Least Squares Algorithms

Georgy Gimel'farb

COMPSCI 369 Computational Science

Learning outcomes: Understand the least squares framework

RECOMMENDED READING:

- M. T. Heath: *Scientific Computing: An Introductory Survey*. McGraw-Hill, 2002: Chapters 3, 6
- G. Strang, *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007: Sections 2.3, 2.8
- W. H. Press et al., *Numerical Recipes: The Art of Scientific Computing*. Cambridge Univ. Press, 2007: Chapter 15
- C. Woodford, C. Phillips: *Numerical Methods with Worked Examples*. Chapman & Hall,1997: Chapter 3

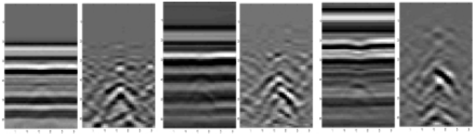
ACKNOWLEDGEMENTS: Facial images from the MIT-CBCL face recognition database:

- B. Weyrauch, J. Huang, B. Heisele, and V. Blanz: "Component-based face recognition with 3d morphable models", in *Proc. of CVPR Workshop on Face Processing in Video (FPIV'04)*, Washington DC, 2004.

Least Squares Methods in Practice

Vehicle mounted Ground Penetrating Radar mine detection system

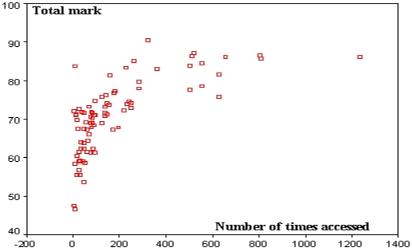
Center for Geospatial Intelligence, Univ. of Missouri-Columbia, USA (geoint.missouri.edu/CGI2/research10.aspx)

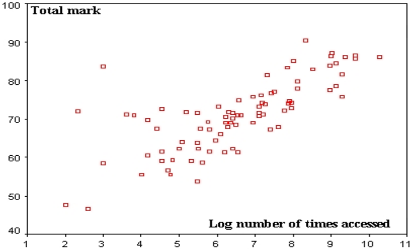


Raw data and results of linear prediction pre-processing for a plastic mine at 2 *in*, metal mine at 4 *in*, and plastic mine in 6 *in* in deep

P. Suanpang e.a.: Relationship between learning outcomes and online accesses

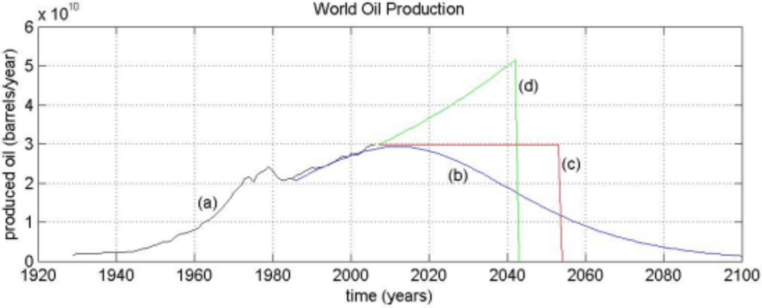
Australasian Journal of Educational Technology, vol. 20 (3), 371–387, 2004





Least Squares Methods in Practice

Various least-squares predictors



http://www.inf.ethz.ch/personal/fcellier/Pubs/World/tod_10i.png

Rectangular Matrices

Overdetermined linear system

- **More equations than unknowns!**
- $\mathbf{Au} = \mathbf{b}$: the $m \times n$ matrix $\mathbf{A}$; $m > n$:

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

  - $\mathbf{A}^{-1}$ does not exist: **no solution!**
- Goal: to find the best solution $\mathbf{u}^*$ when the system $\mathbf{Au} = \mathbf{b}$ is overdetermined
  - Too many equations; exact solutions are unlikely

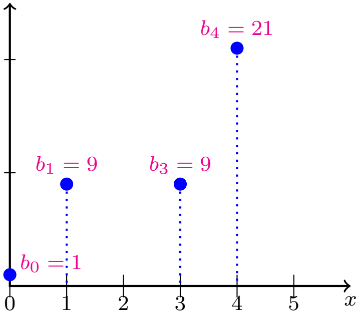
Rectangular Matrices

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

- Straight line $b_x = u_1 + u_2 x$

$$\left\{ \begin{array}{lcr} u_1 + u_2 \cdot 0 & = & 1 \\ u_1 + u_2 \cdot 1 & = & 9 \\ u_1 + u_2 \cdot 2 & = & 9 \\ u_1 + u_2 \cdot 3 & = & 21 \end{array} \right. \Leftrightarrow$$

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



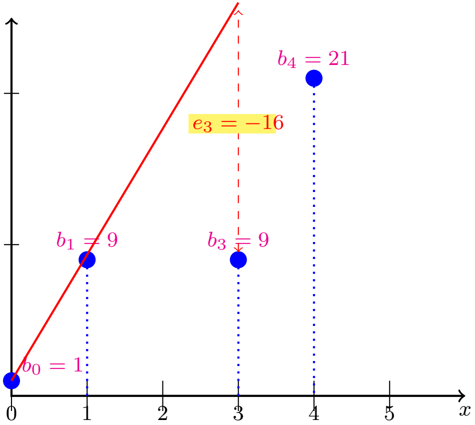
Principle of Least Squares

Equations in Example 1 have no solution:

- Vector $\mathbf{b}$ is not a linear combination of the two column vectors from $\mathbf{A}$:

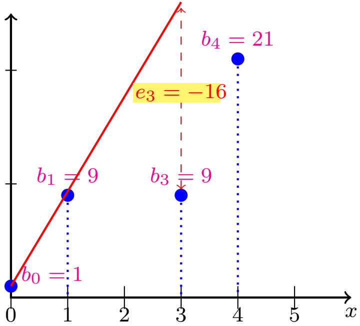
$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 9 \\ 9 \\ 21 \end{bmatrix}$$

- Line $1 + 8x$ through the first two points is almost certainly not the best line



Principle of Least Squares

- The error $e_x = b_x - (1 + 8x)$ is large for other two points: $e_3 = 16$ and $e_4 = 12$
- The squared error is $E = 0 + 0 + 256 + 144 = \mathbf{400}$!



$$\underbrace{\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} 1 \\ 9 \\ 9 \\ 21 \end{bmatrix}}_{\mathbf{b}} \Rightarrow \underbrace{\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{u}}_{\text{residual error}}$$

Total squared error $E(\mathbf{u}) = \mathbf{e}^\mathsf{T}\mathbf{e} \equiv \| \mathbf{e} \|^2$

$$= (\mathbf{b} - \mathbf{A}\mathbf{u})^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{u})$$

$$\to \min_{\mathbf{u}} \left\{ (\mathbf{b} - \mathbf{A}\mathbf{u})^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{u}) \right\}$$

Principle of Least Squares

(Unweighted) least squares method:

- Choose $\mathbf{u}^*$ to minimise the squared error:

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

- Let's solve for the minimiser:

$$\begin{aligned}
& \min_{\mathbf{u}} \left\{ E(\mathbf{u}) = (\mathbf{b} - \mathbf{A}\mathbf{u})^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{u}) \right\} \\
= \quad & \min_{\mathbf{u}} \left\{ \mathbf{b}^\mathsf{T}\mathbf{b} - 2\mathbf{u}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{b} + \mathbf{u}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{u} \right\} \\
\rightarrow \quad & \frac{\partial E(\mathbf{u})}{\partial \mathbf{u}} = 0 \\
\rightarrow \quad & -2\mathbf{A}^\mathsf{T}\mathbf{b} + 2\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{u} = \mathbf{0} \\
\rightarrow \quad & \mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{u} = \mathbf{A}^\mathsf{T}\mathbf{b}
\end{aligned}$$

Principle of Least Squares

Least squares estimate for $\mathbf{u}$

- Solution $\mathbf{u}^*$ of the **"normal" equation** $\mathbf{A}^\top\mathbf{A}\mathbf{u}^* = \mathbf{A}^\top\mathbf{b}$
  - The left-hand and right-hand sides of the insolvable equation $\mathbf{A}\mathbf{u} = \mathbf{b}$ are multiplied by $\mathbf{A}^\top$
  - Least squares is a projection of $\mathbf{b}$ onto the columns of $\mathbf{A}$
- Matrix $\mathbf{A}^\top\mathbf{A}$ is *square*, *symmetric*, and *positive definite* if $\mathbf{A}$ has independent columns
  - Positive definite $\mathbf{A}^\top\mathbf{A}$: the matrix is invertible; the normal equation produces $\mathbf{u}^* = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{b}$
- Matrix $\mathbf{A}^\top\mathbf{A}$ is *square*, *symmetric*, and *positive semi-definite* if $\mathbf{A}$ has dependent columns
  - If positive semi-definite $\mathbf{A}^\top\mathbf{A}$ (or almost semi-definite, so its determinant is close to zero: $|\mathbf{A}^\top\mathbf{A}| \approx 0$), then the QR factorisation is much safer!

Principle of Least Squares: Completing Example 1

The normal equation $\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{u}^* = \mathbf{A}^\mathsf{T}\mathbf{b}$:

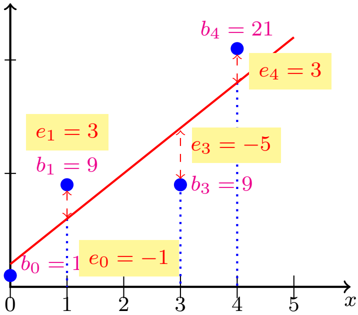
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 9 \\ 9 \\ 21 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 4 & 8 \\ 8 & 26 \end{bmatrix} \begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix} = \begin{bmatrix} 40 \\ 120 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix} = \frac{1}{40} \begin{bmatrix} 26 & -8 \\ -8 & 4 \end{bmatrix} \begin{bmatrix} 40 \\ 120 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} u_1^* \\ u_2^* \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Projection of $\mathbf{b}$ onto columns of $\mathbf{A}$: $\mathbf{A}\mathbf{u}^* = \mathbf{p}$
$\mathbf{e}^* = \mathbf{b} - \mathbf{A}\mathbf{u}^* \equiv \mathbf{b} - \mathbf{p} \perp \mathbf{p}$
Error for the best line: $e_x = b_x - \underbrace{(2 + 4x)}_{p_x}$

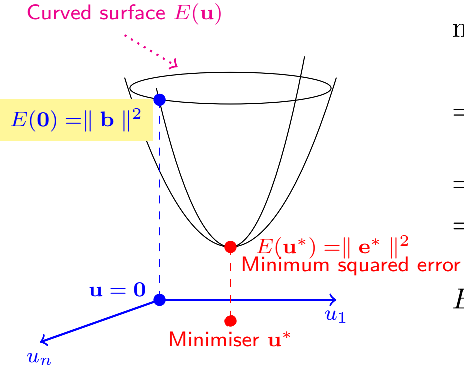


$p_0 = 2;\ p_1 = 6;\ p_3 = 14;\ p_4 = 18 \ \rightarrow \ E(\mathbf{u}^*) = 1 + 9 + 25 + 9 = 44$

Least Squares by Calculus (optional)

Setting to zero the derivative by $\mathbf{u}$ of the squared error:

$$E(\mathbf{u}) = \| \mathbf{e} \|^2 = (\mathbf{b} - \mathbf{A}\mathbf{u})^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{u}) = \mathbf{u}^\mathsf{T} \underbrace{\mathbf{A}^\mathsf{T}\mathbf{A}}_{\mathbf{K}} \mathbf{u} - 2\mathbf{u}^\mathsf{T} \underbrace{\mathbf{A}^\mathsf{T}\mathbf{b}}_{\mathbf{f}} + \mathbf{b}^\mathsf{T}\mathbf{b}$$



$$\min_{\mathbf{u}} \left\{ \mathbf{u}^\mathsf{T}\mathbf{K}\mathbf{u} - 2\mathbf{u}^\mathsf{T}\mathbf{f} \right\}$$

$$\Rightarrow \frac{\partial}{\partial \mathbf{u}} \left(\mathbf{u}^\mathsf{T}\mathbf{K}\mathbf{u} - 2\mathbf{u}^\mathsf{T}\mathbf{f} \right) = \mathbf{0}$$

$$\Rightarrow \mathbf{K}\mathbf{u} = \mathbf{f} \Rightarrow \mathbf{u}^* = \mathbf{K}^{-1}\mathbf{f}$$

$$\Rightarrow \mathbf{u}^* = \left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}\mathbf{b}$$

$$E(\mathbf{u}^*) = (\mathbf{b} - \mathbf{A}\mathbf{u}^*)^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{u}^*) = \| \mathbf{b} \|^2 - \| \mathbf{A}\mathbf{u}^* \|^2$$

Least Squares by Linear Algebra (optional)

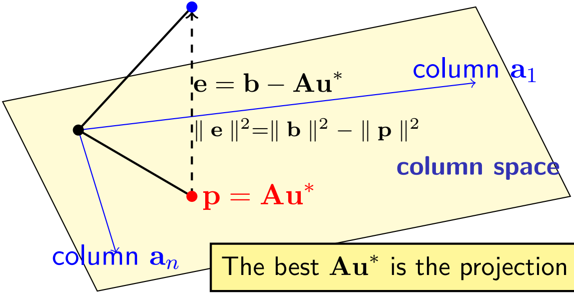
Impossible equation $\mathbf{Au} = \mathbf{b}$:

- An attempt to represent $\mathbf{b}$ in m-dimensional space with a linear combination of the n columns of $\mathbf{A}$
- But those columns only give an n-dimensional plane inside the much larger m-dimensional space
- Vector $\mathbf{b}$ is unlikely to lie in that plane, so $\mathbf{Au} = \mathbf{b}$ is unlikely to be solvable

Least Squares by Linear Algebra (optional)

- The vector $\mathbf{Au}^*$ is the nearest to the $\mathbf{b}$ point in the plane
- Error vector $\mathbf{e}$ is orthogonal to the plane (column space):

$$\begin{aligned} \text{Column 1:} &\quad \mathbf{a}_1^\mathsf{T}\mathbf{e} = 0 \\ \text{Column 2:} &\quad \mathbf{a}_2^\mathsf{T}\mathbf{e} = 0 \\ \ldots \to &\quad \mathbf{A}^\mathsf{T}\mathbf{e} = \mathbf{0} \end{aligned}$$



The best $\mathbf{Au}^*$ is the projection $\mathbf{p}$

Least Squares by Linear Algebra (optional)

Error vector $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{u}^*$ is perpendicular to the column space:

$$\overbrace{\begin{bmatrix} (\textbf{column } 1)^\mathsf{T} \\ \vdots \\ (\textbf{column } n)^\mathsf{T} \end{bmatrix}}^{\mathbf{A}^\mathsf{T}} \mathbf{e} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \Rightarrow \mathbf{A}^\mathsf{T}\mathbf{e} = \mathbf{0}$$

- This geometric equation $\mathbf{A}^\mathsf{T}\mathbf{e} = \mathbf{0}$ finds $\mathbf{u}^*$: the projection is $\mathbf{p} = \mathbf{A}\mathbf{u}^*$ (the combination of columns that is closest to $\mathbf{b}$)
- It gives again the normal equation for $\mathbf{u}^*$:

$$\mathbf{A}^\mathsf{T}\mathbf{e} = \mathbf{A}^\mathsf{T}(\mathbf{b} - \mathbf{A}\mathbf{u}^*) = \mathbf{0} \;\Rightarrow\; \mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{u}^* = \mathbf{A}^\mathsf{T}\mathbf{b}$$

Changing from the minimum in calculus to the projection in linear algebra gives the right triangle with sides $\mathbf{b}$, $\mathbf{p}$, and $\mathbf{e}$

Least Squares by Linear Algebra (optional)

- The perpendicular error vector $\mathbf{e}$ hits the column space in the nearest to $\mathbf{b}$ point $\mathbf{p} = \mathbf{A}\mathbf{u}^*$ where $\mathbf{u}^* = \left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}\mathbf{b}$
- $\mathbf{p}$ is the projection of $\mathbf{b}$ onto the column space:

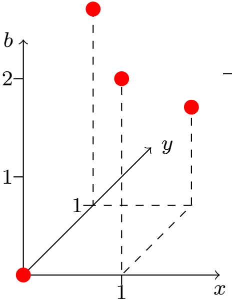
$$\mathbf{p} = \mathbf{A}\mathbf{u}^* = \underbrace{\left[\mathbf{A}\left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}\right]}_{\text{projection matrix } \mathbf{P}}\mathbf{b} \equiv \mathbf{P}\mathbf{b}$$

  - $\mathbf{A}\mathbf{u} = \mathbf{b}$ has no solution, but $\mathbf{A}\mathbf{u} = \mathbf{p}$ has one solution $\mathbf{u}^*$
    - The smallest adjustment $\mathbf{b} \rightarrow \mathbf{p}$ to be in the column space
    - Measurements are inconsistent in $\mathbf{A}\mathbf{u} = \mathbf{b}$, but consistent in $\mathbf{A}\mathbf{u}^* = \mathbf{p}$
- Projection matrix $\mathbf{P} = \mathbf{A}\left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}$ is symmetric
  - $\mathbf{P}^2 = \mathbf{P}$ as repeated projections give the same result
  - $\mathbf{P}$ is $m \times m$ but only of rank n (as all its factors have rank n)

Least Squares: Example 2

The closest plane through 4 points in (x, y, b) space: $b = C + Dx + Ey$:

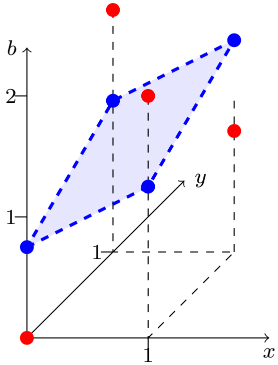
$$\left.\begin{array}{l} C + Dx_1 + Ey_1 = b_1 \\ C + Dx_2 + Ey_2 = b_2 \\ C + Dx_3 + Ey_3 = b_3 \\ C + Dx_4 + Ey_4 = b_4 \end{array}\right\} \Rightarrow \underbrace{\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \\ 1 & x_4 & y_4 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} C \\ D \\ E \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}}_{\mathbf{b}} \rightarrow \left\{\begin{array}{l} \mathbf{A}^\top \mathbf{A} \mathbf{u}^* = \mathbf{A}^\top \mathbf{b} \\ \mathbf{u}^* \equiv \begin{bmatrix} C^* \\ D^* \\ E^* \end{bmatrix} \\ = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} \end{array}\right.$$



x	y	b
0	0	0
0	1	2
1	0	2
1	1	1

$$\Rightarrow \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}; \ \mathbf{b} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

Least Squares: Example 2 (cont.)



$$\mathbf{A}^\mathsf{T}\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

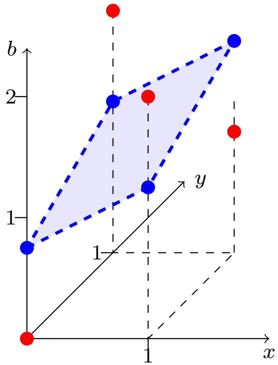
$$\Rightarrow (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1} = \begin{bmatrix} \frac{3}{4} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix}$$

$$\mathbf{u}^* \equiv \begin{bmatrix} C^* \\ D^* \\ E^* \end{bmatrix} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{b}$$

$$= \begin{bmatrix} \frac{3}{4} & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{3}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{4} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$$

Least Squares: Example 2 (cont.)



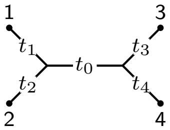
$$\mathbf{u}^* = \begin{bmatrix} \frac{3}{4} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \Rightarrow b^*_{x,y} = \frac{3}{4} + \frac{1}{4}x + \frac{1}{4}y$$

$$\mathbf{p} = \mathbf{A}\mathbf{u}^* = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{3}{4} \\ \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} \frac{3}{4} \\ \frac{5}{4} \\ \frac{5}{4} \\ \frac{7}{4} \end{bmatrix}$$

$$\mathbf{e} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{3}{4} \\ \frac{5}{4} \\ \frac{5}{4} \\ \frac{7}{4} \end{bmatrix} = \begin{bmatrix} -\frac{3}{4} \\ \frac{3}{4} \\ \frac{3}{4} \\ -\frac{3}{4} \end{bmatrix}$$

One More Example from Phylogenetics

Typical problem: Given n nodes and $m = \frac{(n-1)n}{2}$ inter-node distances d_{ij}, find $\nu = 2n - 3$ lengths t_i of tree branches



It is the least-squares problem: for $n = 4$, $m = 6$, and $\nu = 5$

$$\min_{\mathbf{t}=[t_0,\ldots,t_4]^\mathsf{T}} \{F(\mathbf{t}) = (t_1 + t_2 - d_{12})^2 + (t_1 + t_0 + t_3 - d_{13})^2 \\ +(t_1 + t_0 + t_4 - d_{14})^2 + (t_2 + t_0 + t_3 - d_{23})^2 \\ +(t_2 + t_0 + t_4 - d_{24})^2 + (t_3 + t_4 - d_{34})^2\}$$

- Normal equations $\nabla F(\mathbf{t}) = \mathbf{0} \Rightarrow$

$$\begin{bmatrix} 4 & 2 & 2 & 2 & 2 \\ 2 & 3 & 1 & 1 & 1 \\ 2 & 1 & 3 & 1 & 1 \\ 2 & 1 & 1 & 3 & 1 \\ 2 & 1 & 1 & 1 & 3 \end{bmatrix} \begin{bmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} d_{12} \\ d_{13} \\ d_{14} \\ d_{23} \\ d_{24} \\ d_{34} \end{bmatrix}$$

- Solution:

$$\begin{bmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} = \begin{bmatrix} -0.5 & 0.25 & 0.25 & 0.25 & 0.25 & -0.5 \\ 0.5 & 0.25 & 0.25 & -0.25 & -0.25 & 0 \\ 0.5 & -0.25 & -0.25 & 0.25 & 0.25 & 0 \\ 0 & 0.25 & -0.25 & 0.25 & -0.25 & 0.5 \\ 0 & -0.25 & 0.25 & -0.25 & 0.25 & 0.5 \end{bmatrix} \begin{bmatrix} d_{12} \\ d_{13} \\ d_{14} \\ d_{23} \\ d_{24} \\ d_{34} \end{bmatrix}$$

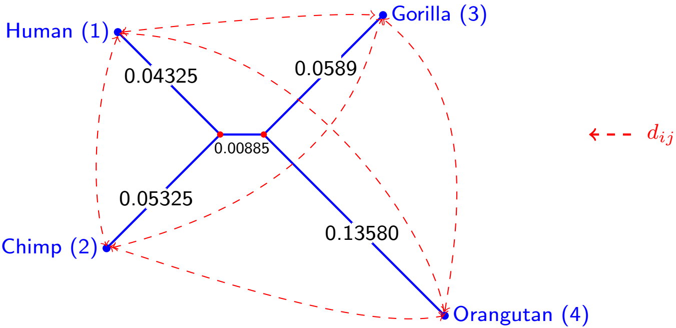
One More Example from Phylogenetics (cont.)

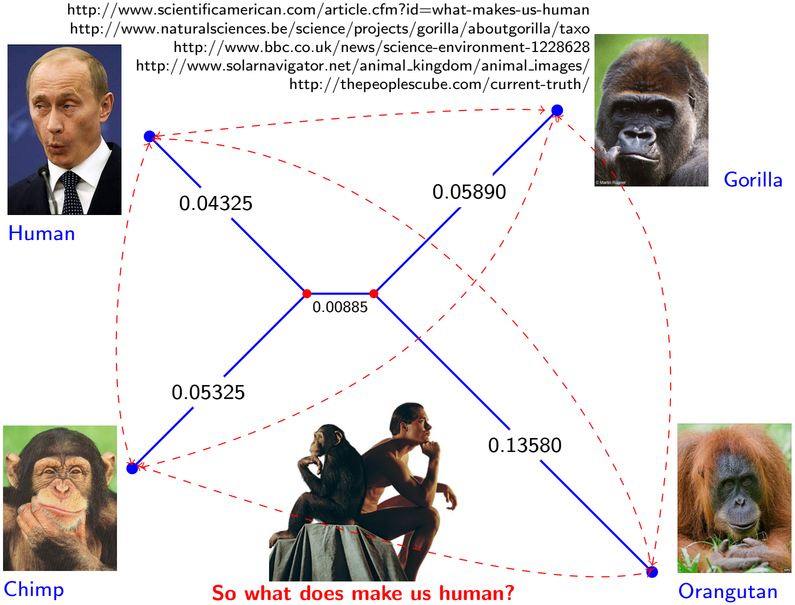
Distances d_{ij}:

	$i = 1_{\text{human}}$	$i = 2_{\text{chimp}}$	$i = 3_{\text{gorilla}}$
$j = 2_{\text{chimp}}$	0.0965		
$j = 3_{\text{gorilla}}$	0.1140	0.1180	
$j = 4_{\text{orangutan}}$	0.1849	0.2009	0.1947

Tree branch lengths found:

t_0	t_1	t_2	t_3	t_4
0.00885	0.04325	0.05325	0.05890	0.13580





http://www.scientificamerican.com/article.cfm?id=what-makes-us-human
http://www.naturalsciences.be/science/projects/gorilla/aboutgorilla/taxo
http://www.bbc.co.uk/news/science-environment-1228628
http://www.solarnavigator.net/animal_kingdom/animal_images/
http://thepeoplescube.com/current-truth/
Human
Gorilla
0.04325
0.05890
0.00885
0.05325
0.13580
Chimp
Orangutan
So what does make us human?

How to Compute the Least Squares Solution $\mathbf{u}^*$?

1) Solving the normal equations by the Gaussian **elimination**

- The elimination $\mathbf{S} = \mathbf{LU}$ () reduces a square matrix $\mathbf{S}$ to an upper triangular matrix $\mathbf{U}$ by using row operations, such that their multipliers form a lower triangular matrix $\mathbf{L}$
- Note that elimination of $\mathbf{S} = \mathbf{A}^\mathsf{T}\mathbf{A}$ may be very unstable!
- Why? – Because the condition number of $\mathbf{A}^\mathsf{T}\mathbf{A}$ is the square of the condition number of $\mathbf{A}$
  - **Condition number** of a positive definite matrix $\mathbf{K}$ is the ratio of its max and min eigenvalues $\frac{\lambda_{\max}(\mathbf{K})}{\lambda_{\min}(\mathbf{K})}$
- Condition number measures sensitivity of a linear system
- The larger the number, the lesser the system's stability...

How to Compute the Least Squares Solution $\mathbf{u}^*$?

2) **Orthogonalization** $\mathbf{A} = \mathbf{QR}$, when stability is in doubt

- $\mathbf{Q}$ is an $m \times n$ matrix with n orthonormal columns ( )
- $\mathbf{R}$ is an $n \times n$ upper triangular matrix ( )
- This factoring reduces the normal equation $\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{u} = \mathbf{A}^\mathsf{T}\mathbf{b}$ to a much simpler one:

$$(\mathbf{QR})^\mathsf{T}\mathbf{QR}\mathbf{u}^* = (\mathbf{QR})^\mathsf{T}\mathbf{b} \Rightarrow \mathbf{R}^\mathsf{T}\underbrace{\mathbf{Q}^\mathsf{T}\mathbf{Q}}_{\mathbf{I}}\mathbf{R}\mathbf{u}^* = \mathbf{R}^\mathsf{T}\mathbf{Q}^\mathsf{T}\mathbf{b}$$

$$\Rightarrow \mathbf{R}^\mathsf{T}\mathbf{R}\mathbf{u}^* = \mathbf{R}^\mathsf{T}\mathbf{Q}^\mathsf{T}\mathbf{b} \Rightarrow \mathbf{R}\mathbf{u}^* = \mathbf{Q}^\mathsf{T}\mathbf{b}$$

- Multiplication $\mathbf{Q}^\mathsf{T}\mathbf{b}$ is very stable
- Back-substitution with the upper triangular $\mathbf{R}$ is very simple
- Producing $\mathbf{Q}$ and $\mathbf{R}$ takes twice as long as the mn^2 steps to form $\mathbf{A}^\mathsf{T}\mathbf{A}$, but that extra cost gives a more reliable solution!

Modified Gram-Schmidt orthogonalisation

Orthonormal columns $\mathbf{q}_1, \ldots, \mathbf{q}_n$ of $\mathbf{Q}$: sequential computation from the columns $\mathbf{a}_1, \ldots, \mathbf{a}_n$ of $\mathbf{A}$

$$\mathbf{q}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \Leftarrow \mathbf{v}_1 = \mathbf{a}_1$$

$$\mathbf{q}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} \Leftarrow \mathbf{v}_2 = \mathbf{a}_2 - \left(\mathbf{a}_2^\mathsf{T}\mathbf{q}_1\right)\mathbf{q}_1$$

$$\ldots$$

$$\mathbf{q}_j = \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|} \Leftarrow \mathbf{v}_j = \mathbf{a}_j - \sum_{i=1}^{j-1}\left(\mathbf{a}_j^\mathsf{T}\mathbf{q}_i\right)\mathbf{q}_i$$

$$\ldots$$

$$\mathbf{q}_n = \frac{\mathbf{v}_n}{\|\mathbf{v}_n\|} \Leftarrow \mathbf{v}_n = \mathbf{a}_n - \sum_{i=1}^{n-1}\left(\mathbf{a}_j^\mathsf{T}\mathbf{q}_i\right)\mathbf{q}_i$$

Example: Orthogonalisation $\mathbf{A} = \mathbf{QR}$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}; \quad \mathbf{q}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$$\mathbf{v}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} - \underbrace{\left(\begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \right)}_{=1} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix};$$

$$\mathbf{q}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

Example: Orthogonalisation $\mathbf{A} = \mathbf{QR}$ (cont.)

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow \mathbf{q}_1 = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix};\ \mathbf{q}_2 = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix};\ \mathbf{q}_3 = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$\mathbf{v}_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} - \underbrace{\left(\begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \right)}_{=1} \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{bmatrix}$$

$$- \underbrace{\left(\begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \right)}_{=0 \text{ by a pure chance!}} \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

Example: Orthogonalisation $\mathbf{A} = \mathbf{QR}$ (cont.)

- Column-orthonormal matrix $\mathbf{Q} = \frac{1}{2}\begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix}$
- Upper triangular matrix $\mathbf{R} = \mathbf{Q}^\mathsf{T}\mathbf{A}$

$$= \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

- $\overbrace{\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}}^{\mathbf{A}} = \overbrace{\frac{1}{2}\begin{bmatrix} 1 & -1 & 1 \\ 1 & -1 & -1 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix}}^{\mathbf{Q}}\overbrace{\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}}^{\mathbf{R}}$

How to Compute the Least Squares Solution $\mathbf{u}^*$?

3) **Singular Value Decomposition** (SVD): $\underbrace{\mathbf{A}}_{m\times n} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T}$

- $\mathbf{U}$ – a column-orthonormal $n \times m$ matrix: $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_n$
- $\mathbf{D} = \mathrm{diag}\{\sigma_1, \ldots, \sigma_n\}$ – a diagonal $n \times n$ matrix of singular values: $\mathbf{D}^\mathsf{T} = \mathbf{D}$
- $\mathbf{V}$ – an orthonormal $n \times n$ matrix: $\mathbf{V}^\mathsf{T} = \mathbf{V}^{-1}$; $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}_n$
- $\mathbf{A}^\mathsf{T}\mathbf{A} = \mathbf{V}\mathbf{D}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T} = \mathbf{V}\mathbf{D}^\mathsf{T}\mathbf{D}\mathbf{V}^\mathsf{T} = \mathbf{V}\mathbf{D}^2\mathbf{V}^\mathsf{T}$
- The most stable computation of $\mathbf{u}^*$!

$$\mathbf{V}\underline{\mathbf{D}^2\mathbf{V}^\mathsf{T}\mathbf{u}^*} = \mathbf{V}\underline{\mathbf{D}\mathbf{U}^\mathsf{T}\mathbf{b}} \;\Rightarrow\; \mathbf{D}^2\mathbf{V}^\mathsf{T}\mathbf{u}^* = \mathbf{D}\mathbf{U}^\mathsf{T}\mathbf{b}$$

$$\Rightarrow\; \mathbf{V}^\mathsf{T}\mathbf{u}^* = \overbrace{\left(\mathbf{D}^2\right)^{-1}\mathbf{D}}^{\mathbf{D}^+}\,\mathbf{U}^\mathsf{T}\mathbf{b} \;\Rightarrow\; \mathbf{u}^* = \mathbf{V}\mathbf{D}^+\mathbf{U}^\mathsf{T}\mathbf{b}$$

How to Compute the Least Squares Solution $\mathbf{u}^*$?

$\mathbf{u}^* = \mathbf{V}\mathbf{D}^+\mathbf{U}^\mathsf{T}\mathbf{b}$

- If $\mathrm{rank}(\mathbf{A}) = n$, i.e. all n singular values are non-zero: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$, then

$$\mathbf{D}^+ = \mathbf{D}^{-1} = \mathrm{diag}\left\{\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_n}\right\}$$

- $\mathbf{D}^+$, called the "**pseudoinverse**" of $\mathbf{D}$, in this case coincides with the inverse diagonal matrix $\mathbf{D}^{-1}$, so that $\mathbf{D}^+\mathbf{D} = \mathbf{I}_n$
- The matrix $\mathbf{A}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^\mathsf{T}$ is the pseudoinverse of $\mathbf{A}$: if $\mathrm{rank}(\mathbf{A}) = n$, then $\mathbf{A}^\mathsf{T}\mathbf{A} = \mathbf{V}\underline{\mathbf{D}^+\underline{\mathbf{U}^\mathsf{T}\mathbf{U}}\mathbf{D}}\mathbf{V}^\mathsf{T} = \mathbf{I}_n$
- Singular values specify stability: the matrix $\mathbf{A}^\mathsf{T}\mathbf{A}$ is ill-conditioned when σ_n is very small
- Extremely small singular values can be removed!

Pseudoinverse

SVD $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\mathsf{T} \Longrightarrow \mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{D}$ or $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$

- If $\mathbf{A}$ is a square matrix such that $\mathbf{A}^{-1}$ exists, then the singular values for $\mathbf{A}^{-1}$ are $\sigma^{-1} = \frac{1}{\sigma}$ and $\mathbf{A}^{-1}\mathbf{u}_i = \frac{1}{\sigma_i}\mathbf{v}_i$
- If $\mathbf{A}^{-1}$ does not exist, then the pseudoinverse matrix $\mathbf{A}^+$ does exist such that:

$$\mathbf{A}^+\mathbf{u}_i = \begin{cases} \frac{1}{\sigma_i}\mathbf{v}_i & \text{if} \quad i \le r = \operatorname{rank}(\mathbf{A}) \text{ i.e. if } \sigma_i > 0 \\ 0 & \text{for} \quad i > r \end{cases}$$

Pseudoinverse $\mathbf{A}^+$ of a matrix $\mathbf{A}$: $\mathbf{A}^+ = \mathbf{V}\mathbf{D}^+\mathbf{U}^\mathsf{T}$

- $\mathbf{D}^+ = \operatorname{diag}\left\{\sigma_1^+, \ldots, \sigma_n^+\right\}$ where

$$\sigma_i^+ = \begin{cases} \sigma_i^{-1} = \frac{1}{\sigma_i} & \text{if} \quad \sigma_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Pseudoinverse: Basic Properties

- Pseudoinverse matrix $\mathbf{A}^{+}$ has the same rank r as $\mathbf{A}$
- Pseudoinverse $\mathbf{D}^{+}$ of the diagonal matrix $\mathbf{D}$:
  each positive singular value $\sigma > 0$ is replaced by $\frac{1}{\sigma}$, and zero singular values remain unchanged
- Product $\mathbf{D}^{+}\mathbf{D}$ is as near to the identity matrix as possible
- The matrices $\mathbf{A}\mathbf{A}^{+}$ and $\mathbf{A}^{+}\mathbf{A}$ are also as near as possible to the $m \times m$ and $n \times n$ identity matrices, respectively
- $\mathbf{A}\mathbf{A}^{+}$ – the $m \times m$ projection matrix onto the column space of $\mathbf{A}$
- $\mathbf{A}^{+}\mathbf{A}$ – the $n \times n$ projection matrix onto the row space of $\mathbf{A}$

Pseudoinverse $\mathbf{A}^+$: Example 1

$$\underset{\text{rank } r=2}{\mathbf{A}} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \overset{\text{SVD}}{\Longrightarrow} \mathbf{A} = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{V}^\top}$$

$$\mathbf{A}^+ = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{3}} & 0 \\ 0 & 1 \end{bmatrix}}_{\mathbf{D}^+} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}}_{\mathbf{U}^\top} = \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \end{bmatrix}$$

$$\mathbf{A}\mathbf{A}^+ = \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

$$\mathbf{A}^+\mathbf{A} = \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Pseudoinverse $\mathbf{A}^+$: Example 2

$$\underset{\text{rank } r=1}{\mathbf{A}} = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \overset{\text{SVD}}{\Longrightarrow} \begin{cases} \mathbf{A}\mathbf{A}^\mathsf{T} = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix} \rightarrow & \lambda_1 = 10;\ \mathbf{u}_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ & \lambda_2 = 0;\ \mathbf{u}_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ \mathbf{A}^\mathsf{T}\mathbf{A} = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix} \rightarrow & \lambda_1 = 10;\ \mathbf{v}_1 = \frac{1}{\sqrt{5}}\begin{bmatrix} 1 \\ 2 \end{bmatrix} \\ & \lambda_2 = 0;\ \mathbf{v}_2 = \frac{1}{\sqrt{5}}\begin{bmatrix} 2 \\ -1 \end{bmatrix} \end{cases}$$

$$\overset{\text{SVD}}{\Longrightarrow} \mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j;\quad j = 1,2 \quad \Longrightarrow \quad \sigma_1 = \sqrt{10};\ \sigma_2 = 0$$

$$\overset{\text{SVD}}{\Longrightarrow} \mathbf{A} = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \sqrt{10} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}}_{\mathbf{V}^\mathsf{T}}$$

Pseudoinverse $\mathbf{A}^+$: Example 2 (cont.)

$$\text{SVD}\ \mathbf{A} = \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{U}} \underbrace{\begin{bmatrix} \sqrt{10} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{D}} \underbrace{\frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}}_{\mathbf{V}^\mathsf{T}}$$

$$\mathbf{A}^+ = \underbrace{\frac{1}{\sqrt{5}}\begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{10}} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathbf{D}^+} \underbrace{\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}_{\mathbf{U}^\mathsf{T}} = \frac{1}{10}\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}$$

$$\mathbf{A}\mathbf{A}^+ = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \frac{1}{10}\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

$$\mathbf{A}^+\mathbf{A} = \frac{1}{10}\begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 0.2 & 0.4 \\ 0.4 & 0.8 \end{bmatrix}$$

Weighted Least Squares (optional)

A small but important extension of the least squares problem

- The same rectangular $\mathbf{A}$ and a square weighting matrix $\mathbf{W}$

Minimise $\|\, \mathbf{W}\,(\mathbf{A}\mathbf{u} - \mathbf{b})\,\|^2 \Leftrightarrow$ Minimise $\|\, (\mathbf{W}\mathbf{A})\,\mathbf{u} - (\mathbf{W}\mathbf{b})\,\|^2$

Normal equations for $\mathbf{u}^*$: $(\mathbf{W}\mathbf{A})^\mathsf{T}(\mathbf{W}\mathbf{A})\mathbf{u}^* = (\mathbf{W}\mathbf{A})^\mathsf{T}(\mathbf{W}\mathbf{b})$, or

$$\mathbf{A}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{A}\mathbf{u}^* = \mathbf{A}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{b}$$

No new math: just replace $\mathbf{A}$ and $\mathbf{b}$ by $\mathbf{W}\mathbf{A}$ and $\mathbf{W}\mathbf{b}$

- Symmetric positive definite combination matrix $\mathbf{C} = \mathbf{W}^\mathsf{T}\mathbf{W}$
  $\Rightarrow \mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{A}\mathbf{u}^* = \mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{b} \Rightarrow \mathbf{A}^\mathsf{T}\mathbf{C}\,(\mathbf{b} - \mathbf{A}\mathbf{u}^*) = \mathbf{0}$

Random measurement errors (noise) $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{u}$:

- Equation system to be solved: $\mathbf{A}\mathbf{u} = \mathbf{b} - \mathbf{e}$
- Expected error $\mathbb{E}[e_i] = 0$
- Error variance $\sigma_i^2 = \mathbb{E}[e_i^2] > 0$

Weighted Least Squares

Independent errors e_i with equal variances $\sigma_i^2 = \sigma^2$:

- Non-weighted least squares: $\mathbf{C} = \mathbf{I}$
- Minimising just $\mathbf{e}^\mathsf{T}\mathbf{e}$

Independent errors e_i with different variances σ_i^2:

- The smaller the σ_i^2, the more reliable the measurement b_i and the higher the weight of that equation ($\mathbf{C} = \mathrm{diag}\{\frac{1}{\sigma_1^2}, \ldots, \frac{1}{\sigma_n^2}\}$
- Minimising $\mathbf{e}^\mathsf{T}\mathbf{C}\mathbf{e}$

Interdependent errors e_i:

- "Covariances" $\sigma_{ij} \equiv \sigma_{ji} = \mathbb{E}[e_i e_j]$ also enter **the inverse** of $\mathbf{C}$:

$$\mathbf{C}^{-1} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \ldots & \sigma_{1n} \\ \sigma_{21} & \sigma_2^2 & \ldots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \ldots & \sigma_n^2 \end{bmatrix}$$

Weighted Least Squares: Probability Model

- The best $\mathbf{u}^*$ accounting for weights: from $\mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{A}\mathbf{u}^* = \mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{b}$
- How reliable is $\mathbf{u}^* = \left(\mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{b}$ comes from the **matrix of variances and covariances** $\left(\mathbf{A}^\mathsf{T}\mathbf{C}\mathbf{A}\right)^{-1}$ in the $\mathbf{u}^*$

Interdependent Gaussian errors:

$$p(\mathbf{u}|\mathbf{A},\mathbf{b},\mathbf{S}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{S}|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(\mathbf{A}\mathbf{u}-\mathbf{b})^\mathsf{T}\mathbf{S}^{-1}(\mathbf{A}\mathbf{u}-\mathbf{b})\right)$$

- Maximum probable $\mathbf{u}^\circ = \arg\max\limits_{\mathbf{u}} p(\mathbf{u}|\mathbf{A},\mathbf{b},\mathbf{S})$ - from $\mathbf{A}^\mathsf{T}\mathbf{S}^{-1}\mathbf{A}\mathbf{u}^\circ = \mathbf{A}^\mathsf{T}\mathbf{S}^{-1}\mathbf{b}$
- $\mathbf{S} = \mathbf{C}^{-1}$ – the covariance matrix $\mathbb{E}[\mathbf{e}\mathbf{e}^\mathsf{T}]$
- The weighted LS solution $\mathbf{u}^*$ coincides with the maximum probable one $\mathbf{u}^\circ$ under such weights

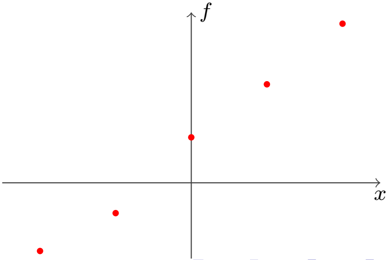
Least Squares for Regression (optional)

Linear regression

Given a data set $\{(x_i, f_i) : i = 1, \ldots, n\}$, find a linear function $f(x) = a + bx$ minimising the sum of squared deviations

$$L(a,b) = \sum_{i=1}^{n} (f_i - f(x_i))^2 \equiv \sum_{i=1}^{n} (f_i - (a + bx_i))^2$$

- Search for the minimiser (a^*, b^*) of the function $L(a, b)$ depending on parameters a and b
  - a – an f-axis segment
  - b – a slope of the line



Linear Regression

Normal equations for the minimiser:

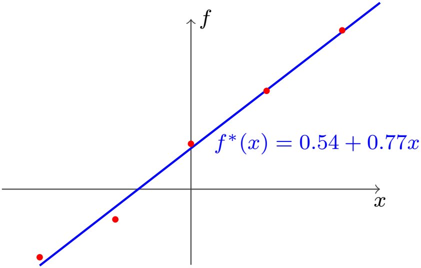
$$\begin{cases} \frac{\partial L(a,b)}{\partial a} = -2\sum_{i=1}^{n}(f_i - (a + bx_i)) = 0 \\ \frac{\partial L(a,b)}{\partial b} = -2\sum_{i=1}^{n}(f_i - (a + bx_i))x_i = 0 \end{cases}$$

$$\Rightarrow \begin{cases} an + b\overbrace{\sum_{i=1}^{n} x_i}^{S_x} = \overbrace{\sum_{i=1}^{n} f_i}^{S_f} \\ a\underbrace{\sum_{i=1}^{n} x_i}_{S_x} + b\underbrace{\sum_{i=1}^{n} x_i^2}_{S_{xx}} = \underbrace{\sum_{i=1}^{n} f_i x_i}_{S_{fx}} \end{cases} \Rightarrow \begin{bmatrix} n & S_x \\ S_x & S_{xx} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} S_f \\ S_{fx} \end{bmatrix}$$

Linear Regression

Least squares solution:

$$\begin{bmatrix} a^* \\ b^* \end{bmatrix} = \frac{1}{nS_{xx} - S_x^2} \begin{bmatrix} S_{xx} & -S_x \\ -S_x & n \end{bmatrix} \begin{bmatrix} S_f \\ S_{fx} \end{bmatrix} \Rightarrow \begin{cases} a^* = \frac{S_{xx}S_f - S_x S_{fx}}{nS_{xx} - S_x^2} \\ b^* = \frac{-S_x S_f + nS_{fx}}{nS_{xx} - S_x^2} \end{cases}$$



i	1	2	3	4	5
x_i	-2	-1	0	1	2
f_i	-0.9	-0.4	0.6	1.3	2.1

$$S_x = 0; \; S_{xx} = 10; \; S_f = 2.7; \; S_{fx} = 7.7$$

$$\begin{cases} a^* = \frac{10 \cdot 2.7 - 0 \cdot 7.7}{5 \cdot 10 - 0^2} = \frac{27}{50} = 0.54 \\ b^* = \frac{-0 \cdot 2.7 + 5 \cdot 7.7}{5 \cdot 10 - 0^2} = \frac{38.5}{50} = 0.77 \end{cases}$$

Linear Regression

Residual sum of squared deviations:

$$L(a^*, b^*) = \underbrace{\sum_{i=1}^{n} f_i^2}_{S_{ff}} - \frac{S_f^2 S_{xx} - 2S_f S_x S_{fx} + nS_{fx}^2}{nS_{xx} - S_x^2}$$

Example: $L(0.54, 0.77) = 7.43 - \frac{2.7^2 \cdot 10 - 2 \cdot 2.7 \cdot 0 \cdot 7.7 + 5 \cdot 7.7^2}{5 \cdot 10 - 0^2} = 7.43 - 7.387 = 0.043$

i	1	2	3	4	5
x_i	-2	-1	0	1	2
f_i	-0.9	-0.4	0.6	1.3	2.1
$f^*(x_i)$	-1.00	-0.23	0.54	1.31	2.08
$f_i - f^*(x_i)$	0.10	-0.17	0.06	-0.01	0.02

$$S_x = 0;\ S_{xx} = 10;\ S_f = 2.7;\ S_{fx} = 7.7;\ S_{ff} = 7.43$$

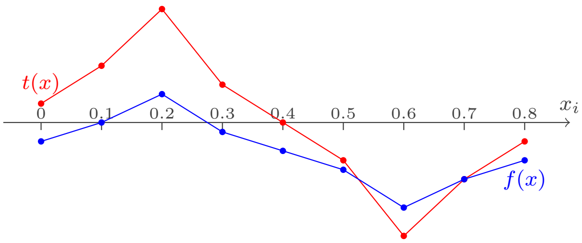
Example: $L(0.54, 0.77) = 0.10^2 + (-0.17)^2 + 0.06^2 + (-0.01)^2 + 0.02^2 = 0.043$

Correlation Matching (optional)

Least squares: 1D signals; constant contrast a and offset b

Given time or spatial data series $\{(t_i = t(x_i), f_i = f(x_i)) :\ i = 1, \ldots, n;\ x_1 < \ldots < x_n\}$, find a "contrast – offset", $f(x) = a + bt(x)$, transformation minimising the sum of squared deviations

$$L(a,b) = \sum_{i=1}^{n} \left(f(x_i) - (a + bt(x_i))\right)^2 \equiv \sum_{i=1}^{n} \left(f_i - (a + bt_i)\right)^2$$



i	x_i	t_i	f_i
1	0	0.5	−0.50
2	0.1	1.5	0.00
3	0.2	3.0	0.75
4	0.3	1.0	−0.25
5	0.4	0.0	−0.75
6	0.5	−1.0	−1.25
7	0.6	−3.0	−2.25
8	0.7	−1.5	−1.50
9	0.8	−0.5	−1.00

Correlation Matching, continued

The minimiser (a^*, b^*) for the matching score $L(a,b)$ is obtained similarly to the linear regression:

- Normal equations:

$$\frac{\partial L}{\partial a} = 0; \ \frac{\partial L}{\partial b} = 0 \ \Rightarrow \begin{bmatrix} n & S_t \\ S_t & S_{tt} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} S_f \\ S_{ft} \end{bmatrix}$$

where $S_t = \sum_{i=1}^{n} t_i$; $S_{tt} = \sum_{i=1}^{n} t_i^2$; $S_f = \sum_{i=1}^{n} f_i$; $S_{ft} = \sum_{i=1}^{n} f_i t_i$

- Solution: $\begin{bmatrix} a^* \\ b^* \end{bmatrix} = \frac{1}{nS_{tt} - S_t^2} \begin{bmatrix} S_{tt} & -S_t \\ -S_t & n \end{bmatrix} \begin{bmatrix} S_f \\ S_{ft} \end{bmatrix}$

$$a^* = \frac{1}{nS_{tt} - S_t^2} \left(S_{tt} S_f - S_t S_{ft}\right); \ b^* = \frac{1}{nS_{tt} - S_t^2} \left(-S_t S_f + n S_{ft}\right)$$

$$\Rightarrow a^* = \frac{S_f}{n} - b^* \cdot \frac{S_t}{n}; \ b^* = \frac{1}{nS_{tt} - S_t^2} \left(-S_t S_f + n S_{ft}\right)$$

$$\Rightarrow f^*(x) = \frac{S_f}{n} + b^* \cdot \left(t(x) - \frac{S_t}{n}\right)$$

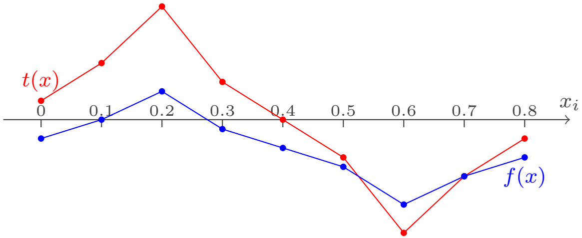
Correlation Matching, continued

Minimum sum of squared deviations ($\bar{f} = \frac{S_f}{n}$; $\bar{t} = \frac{S_t}{n}$ – mean signals):

$$L(a^*, b^*) = \sum_{i=1}^{n} \left(f(x_i) - \bar{f}\right)^2 - \frac{\left(\sum\limits_{i=1}^{n} \left(f(x_i) - \bar{f}\right)\left(t(x_i) - \bar{t}\right)\right)^2}{\sum\limits_{i=1}^{n} \left(t(x_i) - \bar{t}\right)^2}$$

- Signal variances: $\sigma_f^2 = \frac{1}{n} \sum\limits_{i=1}^{n} \left(f(x_i) - \bar{f}\right)^2$, $\sigma_t^2 = \frac{1}{n} \sum\limits_{i=1}^{n} \left(t(x_i) - \bar{t}\right)^2$
- Signal covariance: $\sigma_{ft} = \frac{1}{n} \sum\limits_{i=1}^{n} \left(f(x_i) - \bar{f}\right)\left(t(x_i) - \bar{t}\right)$
- Correlation (matching score): $C_{ft} = \frac{\sigma_{ft}}{\sigma_f \sigma_t}$; $-1 \leq C_{ft} \leq 1$
- Matching distance: $D_{ft}^* \equiv L(a^*, b^*) = n\sigma_f^2 \left(1 - C_{ft}^2\right)$

Correlation Matching: An Example



i	x_i	t_i	f_i
1	0	0.5	−0.50
2	0.1	1.5	0.00
3	0.2	3.0	0.75
4	0.3	1.0	−0.25
5	0.4	0.0	−0.75
6	0.5	−1.0	−1.25
7	0.6	−3.0	−2.25
8	0.7	−1.5	−1.50
9	0.8	−0.5	−1.00

$$S_t = 0;\ S_{tt} = 25;\ S_f = -6.75;\ S_{ff} = 11.3125;\ S_{ft} = 12.5 \Rightarrow$$

$$b^* = \frac{1}{9\cdot 25 - 0^2}\left(-0\cdot(-6.75) + 9\cdot 12.5\right) = 0.5;\ a^* = \frac{-6.75}{9} - 0.5\cdot\frac{0}{9} = -0.75$$

$$\Rightarrow f(x) = -0.75 + 0.5\cdot t(x);\ \bar{f} = -0.75;\ \sigma_f^2 = \frac{6.25}{9};\ \sigma_t^2 = \frac{25}{9};\ \sigma_{ft} = \frac{12.5}{9}$$

$$\Rightarrow C_{ft} = \frac{\frac{25}{9}}{\frac{2.5}{3}\frac{5}{3}} = 1;\ D_{ft}^* = 9\,\frac{6.25}{9}(1 - 1^2) = 0$$

Probability Model of Matching Signals

Correlation matching follows from a simple probability model of f:

- Transformed template t corrupted by a centred independent random Gaussian noise r: for $i = 1, \ldots, n$,

$$f_i = a + bt_i + r_i \Rightarrow p(r_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(f_i - (a+bt_i))^2}{2\sigma^2}\right)$$

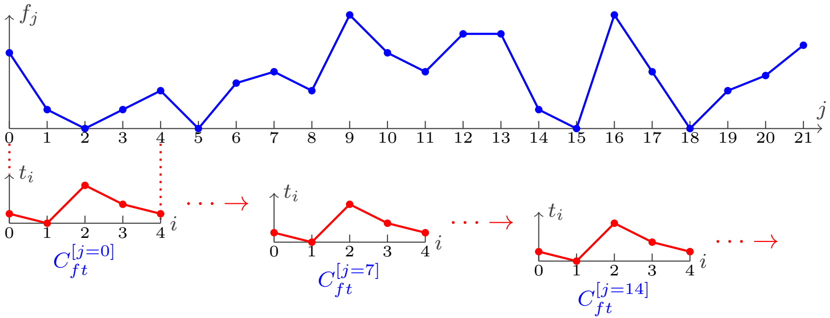
$$\Rightarrow P_{a,b}(f|t) = \prod_{i=1}^{n} p(r_i) = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp\left(-\frac{\sum_{i=1}^{n}(f_i - (a+bt_i))^2}{2\sigma^2}\right)$$

- Maximum likelihood between f and t by transforming parameters a and b results in the correlation matching:

$$\max_{a,b} P_{a,b}(f|t) \Rightarrow \min_{a,b} \sum_{i=1}^{n} (f_i - (a+bt_i))^2$$

Search for the Best Matching Position

- Matching a template $t = [t_i : i = 1, \ldots, n]$ to a much longer data sequence $f = [f_j : j = 1, \ldots, N]$; $N > n$
- Goal position j^* maximises the correlation C_{ft} (or minimises the distance D_{ft}) between t and the segment $[f_{j+i} : i = 1, \ldots, n]$ of f



2D Correlation: Constant Contrast–Offset

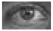
- 2D $m \times n$ template t and $M \times N$ image f; $m < M$; $n < N$:

$$\begin{array}{lcllll} t & = & [t_{i'j'}: & i' = 0, \ldots, n-1; & j' = 0, \ldots, m-1] \\ f & = & [f_{ij}: & i = 0, \ldots, N-1; & j = 0, \ldots, M-1] \end{array}$$

- An example:

  Eye template t 32×18 pixels:
  Facial image f 200×200 pixels:

  Moving window matching:
  Searching for a window position (i^*, j^*) in f such that the correlation C_{ft} (the distance D_{ft}) between the template t and the underlying region of the image f in the moving window is maximal (minimal)



2D correlation: Constant Contrast–Offset

Distance between the template t and the moving window in position (i, j) in the image f:

$$D_{ij} = \sum_{i'=0}^{n-1}\sum_{j'=0}^{m-1} \tilde{f}^2_{i+i',j+j'} - \frac{\left(\sum_{i'=0}^{n-1}\sum_{j'=0}^{m-1} \tilde{f}_{i+i',j+j'}\tilde{t}_{i',j'}\right)^2}{\sum_{i=1}^{n} \tilde{t}^2_{i',j'}}$$

- Centred signals: $\tilde{f}_{i+i',j+j'} = f_{i+i',j+j'} - \bar{f}_{[ij]}$ and $\tilde{t}_{i',j'} = t_{i',j'} - \bar{t}$
  - Mean for the moving window: $\bar{f}_{[ij]} = \frac{1}{mn}\sum_{i'=0}^{n-1}\sum_{j'=0}^{m-1} f_{i+1',j+j'}$
  - Variance for the moving window:

$$\sigma^2_{f:[ij]} = \frac{1}{mn}\sum_{i'=0}^{n-1}\sum_{j'=0}^{m-1}\left(f_{i+i',j+j'} - \bar{f}_{[ij]}\right)^2$$

2D correlation: Constant Contrast–Offset

- Fixed template mean: $\bar{t} = \frac{1}{mn} \sum_{i'=0}^{n-1} \sum_{j'=0}^{m-1} t_{i',j'}$
- Fixed template variance:

$$\sigma_t^2 = \frac{1}{mn} \sum_{i'=0}^{n-1} \sum_{j'=0}^{m-1} \left(t_{i',j'} - \bar{t}\right)^2$$

- Window–template covariance:

$$\sigma_{ft:[ij]} = \frac{1}{mn} \sum_{i'=0}^{n-1} \sum_{j'=0}^{m-1} \left(f_{i+i',j+j'} - \bar{f}_{[ij]}\right) \left(t_{i',j'} - \bar{t}\right)$$

- Correlation matching: $C_{ft:[ij]} = \frac{\sigma_{ft:[ij]}}{\sigma_{f:[ij]}\sigma_t}$; $-1 \leq C_{ft:[ij]} \leq 1$
  - Distance: $D^*_{ft:[ij]} \equiv L(a^*, b^*) = n\sigma^2_{f:[ij]} \left(1 - C^2_{ft:[ij]}\right)$