Information-Theoretic Modeling, Fall 2009

Exercises V, due Friday 16 October.

1. Consider binary sequences $x^{15} = (x_1, x_2, \ldots, x_{15}) \in \{0,1\}^{15}$ of length $n = 15$. Let $\mathcal{M} = \{p_\theta \; ; \; \theta \in [0,1]\}$ be a model class consisting of i.i.d. Bernoulli distributions — hence, the probability of sequence x^{15} is given by $\theta^{\sum x_i}(1-\theta)^{n-\sum x_i}$, where $\sum x_i$ and $n - \sum x_i$ denote the number of 1's and 0's in x^{15}, respectively.

We quantize the parameter space $\Theta = [0,1]$ by choosing 11 points at even intervals, letting the possible quantized parameters be $\theta^q \in \Theta^q = \{0.0, 0.1, 0.2, \ldots, 1.0\}$.

(a) What is the two-part code-length (ignoring the integer requirement) for data sequence $x^{15} = 001000100000001$? Since we are not using the optimal quantization, we need to evaluate the two-part code-length as

$$\min_{\theta^q \in \Theta^q} \left[\log_2 \frac{1}{p_{\theta^q}(D)} + \ell(\theta^q) \right].$$

Use the uniform code for θ^q which implies $\ell(\theta^q) = \log_2 11$ for all $\theta^q \in \Theta^q$.

(b) Compute the mixture code-length,

$$\log_2 \frac{1}{\sum_{\theta^q \in \Theta^q} p_{\theta^q}(x^{15})\, w(\theta^q)},$$

with the uniform prior $w(\theta^q) = \frac{1}{11}$ for all $\theta^q \in \Theta^q$.

Compare these code-lengths. *Optional:* Does the order of the code-lengths depend on the actual sequence x^{15}?

2. Continuation of the first exercise: Compute the normalized maximum likelihood code-length,

$$\log_2 \frac{1}{p_{\hat\theta}(x^{15})/C}, \qquad \text{where } C = \sum_{y^{15} \in \{0,1\}^{15}} p_{\hat\theta}(y^{15}),$$

where the sum is over all the possible 15 bit sequences. Note that each term $p_{\hat\theta}(y^{15})$ in the sum involves the parameters maximizing the probability of sequence y^{15}. The maximizing θ for y^{15} is given by $\hat\theta = \frac{\sum y_i}{n}$. By these observations, we obtain

$$p_{\hat\theta}(y^{15}) = \left(\frac{\sum y_i}{n}\right)^{\sum y_i} \left(1 - \frac{\sum y_i}{n}\right)^{n - \sum y_i}$$

Optional: Can you figure out a way to compute the sum faster than by enumerating all the 2^{15} possible binary sequences?

3. (2 points) Curve fitting. For this exercise you need to use `gnuplot` or some other tool that allows you to fit parametric functions to data. We analyse a sequence of observations given in the `polydata.txt` file, available at the course web-page[1].

First, let's take a look at the data (Fig. 1):

```
gnuplot> plot "polydata.txt"
```

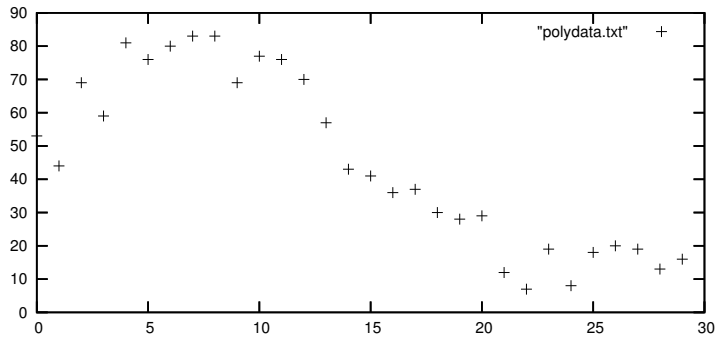

Figure 1: `polydata.txt`

Now, let's fit a linear function $f(x) = a + bx$ to the data using `gnuplot`'s `fit` procedure:

```
gnuplot> func1(x)=a+b*x
gnuplot> fit func1(x) "polydata.txt" via a,b
```

We get a long output giving us all sorts of statistics, including the "final sum of squares of residuals: 5978.96". The fitted curve (Fig. 2) can be plotted by:

```
gnuplot> plot "polydata.txt", func1(x)
```

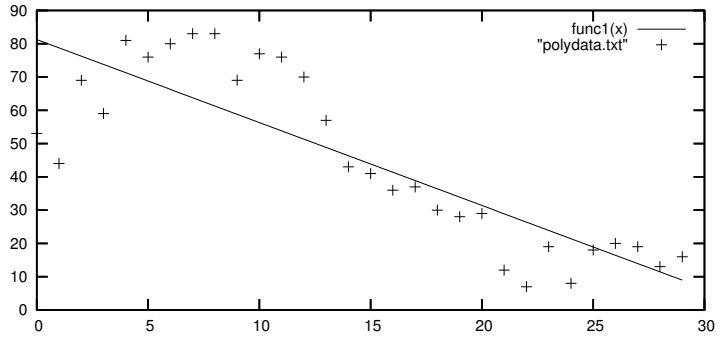

Figure 2: A linear function $f(x) = a + bx$ fitted to the data.

We can also fit, say, a quadratic function, i.e., a second order polynomial:

```
gnuplot> func2(x)=a+b*x+c*x*x
gnuplot> fit func2(x) "polydata.txt" via a,b,c
```

[1] The sequence of values is: 53, 44, 69, 59, 81, 76, 80, 83, 83, 69, 77, 76, 70, 57, 43, 41, 36, 37, 30, 28, 29, 12, 7, 19, 8, 18, 20, 19, 13, 16.

In this case the residual sum of squares is 5195.46, which means that the fit was better. In fact we need not restrict to polynomials: we can fit any function that we can write in gnuplot, including exponentials, logarithms, trigonometric functions, etc.

If we want to encode the data using a this kind of a model, we need to encode

1. the coefficients: we use the asymptotic formula $\frac{k}{2}\log_2 n$ as the code-length for this part,
2. the data: we use a Gaussian distribution.

In the Gaussian density fitted to the data, the mean is given by the fitted curve and the variance is given by the residual sum of squares divided by the sample size: $\hat{\sigma}^2 = \mathrm{RSS}/n$. For instance, in the linear case, the variance is given by $\hat{\sigma}^2 = 5978.96/30 \approx 199.3$.

The fact that the Gaussian distribution is defined as a density, not a probability mass function, is actually of no concern — this will be explained on Friday's lecture. The code-length of the second part becomes then

$$\log_2 \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} e^{-\frac{(f(x)-y)^2}{2\hat{\sigma}^2}} \right)^{-1},$$

where $f(x)$ is the fitted function. This can be re-written as

$$\frac{n}{2}\log_2(2\pi\hat{\sigma}^2) + \sum_{i=1}^{n} \frac{(f(x)-y)^2}{(2\ln 2)\hat{\sigma}^2},$$

where the sum of squared residuals and the ML estimate of the variance $\hat{\sigma}^2$ cancel each other, and the second term becomes a constant (see Lecture 10). We can thus write the code-length as

$$\frac{n}{2}\log_2 \mathrm{RSS} + \text{constant},$$

where constant doesn't depend on the data or the function we are fitting, and can be ignored.

The total code-length which gives the final MDL criterion is therefore

$$\boxed{\tfrac{n}{2}\log_2 \mathrm{RSS} + \tfrac{k}{2}\log_2 n,}$$

where k is given by the number of coefficients in the model *plus one* for the variance parameter.

To give an example, in the case of the linear model, the value of the criterion is given by $\frac{30}{2}\log_2 5978.96 + \frac{3}{2}\log_2 30 \approx 195.5$. For the quadratic function, we get $\frac{30}{2}\log_2 5195.46 + \frac{4}{2}\log_2 30 \approx 195.0$. The latter is smaller, so we prefer the quadratic function.

Your task is to find a function for which the MDL criterion gives as small a value as possible.

Bonus exercise. Generate your own data from some parametric function $f(x)$, and see if that function is identified correctly by the MDL criterion. Try different sample sizes.