EE5585 Data Compression April 4, 2013

Lecture 19

Instructor: Arya Mazumdar *Scribe: Katie Moenkhaus*

Kolmogorov Complexity

A major result is that the Kolmogorov complexity of a random sequence on average is close to the entropy. This captures the notion of compressibility. Also, an algorithmically incompressible binary sequence as defined by the Kolmogorov complexity has approximately the same numer of 1s and 0s. The difference between the number of 0s and 1s gives insight into how compressible a signal is. Kolmogrov complexity can be defined for numbers as well as sequences. For $n \in \mathbb{Z}$,

$$K(n) = \min_{p:U(p)=n} l(p)$$

where $K(n)$ denotes the Kolmogorov complexity of n, and $l(p)$ is the minimum length of the computer program. Some integers have low Kolmogorov complexity. For exmple, let $n = 5^{5^{5^5}}$. Even though this is a very large number, it has a short description. Also, e is easily described. Even though it is an irrational numer, it has a very short description of the basis of the natural logarithm, so it is actually a function such that the derivative of the function is itself.

Note,

$$K(n) \leq \log^* n + c$$

where $\log^* n = \log n + \log\log n + \log\log\log n + \ldots$, so long as the each term is positive.

Theorem 1 *There exists an infinite number of integers for which $K(n) > \log n$.*

Proof (by contradiction): Assume there exist a finite number of integers for which $K(n) > \log n$. From Kraft's Inequality,

$$\sum_n 2^{-K(n)} \leq 1$$

If the assumption is true, then there will be only a finite number of integers for which $K(n) > \log n$. Then there is a number n_0 for which all $n \geq n_0 \implies K(n) \leq \log n$. Then

$$\sum_{n \geq n_0} 2^{-K(n)} \geq \sum_{n \geq n_0} 2^{-\log n} = \sum_{n \geq n_o} \frac{1}{n}$$

The series $\sum_n \frac{1}{n}$ doesn't converge, so

$$\sum_{n \geq n_o} \frac{1}{n} = \infty$$

It follows that

$$\sum_n 2^{-K(n)} = \infty$$

This contradicts with $\sum_n 2^{-K(n)} \leq 1$, and thus the assumption is false. $\therefore \exists$ a finite number of integers for which $K(n) > \log n$. ■

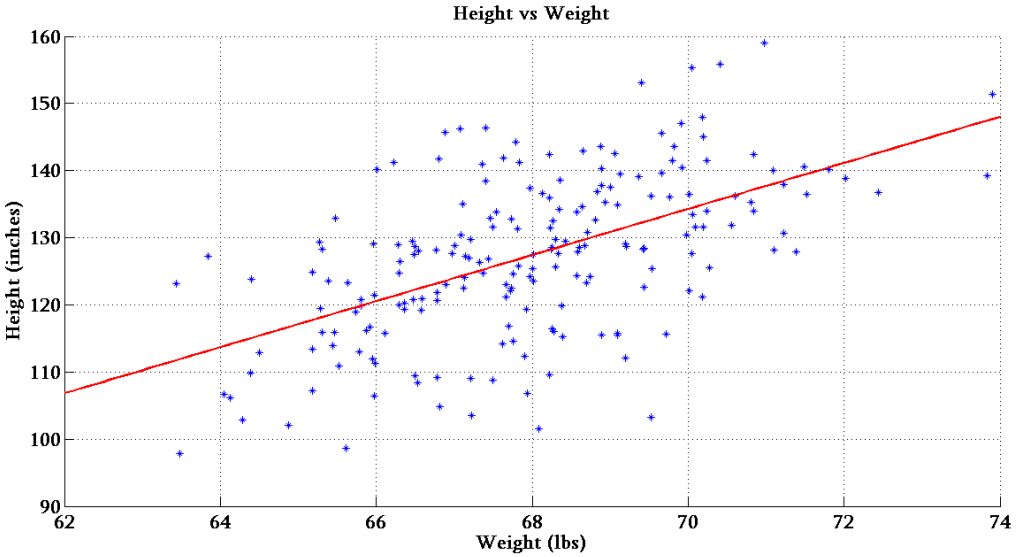
This illustrates that there is an infinite number of integers that are not simple to describe, i.e. they will take more than $\log n$ bits to describe. Given a binary sequency, if

$$|\#1s - \#0s| \geq \epsilon_n$$

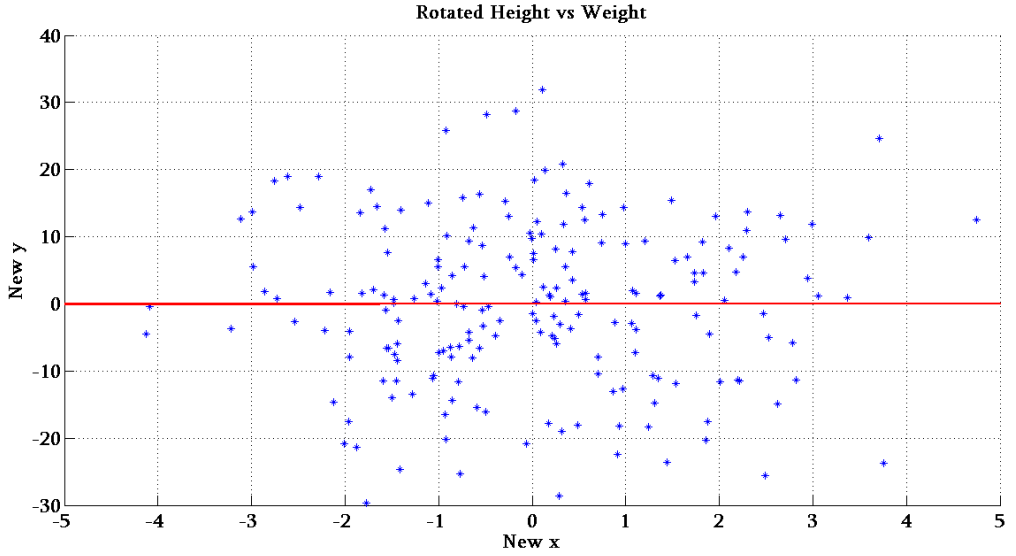
then the sequence can be compressed.

Transform Coding

Consider a height/weight data set.



Because the data is highly correlated, a clockwise rotation can be done to obtain new axes such that the line of best fit becomes the new prominent axis.



Thus the difference between any data point and the axis is small, and fewer bits are needed to describe them. In the new data set, the entries are uncorrelated. Ideally, there will be 0 correlation in the new coordinates. Note that the correlation between random variables X and Y is $E[(X - E(X))\,(Y - E(Y))]$, where $E(X)$ denotes the expected value of X. If X and Y are independent, then $Cor = 0$. Similarly, if the correlation is large, the variables are highly correlated. Without loss of generality, $E(X) = E(Y) = 0$.

This is just subtracting the mean from each set. Let

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}; \quad E(X) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

In the height/weight data, there were only two variables, i.e. height is x_1 and weight is x_2. The rotation is done by multiplying by a matrix:

$$Y = AX$$

Y is another N-dimensional vector, and A is $N \times N$. It is desirable to have A be an orthonormal matrix, i.e. $A^T A = I$. If this holds, then the result is an orthogonal transformation. For any orthogonal transformation, Parseval's Theorem is true. This says that

$$\begin{aligned} \sum_i \sigma_{y_i}^2 &= \sum_i E(y_i^2) \\ &= E(\|Y\|_2^2) \\ &= E(Y^T Y) \\ &= E(X^T A^T A X) \\ &= E(X^T X) \\ &= E(\|X\|_2^2) \\ &= \sum_i E(x_i^2) = \sum_i \sigma_{x_i}^2 \end{aligned}$$

If σ_x is the variance of x, then $\sum_i E(x_i^2) = \sum_i \sigma_{x_i}^2$ stems from the fact that X is a zero-mean, random variable. Thus, the transformation conserves the total energy. The transformation should ensure that the new data are uncorrelated. The correlation is representated by the covariance matrix, C_x, defined as

$$\begin{aligned} C_x = E(XX^T) &= E \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix} \\ &= E \begin{bmatrix} x_1^2 & x_1 x_2 & \dots & x_1 x_N \\ x_1 x_2 & x_2^2 & \dots & x_2 x_N \\ \vdots & \vdots & \ddots & \vdots \\ x_1 x_N & x_2 x_N & \dots & x_N^2 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{x_1}^2 & E(x_1 x_2) & \dots & E(x_1 x_N) \\ E(x_1 x_2) & \sigma_{x_2}^2 & \dots & E(x_2 x_N) \\ \vdots & \vdots & \ddots & \vdots \\ E(x_1 x_N) & E(x_2 x_N) & \dots & \sigma_{x_N}^2 \end{bmatrix} \end{aligned}$$

Note that this is a symmetric matrix. From the definition of the trace of a matrix,

$$Trace(C_x) = \sum_i \sigma_{x_i}^2$$

which is the total energy of the signal. The transform matrix A should minimize the off-diagonal elements of the covariance matrix.

$$\begin{aligned} C_y &= E(YY^T) \\ &= E(AXX^TA^T) \\ &= AE(XX^T)A^T \\ &= AC_xA^T \end{aligned}$$

Because $A^TA = I$ and A is an orthogonal matrix, $A^T = A^{-1}$. Thus,

$$\begin{aligned} C_y &= AC_xA^{-1} \\ C_yA &= AC_x \end{aligned}$$

The ideal covariance matrix C_y is a diagonal matrix. Suppose

$$C_y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix}$$

Thus

$$\begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix} A = AC_x$$

Because X is zero mean, Y is also zero mean. Suppose

$$\begin{aligned} A &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \\ \implies C_yA &= \begin{bmatrix} \lambda_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{N1} \end{bmatrix} & \lambda_2 \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{N2} \end{bmatrix} & \dots & \lambda_N \begin{bmatrix} a_{1N} \\ a_{2N} \\ \vdots \\ a_{NN} \end{bmatrix} \end{bmatrix} \\ &= \begin{bmatrix} C_x \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{N1} \end{bmatrix} & C_x \begin{bmatrix} a_{12} \\ a_{22} \\ \vdots \\ a_{N2} \end{bmatrix} & \dots & C_x \begin{bmatrix} a_{1N} \\ a_{2N} \\ \vdots \\ a_{NN} \end{bmatrix} \end{bmatrix} \end{aligned}$$

For ease of notation, let

$$A_i = \begin{bmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{Ni} \end{bmatrix}$$

In general,

$$C_x A_i = \lambda_i A_i$$

λ_i is an eigenvalue of C_x, and A_i is the corresponding eigenvector.

$$A = \begin{bmatrix} A_1 & A2 & \dots & A_N \end{bmatrix}$$

Thus, the transform matrix A is

$$A = \begin{bmatrix} eigenvectors\ \ of\ \ C_x \end{bmatrix}$$

The eigenvalues are simply $\lambda_i = \sigma_{y_i}^2$. This is called the **Karhunen-Loève Transform** (KLT). The process for finding the KLT is

1. Subtract off the mean:
   $x_1 \to x_1 - E(x_1)$
   $x_2 \to x_2 - E(x_2)$
   $\vdots$
   $x_N \to x_N - E(x_N)$

2. Find the covariance matrix. If there are M elements in each vector, then the total energy is

   $$\sigma_{x_n}^2 = \frac{1}{M} \sum_{i=1}^{M} x_{ni}^2$$

   for each n going from 1 to N. The correlations are then

   $$\rho_{x_n, x_m} = \frac{1}{M} \sum_{i=1}^{M} x_n x_m$$

   The covariance matrix is then

   $$C_x = \begin{bmatrix} \sigma_{x_1}^2 & \rho_{x_1,x_2} & \cdots & \rho_{x_1,x_N} \\ \rho_{x_1,x_2} & \sigma_{x_2}^2 & \cdots & \rho_{x_2,x_N} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{x_1,x_N} & \rho_{x_2,x_N} & \cdots & \sigma_{x_N}^2 \end{bmatrix}$$

3. Knowing the covariance matrix C_x, find the eigenvalues and eigenvectors. Form the transform matrix A as

   $$A = \begin{bmatrix} A_1 & A2 & \dots & A_N \end{bmatrix}$$

   where A_i represents the i^{th} eigenvector of C_x.

The process outlined above is called **Principal Component Analysis**. The goal of this analysis is to find a matrix that transforms the vectors to a new coordinate system in which they are uncorrelated. So ideally,

$$C_y = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_{N-1} \\ & & & & \lambda_N \end{bmatrix}$$

Keep only the eigenvalues that are largest. The energy lost in this process is the sum of the eigenvalues that are not used in the compression scheme. This is the mean square error. Suppose that the eigenvalues are ordered, i.e. $\lambda_1 > \lambda_2 > \cdots > \lambda_N$. If all $\{\lambda_i\}_{i=N_0}^{N}$ are unused, then the mean square error is

$$MSE = \sum_{i=N_0}^{N} \lambda_i$$

This is one measure of the distortion induced by compressing the signal. KLT optimally minimizes $E(y_i y_j)$, but it is computationally inefficient because for each new data set, a new transform matrix has to be computed. Instead, some standard transform matrices are used. One such of these is

$$F = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \dots & \omega^N \\ 1 & \omega^2 & \omega^4 & \omega^6 & \dots & \omega^{2N} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \dots & \omega^{3N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix}$$

This is an example of a generic transform matrix. Typically, $\omega = e^{-i\frac{2\pi}{N}}$. This is the **Discrete Fourier Transform** (DFT) matrix. The DFT projects data along the rows, and each row has a frequency kernel. Thus, it produces the frequency components of the data.

$$\begin{aligned} Y &= FX \\ &= \begin{bmatrix} Y1 \\ Y2 \\ \vdots \\ Y_N \end{bmatrix} \end{aligned}$$

The values $\{Y_i\}_{i=1}^{N}$ are the frequency components. Compressing a signal by omitting any of the low amplitude $Y_i s$ (usually high frequency) effectively filters the signal.

If the low amplitude $Y_i s$ are scattered about the vector, filtering is still being done, but it is not necessarily characterizable as a low, high, or band-pass filter. This is a widely used compression scheme. In the case of images, most frequently a **Discrete Cosine Transform** (DCT) is used.

$$A_{ij} = \begin{cases} \sqrt{\frac{1}{N}}\, cos\left(\frac{j\pi(2i+1)}{2N}\right) & i = j \\ \sqrt{\frac{2}{N}}\, cos\left(\frac{j\pi(2i+1)}{2N}\right) & i \neq j \end{cases}$$

For correlated data that forms a first-order Markov chain, then the energy compaction factor is very close to the energy compaction factor of KLT. The DCT has very good performance for highly correlated data.

A problem with modern technology is that image capturing systems, for example, lack hardware capable of these data compression techniques, so much more information is captured than is actually used after the compression. The process is outlined below.

Ideally, the sensing and compression would be combined into a compressed sensing step.

The compressed sensing block contains a matrix Φ. If the discrete signal is X, ideally the sensors will be able to do the transform matrix multiplication ΦX. Given any signal to be sensed, X, there is hardware to implement the linear combination of the rows of Φ. The multiplication is

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{M1} & \phi_{M2} & \cdots & \phi_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}$$

where $M \ll N$ for compression. Note that Φ is no longer a square matrix; it is short and fat. Thus only M samples are taken, and compression and sensing occur all in the same step. This can be done in hardware.

Decoding Transform Codes

In decoding, X needs to be recovered from Y, but there is not a unique solution for X given Φ and Y. However, some more information is known about X. Given any transform matrix F, FX is a vector that has few non-zero entries. F is known.

$$\Phi = \tilde{\Phi} F$$

Note that Φ is $M \times N$, $\tilde{\Phi}$ is $M \times N$, and F is $N \times N$.

$$\begin{aligned} Y &= \Phi X \\ &= \tilde{\Phi} F X \\ &= \tilde{\Phi} \tilde{X} \end{aligned}$$

where $\tilde{X} = FX$. $\tilde{X}$ is a sparse vector, i.e. it has few non-zero values. Let k be the number of non-zero entries of $\tilde{X}$. $k \ll N$. Knowing $\tilde{\Phi}$ is equivalent to knowing Φ, since F is known. The formulation of the problem is then: Given $Y = \Phi X$ and Φ, find X, where X has $k \ll N$ non-zero values. Also find the matrix Φ for which this problem is solvable.