8.7 Examples

See associated slides for population structure in Europe (Novembre et al, Nature 2008, `http://www.nature.com/nature/journal/v456/n7218/full/nature07331.html`)and Eigenfaces.

The "eigenfaces" example in the slides was developed by Matthew Turk and Alex Pentland (Journal of Cognitive Neuroscience, 1991, v3 (1)). The following quote is from their abstract:

> We have developed a near-real-time computer system that can locate and track a subject's head, and then recognize the person by comparing characteristics of the face to those of known individuals. ... The system functions by projecting face images onto a feature space that spans the significant variations among known face images. The significant features are known as "eigenfaces," because they are the eigenvectors (principal components) of the set of faces; they do not necessarily correspond to features such as eyes, ears, and noses. The projection operation characterizes an individual face by a weighted sum of the eigenface features, and so to recognize a particular face it is necessary only to compare these weights to those of known individuals. Some particular advantages of our approach are that it provides for the ability to learn and later recognize new faces in an unsupervised manner, and that it is easy to implement using a neural network architecture.

8.8 What is connection between PCA and SVD?

Given $\mathbf{A}$ such that the rows of $\mathbf{A}$ have zero mean, define $\mathbf{Y} = \frac{1}{\sqrt{n-1}}\mathbf{A}^T$ (which has columns with zero mean). Then $\mathbf{Y}^T\mathbf{Y} = \mathbf{\Sigma}$, the covariance of $\mathbf{A}$. We have seen that the principal components of $\mathbf{A}$ are the eigenvectors of $\mathbf{\Sigma}$.

Now, if we calculate the SVD of $\mathbf{Y}$ to get $\mathbf{Y} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{Y}^T\mathbf{Y} = \mathbf{\Sigma}$. Therefore, the columns of $\mathbf{V}$ are the principal components of $\mathbf{A}$.

8.9 Problems with SVD and PCA

As we have seen, SVD and PCA are powerful analysis tools and SVD is a very stable procedure. They do not, however, come free of cost.

The time complexity of SVD is $O(m^2n + n^3)$ to calculate all of $\mathbf{U}, \mathbf{V}$ and $\mathbf{D}$ (where, typically, $m \gg n$) while faster algorithms are available when some elements of the SVD are not required.

However, the matrices $\mathbf{U}$ and $\mathbf{V}$ are not at all *sparse*, where we say a matrix is sparse when it mainly consists of zeros. Spareness is a commonly assumed property in large systems as it reflects the observation that most effects are local and do not influence all

parameters in the system — a large world with small neighbourhoods. Sparse matrices are typically computationally efficient to work with and store.

A second potential set-back is that SVD and PCA only work with data that can be (coherently) expressed as a two dimensional array (that is, a matrix). When data naturally has 3 or 4 dimensions arrays (*tensors*), as is common in many engineering applications, there is no perfect analogue to SVD or PCA or even eigenvectors.

Finally, when using PCA for data analysis, you should be aware of the strong assumptions being made. In particular, dependencies in the data are assumed to be linear, which may not be the case. PCA and SVD will always give an answer but it is up to the user to interpret whether or not it is a valid answer to any question they are interested in.

9 Least squares

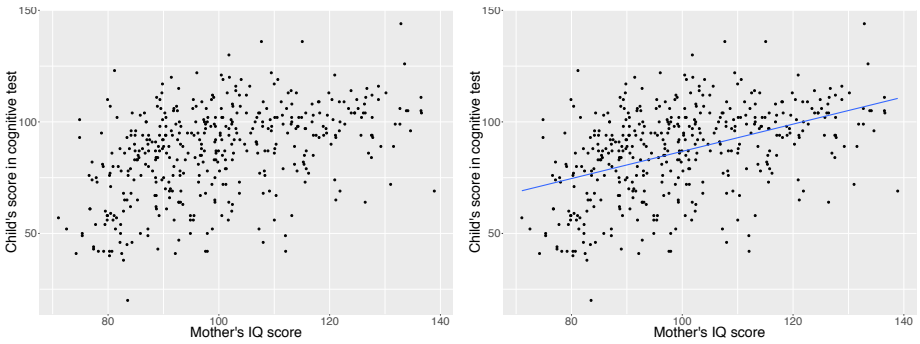


Figure 5: Left: Relationship between cognitive test scores for 3-4 year old children and mother's IQ score. Right: The same data with a least squares best fit line added. Discussed in Gelman and Hill, 2007, Cambridge University Press, data at http://www.stat.columbia.edu/ gelman/arm/examples/child.iq/kidiq.dta

You are probably familiar with the basic idea of least squares: we have a set of measurements and we want to fit a model to them. But no sufficiently simple model exactly fits all of the points at the same time. So how choose the model that is most satisfactory? The answer often given is that we chose the model that satisfies the *least squares* criterion: that is, the model for which the sum of the squares of differences between the predictions from the model and the actual observations is minimised.

For example, in Figure 5, we might want to fit a linear model to the relationship between a mother's IQ score and her young child's score in a cognitive test. This should be familiar to you as the linear regression problem in statistics.

This problem arises when we have an *overdetermined* linear system: recall that $\mathbf{Au} = \mathbf{b}$

is overdetermined when $\mathbf{A}$ is $m \times n$ matrix with $m > n$:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

In this case, $\mathbf{A}^{-1}$ does not exist and there is no $\mathbf{u}$ that solves this problem. (We ignore the highly unusual cases where a solution does exist.)

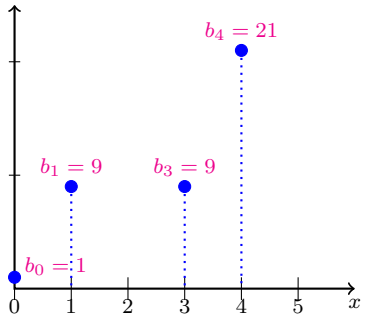
The goal, then, is to find the best solution $\mathbf{u}^*$ to the problem.

Example 1: Fitting $m = 4$ measurements by a small number $n = 2$ of parameters (e.g. linear regression in statistics)

Want to find the straight line $b_x = u_1 + u_2 x$ where we have observed the points b_x at x.

$$\begin{cases} u_1 + u_2 \cdot 0 &= 1 \\ u_1 + u_2 \cdot 1 &= 9 \\ u_1 + u_2 \cdot 2 &= 9 \\ u_1 + u_2 \cdot 4 &= 21 \end{cases} \Leftrightarrow$$

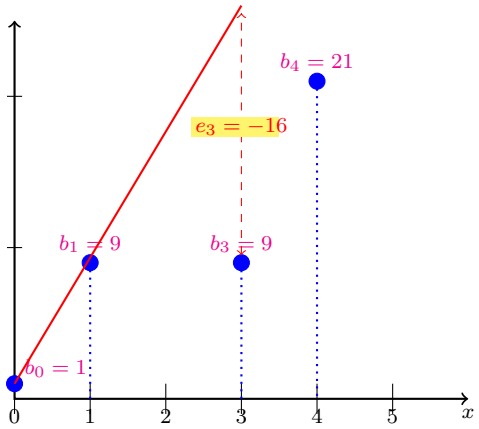
$$\begin{cases} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 9 \\ 9 \\ 21 \end{bmatrix} \end{cases}$$



The above set of equations clearly has no solution as vector $\mathbf{b}$ is not a linear combination of the two column vectors from $\mathbf{A}$:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 9 \\ 9 \\ 21 \end{bmatrix}$$

For example, The line $b = 1 + 8x$ through the first two points is almost certainly not the best line:



But why is this not the best line: look at the *error* or *residual* , $\mathbf{e} = \mathbf{b} - \mathbf{Au}$. For the two points the line does not pass through the error is $e_x = b_x - (1+8x)$ is large: $e_3 = 16$ and $e_4 = 12$. The *Total square error*, $E(\mathbf{u}) = 0 + 0 + 256 + 144 = 400$.

Notice that the *total square error* is given by.

$$E(\mathbf{u}) = \mathbf{e}^\mathsf{T}\mathbf{e} \equiv \| \mathbf{e} \|^2 = (\mathbf{b} - \mathbf{Au})^\mathsf{T}(\mathbf{b} - \mathbf{Au})$$

The Least Squares method to find the chooses a solution $\mathbf{u}^*$ that minimises $E(\mathbf{u})$.

How do we find $\mathbf{u}^*$? To find the minimum of $E(\mathbf{u})$, we can differentiate with respect to $\mathbf{u}$, set to 0 and attempt to solve for $\mathbf{u}$:

$$\begin{aligned} E(\mathbf{u}) &= (\mathbf{b} - \mathbf{Au})^\mathsf{T}(\mathbf{b} - \mathbf{Au}) \\ &= \mathbf{b}^\mathsf{T}\mathbf{b} - 2\mathbf{u}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{b} + \mathbf{u}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{Au} \end{aligned}$$

Differentiating and setting to 0:

$$\begin{aligned} \frac{\partial E(\mathbf{u})}{\partial \mathbf{u}} &= 0 \\ \implies -2\mathbf{A}^\mathsf{T}\mathbf{b} + 2\mathbf{A}^\mathsf{T}\mathbf{Au} &= \mathbf{0} \\ \implies \mathbf{A}^\mathsf{T}\mathbf{Au} &= \mathbf{A}^\mathsf{T}\mathbf{b} \end{aligned}$$

This equation, $\mathbf{A}^\mathsf{T}\mathbf{Au} = \mathbf{A}^\mathsf{T}\mathbf{b}$ is called the *normal equation*.

The least squares estimate, $\mathbf{u}^*$, is the solution to the normal equation.

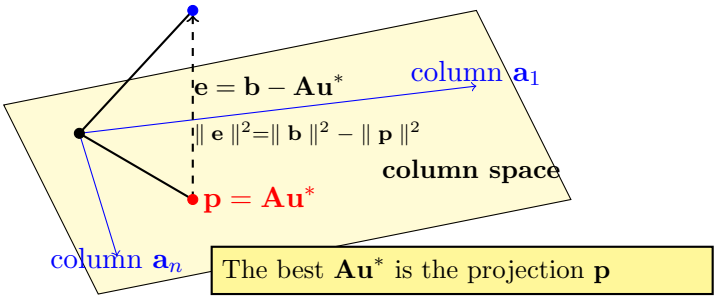
Notice that $\mathbf{A}^T\mathbf{A}$ is square and symmetric. In some cases it may be possible to directly find the inverse (in particular, when $\mathbf{A}$ has independent columns, then $\mathbf{A}^T\mathbf{A}$ is positive definite and $\mathbf{A}^T\mathbf{A}$ is invertible in which case $\mathbf{u}^* = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{b}$). In other cases, this approach may be highly unstable, so stable numerical techniques need to be employed.

9.1 Understanding the Least Squares solution

This subsection is not examined. The main point of this section is to add some geometric and algebraic understanding to our discussion. You are not expected to understand all

the detail in this section, but do familiarise yourself with the concept and definition of the projection matrix $\mathbf{P}$ defined below.

The equation $\mathbf{Au} = \mathbf{b}$ can be seen as attempting to represent $\mathbf{b}$ as a linear combination of the n columns of $\mathbf{A}$. This is impossible, since the n columns of $\mathbf{A}$ describe, at most, an n-dimensional plane inside the much larger m dimensional space (recall that $n < m$). Thus b is unlikely to fall on that plane. The plane is called the *column space* of $\mathbf{A}$.



The best solution, $\mathbf{Au}^*$, is the nearest point to $\mathbf{b}$ on that plane. Call this point $\mathbf{p} = \mathbf{Au}^*$.

Now, from a geometric argument, you can see that the error vector $\mathbf{e}$ is orthogonal (perpendicular) to this plane. Thus $\mathbf{A}^T e = 0$.

Notice that $0 = \mathbf{A}^T\mathbf{e} = \mathbf{A}^T(\mathbf{b} - \mathbf{Au}^*) = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{Au}^* \implies \mathbf{A}^T\mathbf{b} = \mathbf{A}^T\mathbf{Au}^*$. This is a geometric derivation of the normal equation that we earlier saw derived from calculus.

The point $\mathbf{p}$ $(= \mathbf{Au}^*)$ is the projection of $\mathbf{b}$ onto the column space of $\mathbf{A}$:

$$\mathbf{p} = \mathbf{Au}^* = \underbrace{\left[\mathbf{A}\left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}\right]}_{\text{projection matrix } \mathbf{P}}\mathbf{b} = \mathbf{Pb},$$

where we define the *Projection matrix*, $\mathbf{P} = \mathbf{A}\left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}$. $\mathbf{P}$ is symmetric and of size $m \times m$ but the rank of $\mathbf{P}$ is only n (as all the factors of $\mathbf{P}$ in the definition above have rank n).

9.2 Computing the Least Squares solution, $\mathbf{u}^*$

We consider three methods for computing the least squares solution to a linear system. They are Gaussian elimination, QR Decomposition (aka Orthogonalisation) and computation of the pseudo-inverse via SVD.

9.3 Computing $\mathbf{u}^*$ via Gaussian elimination

Given the normal equation $\mathbf{A}^T\mathbf{Au} = \mathbf{A}^T\mathbf{b}$, we may be tempted to find the solution by Gaussian elimination, where we reduce the the matrix $\mathbf{A}^T\mathbf{A}$ to upper triangular form using elementary row operations.

This solution can work but is highly unstable. To see why it is unstable, consider the condition number of the matrix $\mathbf{A}^T\mathbf{A}$. It can be shown that the condition number of

$\mathbf{A}^T\mathbf{A}$ is the square of the condition number of $\mathbf{A}$ (if we take $\sigma_{\min}$ to be the smallest non-zero singular value in the definition of condition number). So even if $\mathbf{A}$ has only moderately widely spread singular values, $\mathbf{A}^T\mathbf{A}$ can have a very large condition number and solution by row reduction can be very unstable.