hyperparameters, their contribution to the overall code length should be small.
An important ingredient in our code length (0.55) is the prior variance-covariance matrix, $\Sigma$. As mentioned above, for most values of $\Sigma$ we cannot find a closed-form expression for $\hat{c}$ and must instead rely on an iterative scheme. (A general form for the procedure is outlined in the appendix.) Rissanen (1989) gives details for the special case $\Sigma = I_{k \times k}$. We refer to the criterion derived under this specification as $iMDL$, where $i$ refers to its use of the identity matrix.
In the Bayesian literature on linear models, several authors have suggested a computationally attractive choice for $\Sigma$; namely $\Sigma = (X^t X)^{-1}$. Zellner (1986) christens this specification the $g$-prior. In our context, this value of $\Sigma$ provides us with a closed-form expression for $\hat{c}$. After substituting $\hat{a} = R_c/n$ for $a$ in (0.54), it is easy to see that

$$1/\hat{c} = \max (F - 1, 0) \quad \text{with} \quad F = \frac{(y'y - RSS)}{kS},$$

(0.56)

where $F$ is the usual $F$-ratio for testing the hypothesis that each element of $\beta$ is zero, and $S = RSS/(n - k)$. 

235
The truncation at zero in (0.56) rules out negative values of the prior variance. Rewriting (0.56), we find that \( \hat{c} \) is zero unless \( R^2 > k/n \), where \( R^2 \) is the usual squared multiple correlation coefficient. When the value of \( \hat{c} \) is zero, the prior on \( \beta \) becomes a point mass at zero, effectively producing the “null” mixture model\(^a\) corresponding to \( \gamma = (0, \ldots, 0) \).

\(^a\)The null model is a scale mixture of normals, each \( N(0, \tau) \) and \( \tau \) having an inverse-gamma prior.
Substituting the optimal value of $\hat{c}$ into (0.54) and adding the cost to code the hyperparameters as in (0.55), we arrive at a final mixture form

$$g_{MDL} = \begin{cases} 
\frac{n}{2} \log S + \frac{k}{2} \log F + \log n , & R^2 > k/n \\
\frac{n}{2} \log \left( \frac{y'y}{n} \right) + \frac{1}{2} \log n , & \text{otherwise}.
\end{cases}$$

(0.57)

which we will refer to as $g_{MDL}$ for its use of the $g$-prior.

$$F = \frac{(y'y - RSS)}{kS} , \quad S = RSS/(n - k).$$
From this expression, we have dropped a single bit that is required to indicate whether the condition $R^2 < k/n$ is satisfied and hence which model was used to code the data. When $R^2 < k/n$, we apply the null model which does not require communicating the hyperparameter $\hat{c}$. Hence a $\frac{1}{2} \log n$ term is missing from the lower expression.
Unlike most choices for $\Sigma$, the $g-$prior structure provides us with an explicit criterion that we can study theoretically. First, since $n/n = 1 \geq R^2$, this version of mixture MDL can never choose a model with dimension larger than the number of observations. After a little algebra, it is also clear that $gMDL$ orders models of the same dimension according to $RSS$; that is, holding $k$ fixed, the criterion (0.57) is an increasing function of $RSS$. This property is clearly shared by $AIC$, $BIC$ and $C_p$. 
Unlike these criteria, however, \( gM DL \) applies an adaptively determined penalty on model size. Rewriting (0.57) in the form

\[
\frac{n}{2} \log RSS + \frac{\alpha}{2} k
\]  

(0.58)

we find that \( \alpha \) depends on the \( F \)-statistics, so that “poor fitting” models receive a larger penalty.
Normalized Maximised Likelihood

The normalized maximum likelihood form of MDL (cf. Rissanen, 1996, and Barron et al., 1998) is recent and only some of its theoretical properties are known. It is motivated by the maximum likelihood code introduced by Shtarkov (1987). Recall that the maximum likelihood estimates of $\beta$ and $\tau = \sigma^2$ are given by (0.48). Let $f(y|X, \beta, \tau)$ be the joint Gaussian density of the observed data $y$, so that the normalized maximum likelihood function is

$$
\hat{f}(y) = \frac{f(y|X, \hat{\beta}(y), \hat{\tau}(y))}{\int_{\mathcal{Y}(r, \tau_0)} f(z|X, \hat{\beta}(z), \hat{\tau}(z)) dz}, \quad (0.59)
$$

where $\mathcal{Y}(r, \tau_0) = \{ z | \hat{\beta}'(z) X' X \hat{\beta}(z)/n \leq r, \hat{\tau}(z) \geq \tau_0 \}$. 

241
In this case, the maximized likelihood is not integrable, and our solution is to simply restrict the domain of $\hat{f}$ to $\mathcal{Y}$. Recall that we did not encounter this difficulty with the Bernoulli model studied in Section ??; where given the number of 1’s, the binary sequences had a uniform distribution over the type class. Using the sufficiency and independence of $\hat{\beta}(y)$ and $\hat{\tau}(y)$, one obtains

$$- \log \hat{f}(y) = \frac{n}{2} \log RSS - \log \Gamma \left( \frac{n-k}{2} \right) - \log \Gamma \left( \frac{k}{2} \right) + \frac{k}{2} \log \frac{r}{\tau_0} - 2 \log(2k).$$

(0.60)
To eliminate the hyper-parameters $r$ and $\tau_0$, we again minimize the above code length for each model by setting

$$\hat{r} = \frac{\hat{\beta}'(y) X' X \hat{\beta}(y)}{n} = \frac{y'y - RSS}{n}$$

and

$$\hat{\tau}_0 = \frac{RSS}{n}.$$

By substituting these values for $r$ and $\tau_0$ into (0.60), we obtain the selection criteria $nM DL$ ($n$ for “normalized maximum likelihood”),

$$nM DL = \frac{n}{2} \log RSS - \log \Gamma \left( \frac{n - k}{2} \right) - \log \Gamma \left( \frac{k}{2} \right) + \frac{k}{2} \log \frac{y'y - RSS}{RSS} - 2 \log(2k)$$

(0.61)
Technically, we should also add $\frac{1}{2} \log n$ for each of the optimized hyperparameters as we had done for $gMDL$. In this case, the extra cost is common to all models and can be dropped. Rewriting this expression, we find that

$$nMDL = \frac{n}{2} \log S + \frac{k}{2} \log F$$

$$+ \frac{n - k}{2} \log(n - k) - \log \Gamma \left( \frac{n - k}{2} \right) + \frac{k}{2} \log(k) - \log \Gamma \left( \frac{k}{2} \right) - 2$$

up to an additive constant that is independent of $k$. Applying Stirling’s approximation to each $\Gamma(\cdot)$ yields

$$nMDL \approx \frac{n}{2} \log S + \frac{k}{2} \log F + \frac{1}{2} \log(n - k) - \frac{3}{2} \log k.$$ 

We recognize the leading two terms in this expression as the value of $gMDL$ (0.57) when $R^2 > k/n$. This structural similarity is interesting given that these two MDL forms were derived from very different codes.
Comments:

1. As mentioned at the beginning of this section, there are alternatives to our use of the Bernoulli($\frac{1}{2}$) model for coding the index $\gamma$. For example, George and Foster (1999) take the elements of $\gamma$ to be a priori independent Bernoulli random variables with success probability $p$. They then select a value for $p$ by maximum likelihood (in the same way we treated the parameters $a$ and $c$). In early applications of model selection to wavelet expansions, the value of $p$ was fixed at some value less than a half to encourage small models (Clyde, Parmigiani and Vidakovic, 1998).
The use of a normal-inverse-gamma prior with $\Sigma = (X^t X)^{-1}$ appears several times in the literature in Bayesian model selection. For example, Akaike (1977) essentially derives $gMDL$ for orthogonal designs. Smith and Spiegelhalter (1980) use this prior when considering model selection based on Bayes factors where $a = 0$ and $c = c(n)$ is a deterministic function of sample size. These authors were motivated by a “calibration” between Bayes factors and penalized selection criteria in the form of $BIC$ and $AIC$ (see also Smith, 1996; and Smith and Kohn, 1996). Finally, Peterson (1986) builds on the work of Smith and Spiegelhalter (1980) by first choosing $\Sigma = (X^t X)^{-1}$ and then suggesting that $c$ be estimated via (marginal) maximum-likelihood based on the same mixture (0.54). This is essentially Rissanen’s (1989) prescription.
Throughout our development of the various MDL criteria, we have avoided the topic of estimating the coefficient vector $\beta$ once the model has been selected. In the case of $AIC$ and $BIC$, it is common practice to simply rely on OLS. The resemblance of mixture MDL to Bayesian schemes, however, suggests that for this form a shrinkage estimator might be more natural. For example, the criterion $gMDL$ is implicitly comparing models not based on $\hat{\beta}$, but rather the posterior mean (conditional on our choice of model)

$$\max \left( 1 - \frac{1}{F}, 0 \right) \hat{\beta}$$

associated with the normal-inverse-gamma prior and the regression model (0.45). Here, $F$ is defined as in the $gMDL$ criterion (0.57). Recall that the condition that $F > 1$ is equivalent to the multiple $R^2$ being larger than $k/n$. Interestingly, this type of shrinkage estimator was studied by Sclove (1968) and Sclove, Morris and Radhakrishnan (1972), where it was shown to have improved mean squared error performance over OLS and other shrinkage estimators. In the case of $iMDL$, the coefficient vector $\beta$ is estimated via classical ridge regression. Of
course, Bayesian methods can be applied more generally within the MDL framework. For example, in Section ?? we found that any $\sqrt{n}$—consistent estimator can be used in the two-stage coding scheme. This means that we could even substitute Bayesian estimators for $\sigma^2$ and $\beta$ in the two-stage criterion (0.50) rather than $\hat{\beta}$ and $\hat{\sigma}^2$. The beauty of MDL is that each such scheme can be compared objectively, regardless of its Bayesian or frequentist origins.
Next, in several places we are forced to deal with hyperparameters that need to be transmitted so that the decoder knows which model to use when reconstructing the data \( y \). We have taken a two-stage approach, attaching a fixed cost of \( \frac{1}{2} \log n \) to each such parameter. Rissanen (1989) proposes using the universal prior on integers \( L^* \) after discretizing range of the hyperparameters in a model-independent way. If prior knowledge suggests a particular distribution, then naturally it should be used instead. In general, the value of the hyperparameters are chosen to minimize the combined code length

\[
(\hat{a}, \hat{c}) = \min_{(a,c)} \left\{ L(y|X, a, c) + L(a) + L(c) \right\}
\]  (0.62)

where the first term represents the cost of coding the data given the value of the hyperparameters, \( \hat{a} \) and \( \hat{c} \), and the second term accounts for the overhead in sending them. In our derivation of \( iMDL \) and \( gMDL \), we took the latter terms to be constant so that we essentially selected the hyperparameters via maximum (mixture or marginal) likelihood.