Complexity Lab 1: Information Theory for Processes

[redacted]

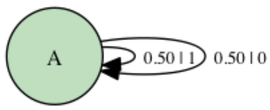
last edited on June 30, 2011 10:27 AM by autoplectic


```
%hide
```

Let us begin by looking at generators for five different processes. The processes we have available are *faircoin*, *biasedcoin*, *even*, *goldenmean*, and *periodic*. The periodic process repeats the word "11010" indefinitely. In the next five cells, create a picture of each one by running the "draw" method on each.

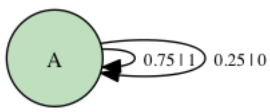
For example, here we draw the *faircoin* generator:

```
faircoin.draw()
```



Draw the *biasedcoin* generator:

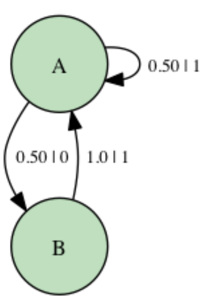
```
biasedcoin.draw()
```



How does the biased coin differ from the fair coin? Would you consider one more complex than the other?

Draw the *goldenmean* generator:

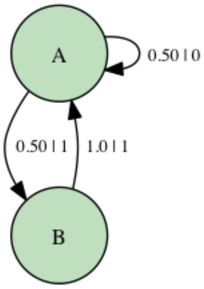
```
goldenmean.draw()
```



There is something new here. What is it? Would you consider the golden mean generator as more complex?

Next draw the *even* generator:

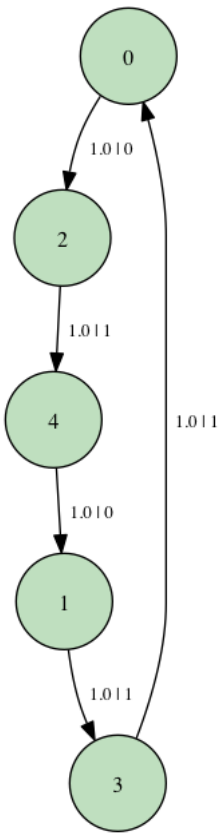
```
even.draw()
```



How are the even and the golden mean similar? How are they different?

Draw the *periodic* generator:

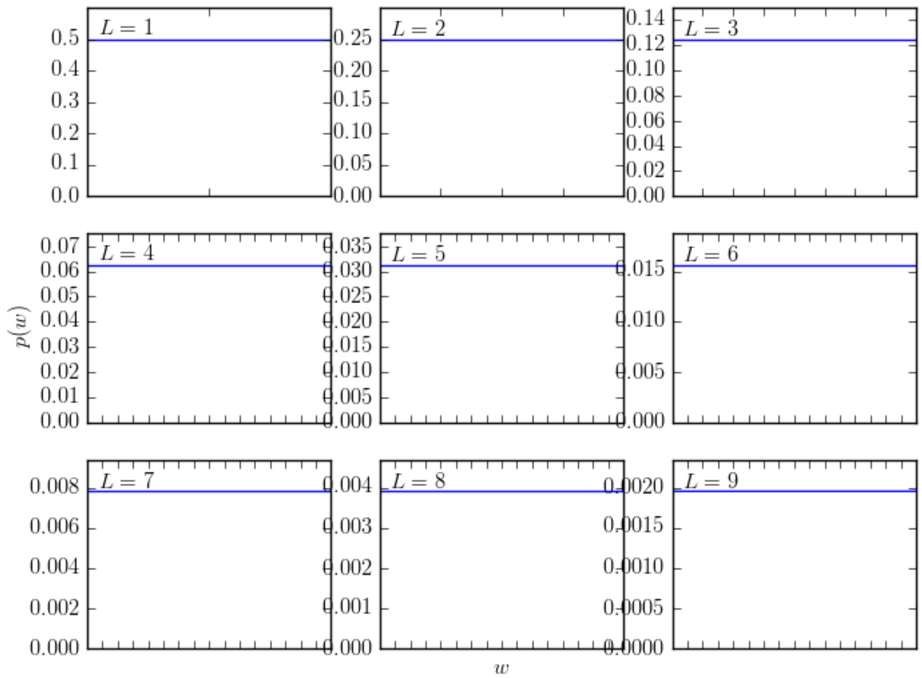
```
periodic.draw()
```



Do you notice anything different about the transitions in the periodic generator? How do they differ from the previous generators? What is the period of the process it generates?

One way of analyzing a process is by looking at its word distributions. Here we will plot the word distributions up to words of length 9 for each of the processes we looked at previously. Plotting the word distributions is done using the *word_distribution_grid* function. For example, here is how to plot the word distributions of the fair coin process:

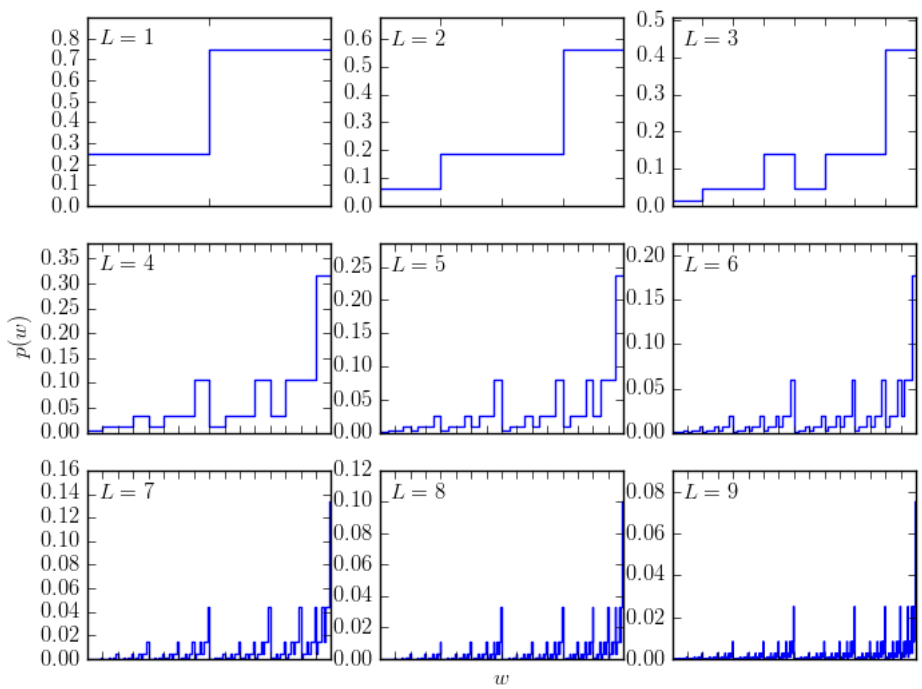
```
word_distribution_grid(faircoin)
```



Horizontally we see each word as a bin expansion (e.g. binary 101 is decimal 5, so it maps to 5/8 = 0.625). What do you notice about the fair coin word distributions?

Next plot the biased coin word distributions:

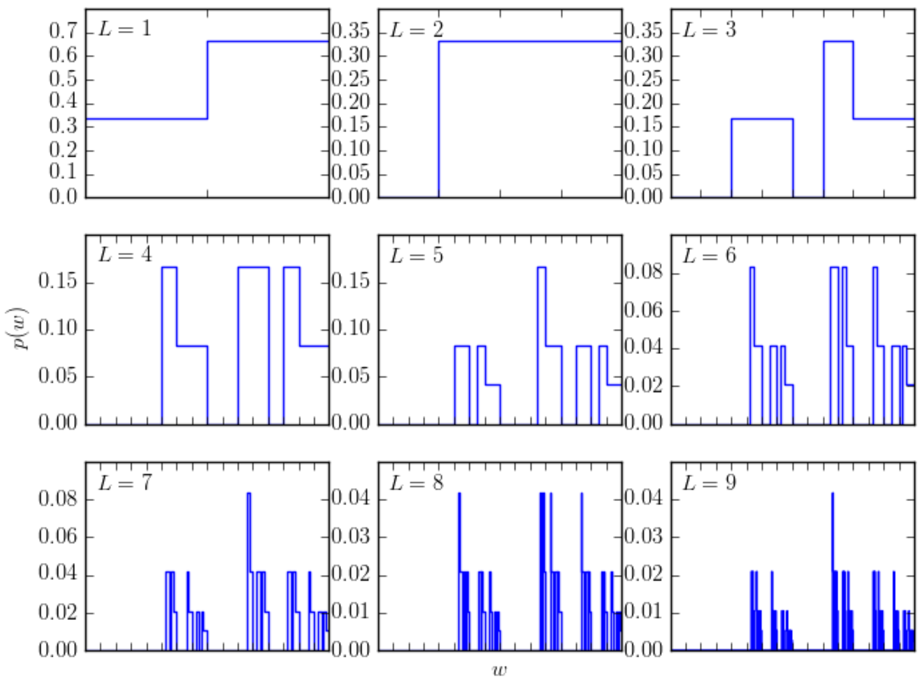
```
word_distribution_grid(biasedcoin)
```



How is this similar to the fair coin distributions? How is this different from those of the fair coin?

Plot the golden mean process word distributions:

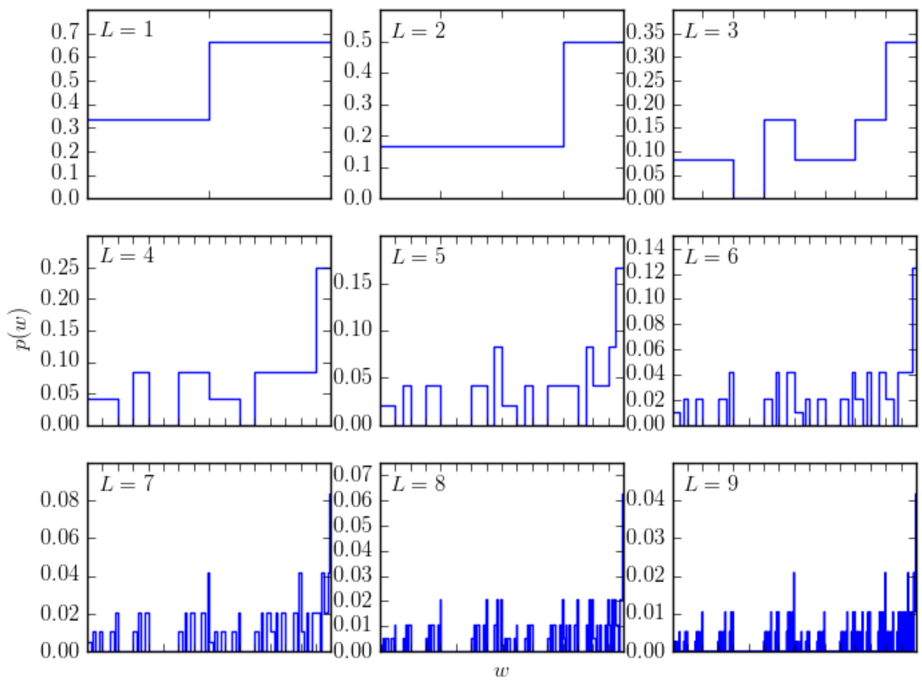
```
word_distribution_grid(goldenmean)
```



What do you notice happening at L=2? Which word is forbidden? Look at the forbidden words at L=3, notice that they contain the forbidden word '00'.

Plot the even process word distributions:

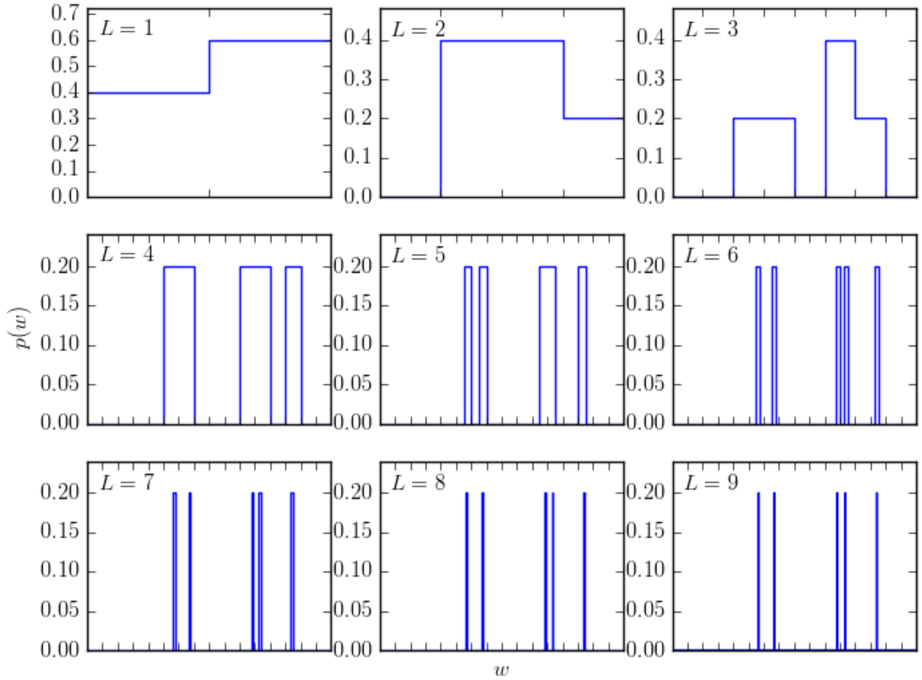
```
word_distribution_grid(even)
```



What is the forbidden word at L=3? List out the forbidden words at L=4, notice that they contain the forbidden word at L=3, w='010'. At L=5 there's a new forbidden word: '01110'. Can you seen this for yourself looking at the generator earlier in the lab?

Now draw the word distributions for the periodic process:

```
word_distribution_grid(periodic)
```

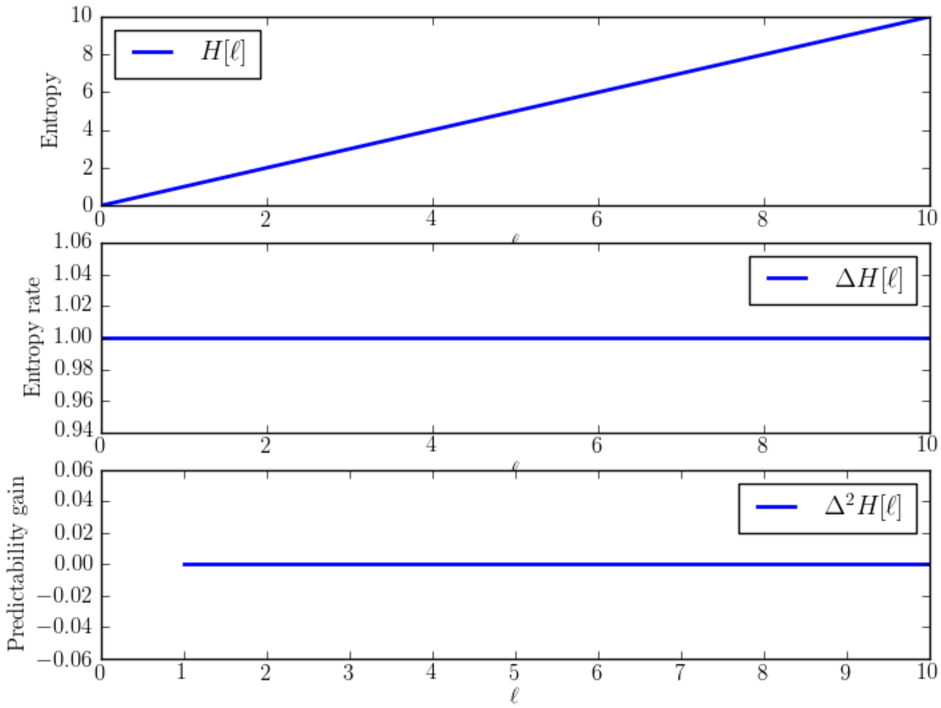


Notice at L=4 that there are five words possible. This is true for all larger L as well.

Have you noticed that it gets difficult to analyze the word distribution plots at large L? Instead of using the word distributions, we can summarize some of their structure by looking at the entropy of the word distribution with increasing length.

First let's look at the fair coin:

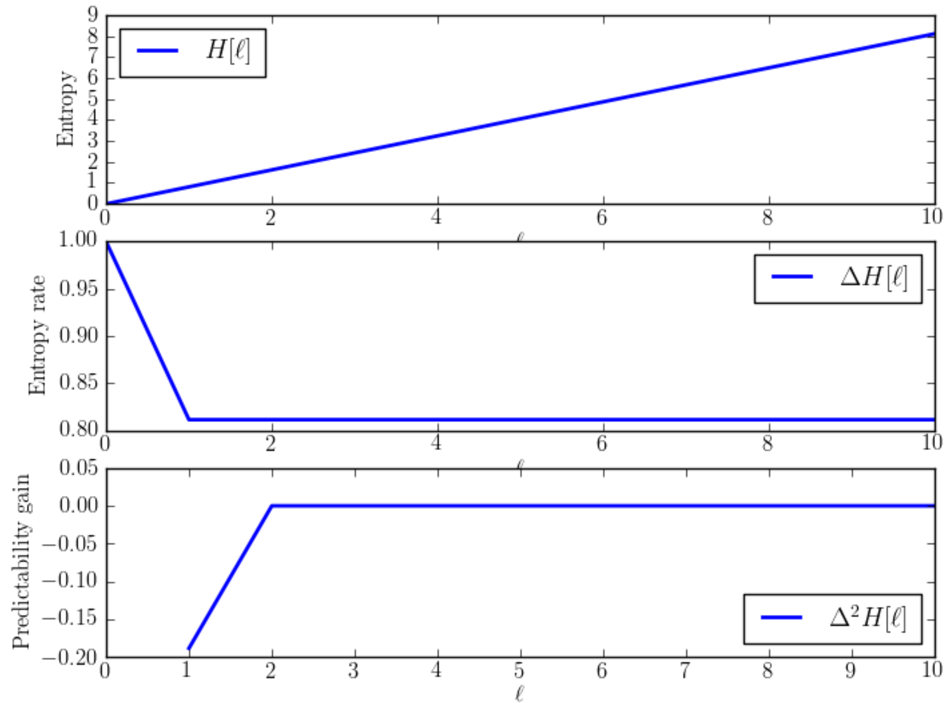
```
block_entropy(faircoin)
```



Notice that there are three plots: the block entropy, the derivative of the block entropy, and the second derivative of the block entropy. For the fair coin, these are all pretty simple -- the entropy simply grows linearly with L at a slope of 1. As a consequence, the slope is constant (value of 1) and therefor we have a constant entropy rate in the second plot.

Now let's do the same for the biased coin:

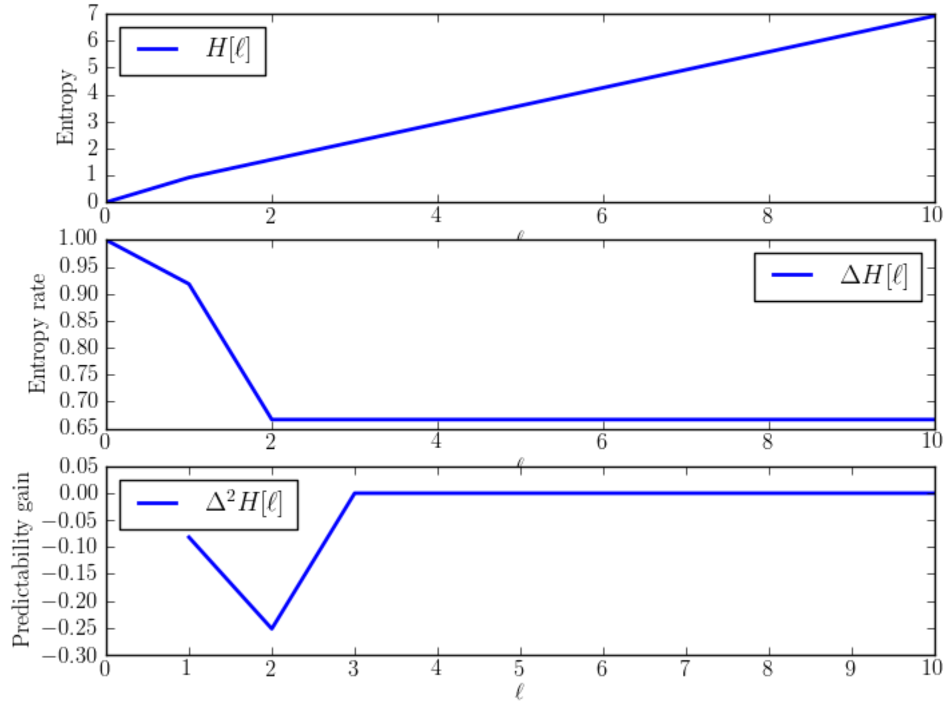
```
block_entropy(biasedcoin)
```



Notice that here also the block entropy curve has a constant slope. This time less than 1, which can be seen in the entropy rate plot as having a value slightly above 0.8. Notice further that the entropy rate plot doesn't have its asymptotic value until L=1.

Next plot the golden mean process:

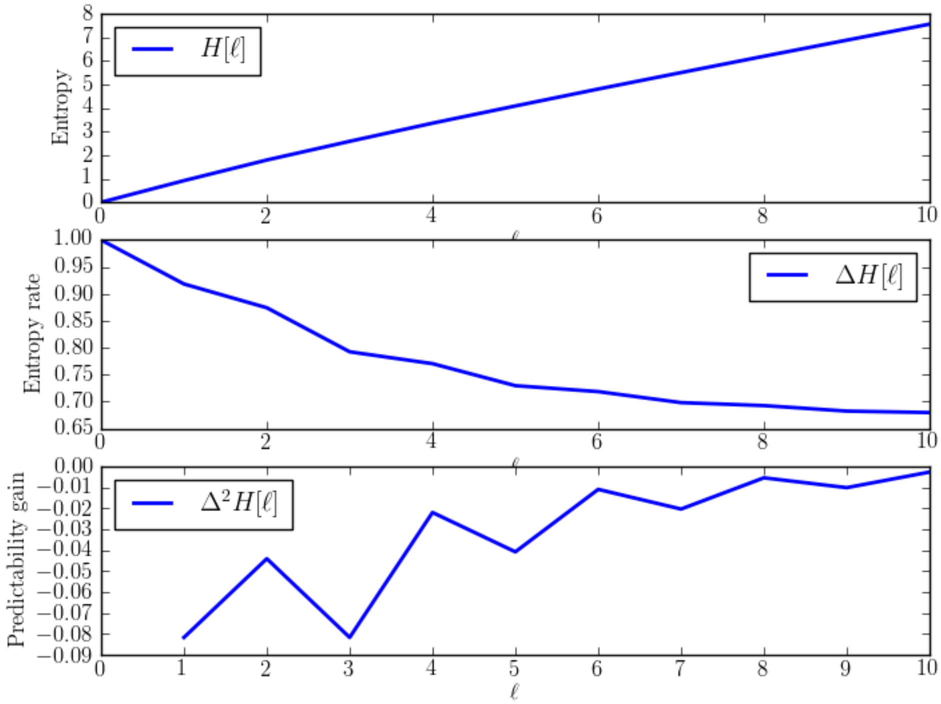
```
block_entropy(goldenmean)
```



Here we can see that the block entropy curve is no longer constant, it has a knee at L=1. This is also reflected in the entropy rate plot where the convergence doesn't happen until L=2. This is consistent with the structure of the golden mean process being based around the forbidden word '00'. In the predictability gain curve, we notice that the most informative measurement is at L=2, which is where we first find the forbidden word and as a consequence that measurement provides the largest increase in one's ability to predict.

Next let us look at the even process:

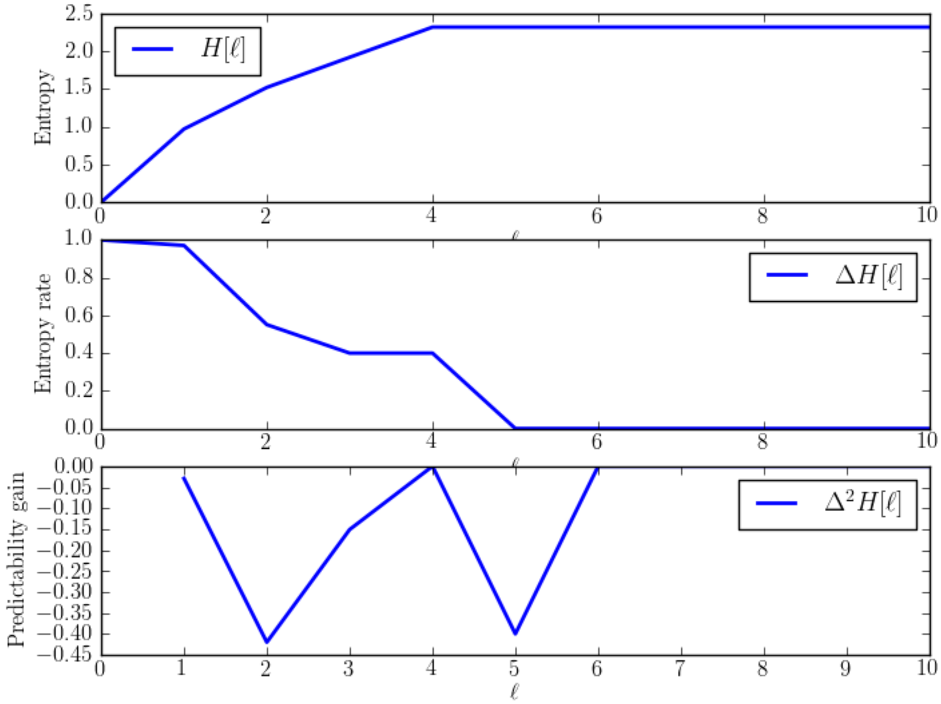
```
block_entropy(even)
```



We notice immediately that the even process, though having a generator very similar to that of the golden mean, has strikingly different behavior in the block entropy curve and its derivatives. One thing to notice is that the entropy rate curve converges slowly, and does not actually reach its asymptotic value of 0.66667 bits per symbol in a finite number of observations. In the predictability gain curve, we also notice that it is the odd length words which provide the biggest boon to prediction. This is due to observing the forbidden words '010', '01110', '0111110', ...

Lastly let us look at the periodic process:

```
block_entropy(periodic)
```



The first thing to notice is that the block entropy curve reaches its maximal value ($\log_2(5)$) and from that point forward is flat (zero slope). This is reflected on the entropy rate plot by seeing that the entropy rate is 0 at L=5 and larger. Can you explain the informative words at L=2 and L=5?

evaluate