Correlation in Statistics and in Data Compression

Added material to *Data Compression: The Complete Reference*

The aim of this document is to illuminate the concept of correlation as used in data compression. The correlation between pixels is what makes image compression possible, so this type of correlation deserves a better understanding. Such an understanding is provided here in the form of a quantitative treatment of this concept.

We start with a discussion of how the correlation between two variables (arrays of numbers) is measured in statistics. The Pearson correlation coefficient R is then introduced and explained. This is compared to the term "correlation" as used in data compression, i.e., the correlation between elements (pixels or audio samples) of a *single* variable (an image row or column or an audio file). Next, we propose two ways to quantitatively measure the correlation between elements of a single variable, and support these proposals with experiments performed on real data.

The Correlation Coefficient. We start with a discussion of the concept of correlation and how it is measured statistically.

In statistics, correlation is measured between two random variables (arrays of numbers) a and b. We say that the two variables are positively correlated if the numbers feature the same behavior. The two variables

$$a = (1, 2, 3, 4, 3, 2, 1) \quad \text{and} \quad b = (3, 5, 7, 9, 7, 5, 3)$$

are strongly (and positively) correlated, since $a_i > a_{i-1}$ implies $b_i > b_{i-1}$ and $a_i < a_{i-1}$ implies $b_i < b_{i-1}$. In other words, knowing a helps in predicting b. The correlation coefficient should be defined such that its value for these two variables would be large positive. If we reverse one of the variables, they become negatively correlated, since $a_i > a_{i-1}$ now implies $b_i < b_{i-1}$. Knowing a in this case also helps in predicting b, but we want the correlation coefficient to be negative. If the relation $a_i > a_{i-1}$ tells us nothing about the relation between b_i and b_{i-1}, then there is no association between the variables, they are decorrelated and their correlation coefficient should be zero.

The English statistician Karl Pearson [1857–1936] was the first to approach the study of correlation scientifically. He measured the heights of 1078 fathers and their sons (at maturity) and arranged the results in a *scatter diagram*, similar to those of Figure 1b,c,d. Each point on the diagram corresponds to the heights of a father-son pair. Pearson realized that such a diagram can be the basis for defining a number (today denoted by R) that measures the correlation between the two arrays of values. The points of Figure 1b are clustered around the main diagonal. This means that the larger the x coordinate, the larger also the y coordinate. Thus, the diagram illustrates a strong positive association between the variables. Similarly, the points of Figure 1c are clustered around the secondary diagonal. This means that the larger the x coordinate, the smaller the y coordinate. Thus, this diagram illustrates a strong negative association between the variables. Figure 1d is an example of no correlation. Knowing the values of variable a does not help in predicting the values of b, so R should be zero in this case.

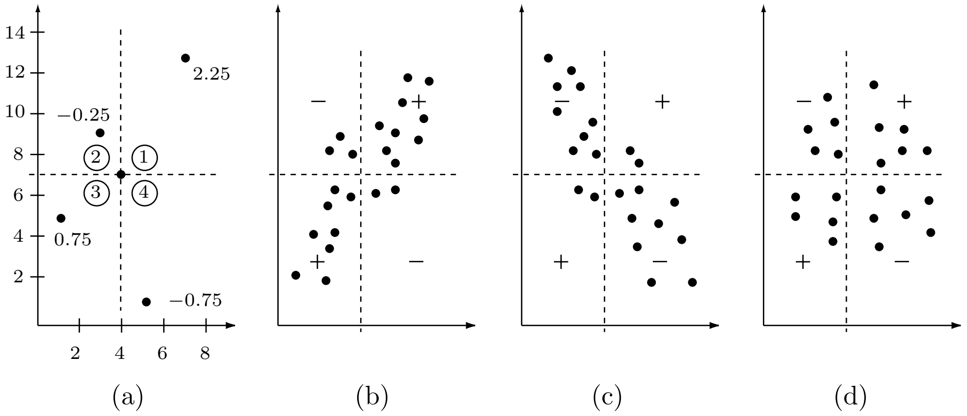

Figure 1: The correlation coefficient as a scatter diagram

It is also intuitively clear from these diagrams that the value of the correlation coefficient should depend on the thickness of the cloud of points. The thinner the cloud (the more concentrated it is around one of the diagonals), the stronger is the association between the variables, and the larger (positive or negative) the value of R should be. One way to measure the thickness of the cloud is to measure the distance between each point and the diagonal, and use the average. This, however, is complicated, and the correlation measure defined by Pearson uses the distance between each point (x, y) and the point of averages $(\bar{x}, \bar{y})$.

From the earlier discussion of variance, it is clear that these distances are measured by the covariance. The covariance s_{ij} of two variables i and j is therefore a measure of the correlation between them. The actual definition of correlation divides s_{ij} by the standard deviations of the two variables, since this normalizes R_{ij} and limits its value to the range $[-1, +1]$. Thus, the traditional definition of the correlation coefficient R is

$$R_{ij} = \frac{s_{ij}}{s_i \cdot s_j}.$$

The proof that R is normalized uses the *Schwartz inequality*

$$\left|\sum a_i b_i\right| \le \sqrt{\sum a_i^2}\sqrt{\sum b_i^2}.$$

Employing this inequality, it is easy to see that

$$|R_{xy}| = \frac{|\frac{1}{n}\sum(x_i-\bar{x})(y_i-\bar{y})|}{\sqrt{\frac{1}{n}\sum(x_i-\bar{x})^2}\sqrt{\frac{1}{n}\sum(y_i-\bar{y})^2}} \le \frac{\sqrt{\frac{1}{n}\sum(x_i-\bar{x})^2}\sqrt{\frac{1}{n}\sum(y_i-\bar{y})^2}}{\sqrt{\frac{1}{n}\sum(x_i-\bar{x})^2}\sqrt{\frac{1}{n}\sum(y_i-\bar{y})^2}} = 1.$$

This section, however, approaches R from a different direction.

The first step in defining the correlation coefficient is to *standardize* the values of the two variables. This eliminates any differences due to the use of a particular scale or units. Imagine that the values of variable $a = (1, 2, 3, 4, 3, 2, 1)$ are in kilograms and that variable $b = (3.2, 5.4, 7.6, 9.8, 7.6, 5.4, 3.2)$ contains the same values in pounds (and also incremented by 1). Thus, variables a and b differ by scale (a factor of 2.2) and origin (one unit), but express the same quantities (seven weights). Given another variable c, we therefore intuitively feel that the two correlation coefficients R_{ac} and R_{bc} should be equal. Standardizing a variable should therefore be done by changing its mean and variance to fixed values, and it has been agreed that the mean of a standardized variable should be zero, while its variance should be 1. Standardizing a variable v is done in two steps. First, its mean and variance are computed and the mean $\bar{v}$ is subtracted from all the values v_i, then the resulting values are divided by the variance. When variables a and b above are standardized in this way, they are both transformed into the array

$$(-1.15549, -0.256776, 0.641941, 1.54066, 0.641941, -0.256776, -1.15549).$$

An an example, consider the variables $x = (1, 3, 4, 5, 7)$ and $y = (5, 9, 7, 1, 13)$. The average of the x values is 4 and their standard deviation is 2. The standardized values of x are therefore

$$(1-4)/2 = -1.5, (3-4)/2 = -0.5,\ (4-4)/2 = 0,\ (5-4)/2 = 0.5,\ \text{and}\ (7-4)/2 = 1.5.$$

These standardized values tell how far, in units of standard deviation, the original values of x are above or below the average. Thus, the standardized value -1.5 implies that the first original value of x $(= 1)$ is 1.5 standard deviations $(= 1.5{\cdot}2$ units) below the average 4. Similarly, the average of the y values is 7 and their standard deviation is 4/3, leading to the standardized values $(-0.5, 0.5, 0.0, -1.5, 1.5)$.

The second step is to calculate the correlation coefficient R_{xy} as the average of the products $x_i y_i$ of the standardized values of x and y. Thus

$$R_{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i. \tag{1}$$

In our example

$$R = \big[(-1.5)(-0.5) + (-0.5)0.5 + 0{\cdot}0 + 0.5(-1.5) + 1.5\cdot 1.5\big]/5 = 0.40,$$

indicating a weak positive correlation between the two variables.

This definition makes it easy to see why R measures association between variables. Figure 1a shows a zero point (an origin) placed at the point of averages $(4, 7)$ of the original values. The first pair $(1, 5)$ of original (x, y) values is standardized to the two negative values $(-1.5, -0.5)$ because both 1 and 5 are below their averages. The product $(-1.5)(-0.5)$ is the positive value 0.75 plotted in the third quadrant (quadrant numbers are shown circled). Similarly, the last pair $(7, 13)$ of the original (x, y) values is standardized to the two positive values $(1.5, 1.5)$ because both 7 and 13 are above their averages. The product 1.5·1.5 is the positive value 2.25 plotted in the first quadrant. However, the second pair of values $(3, 9)$ is standardized to one negative and one positive value, and therefore produces a negative product −0.25 that's plotted in the second quadrant. Similarly, the fourth pair of values produces the point −0.75 in the fourth quadrant.

We therefore conclude that positive association between values of the two variables (both x_i and y_i are above or both are below their averages) produces points in quadrants 1 and 3 (the positive quadrants of Figure 1b) and thus results in a positive correlation coefficient. On the other hand, values that are negatively associated (one above and one below the averages) produce points on quadrants 2 and 4 (the negative quadrants of Figure 1c) and result in a negative R.

For an even deeper understanding of R, we provide another interpretation for it. The *dot product* (or scalar product) of two vectors $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ is defined as

$$x \cdot y = \sum x_i y_i.$$

Its value is the product of three quantities, the magnitudes of x and y and the cosine of the angle between them. Standardized vectors have a magnitude of 1, so their dot product is closely related to the angle between them. If the vectors point in the same direction, the angle between them is zero and their dot product is 1. If they point in opposite directions, their dot product is $\cos 180° = -1$, and if they are perpendicular, their dot product is $\cos 90° = 0$. Thus, the correlation coefficient of two variables can be viewed as a measure of the angle between the "directions" of the variables.

The definition of R implies that it has the following useful properties:

1. It is a pure number. This is because standardized values are pure numbers. Standardization eliminates all the effects of units and origin. Adding a constant to the values of a variable or multiplying them by a constant does not change R because these transformations are cancelled out when the values are standardized.

2. It is symmetric, $R_{ij} = R_{ji}$. This is because the products $x_i y_i$ are commutative.

A detailed discussion of correlation can be found in: Freedman, D., R. Pisani et al., *Statistics*, 2nd edition, W. W. Norton, 1991.

Correlation in Data Compression

The image, video, and audio compression literature favors the term *correlation*. Expressions such as "consecutive audio samples are correlated" and "in images of interest, the pixels are correlated" abound. In contrast with statistics, however, no attempt is made to quantify the correlation between pixels or audio samples and assign it a numerical value. The problem is that the correlation coefficient used in statistics measures the correlation between two arrays of numbers, whereas in data compression the interest is in correlation between neighbors in the same array.

This document proposes two measures to quantify the correlation between pixels in an image. The first measure is one-dimensional and can therefore be also applied to audio samples. This measure applies the Pearson correlation coefficient

R to assign a numeric value to the correlation between elements of a single array. Given an array $a = (a_1, a_2, \ldots, a_n)$ of n values, we construct the two arrays $x = (a_1, a_2, \ldots, a_{n-1})$ and $y = (a_2, a_3, \ldots, a_n)$ of $n-1$ values each, and compute R_{xy}. Array x is a minus its last element and array y is a shifted version of a with its first element dropped. The following arguments justify the use of this measure.

1. From the definition of R [Equation (1)] it is clear that our proposal computes the sum $a_1a_2 + a_2a_3 + \cdots + a_{n-1}a_n$ (performed on standardized a_i). If the values are correlated, a_i and a_{i+1} tend to be close, bringing this sum close to 1.

2. When applied to the highly-correlated array $a = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$, the proposed measure produces 1, but when applied to the same numbers arranged randomly $a = (4, 9, 3, 7, 10, 2, 6, 1, 5, 8)$ it results in -0.3773. When applied to arrays of alternating elements such as $a = (0, 1, 0, 1, 0, 1, 0, 1, 0, 1)$, the result is -1, as expected from the previous argument.

row	$R^{(1)}$	$R^{(2)}$	$R^{(3)}$	$R^{(4)}$	$R^{(5)}$	$R^{(6)}$	$R^{(7)}$	$R^{(8)}$
62	0.8689	0.6881	0.5536	0.4527	0.3622	0.2800	0.2024	0.1198
63	0.8563	0.6527	0.5436	0.4651	0.3523	0.2203	0.1078	0.0145
64	0.8698	0.7050	0.5682	0.4428	0.2978	0.1701	0.0362	−0.0673
65	0.8800	0.6897	0.5016	0.3122	0.1294	−0.0258	−0.1452	−0.2486
66	0.8387	0.5847	0.4019	0.2677	0.1072	−0.0640	−0.1652	−0.1865
ave	0.8628	0.6640	0.5138	0.3881	0.2498	0.1520	0.1314	0.1273

Table 2. Correlations of five image rows and eight distances

```
clear
filename='lena128'; dim=128;
fid=fopen(filename,'r');
img=fread(fid,[dim,dim])';
clm=1;
for r=62:66
 for i =1:8
  a=img(r,1:128-i);
  b=img(r,i+1:128);
  c=corrcoef(a,b);
  d(clm,i)=c(1,2);
 end %i
 clm=clm+1;
end %r
d
sum(abs(d),1)/5 % averages
```

Figure 3. Matlab code for the proposed correlation

3. We feel intuitively that a pixel is expected to be strongly correlated only with its immediate neighbors. The correlation of a pixel with other neighbors should

drop quickly with distance. Thus, we can generalize the definition of our measure and define a quantity $R^{(k)}$ to measure the correlation between (standardized) a_i values separated by k units of distance as

$$R^{(k)} = a_1 a_k + a_2 a_{k+1} + \cdots + a_{n-k} a_n.$$

The following experiment illustrates this type of correlation. We use the well-known "Lena" image in grayscale and at a size of 128×128 pixels. Applying our measure to the five center rows (rows 62 through 66) of this image, and repeating each calculation eight times, to compute $R^{(1)}$ through $R^{(8)}$, we end up with the results summarized in Table 2, with the Matlab code that created it listed in Figure 3. It is obvious from the table (especially from its last row, the averages) that the correlation drops quickly as we compare a pixel to neighbors that get more and more distant. Neighbor pixels separated by seven or more units are for all practical purposes decorrelated.

The second measure proposed here is denoted by R_{xy} and is two-dimensional. It depends on two shift parameters x and y, and it produces one number, normalized to the range $[-1, +1]$, that describes the amount of correlation between the pixels in the entire image. Computing this measure is a multistep process that compares each row and each column in an image to their shifted versions and employs the Pearson correlation coefficient to compute a single number R_{xy}. The measure is defined such that for $x = y = 0$ (no shifts), it results in $R_{xy} = 1$.

We denote the pixels of the image by $I[i, j]$ where $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Based on the original definition of R by means of covariance, the proposed measure is computed in the following steps:

$$\begin{aligned}
\bar{I} &= \frac{1}{n \cdot m} \sum_{i=1}^{n-x} \sum_{j=1}^{m-y} I[i, j],\\
\bar{S} &= \frac{1}{n \cdot m} \sum_{i=1}^{n-x} \sum_{j=1}^{m-y} I[i + x, j + y],\\
\mathrm{SQI} &= \sqrt{\sum_{i=1}^{n-x} \sum_{j=1}^{m-y} (I[i, j] - \bar{I})^2},\\
\mathrm{SQS} &= \sqrt{\sum_{i=1}^{n-x} \sum_{j=1}^{m-y} (I[i + x, j + y] - \bar{S})^2},\\
R_{x,y} &= \frac{\sum_{i=1}^{n-x} \sum_{j=1}^{m-y} (I[i, j] - \bar{I})(I[i + x, j + y] - \bar{S})}{\mathrm{SQI} \cdot \mathrm{SQS}}.
\end{aligned}$$

Applying this measure to the entire Lena image (grayscale at 128×128 pixels) while varying x and y independently from 0 to 7 has resulted in the values of Table 4. For comparison, Table 5 lists the results obtained by this method for a random image of the same size. As an example, the value 0.1368 at the bottom-right corner of

$x \rightarrow$	0	1	2	3	4	5	6	7
y 0:	1.0000	0.8851	0.7066	0.5695	0.4541	0.3672	0.2957	0.2331
↓ 1:	0.9504	0.8443	0.6856	0.5503	0.4372	0.3543	0.2836	0.2214
2:	0.8703	0.7930	0.6595	0.5306	0.4233	0.3434	0.2716	0.2094
3:	0.8048	0.7426	0.6272	0.5094	0.4093	0.3312	0.2579	0.1956
4:	0.7472	0.6944	0.5942	0.4878	0.3930	0.3148	0.2414	0.1806
5:	0.6958	0.6494	0.5618	0.4645	0.3721	0.2941	0.2225	0.1651
6:	0.6493	0.6069	0.5279	0.4362	0.3465	0.2712	0.2031	0.1501
7:	0.6047	0.5659	0.4924	0.4043	0.3188	0.2471	0.1847	0.1368

Table 4. Correlations for the entire Lena image

$x \rightarrow$	0	1	2	3	4	5	6	7
y 0:	1.0000	−0.0128	−0.0064	0.0040	−0.0063	0.0065	0.0123	0.0151
↓ 1:	0.0060	−0.0009	−0.0044	0.0184	0.0050	0.0262	0.0130	0.0044
2:	−0.0008	0.0021	−0.0019	−0.0053	0.0119	0.0042	0.0007	−0.0002
3:	−0.0028	−0.0016	0.0083	0.0024	−0.0015	0.0063	0.0070	−0.0069
4:	0.0106	−0.0030	−0.0051	0.0017	0.0131	0.0063	−0.0011	−0.0021
5:	0.0130	−0.0045	−0.0052	−0.0057	0.0044	0.0070	−0.0063	0.0005
6:	−0.0132	−0.0036	−0.0035	0.0004	0.0069	−0.0065	0.0158	0.0144
7:	0.0096	0.0028	0.0138	−0.0019	−0.0016	0.0007	−0.0069	0.0080

Table 5. Correlations for a random 128×128 image

```
% A single correlation measure for rows and cols
% of an image. Use with various values of x, y
clear
filename='lena128'; dim=128;
fid=fopen(filename,'r');
img=fread(fid,[dim,dim])';
%img=rand(128); a random image for comparison
for x =0:7
 for y =0:7
  iimg=img(1:dim-x,1:dim-y); % delete last x,y rows cols
  simg=img(1+x:dim,1+y:dim); % delete first x,y rows cols
  Ibar=sum(sum(iimg,1),2)/((dim-x)*(dim-y));
  Sbar=sum(sum(simg,1),2)/((dim-x)*(dim-y));
  timg=(iimg-Ibar).*(iimg-Ibar);
  SQI=sqrt(sum(sum(timg,1),2));
  timg=(simg-Sbar).*(simg-Sbar);
  SQS=sqrt(sum(sum(timg,1),2));
  timg=(iimg-Ibar).*(simg-Sbar);
  R(x+1,y+1)=sum(sum(timg,1),2)/(SQI*SQS);
 end
end
R
```

Figure 6. Matlab code for the proposed correlation

Table 4 is obtained when all the rows and columns of the image are shifted seven positions and the shifted image is correlated with the original one. The reader should compare this value with the 0.1314 found on the bottom row of Table 2.

Another way to look at correlation in an image is to compute the Pearson correlation of every row with every other row and of every column with every other column. Assuming an image with m rows and n columns, this results in two symmetric matrices, for the row and column correlations, respectively. If the image is square, these symmetric matrices have the same size and can be combined by dropping a triangular half of each and concatenating the remaining triangles. The Matlab code of Figure 7 generates such a matrix. Element (i,j) in the upper half of this matrix is the correlation of row i with row j, while element (i,j) in the lower half is the correlation of columns i and j. However, since most images have at least a few hundred rows and columns, such a correlation matrix is too big to be evaluated visually.

```
filename=’lena128’; dim=128;
fid=fopen(filename,’r’);
img=fread(fid,[dim,dim])’;
upper=triu(corrcoef(img’)) % correlate rows
lower=tril(corrcoef(img),-1) % correlate cols
% -1 produces zeros on main diagonal
upper+lower
```

Figure 7. Matlab code for a complete correlation matrix of an image