Last lecture I introduced linear predictive coding. Continuing on...

Linear predictive coding (continued)

Where do the coefficients a_m come from ? The encoder has a sequence of length n and wishes to choose good a_m's for that sequence. One way to choose good a_m's is to make the difference between the true x_{j+1} and predicted x_{j+1} as small as possible, for example, minimize the sum of absolute differences between the true values and the predicted values:

$$\sum_{j=m}^{n-1} (x_{j+1} - \sum_{m=0}^{k-1} a_m \hat{x}_{j-m})^2 \quad . \tag{1}$$

Finding such a_m's is highly non-trivial, however. The $\hat{x}_{j-m}$'s themselves depend on the a_i's.

A simpler method, which can be explained in a few lines, is to find a_m's that minimize

$$\sum_{j=m}^{n-1} (x_{j+1} - \sum_{m=0}^{k-1} a_m x_{j-m})^2 \quad . \tag{2}$$

Here the $\hat{x}$ terms of Eq. (1) have been replaced by x terms. The intuition for why this is reasonable is that if the a_m's give good predictions of x_{j+1}, then $\hat{x}_{j+1} \approx x_{j+1}$ and so the a_m's that minimize Eq. (1) should be similar to the a_m's that minimize Eq. (2).

We therefore solve for these a_m's by minimizing Eq. (2). We do so by taking the partial derivative of the sum of squares with respect to each a_i and setting the partial derivative to zero. It is easy to see that this indeed gives a minimum for each a_i, since Eq. (2) is a quadratic function of each a_i and goes to $+\infty$ when $a_i \to \pm\infty$. Taking the partial derivative with respect to a particular a_i gives:

$$\sum_{j=k}^{n-1} (x_{j+1} - \sum_{m=0}^{k-1} a_m x_{j-m}) x_{j-i} = 0$$

where the variable i corresponds to the i^{th} partial derivative. That is, for each $i = 0, \ldots, k-1$, we get an equation:

$$\sum_{j=k}^{n-1} x_{j+1} x_{j-i} = \sum_{m=0}^{k-1} a_m \sum_{j=k}^{n-1} x_{j-m} x_{j-i} \tag{3}$$

So we get k linear equations in k unknowns (the a_m's)

$$\begin{bmatrix} & \\ & \\ & \end{bmatrix} = \begin{bmatrix} & \\ & \\ & \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ : \\ a_{k-1} \end{bmatrix}$$

We can solve for a_m using basic linear algebra. Typically k will be very small (2 or 3).

Autocorrelation

This least squares minimization requires computing $k^2 + k$ summations, each of which involved approximately n terms. That is, we sum roughly from $j = k$ to $n - 1$ where k is very small and n is very large. There were two sets of sums. For the sum of form

$$\sum_{j=k}^{n-1} x_{j+1}\ x_{j-i}\ ,$$

the samples in each product are separated by a distance $i + 1$. For the sums of form

$$\sum_{j=k}^{n-1} x_{j-k}\ x_{j-i}\ ,$$

the samples are separated by a distance $i - m$. Recall that both $i, m \in \{0, 1, \ldots, k-1\}$.

Consider the expected value of the product of two random variables separated by distance i:

$$R(d) \equiv \mathcal{E}\{\ X_j\ X_{j-d}\ \}.$$

This expected value is called the *autocorrelation function*. If the random variables $X_1, X_2, \ldots, X_n$ are stationary, then the autocorrelation function depends only on separation distance i.

If we divide each of the sums above by the number of terms $n-m$, then we obtain an estimate of the autocorrelation function, for various values of the separation distance between samples. These estimates are approximately the same as the following function:

$$\hat{R}(i) \equiv \frac{1}{n-k} \sum_{j=k+1}^{n} x_j\ x_{j-i} \tag{4}$$

where $i = 0, \ldots, k-1$ is the separation distance. The only difference is that the end points of the indices may be slightly shifted. But if $n \gg k$ then this will have little effect.

On the previous page, we estimated the a_m's by solving a linear system of equations. Now we take Eq. (3) and approximate the sums by the estimate of the autocorrelation function i.e. Eq. (4). This gives:

$$\hat{R}(i+1)\ \approx\ \sum_{m=0}^{k-1} a_m\ \hat{R}(\ |\ i-m\ |\ ), \qquad i = 0 \text{ to } k-1$$

where we are ignoring the factor $n - k$ on each side of the equation, since this factor cancels. The approximation is very good when n is large and k is small.

Take the case $m = 3$, the a's are estimated by solving the system of equations:

$$\begin{bmatrix} \hat{R}(1) \\ \hat{R}(2) \\ \hat{R}(3) \end{bmatrix} = \begin{bmatrix} \hat{R}(0) & \hat{R}(1) & \hat{R}(2) \\ \hat{R}(1) & \hat{R}(0) & \hat{R}(1) \\ \hat{R}(2) & \hat{R}(1) & \hat{R}(0) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

For more general m, the pattern is the same. We have $\hat{R}(0)$ on the diagonal, $\hat{R}(1)$ on the first off diagonal, $\hat{R}(2)$ on the second off diagonal, etc. These are called the *Yule-Walker* equations.

We will have more to say about autocorrelation in the coming lectures.