1 The Correlation Coefficient

The Pearson Correlation Coefficient \( \rho(X,Y) \) is a way to measure how linearly correlated (in other words, how well a linear model captures the relationship between) distributions \( X \) and \( Y \).

\[
\rho(X,Y) = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}
\]

Here are some important facts about it:

- It is commutative. \( \rho(X,Y) = \rho(Y,X) \)
- It always lies between -1 and 1. \( -1 \leq \rho(X,Y) \leq 1 \)
- It is completely invariant to affine transformations.

\[
\rho(aX + b, cY + d) = \frac{\text{Cov}(aX + b, cY + d)}{\sqrt{\text{Var}(aX + b)\text{Var}(cY + d)}} = \frac{\text{Cov}(aX, cY)}{\sqrt{\text{Var}(aX)\text{Var}(cY)}} = \frac{a \cdot c \cdot \text{Cov}(X,Y)}{\sqrt{a^2\text{Var}(X) \cdot c^2\text{Var}(Y)}} = \frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}} = \rho(X,Y)
\]

For \( n \) data points \( X \) and \( Y \) with means \( \bar{X} \) and \( \bar{Y} \), the sample Pearson Correlation Coefficient \( r \) is given by

\[
\frac{\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{X})^2 \cdot \sum_{i=1}^{n} (y_i - \bar{Y})^2}} = \frac{\bar{X}^T\bar{Y}}{\sqrt{\bar{X}^T\bar{X} \cdot \bar{Y}^T\bar{Y}}}, \quad \bar{X} = X - \bar{X}, \bar{Y} = Y - \bar{Y}
\]
If you’ve ever used your graphing calculator or Excel to generate “best-fit lines” for reports in high school, you may recall reporting an \( r^2 \) value for the best-fit line. This \( r \) is exactly the sample correlation coefficient!

Here are some 2-D scatterplots and their corresponding correlation coefficients:

You should notice that:

- The magnitude of \( r \) increases as \( X \) and \( Y \) become more linearly correlated.
- The sign of \( r \) tells whether \( X \) and \( Y \) have a positive or negative relationship.
- The correlation coefficient is undefined if either \( X \) or \( Y \) has 0 variance (horizontal line).

2 Correlation and Gaussians

Here’s a neat fact: if \( X \) and \( Y \) are jointly Gaussian, we can define a distribution on normalized \( X \) and \( Y \) and have their relationship entirely captured by \( \rho(X,Y) \).

\[
\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}(0, \Sigma) \\
\rho(X,Y) = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \\
\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & \rho \sigma_x \sigma_y \\ \rho \sigma_x \sigma_y & \sigma_y^2 \end{bmatrix}
\]

\[
\begin{bmatrix} \sigma_x^{-1} & 0 \\ 0 & \sigma_y^{-1} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} \sigma_x^{-1} & 0 \\ 0 & \sigma_y^{-1} \end{bmatrix} \Sigma \begin{bmatrix} \sigma_x^{-1} & 0 \\ 0 & \sigma_y^{-1} \end{bmatrix}^T ) \\
\sim \mathcal{N}(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix})
\]

3 Bringing it all together: CCA

**CCA: Canonical Correlation Analysis** is a method of modeling the relationship between two point sets by making use of the correlation coefficient. Formally, given data matrices \( X_{n \times p} \) and...
Y_{n\times q}$, we want to find projection vectors $u_{p\times 1}$ and $v_{q\times 1}$ (for $X$ and $Y$ respectively) that **maximizes the correlation between $Xu$ and $Yv$.**

$$\max_{u,v} \rho(Xu, Yv) = \max_{u,v} \frac{u^T X^TYv}{\sqrt{u^T X^TXu \cdot v^T Y^TYv}}$$

An advantage of CCA over PCA is that it is invariant to scalings and affine transformations of $X$ and $Y$.

Take a simplified scenario, with two matrix-valued random variables $X, Y$. For simplicity, say $Y = X + \epsilon$ where noise $\epsilon$ has huge variance. What happens when we run PCA on $Y$? Since PCA maximizes variance, it will actually project $Y$ (largely) into the column space of $\epsilon$! However, we’re interested in $Y$’s relationship to $X$, not its dependence on noise. How can we fix this? As it turns out, CCA solves this issue. Instead of maximizing variance of $Y$, we maximize correlation between $X$ and $Y$. In some sense, we want the maximize ”predictive power” of information we have.

Let’s try to massage the maximization problem into a form that we can reason with more easily!

1. First, let’s choose matrices $W_x, W_y$ to **whiten** $X$ and $Y$. This will make the (Co)variance matrices $(XW_x)^T(XW_x)$ and $(YW_y)^T(YW_y)$ become identity matrices and simplify our expression. To do this, we’ll make use of the SVD:

$$X^TX = U_xS_xU_x^T$$

$$(U_xS_1^{1/2}U_x^T)^2 = X^TX$$

We can write this because $X^TX$ is symmetric. Can you see a choice of $W_x$ that will whiten $X$?
The answer is $W_x = U_xS^{-1/2}U_x^T$! Expand out the calculation of $(XW_x)^T(XW_x)$ (which we wanted to be $I$) to see for yourself. We’ll do the exact same thing to choose $W_y$ to whiten $Y$.

Let’s denote our whitened versions of $X$ and $Y$ as $X_w$ and $Y_w$ respectively. Since these are just affine transformations of $X$ and $Y$, our goal to maximize the correlation between $X$ and $Y$ is equivalent to the same goal for $X_w$ and $Y_w$. Namely:

$$\max_{u,v} \rho(Xu, Yv) = \max_{u,w, v_w} \rho(X_wu_w, Y_wv_w) = \max_{u,v} \frac{u_w^T X_w^TY_wv_w}{\sqrt{u_w^T X_w^TX_wu_w \cdot v_w^T Y_w^TY_wv_w}} = \max_{u,v} \frac{u_w^T X^TYv_w}{\sqrt{u_w^T u_w \cdot v_w^T v_w}}$$

2. Second, let’s choose matrices $D_x, D_y$ to **decorrelate** $X_w$ and $Y_w$. This will let us simplify the covariance matrix $(X_wD_x)^T(Y_wD_y)$ into a **diagonal** matrix. To do this, we’ll once again make
use of the SVD:

\[ X_w^T Y_w = USV^T \]

This is a little harder to see, but the choice of \( U \) for \( D_x \) and \( V \) for \( D_y \) will accomplish exactly what we want.

\[ (X_wU)^T (Y_wV) = U^T X_w Y V = U^T (USV^T) V = S \]

Let’s denote our new decorrelated \( X \) and \( Y \) as \( X_d \) and \( Y_d \) respectively. Once again, because all we did was an affine transformation, the correlation coefficient remains unchanged, and our new maximization problem becomes:

\[
\max_{u,v} \rho(Xu, Yv) = \max_{u_d,v_d} \rho(X_d u_d, Y_d v_d) = \max_{u_d,v_d} \frac{u_d^T S v_d}{\sqrt{u_d^T u_d \cdot v_d^T v_d}}
\]

We’re finally ready to start reasoning about the expression that we’re trying to maximize. Recall for a moment that we are choosing vectors \( u \) and \( v \) to project data matrices \( X \) and \( Y \) onto.

Here’s a fact: You can rewrite the numerator of this expression as \( \sum_i S_{ii} u_i v_i \). Without loss of generality, let’s force \( u_d \) and \( v_d \) to be unit vectors, allowing us to ignore the denominator. Recall that \( S \) is diagonal matrix of singular values (of \( X_w^T Y_w \)) arranged in descending order. The choice of \( u \) and \( v \), therefore, becomes a weighted sum of the singular values, and if we want to maximize this sum, all we need to do is put all our eggs in one basket, if you will, and extract \( S_{11} \) by choosing \( u_d = v_d = [1 \ 0 \ 0 \ ... \ 0] \). Lots of work for a fairly simple result, huh?

To bring it back to our original optimization problem, all we have to do is remember the changes of basis we made (namely, \( W_x \) and \( W_y \) to whiten and \( D_x \) and \( D_y \) to decorrelate) and apply them to \( u_d \) and \( v_d \):

\[
\begin{align*}
\ u & = W_x u_w = W_x D_x u_d \\
\ v & = W_y v_w = W_y D_y v_d
\end{align*}
\]