ECON5529
Bayesian Theory

Lecture 8

Ellis Scharfenaker

Model Selection

- Once we have constructed a probability model and computed the posterior distribution the next step to good statistical inference is to check and criticize our model.
- There are a variety of standard model checks for Bayesian inference, such as *sensitivity analysis*, as well as more nuanced approaches such the *Minimum Description Length Principle* (MDL), which emerges out of computer science and information theory.
- In essence the question facing the researcher is: *which model is most likely given the data*? Recall, this is exactly how the Bayesian method was introduced. That is, we want to think in terms of models, or hypothesis, rather than parameters θ.
- When there are multiple competing model specifications, a set of posterior distributions is produced requiring some method for comparison.

Sensitivity Analysis

- It is reasonable to believe that a number of probability models can provide an adequate fit to our data.
- In general, the process of evaluating models and determining the sensitivity of the posterior distribution to the assumptions is an interactive and iterative process that continues until there is reasonable and reliable evidence for a particular hypothesis.
- *Sensitivity analysis* is an informal process of altering assumptions according to researcher intuition with the objective of determining the extent to which such changes modify the posterior distribution.
- The purpose of sensitivity analysis is to explore how much does posterior inference change when we make reasonable changes to our model? Such changes typically include a different prior or likelihood function.
- *Robustness* is posterior insensitivity to broad classes of user-specified assumptions and is a systematic process of determining the degree to which posterior inferences are affected by both potential misspecification of the prior and influential data points.
  - *Global robustness* can be summarized as an evaluation of a large class of priors to determine the subsequent range of inferences and asks, “should we abandon the current framework completely?”
  - *Local Robustness* uses methods for determining the volatility of specific reported results and asks, “should the model be modified or extended in any particular direction?”

Posterior Predictive Checking

- A standard way to check for model adequacy is simply to investigate whether the predictions on new data are accurate.
- The idea behind *posterior predictive checking* is that if the model fits the data well, then simulated data from the model ought to be similar to the observed data.
- Any observed discrepancy between the posterior prediction and the actual data may then be due to model misspecification.

Posterior Predictive Checking

- Lets look at the employment and output data.
- To check the posterior predictive performance now entails extracting the simulated values from the model object, and taking a random draw from the normal distribution based on our estimated coefficients β and σ that are drawn to produce our replicated data y^{rep}.
- With the y^{rep} in hand we can calculate all manner of statistics that might be of interest. As a starting point, we can check our minimum value among the replicated data sets versus that observed in the data.
- In this case we see our average minimum not much lower than that seen in the data.

Bayes Factor

- The most straightforward method for comparing a set of posterior distributions is to calculate the ratio of their posterior probabilities. This is a common Bayesian *measure of evidence* that simply calculates the posterior odds.
- The posterior odds ratio gives the odds of one model relative to another and is called the Bayes Factor. The benefits of using the Bayes Factor for model comparison are that it is usually easy to calculate and it has a naturally intuitive interpretation.
- Suppose we have observed data x and wish to test two competing models, M_1 and M_2 which relate these data to separate sets of parameters θ_1 and θ_2. To decide between the two families of density specifications:

  $M_1 : p_1[x \mid \theta_1]$ and $M_2 : p_2[x \mid \theta_2]$
- Following the standard Bayesian framework, specify a prior over the parameter vectors $p[\theta_1]$ and $p[\theta_2]$, and as a result the prior probability of the two models $p[M_1]$ and $p[M_2] = 1 - p[M_1]$. Given the data we produce the posterior probabilities:

  $p[M_1 \mid x]$ and $p[M_2 \mid x] = 1 - p[M_1 \mid x]$
- Converting to an odds scale where odds = $\frac{p}{1-p}$ and using Bayes' theorem:

$$p[M_i \mid x] = \frac{p[x \mid M_i]\, p[M_i]}{p[x \mid M_1]\, p[M_1] + p[x \mid M_2]\, p[M_2]} \text{ for } i = 1, 2$$

- so that the posterior odds ratio in favor of model 1 versus model 2 is then the ratio:

$$\frac{p[M_1 \mid x]}{p[M_2 \mid x]} = \frac{p[x \mid M_1]\, p[M_1]}{p[x \mid M_2]\, p[M_2]}$$

- Which is equivalent to expanding each posterior $p[x \mid M_i]\, p[M_i]$ via Bayes' theorem (the denominator $p[x]$ cancels out) and which says the posterior odds ($\frac{p[M_1 \mid x]}{p[M_2 \mid x]}$) is equal to the prior odds ($\frac{p[M_1]}{p[M_2]}$) times the Bayes Factor ($\frac{p[x \mid M_1]}{p[x \mid M_2]} = \frac{\int_{\theta_1} p_1[x \mid \theta_1]\, p[\theta_1]\, d\theta_1}{\int_{\theta_2} p_2[x \mid \theta_2]\, p[\theta_2]\, d\theta_2}$).
- The quantity of interest is this ratio of marginal likelihoods from the two models. By rearranging we get the standard form of the Bayes factor, which is the magnitude of the evidence for model 1 over model 2 contained in the data:

$$B[x; M_i] = \frac{p[M_1 \mid x] / p[M_1]}{p[M_2 \mid x] / p[M_2]}$$

- In the case that we are willing to put equal prior probabilities on the two models the Bayes factor reduces to the standard likelihood ratio. Note that Bayes factors do not have an inherent scale. However, a typical arbitrary decision threshold is:

$B[x] \geq 1$: model 1 supported

$1 > B[x] \geq 10^{-1/2}$: minimal evidence against model 1

$10^{-1/2} > B[x] \geq 10^{-1}$: substantial evidence against model 1

$10^{-1} > B[x] \geq 10^{-2}$: strong evidence against model 1

$10^{-2} > B[x]$: decisive evidence against model 1

Example

- Suppose that two economists, Paul Krugman and Larry Summers are interested in public opinion about a $15 minimum wage. Paul believes that 70% of the public support the $15 minimum wage and Larry believes that the support is only 60%. Paul and Larry decide to ask 100 randomly selected people whether they support the $15 minimum wage.
- Clearly Paul and Larry have predictions about how the sample will turn out where Paul's best guess is that 70 out of 100 will support the wage while Larry's best guess is 60 will support it.
- Since the data will be binomial distributed we can show Paul and Larry's priors.

In R

- Having made predictions, Paul and Larry collect their random sample. Of the 100 people in the sample, 62 are supportive. This seems to support Larry because the observation is closer to Larry's average prediction; however, Paul points out that he also predicted that 62 was possible, and so his hypothesis is not ruled out. The question is by how much does the data support Larry's hypothesis?
- What is the probability of seeing 62 supporters given Paul's hypothesis and Larry's hypothesis?
- So Larry thought the observation $x = 62$ had a probability of 0.0754 and Paul thought this observation has a probability $p[x = 62] = 0.0191$.
- The Bayes factor, which calculates the evidence in the data for a particular hypothesis is the factor by which Larry's line is taller than Paul's.
- So the observed data favors Larry by a factor of about 4.

Minimum Description Length (MDL)

- There is a close relationship between information theory (Shannon 1947) and probability (Jaynes 2003). The posterior probability distribution can be understood as a way of *coding the information* in the data x, using the hypothesis as a model.

- We can think of the problem as conveying the information in the data to some receiver over a *communication channel*.
- A communication channel is a means used to convey an information signal. Examples of communication channels are telecommunication channels, radio channels, and computer networks and hardrives.
- Communication is not required to be simultaneous as in telecommunication. For example, when we write a file on a disk drive, we may read it off in the same location at a later time.
- It is always possible to transmit the information in the data directly by coding the numbers in computer characters and sending them one at a time.
- A *fixed-point* is designed to represent and manipulate integers typically with a minimum of 16 bits. This yields up to 2^{16} = 65,536 possible bit patterns (A bit is just a binary digit). A *floating-point* is designed to represent and manipulate rational numbers usually with a *minimum* of 32 bits and thus can support a much wider range of values than fixed point.
- If we have data $x \in \mathbb{R}$ we can code the data in a computer using *floating points*. *Mathematica* uses 16 bytes (16×8=128 bits) to code a *floating point*.
- By simply sending the data by conveying each number individually we achieve zero *data compression.* If we had 100 observations in our data it would take 128×100=12,800 bits to code and send. If we can reduce these 12,800 bits we achieve data compression.

```
Transpose@{Table[i, {i, 1, 10}], Array[BaseForm[#, 2] &, 10]} // TableForm
```

```
1     1_2
2     10_2
3     11_2
4     100_2
5     101_2
6     110_2
7     111_2
8     1000_2
9     1001_2
10    1010_2
```

```
BaseForm[RandomReal[10], 2]
```

```
10.011100001101001_2
```

Data Compression

- Say we have the following data string of 10,000 bits.

  x = {1, 0, 1, 0, 1, 0, 1, 0, ..., 1, 0, 1, 0}
- In order to communicate this string of data to a receiver over a communication channel we can send the entire string of data which will cost us 10,000 bits.
- However, we quickly notice a pattern in the data, namely a one and then a zero are being repeated. Noticing this redundancy in the data we realize we can send the information in the data using far fewer bits. We write the simple computer code

```
for j in 1 : 5000; print (1, 0); halt
```

- Assuming this brief program costs 10 bits to write and adding the fact that we need to convey the two numbers (1,0) which each cost a bit, we can now send the program code as well as the (1,0) for a total cost of 12 bits, thus achieving a great amount of data compression.

- Intuitively, compression works by taking advantage of the predictability (redundancy/patterns) of a data string.
- Data compression takes advantage of redundancy in the structure of the data. *Lossy* data compression, such as JPEG, compresses the data while throwing out some of the information about the structure of the data. *Lossless* data compression, such as ZIP, retains all information and can thus be completely and accurately uncompressed.

Inductive Inference and Data Compression

- The task of *inductive inference* is to find *laws* or *regularities* underlying some given set of fixed data. These laws are then used to gain insight into the data or to classify or predict future data.
- The more we are able to compress a set of data the more redundancies, regularities, or patters we have found. The more we can describe the data with these regularities the more we have *learned* from the data.
- From this perspective we view learning as data compression such that for a given hypothesis *H,* and data *D,* we should try to find the hypothesis or combination of hypothesis in *H* that compress *D* the most.
- On the reverse side, we may consider quantifying the lack of regularities in a data set, that is what we might consider the randomness or complexity of data.
- One of the most well-known complexity measures is called ***Kolmogorov complexity***.
- The Kolmogorov complexity of some finite binary string *x* is the length of the shortest computer program that produces *x* on a universal Turing machine.
- Kolmogorov complexity quantifies how well a string can *in principle* be compressed and may be thought of as the "true" model.

Shannon's theorem

- The key to understanding the relation between probability and data compression and transmission is *Shannon's Theorem*, the central result in information theory formalized by the mathematician Claude Shannon.
- Claude Shannon, working on problems of digital communications at Bell Labs in 1948, had a very practical interest in developing a consistent measure of the "amount of information" in the outcome of a random variable transmitted across a communication channel.
- He wanted certain intuitive conditions to be satisfied in constructing a consistent measure of the amount of uncertainty associated with a random variable X using only the probabilities $p_i[x_i]$, $x_i \in X$.

Uncertainty

- The central idea of information theory is to measure the ***uncertainty*** associated with random variables. In general, a random variable X has a certain number of outcomes x_i which have probabilities p_i of occurring.
- The simplest setting in which to understand Shannon's theorem is a case where there is a finite set of n messages known to both the transmitter and receiver.
- For example Paul Revere alerting Colonial militia to the approach of British troops preceding the battles of Lexington and Concord. His problem was to communicate $n = 2$ possible messages "the British troops are traveling by water" and "the British troops are traveling by land."

- In order to transmit the message there could be agreement on a numbering of the two messages and transmission the corresponding number over the transmission channel (by lantern) is straightforward.

Minimizing the Transmission Rate

- In general, this method requires $\text{Log}_2\, n$ bits of information, since $n = 2^{\text{Log}_2\, n}$. The time it will take to send the message depends on how many bits of information the channel can transmit in a given time, such as a second.
- Shannon noted that if some messages were going to be sent more often than others, it would be possible to improve the *average* transmission rate by assigning short code words to the more frequently transmitted messages.

Average Transmission Time

- For example, messages that send only letters from the English alphabet $\mathcal{A} = \{a, b, \ldots., z\}$ could be coded in a way such that the letters "e" and "t" have a very high probability and thus low code length as in Morse code.
- If there is a frequency $p_i > 0$ assumed for each message i, with $\sum_i p_i = 1$, Shannon proved that it is possible to devise a coding in which the ith message is assigned a code word of $\left\lceil \text{Log}_2 \frac{1}{p_i} \right\rceil = \left\lceil -\text{Log}_2\, p_i \right\rceil$ bits.
- Furthermore Shannon showed that this is the best one can do to minimize the ***average transmission time*** as the set of messages becomes large, that is, as $n \to \infty$.
- This means that for every frequency distribution there corresponds a code length assignment, and, equally important, to every coding of messages there corresponds an implicit frequency distribution over the messages.

Ensembles

- A set of messages with specified frequencies is called an ***ensemble***.
- An ensemble is defined as $X = \{x, A_X, P_X\}$ where X is a triple containing the outcomes $x \in X$ from the set of possible values $A_X = \{a_1, a_2, \ldots, a_n\}$ with probabilities $P_X = \{p_1, p_2, \ldots, p_n\}$ where:

  $p[x = a_i] = p_i,\ p_i \geq 0\ \forall\ i,\ \text{and}\ \sum_{a_i \in \mathcal{A}} p[x = a_i] = 1$

- One example of an ensemble is a letter that is randomly selected from an English document. In this ensemble there are twenty-seven possible letters: a–z, and a space “-“.

p_i	a_i
0.0575	a
0.0128	b
0.0263	c
0.0285	d
0.0913	e
0.0173	f
0.0133	g
0.0313	h
0.0599	i
0.0006	j
0.0084	k
0.0335	l
0.0235	m
0.0596	n
0.0689	o
0.0192	p
0.0008	q
0.0508	r
0.0567	s
0.0706	t
0.0334	u
0.0069	v
0.0119	w
0.0073	x
0.0164	y
0.0007	z
0.1928	-

- The probability of a subset $T \subset \mathcal{A}$ is $p[T] = p[x \in T] = \sum_{a_i \in T} p[x = a_i]$ e.g. the vowels $V = \{a, e, i, o, u\}$ is:

  $P[V] = .06 + .09 + .06 + .07 + .03 = .31$

Coding Information

- Suppose we have a random variable X and we wish to efficiently communicate its values $x \in \mathcal{A}$ as they are determined. We can solve this problem practically by assigning a code $c[x]$ to each possible x and then transmitting the sequence of codes as they are ascertained.
- Since each code has a certain (bit)length $l[c[x]]$ the question we wish to answer is: How can the expected length of the transmission per random realization be minimized and what is this minimum expected length?
- Formally, how do we minimize average code length?

  $L = \sum_{x \in \mathcal{A}} l[c[x]]\, p[x]$

- Shannon's theorem says that if we want a shorter transmission of codes, choose $l[c[x]]$ smaller when $p[x]$ is larger.
- Take the example of the following alphabet: $\mathcal{A} = \{a, b, c, d\}$ and four possible ways of coding these messages where we choose short codes for the high probabilities and long codes for the low probabilities.

x	p[x]	code 1	code 2	code 3	code 4
a	0.5	0	0	10	0

b	0.25	0	010	00	10
c	0.125	0	01	11	110
d	0.125	0	10	110	111

```
Log[2, 2] * 0.5 + Log[2, 2] * .25 + Log[2, 2] * .125 + Log[2, 2] * .125
```

```
1.
```

- Code 1: has L = 1 bit and so is very short but is also not useful because it cannot be decoded at the other end. That is, all messages have the same code.

```
Log[2, 2] * 0.5 + Log[2, 2^3] * .25 + Log[2, 2^2] * .125 + Log[2, 2^2] * .125
```

```
1.75
```

- Code 2: has L = 1.75 bits and so is longer but it does have a unique code for each outcome/letter. However, consider the transmission 010. This could be either *b* or it could be *c a* so it cannot be ***uniquely decoded*** either.

```
Log[2, 2^2] * 0.5 + Log[2, 2^2] * .25 + Log[2, 2^2] * .125 + Log[2, 2^3] * .125
```

```
2.125
```

- Code 3: has L = 2.125 bits, longer still, but this time can be uniquely decoded. One must however wait to the end of the message before decoding so it is not ***instantaneously uniquely decodable***. Consider 11000 whose first four symbols could be *c b* but the last 0 tells us that instead the answer is *d b*.

```
Log[2, 2] * 0.5 + Log[2, 2^2] * .25 + Log[2, 2^3] * .125 + Log[2, 2^3] * .125
```

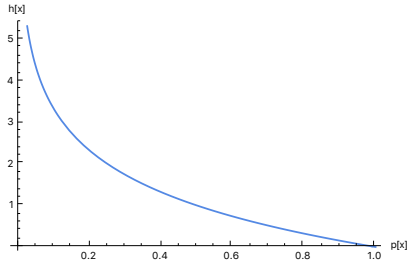
```
1.75
```

- Code 4: has L = 1.75 bits which is the same length as Code 2 but is now instantaneously uniquely decodable (no code is a ***prefix*** of another, whereas for code 3, *c* is a prefix of *d*). This last coding scheme is called a ***prefix free code***.

Consequences of Shannon's theorem

- With a Shannon coding the *information content* of an outcome $x \in X$ is

  $h[x] = \text{Log}_2\left[\frac{1}{p[x]}\right] = -\text{Log}_2[p[x]]$ which is measured in bits.



- The information content of each letter in the English alphabet $\mathcal{A}$ is:

p_i	a_i	$h[a_i]$
0.0575	a	4.12029
0.0128	b	6.28771
0.0263	c	5.24879
0.0285	d	5.13289
0.0913	e	3.45324
0.0173	f	5.85308
0.0133	g	6.23243
0.0313	h	4.99769
0.0599	9	4.0613
0.0006	j	10.7027
0.0084	k	6.89539
0.0335	l	4.8997
0.0235	m	5.4112
0.0596	n	4.06854
0.0689	o	3.85935
0.0192	p	5.70275
0.0008	q	10.2877
0.0508	r	4.29903
0.0567	s	4.14051
0.0706	t	3.82419
0.0334	u	4.90401
0.0069	v	7.17919
0.0119	w	6.39289
0.0073	x	7.09789
0.0164	y	5.93016
0.0007	z	10.4804
0.1928	-	2.37482

- The *average code length* is:

$$H[X] = \sum_{x \in \mathcal{A}} p[x] \, \text{Log}\Big[\frac{1}{p[x]}\Big] = \sum_i p_i \, \text{Log}\Big[\frac{1}{p_i}\Big] = -\sum_i p_i \, \text{Log}[p_i]$$

- $H[X]$ is called the ***informational entropy*** of the frequency distribution $p = \{p_1, \ldots, p_n\}$. Informational entropy is a measure of information that is to be understood as a measure of *uncertainty* or *ignorance*.
- That is, the probability assignment $p = \{p_1, \ldots, p_n\}$ describes a state of knowledge in the sense common to Bayesian reasoning.
- The average code length of the English alphabet $\mathcal{A}$ is $-\sum_i p_i \, \text{Log}[p_i] = 4.1$.
- If logarithms are taken to base 2, the code lengths will be expressed in *bits*. With logarithms to base 10 the code lengths are expressed in *digits* (1 digit = $\frac{1}{\log_{10}[2]}$ = 3.3219 bits), and with logarithms to the natural base e, the code lengths are expressed in *nats* (1 nat = $\frac{1}{\log_e[2]}$ = 1.4427 bits). Unless specified otherwise, Log[x] will implicitly be understood as $\text{Log}_2[x]$.

Information and Uncertainty

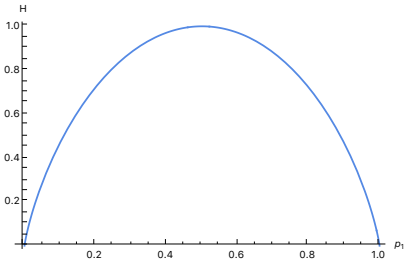
- If the n messages are sent equally often, the frequencies are uniform $p_i = \frac{1}{n}$, all the optimal code lengths are equal to $\lceil -\text{Log}[\frac{1}{n}] \rceil = \lceil \text{Log}[n] \rceil$, and the informational entropy of the ensemble is also $\lceil \text{Log}[n] \rceil$.
- Intuitively, if we know nothing about the probability of any particular message being received we might as well employ the “principle of insufficient reason” and assign equal probabilities to all possible messages.
- In this case $\lceil \text{Log}[n] \rceil$ is the average code length that results from simply numbering the messages and transmitting them by number such that no data compression is achieved.
- For example, if $\mathcal{A} = \{1, 2, \ldots, 6\}$ and we believe all possible messages are equally likely then the informational entropy is equal to:

$$H = -\sum p_i \, \text{Log}[p_i] = -\left(\frac{1}{6}\text{Log}\left[\frac{1}{6}\right] + \ldots + \frac{1}{6}\text{Log}\left[\frac{1}{6}\right]\right) = \text{Log}[6] = 2.58496 \text{ bits}$$

- Which is the average code length of the possible messages. Notice that the measure of information only requires specified probabilities, and not the realization of a message.
- As a sanity check, we can see that informational entropy intuitively measures uncertainty by noticing that when a message is certain, i.e. $p_i = 1$ and $p_j = 0 \; \forall \; j \neq i$, then $H = -\sum_i p_i \, \text{Log}[p_i] = -1 \, \text{Log}[1] = 0$. On the other hand, when all possible messages are equally likely we are in a state of maximum uncertainty and thus $H = \lceil \text{Log}[n] \rceil$ is at its maximum.
- As an example consider the case when there are only two outcomes, p_1 and $p_2 = 1 - p_1$. In this case the entropy function reduces to the *binary entropy function*:

$$H = -p_1 \, \text{Log}[p_1] - (1 - p_1) \, \text{Log}[1 - p_1]$$

- Which if we plot confirms our intuitive notion of entropy as $p_1 = 0.5 \rightarrow p_2 = 0.5$, i.e. the uniform distribution where both outcomes are equally likely giving us maximum informational entropy (uncertainty). At either $p_1 = 0$ or $p_1 = 1$ we have minimum entropy since one of the outcomes is certain.



- Entropy can be calculated for any probability distribution and represents the uncertainty of that distribution. For example, the entropy of the exponential distribution $f[x] = \lambda e^{-\lambda x}$ is:

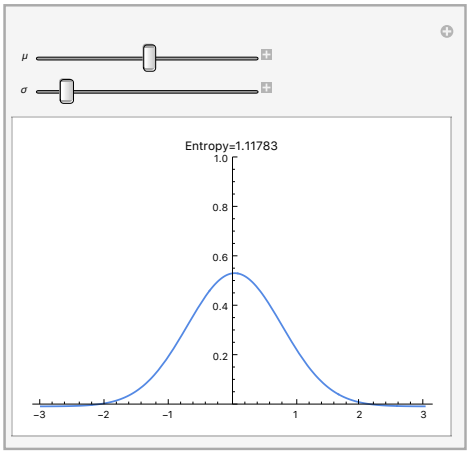
$$-\int f[x] \, \text{Log}[f[x]] \, dx = -\int \lambda e^{-\lambda x} \, \text{Log}\left[\lambda e^{-\lambda x}\right] dx$$

$$-\int_0^\infty \lambda e^{-\lambda x} \, \text{Log}\left[\lambda e^{-\lambda x}\right] dx$$

```
ConditionalExpression[1 - Log[λ], Re[λ] > 0]
```

- Equally we can show that the entropy of the Normal distribution is:

$$\frac{1}{2}\text{Log}[2\,\sigma^2\,\pi\,e]$$



Entropy Measures in Economics

- Informational entropy has been used in a variety of interesting ways in economics, including measuring inequality and segregation (Theil index), reconceptualizing bounded rationality (Sims' *Rational Inattention*), as a metric for determining the fit of a distribution (Soofi's *Information Distinguishability Statistics*), and as a prior in Bayesian reasoning (Jaynes' *Maximum Entropy Prior*).

The Minimum Description Length (MDL) and Two-part codes

- The basic idea behind the Minimum Description Length Principle (MDL) advocated by Jorma Rissanen is as mentioned above: The task of *inductive inference* is to find *laws* or *regularities* underlying some given set of fixed data. The more we are able to compress a set of data the more redundancies, regularities, or patters we have found. The more we can describe the data with these regularities the more we have *learned* from the data.
- The MDL is not a unique single method of inductive inference, but a general principle of inductive inference. The central idea behind the MDL principle is that any regularity in data may be used to compress that data and that *learning* is understood as finding these regularities in the data.
- From the MDL perspective, the goal of inductive inference is to "squeeze out as much regularity as possible" from the given data and to distill the meaningful information (signal) from the noise. As a corollary we interpret models as sets of hypothesis that are *languages* for describing useful properties of the data.
- In the MDL view the "noise" is only defined relative to the model or hypothesis as the residual number of bits needed to encode the data once the model is given. Therefore, noise is not a random variable as we are used to thinking, rather it is a function of the chosen model and the actually observed data.
- As Peter Grundwald describes it, there is no place for a "true distribution" or a "true state of nature" from this perspective, there are only models and data. If we consider the statement "these data are quite noisy," the conventional statistical interpretation is that the data were generated by a distribution with high variance.

- From the MDL perspective, this phrase means only that the data are not compressible with the currently hypothesized model. In principle we cannot rule out the possibility that there exists another model under which the data are compressible.

Mind Projection Fallacy

- Conventional econometrics can easily lead to the belief that "noise" or "randomness" is some kind of real property existing in Nature. Jaynes argued that this complex is form of the ***Mind Projection Fallacy*** which says, in effect, "I don't know the detailed causes therefore Nature does not know them."
- The Mind Projection Fallacy occurs in two complementary forms:
  - (1) My own imagination projected as a real property of nature.
  - (2) My own ignorance projected as indeterminacy of nature.
- The Mind Projection Fallacy is rampant in economics and frequentist econometrics and leads to a completely misguided perspective on the point of inference.

Two Part Codes

- The two part code version of the MDL that allows us to pursue inductive inference can be stated as follows.
- Let H_1, H_2, ... be a list of candidate models. The best hypothesis to explain the data D is the one which minimizes the sum $L[H] + L[D \mid H]$, where $L[H]$ is the bit length of the description of the hypothesis and $L[D \mid H]$ is the bit length of the description of the data when encoded with the help of the hypothesis.
- If we take logarithms on both sides of Bayes' Theorem, we see that:

  $\text{Log}[p[H \mid D]] \propto \text{Log}[p[H]] + \text{Log}[p[D \mid H]]$

- This suggest the following scheme for transmitting the data, or, equivalently, compressing it, or possibly equivalently, explaining it.
- $-\text{Log}[p[D \mid H]]$ is the length of the Shannon code for the data D based on the conditional probability of the data given the model represented by the hypothesis, and also a measure of the *fit*, since the more probable the data is conditional on the model, the better the model fits the data. If the receiver knew the model hypothesis, we could transmit the data through this code.
- Therefore one scheme for coding the data is first to transmit the hypothesis. The prior probability over the hypothesis space implicitly defines a Shannon coding that costs $-\text{Log}[p[H]]$ bits.
- In this case we send the data in the form of a coding of the residuals given the model, $p[D \mid H]$, from which the receiver can reconstruct the data given the model, which costs $-\text{Log}[p[D \mid H]]$ bits.
- The whole scheme forms a "two-part code" for the data, the first part being the transmission of the model as a hypothesis, and the second part being the transmission of the data in terms of its residual deviations from the prediction of the model.
- The maximum posterior probability hypothesis corresponds to the shortest two-part coding of the data given the set of hypotheses we are allowing as messages.

What does the two-part code length mean?

- The difference in bits between the two-part code lengths of two models for the same data is also the negative of the difference between their log posterior probabilities. Thus if one model has a two-part code length 1 bit greater than the other, the first model has half the posterior probability of the other given the data.
- Each bit of two-part code length thus represents a halving or doubling of the posterior probability of the models, and several bits of difference represent quite strong evidence for the superiority of the model with the shorter two-part code.

Overfitting

- The two-part code concept addresses the fundamental question of *overfitting*, which is a recurring problem in econometric analysis.
- One simple way to see the problem of overfitting is to observe that for a finite data set it is always possible to specify a model that fits the data exactly.
- If there are n pairs of data points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, it is possible to choose coefficients, $\beta_0, \beta_1, \ldots, \beta_n$ so that the polynomial passes through all of the data points:

  $$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n$$

- The problem is that without some systematic way of comparing the overall performance of models, it is sometimes very hard to tell when a researcher has “gone too far” in adding parameters to a model to improve its fit to the data.
- $-\text{Log}[P[H]]$ measures the number of bits it takes to transmit the hypothesis (model) H, while $-\text{Log}[P[D \mid H]]$ measures the goodness of fit. If we add a parameter to the model the modified model, H' is going to require more bits to transmit, $-\text{Log}[P[H']] > -\text{Log}[P[H]]$.
- We gain in explanatory power only if the new model reduces the total number of bits required to transmit or explain the data, that is, only if

  $$-\text{Log}[P[H']] - \text{Log}[P[D \mid H']] < -\text{Log}[P[H]] - \text{Log}[P[D \mid H]]$$

- But this is just the same as saying that the posterior probability of H' is higher than the posterior probability of H.

Simplicity vs fit

- The two-part code interpretation of Bayes’ Theorem reveals that the problem of overfitting is an aspect of the philosophical problem of the tradeoff between *simplicity* (sometimes called *elegance*) of theories and their *explanatory power* (their ability to *fit* the data).
- In the philosophy of science the preference for simpler explanations over more complex ones is called *Occam's Razor*, after the English philosopher William of Occam who enunciated it.
- The two-part code uses information theory to frame the tradeoff between simplicity and explanatory power in quantitative terms, namely the total information or number of bits required to transmit or explain the data.
- From the philosophical point of view, the problem with overfitting is that it is the self-defeating result of a one-sided preoccupation with explanatory power or fit.
- The self-defeat lies in the fact that in pursuit of fit the researcher is creating needlessly complex models that actually increase the total length of the code needed to transmit or explain the data.

- The practical difficulty is that without the framework of Bayes' Theorem, it is often difficult to keep track of the complexity of an explanatory model as it develops incrementally in an ongoing research project.

Problems of Overfitting

- In practical terms the drawback of overfitted models is that they perform badly in predicting out-of-sample observations.
- If we regard the data as composed of a *signal* representing the underlying replicable features of the phenomenon under investigation, and *noise* representing non-replicable features of the particular data set we are working with, overfitting incorporates some of the noise into the model.
- Recognizing this problem raises two awkward questions for the econometric literature.
- The first is that many published papers are instances of overfitting. This is the result of the convergence of several factors. Editors and referees have a well-known bias toward papers that claim positive results.
- As a result researchers have a strong conscious or unconscious motivation to look for models that fit the data rather than demonstrating that a class of models has really nothing much to say about the data at all.
- Second, looking at this from the point of view of the data, it may be that many economic data sets contain little evidence for or against the hypotheses that researchers hope to use them to prove or disprove.