Mathematical Tools
for Neural and Cognitive Science

Fall semester, 2021

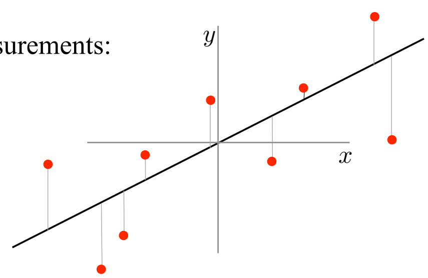
Section 2: Least Squares

Least squares regression:

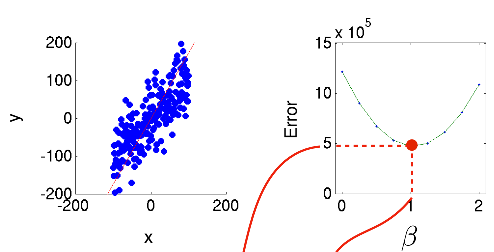
"objective" or "error" function

$$\min_{\beta} \sum_n (y_n - \beta x_n)^2$$

In the space of measurements:



[Gauss, 1795 - age 18]



$$\hat{\beta} = \arg\min_{\beta} \sum_n (y_n - \beta x_n)^2$$

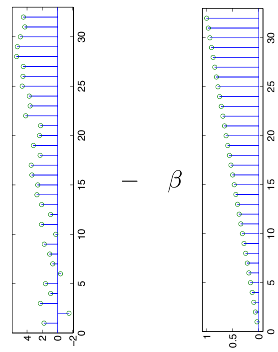
$$\min_{\beta} \sum_n (y_n - \beta x_n)^2$$

$$\min_{\beta} \sum_n (y_n - \beta x_n)^2$$

can solve this with calculus… *[on board]*

... or, with linear algebra!

$$\min_{\beta} ||\vec{y} - \beta\vec{x}||^2$$

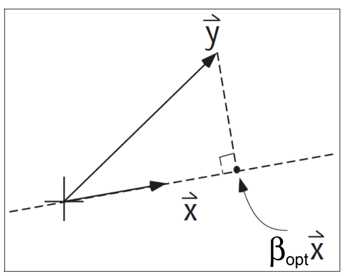


Observation $\vec{y}$ Regressor $\vec{x}$

$$\min_{\beta} ||\vec{y} - \beta\vec{x}||^2$$

Geometry:

Note: this is a 2-D cartoon of the N-D vectors, not the two-dimensional *(x,y)* measurement space of previous plots!



Note: partition of sum of squared data values:

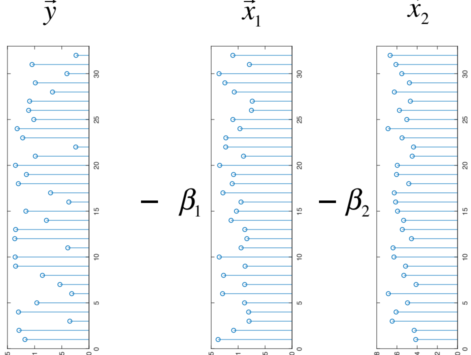
$$||\vec{y}||^2 = ||\beta_{\rm opt}\vec{x}||^2 + ||\vec{y} - \beta_{\rm opt}\vec{x}||^2$$

Multiple regression:

$$\min_{\vec{\beta}} ||\vec{y} - \sum_k \beta_k \vec{x}_k||^2 = \min_{\vec{\beta}} ||\vec{y} - X\vec{\beta}||^2$$

2D example:

$\vec{y}$ $\vec{x}_1$ $\vec{x}_2$

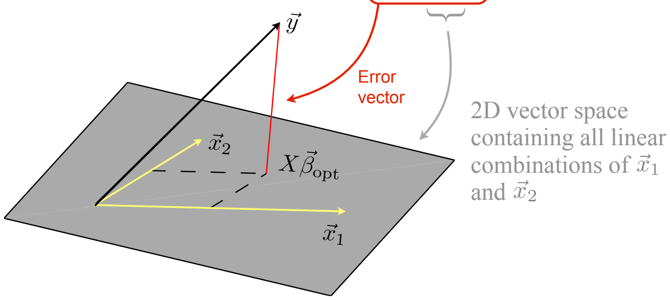


(observation) (regressor 1) (regressor 2)

Solution via the "Orthogonality Principle":

Construct matrix X, containing columns $\vec{x}_1$ and $\vec{x}_2$

Orthogonality: $X^T\left(\vec{y} - X\vec{\beta}\right) = \vec{0}$



Alternatively, can solve using SVD...

$$\min_{\vec{\beta}} ||\vec{y} - X\vec{\beta}||^2 = \min_{\vec{\beta}} ||\vec{y} - USV^T\vec{\beta}||^2$$
$$= \min_{\vec{\beta}} ||U^T\vec{y} - SV^T\vec{\beta}||^2$$
$$= \min_{\vec{\beta}^*} ||\vec{y}^* - S\vec{\beta}^*||^2$$

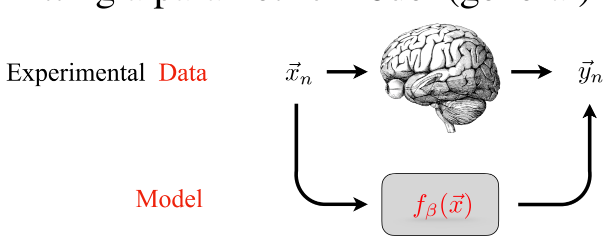
where $\vec{y}^* = U^T\vec{y}, \quad \vec{\beta}^* = V^T\vec{\beta}$

Solution: $\beta^*_{\text{opt},k} = y^*_k / s_k, \quad$ for each k

or $\vec{\beta}^*_{\text{opt}} = S^{\#}\vec{y}^* \quad \Rightarrow \vec{\beta}_{\text{opt}} = VS^{\#}U^T\vec{y}$

[on board: transformations, elliptical geometry]

Fitting a parametric model (general)



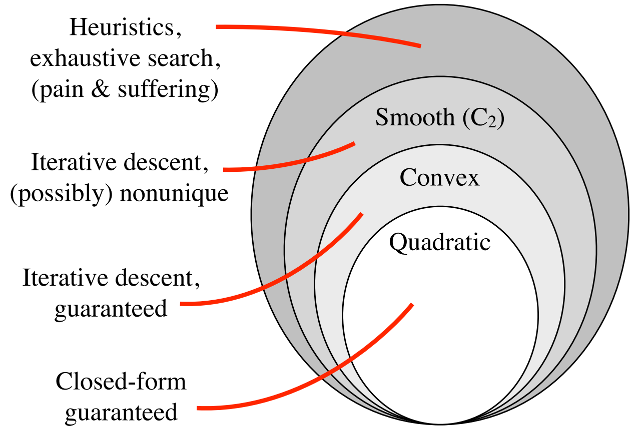
To fit model $f_\beta(\vec{x})$ to data $\{\vec{x}_n, \vec{y}_n\}$,

optimize parameters β to minimize an error function:

$$\min_\beta \sum_n E\left(\vec{y}_n, f_\beta(\vec{x}_n)\right)$$

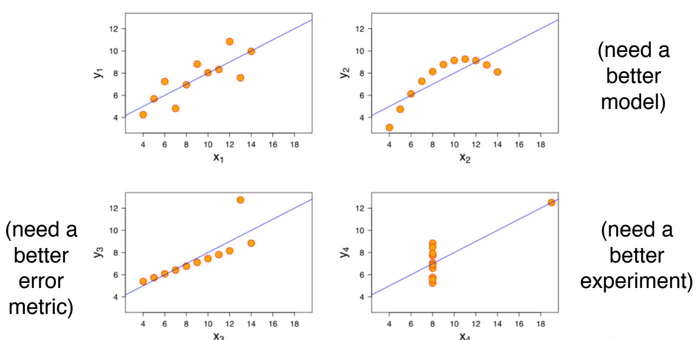
Ingredients: data, model, error function, optimization method

Optimization



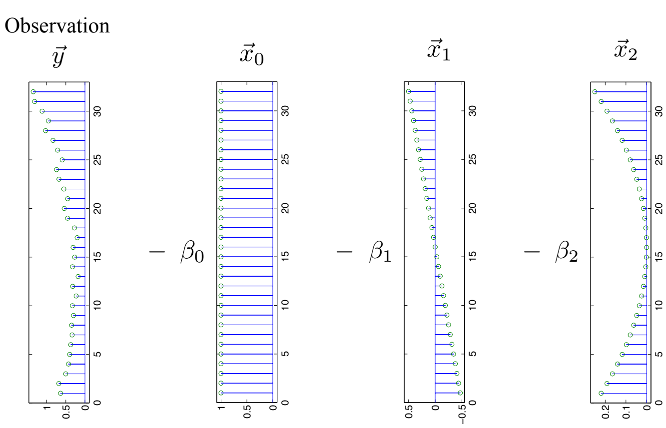
Interpretation warning: fitting a line does not guarantee data actually lie along a line

These 4 data sets give the same regression fit, and same error:

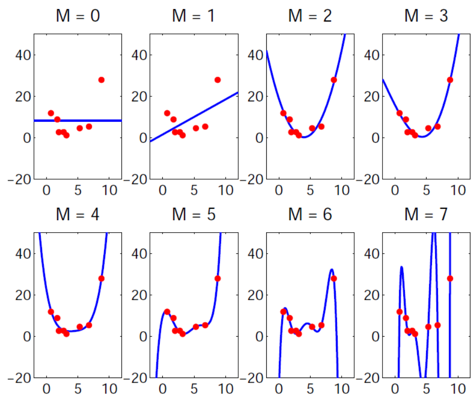


[Anscombe, 1973]

Polynomial regression



Polynomial regression - how many terms?



(to be continued, when we get to "statistics"...)

Weighted Least Squares

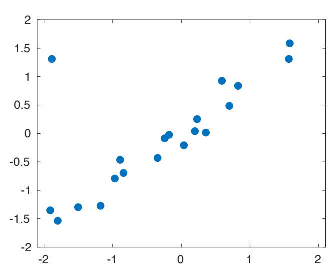
$$\min_{\beta} \sum_{n} \left[w_n(y_n - \beta x_n)\right]^2$$

$$= \min_{\beta} ||W(\vec{y} - \beta\vec{x})||^2$$

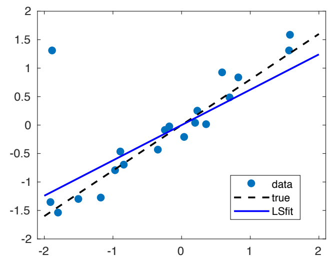
diagonal matrix

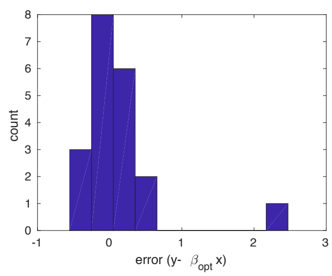
Solution via simple extensions of basic regression solution
(i.e., let $\vec{y}^* = W\vec{y}$ and $\vec{x}^* = W\vec{x}$ and solve for β)

Outliers

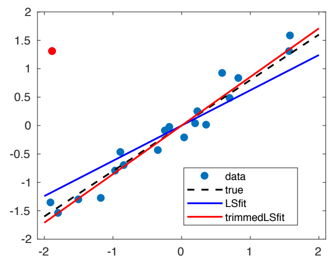


Outliers



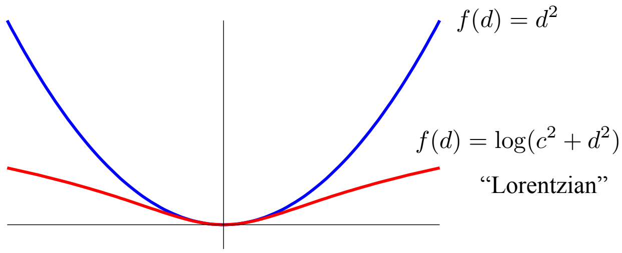


"Trimming"… discard points with large error.
Note: a special case of weighted least squares.



Trimming can be done iteratively (discard outlier, re-fit, repeat), a so-called "greedy" method. When do you stop?

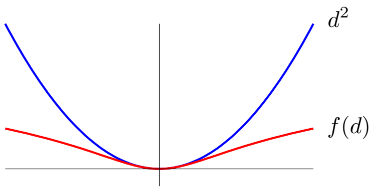
More generally, use a "robust" error metric.
For example:



Note: generally can't obtain solution directly (i.e., requires an iterative optimization procedure).

In some cases, can use iteratively re-weighted least squares (IRLS)...

Iteratively Re-weighted Least Squares (IRLS)



initialize: $w_n^{(0)} = 1$

$$\beta^{(i)} = \arg\min_{\beta} \sum_n w_n^{(i)} \left(y_n - \beta^{(i)} x_n \right)^2$$

iterate

$$w_n^{(i+1)} = \frac{f'(y_n - \beta^{(i)} x_n)}{|y_n - \beta^{(i)} x_n|}$$

iterate

(one of many variants)

Constrained Least Squares

$$\min_{\vec{\beta}} \|\vec{y} - X\vec{\beta}\|^2, \text{ where } \vec{c} \cdot \vec{\beta} = \alpha$$

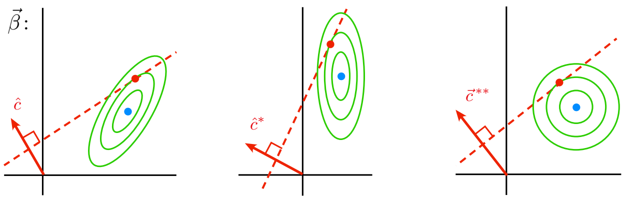
Can be solved exactly using linear algebra (SVD)...
[on board, with geometry]

rotate by V^T

$\vec{\beta}^* = V^T \vec{\beta}$

stretch/squeeze by S^* (nonzero rows of S)

$\vec{\beta}^{**} = S^* \vec{\beta}^*$



$\min_{\vec{\beta}} \left\| \vec{y} - USV^T\vec{\beta} \right\|$ | $\min_{\vec{\beta}^*} \left\| \vec{y}^* - S\vec{\beta}^* \right\|$ | $\min_{\vec{\beta}^{**}} \left\| \vec{y}^{**} - \vec{\beta}^{**} \right\|$

unconstrained: $\vec{\beta}_{\text{opt}} = VS^{\#}U^T\vec{y}$ | $\vec{\beta}^*_{\text{opt}} = S^{\#}\vec{y}^*$ | $\vec{\beta}^{**}_{\text{opt}} = \vec{y}^{**}$

constraint: $\hat{c}^T\vec{\beta} = \alpha$ | $(\hat{c}^*)^T\vec{\beta}^* = \alpha$ | $(\vec{c}^{**})^T\vec{\beta}^{**} = \alpha$

$\vec{y}^* = U^T\vec{y}$

$\hat{c}^* = V^T\hat{c}$

$\vec{y}^{**}$ = top two elements of $\vec{y}^*$

$\vec{c}^{**} = (S^*)^{-1}\hat{c}^*$

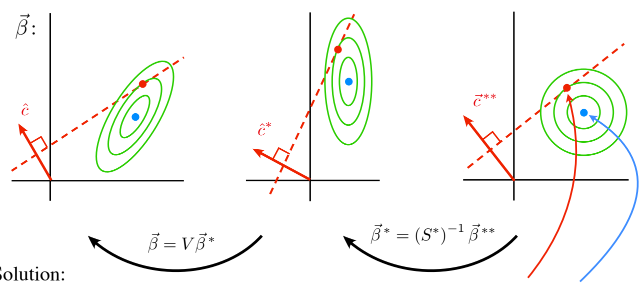
---

rotate by V^T

$\vec{\beta}^* = V^T \vec{\beta}$

stretch/squeeze by S^* (nonzero rows of S)

$\vec{\beta}^{**} = S^* \vec{\beta}^*$



$\vec{\beta} = V\vec{\beta}^*$

$\vec{\beta}^* = (S^*)^{-1}\vec{\beta}^{**}$

Solution:

$\vec{\beta}_{\text{c,opt}} = V\,(S^*)^{-1}\,(\vec{y}^{\,**} + \gamma\vec{c}^{\,**})$

Write solution as: $\vec{\beta}^{**}_{\text{c,opt}} = \vec{y}^{\,**} + \gamma\vec{c}^{**}$

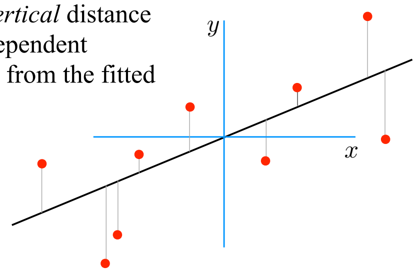
Solve for γ:

$$(\vec{c}^{\,**})^T\,\vec{\beta}^{**} = (\vec{c}^{\,**})^T\,(\vec{y}^{\,**} + \gamma\vec{c}^{\,**}) = \alpha$$

$$\Rightarrow \quad \gamma = \frac{\alpha - (\vec{c}^{\,**})^T\,\vec{y}^{\,**}}{(\vec{c}^{\,**})^T\,\vec{c}^{\,**}}$$

Standard Least Squares regression

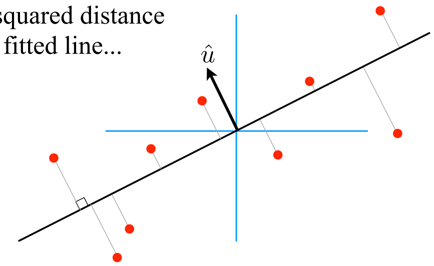
Error is *vertical* distance (in the "dependent variable") from the fitted line...



$$\arg\min_{\beta} ||\vec{y} - \beta\vec{x}||^2$$

Total Least Squares Regression
(a.k.a "orthogonal regression")

Error is squared distance from the fitted line...



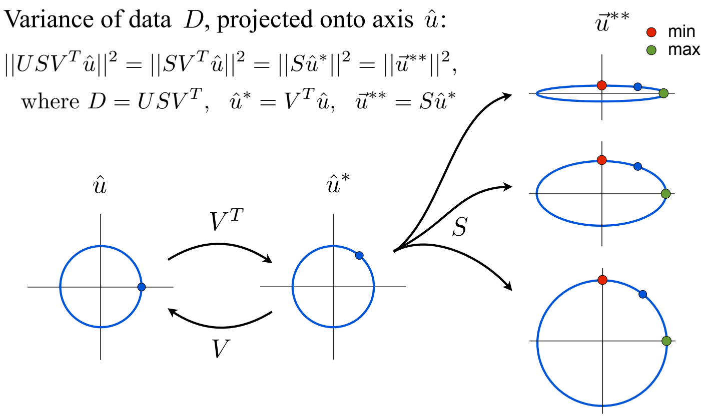
expressed as: $\min_{\hat{u}} ||D\hat{u}||^2, \quad \text{where } ||\hat{u}||^2 = 1$

Note: "data" matrix D now includes both x and y coordinates

Variance of data D, projected onto axis $\hat{u}$:

$$||USV^T\hat{u}||^2 = ||SV^T\hat{u}||^2 = ||S\hat{u}^*||^2 = ||\vec{u}^{**}||^2,$$

where $D = USV^T, \quad \hat{u}^* = V^T\hat{u}, \quad \vec{u}^{**} = S\hat{u}^*$

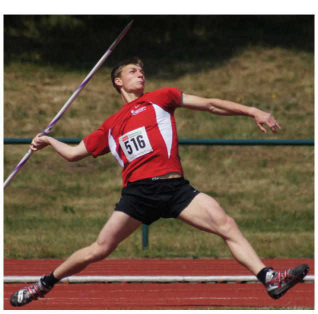


Set of $\hat{u}$'s of length 1 (i.e., unit vectors)

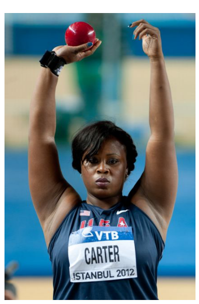
Set of $\hat{u}^*$'s of length 1 (i.e., unit vectors)

First two components of $\vec{u}^{**}$ (rest are zero!), for three example S's.

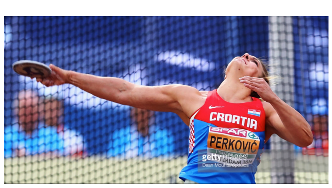
Olympic gold medalists (Rio, 2016)



Thomas Röhler (Germany)



Michelle Carter (USA)



Sandra Perković (Croatia)

3D geometry:
Javelin, Discus, Shotput…

Eigenvectors/eigenvalues

Define symmetric matrix:

$$\begin{aligned} C &= D^T D \\ &= (USV^T)^T (USV^T) \\ &= VS^T U^T USV^T \\ &= V(S^T S)V^T \end{aligned}$$

- "rotate, stretch, rotate back"
- The matrix C "summarizes" the shape of the data with an ellipsoid: principal axes are columns of V, dimensions are diagonal elements of S

- An *eigenvector* is a vector that is rescaled by a matrix (i.e., direction is unchanged)
- The corresponding scale factor is called the *eigenvalue*
- The columns of V (denoted $\hat{v}_k$) are eigenvectors of C, with corresponding eigenvalues s_k^2 :

$$\begin{aligned} C\hat{v}_k &= V(S^T S)V^T \hat{v}_k \\ &= V(S^T S)\hat{e}_k \\ &= s_k^2 V \hat{e}_k \\ &= s_k^2 \hat{v}_k \end{aligned}$$

Principal Component Analysis (PCA)

The shape of a data cloud can be summarized with an ellipse (ellipsoid) using a simple procedure:

(1) Subtract mean of all data points, to re-center around origin

(2) Assemble centered data vectors in rows of a matrix, D

(3) Compute the SVD of D:

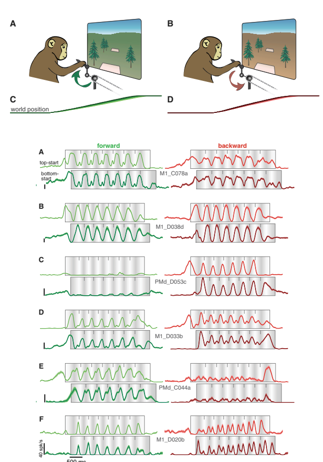
$$D = USV^T$$

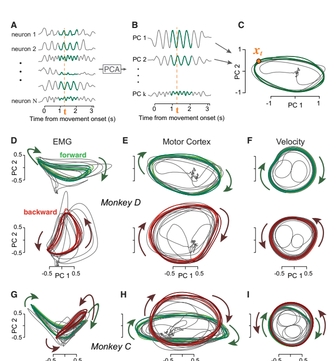
or equivalently compute eigenvectors of $C = D^T D$:

$$C = V \Lambda V^T$$

(4) Columns of V are the *principal components* (axes) of the ellipsoid, diagonal elements s_k or $\sqrt{\lambda_k}$ are the corresponding principle radii

Example: PCA for dimensionality reduction and visualization





[Russo et. al., 2018]