Computational Physics III

Lecture 13:
Singular value decomposition (SVD).
Least squares regression.

May 28, 2020

Outline

- Singular value decomposition (SVD) example
- Linear systems revisited
- Least squares regression
- Weighted least squares regression

Singular Value Decomposition (SVD)

Any $m \times n$ $(m \geq n)$ matrix A has a *singular value decomposition* (SVD)

$$A = U \Sigma V^*$$

where (in full-form SVD)

U is the $m \times m$ unitary matrix of left-singular vectors,

V is the $n \times n$ unitary matrix of right-singular vectors,

Σ is the $m \times n$ diagonal matrix of singular values.

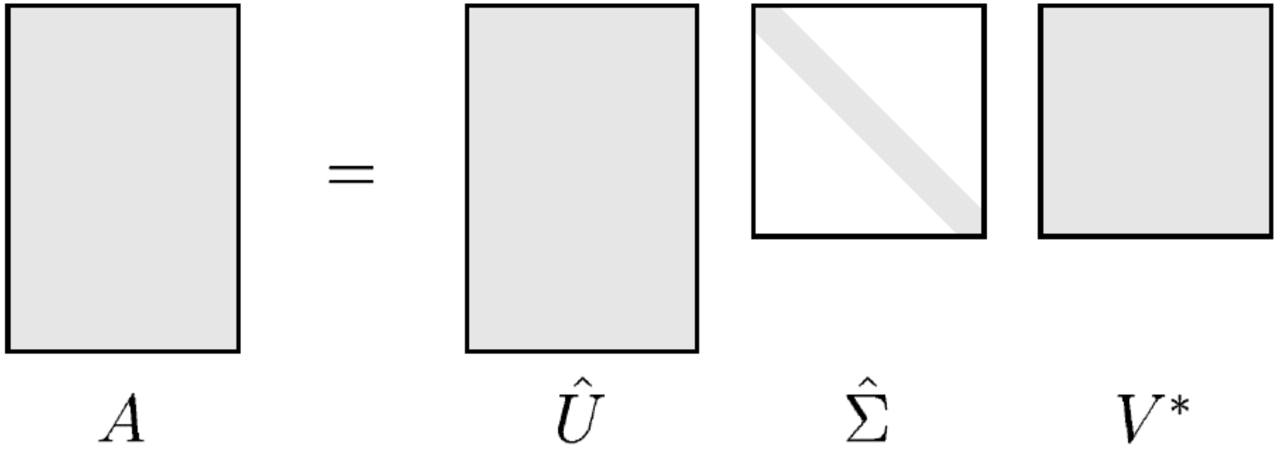
By convention, singular values are ordered as

$$\sigma_1 \geq \sigma_2 \geq \ldots \sigma_p > \sigma_{p+1} = \ldots = \sigma_n = 0$$

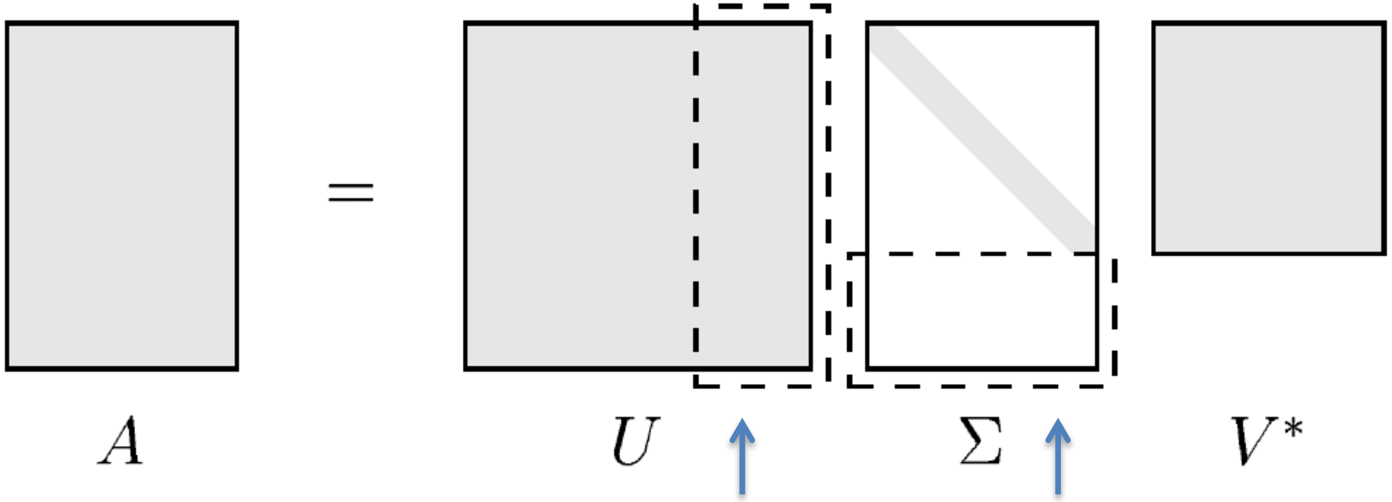
with p the rank matrix A. If $p = n$ the matrix is full-rank.

Singular Value Decomposition (SVD)

Reduced form SVD:



Full form SVD:



orthogonal vectors

zeros

Matrix properties from SVD

A single SVD decomposition of A provides the following information:

- Rank of A (number of non-zero singular values)
- $\text{range}(A) = \langle u_1, \ldots, u_p \rangle$ and $\text{null}(A) = \langle u_{p+1}, \ldots, u_n \rangle$
- Induced 2-norm and Frobenius norm
- Absolute values of eigenvalues of A (if $A = A^*$)
- Determinant $|\det(A)| = \prod_{i=1}^{m} \sigma_i$
- Condition number $K(A) = \dfrac{\sigma_{\max}}{\sigma_{\min}}$
- Least squares solution, dyadic expansion, matrix approximation
- …and more

Solving a linear system with SVD

Solve a linear system $Ax = b$ of m equations with n unknowns ($m \geq n$) using singular value decomposition (full form)

$$U\Sigma V^T x = b \qquad \Sigma V^T x = U^T b \qquad \begin{cases} d = U^T b \\ \Sigma z = d \\ z = V^T x \end{cases}$$

$$\sigma_j z_j = d_j \quad j \leq n \ \text{ and } \ \sigma_j \neq 0 \qquad\qquad z_j = d_j/\sigma_j \quad j \leq n \ \text{ and } \ \sigma_j \neq 0$$

$$0 \cdot z_j = d_j \quad j \leq n \ \text{ and } \ \sigma_j = 0 \qquad\qquad z_j = \text{whatever} \quad j \leq n \ \text{ and } \ \sigma_j = 0$$

$$0 = d_j \quad j > n \qquad\qquad \text{Must be satisified}$$

Solving a linear system with SVD: an example

$$A = \begin{pmatrix} 1 & 6 & 11 \\ 2 & 7 & 12 \\ 3 & 8 & 13 \\ 4 & 9 & 14 \\ 5 & 10 & 15 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 35.127 & 0 & 0 \\ 0 & 2.47 & 0 \\ 0 & 0 & 0.0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$U = \begin{pmatrix} -0.355 & 0.689 & 0.57 & -0.176 & 0.21 \\ -0.399 & 0.376 & -0.745 & 0.224 & 0.307 \\ -0.443 & 0.0624 & -0.17 & -0.365 & -0.799 \\ -0.487 & -0.251 & 0.297 & 0.765 & -0.163 \\ -0.531 & -0.564 & 0.049 & -0.447 & 0.445 \end{pmatrix}$$

$$V = \begin{pmatrix} -0.202 & -0.89 & 0.408 \\ -0.517 & -0.257 & -0.816 \\ -0.832 & 0.376 & 0.408 \end{pmatrix}$$

Solving a linear system with SVD: an example

Let's solve $Ax = b$ with:

$$b = \begin{pmatrix} 5 \\ 5 \\ 5 \\ 5 \\ 5 \end{pmatrix} \qquad d = U^T b = \begin{pmatrix} -11.1 \\ 1.56 \\ 0 \\ \boxed{\begin{matrix} 0 \\ 0 \end{matrix}} \end{pmatrix}$$

$$\begin{aligned} \sigma_1 z_1 &= d_1 \\ \sigma_2 z_2 &= d_2 \\ z_3 &= \text{whatever} = 0 \end{aligned}$$

Zeros, thus b is in range(A)

$$x = Vz = \begin{pmatrix} -0.5 \\ 0 \\ 0.5 \end{pmatrix}$$ is a solution.

But, $\sigma_3 = 0$, this means that the $\mathcal{N}(A) \neq 0$

In fact, $\mathcal{N}(A) = \text{span}(v_3)$

$$x = \begin{pmatrix} -0.5 \\ 0 \\ 0.5 \end{pmatrix} + \alpha \begin{pmatrix} 0.408 \\ -0.816 \\ 0.408 \end{pmatrix}$$

with $\alpha \in R$

Solving a linear system with SVD: an example

Let's change b:

$$b = \begin{pmatrix} 4 \\ 5 \\ 5 \\ 5 \\ 5 \end{pmatrix} \qquad d = U^T b = \begin{pmatrix} -10.7 \\ 0.872 \\ -0.57 \\ \boxed{\begin{matrix} 0.176 \\ -0.21 \end{matrix}} \end{pmatrix}$$

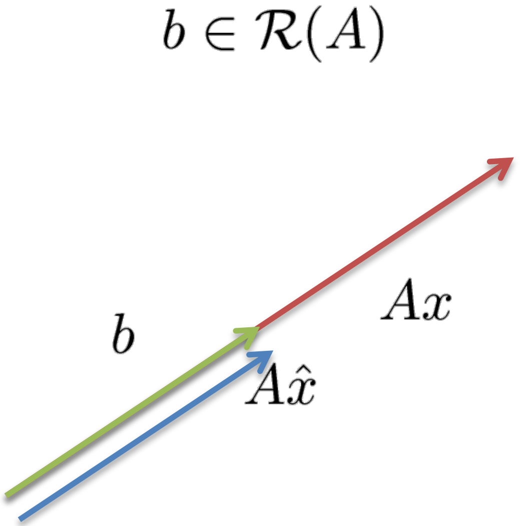
Non-zero elements

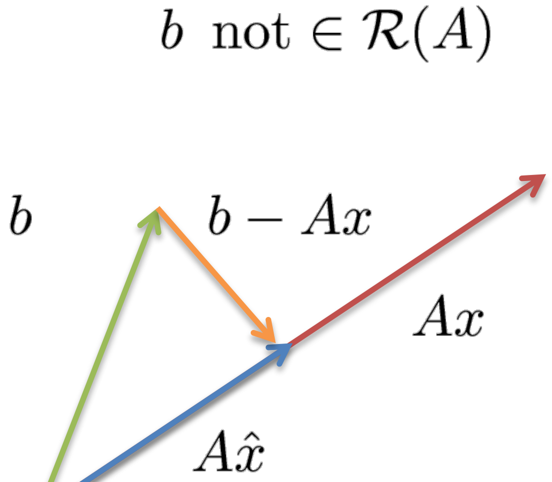
b is not in the $\mathcal{R}(A)$... the system is inconsisten and there is no solution,

Approximate (least square) solution

$$\min_x ||Ax - b||^2$$

Solving a linear system with SVD





We can solve system in the classical sense

$$A\hat{x} = b$$

We can solve system in the *least squares* sense, i.e. find

$$\min_x \left|\left|Ax - b\right|\right|^2$$

Least squares solution of linear systems

$$\left\|Ax-b\right\| = \left\|AVV^Tx-b\right\| = \left\|\overset{\Sigma}{\boxed{U^T(AV}}V^Tx-b)\right\| = \left\|\Sigma z-d\right\|$$

$$z = V^Tx$$

$$d = U^Tb$$

We want the minimum:

$$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_p > \sigma_{p+1} = \sigma_{p+2} = \cdots = \sigma_m$$

$$||\Sigma z-d|| = (\sigma_1 z_1 - d_1)^2 + (\sigma_2 z_2 - d_2)^2 + \cdots + (\sigma_p z_p - d_p)^2 + (d_{p+1})^2 + \cdots + (d_n)^2$$

Minimized by:

$$z_j = d_j/\sigma_j \quad j \leq n \;\text{ and }\; \sigma_j \neq 0$$

$$z_j = whatever \quad j \leq n \;\text{ and }\; \sigma_j = 0$$

This provides the least squares solution

Matrix pseudoinverse

Theorem. A vector x minimizes norm $\|r\| = \|b - Ax\|$ iff $r \perp \mathrm{range}(A)$, i.e.

$$A^* r = 0$$

i.e.

$$A^* A x = A^* b$$

So, if A is full-rank

$$x = \left(A^* A\right)^{-1} A^* b = \underset{\uparrow}{A^+} b$$

Pseudoinverse (or Moore-Penrose inverse, `pinv` in Matlab)

Pseudoinverse with SVD

$$A^+ = V\Sigma^+ U^*$$

If A is full rank

$$\sigma_i^+ = \frac{1}{\sigma_i}$$

If A is not full rank

$$\sigma_i^+ = \frac{1}{\sigma_i} \text{ if } \sigma_i \neq 0$$

$$\sigma_i^+ = 0 \text{ if } \sigma_i = 0$$

Least squares solution via pseudoinverse

The solution is now given by

$$\hat{x} = A^{+}b$$

If A is full rank

$\hat{x}$ is unique; it minimizes $\|b - Ax\|$

If A is not full rank

$\hat{x}$ is not unique; we can add any vector $v_i \in \mathrm{null}(A)$ and still have a solution $x^* = \hat{x} + \alpha_1 v_{p+1} + \alpha_2 v_{p+2} + \ldots$

In this case we are talking about *least norm* solution

Our example revised

$$A = \begin{pmatrix} 1 & 6 & 11 \\ 2 & 7 & 12 \\ 3 & 8 & 13 \\ 4 & 9 & 14 \\ 5 & 10 & 15 \end{pmatrix}$$

$$A^+ = \begin{pmatrix} -0.247 & -0.133 & -0.02 & 0.0933 & 0.207 \\ -0.0667 & -0.0333 & 1.47 \cdot 10^{-17} & 0.0333 & 0.0667 \\ 0.113 & 0.0667 & 0.02 & -0.0267 & -0.0733 \end{pmatrix}$$

$$A^+ b = \hat{x}$$

$$b = \begin{pmatrix} 4 \\ 5 \\ 5 \\ 5 \\ 5 \end{pmatrix}$$

$$\hat{x} = \begin{pmatrix} -0.253 \\ 0.0667 \\ 0.387 \end{pmatrix}$$

This is the least norm solution

$$x^* = \hat{x} + \alpha v_3$$

General solution

$$A * \hat{x} = \begin{pmatrix} 4.4 \\ 4.6 \\ 4.8 \\ 5.0 \\ 5.2 \end{pmatrix}$$

Let's check the least square solution

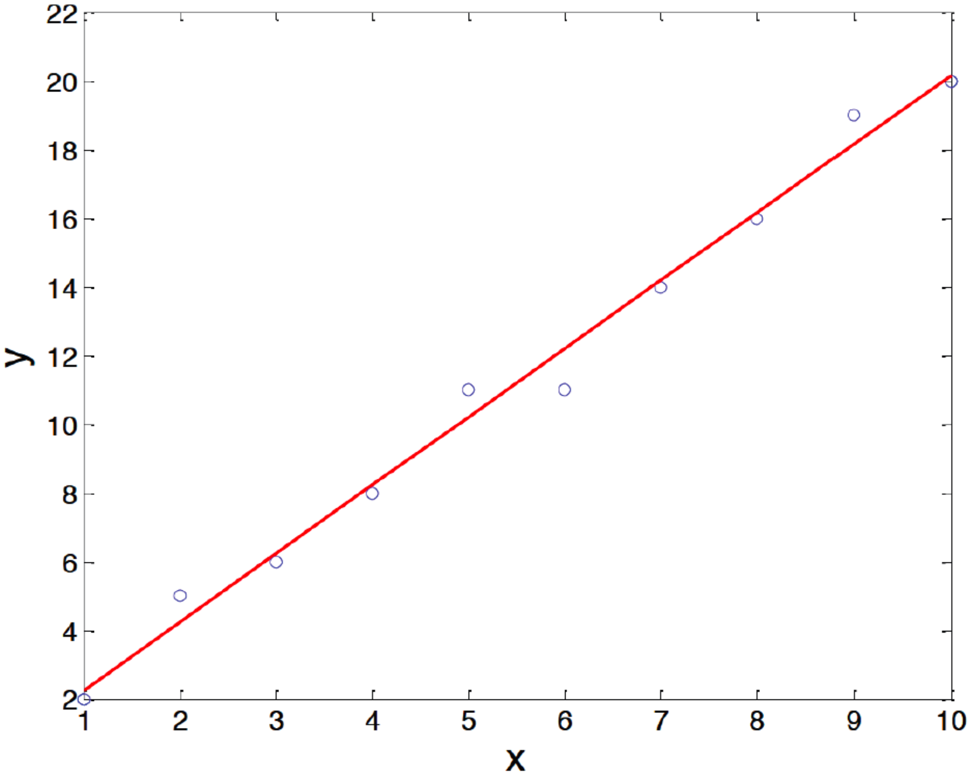
$$||b - A\hat{x}|| = 0.6325$$

You cannot do any better than this!

Linear fit: an example

$$\left[\begin{array}{cc} x & y \\ \hline & \\ 1 & 2 \\ 2 & 4 \\ 3 & 7 \\ 4 & 7 \\ 5 & 10 \\ 6 & 11 \\ 7 & 13 \\ 8 & 16 \\ 9 & 19 \\ 10 & 20 \end{array}\right]$$

$$\begin{aligned} 1 &= 2a + b \\ 2 &= 4a + b \\ 3 &= 7a + b \\ &\vdots \\ 10 &= 20a + b \end{aligned}$$



Least squares solution provides the best fit

$$y_i = ax_i + b$$

$$\begin{pmatrix} | & | \\ x & 1 \\ | & | \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} | \\ y \\ | \end{pmatrix}$$

Minimizes

$$r^2 = \sum_{i=1}^{m} |ax_i + b - y_i|^2$$

Polynomial fit

Fit to $n-1$ degree polynomial ($n<m$)

$$p(x) \; = \; c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

Rectangular Vandermonde system

$$\begin{bmatrix} 1 & x_1 & & x_1^{n-1} \\ 1 & x_2 & \cdots & x_2^{n-1} \\ 1 & x_3 & & x_3^{n-1} \\ & \vdots & & \vdots \\ 1 & x_m & \cdots & x_m^{n-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} \approx \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

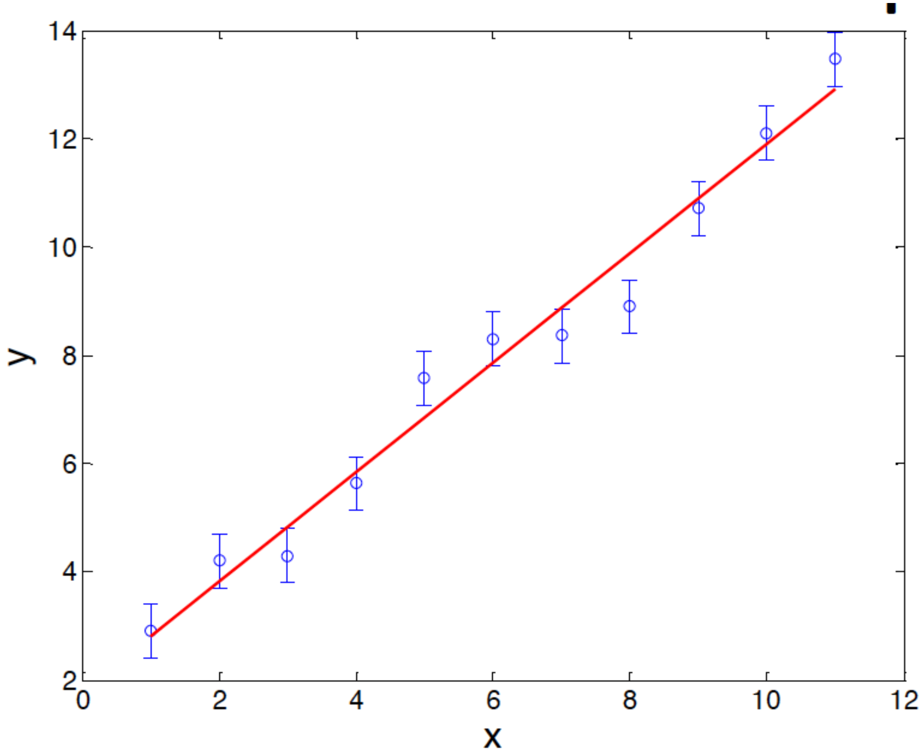
Statistical interpretation

Consider a simple linear model

$$y_i = ax_i + b + \varepsilon_i$$

where x is independent variable, y is dependent variable and ε_i are

- independent and identically distributed variables (i.i.d.)
- belong to a distribution with mean $\mu = 0$ and finite standard deviation σ^2



The least squares method is equivalent to minimizing

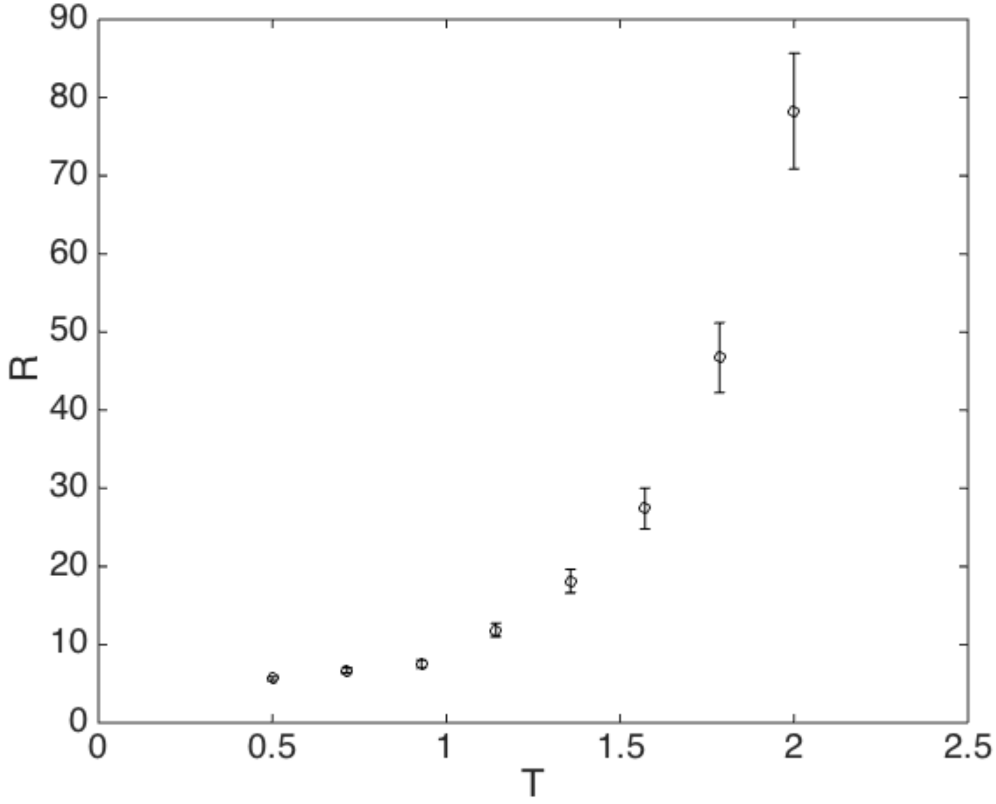
$$\sum_i \varepsilon_i^2 = \sum_i \left[y_i - (ax_i + b)\right]^2$$

Data with errors

An exercise from one of previous years: measurements of resistance *R* vs. temperature *T* with error bars varying the range of *T*

$R(T) = A + BT + CT^2 + \ldots$

A – scattering by impurities
C – electron-electron interactions
…



Data with errors

Consider a situation when with each y_i there is an associated uncertainty σ_i. In this case, we have to minimize

$$\chi^2 = \sum_i \left[\frac{y_i - (ax_i + b)}{\sigma_i} \right]^2$$

In other words, $\{y_i, x_i\}$ have to be weighted by σ_i.

This is formally justified as maximize the probability of measurements $\{y_1, \ldots, y_N\}$ given the <u>normally</u> distributed PDF

$$f(y_1, \ldots, y_N \mid a, b) \propto \exp\left[-\sum_i \frac{\left(y_i - (ax_i + b)\right)^2}{2\sigma_i} \right]$$

or equivalently, the corresponding *log-likelihood function*

$$\ln L(a, b \mid y_1, \ldots, y_N) = -\frac{1}{2} \sum_i \frac{\left(y_i - (ax_i + b)\right)^2}{\sigma_i} = -\frac{1}{2}\chi^2$$

Weighted least squares regression

$$y(x) = \sum_{k=0}^{M-1} a_k X_k(x)$$

$$\chi^2 = \sum_{i=0}^{N-1} \left[\frac{y_i - \sum_{k=0}^{M-1} a_k X_k(x_i)}{\sigma_i} \right]^2$$

$$\begin{array}{cc} & \xleftarrow{\quad} \text{basis functions} \xrightarrow{\quad} \\ & \begin{array}{cccc} X_0(\ ) & X_1(\ ) & \cdots & X_{M-1}(\ ) \end{array} \\ \text{data points} & \begin{array}{c} x_0 \\ \\ x_1 \\ \\ \vdots \\ \\ \\ \\ x_{N-1} \end{array} \left(\begin{array}{cccc} \frac{X_0(x_0)}{\sigma_0} & \frac{X_1(x_0)}{\sigma_0} & \cdots & \frac{X_{M-1}(x_0)}{\sigma_0} \\ \frac{X_0(x_1)}{\sigma_1} & \frac{X_1(x_1)}{\sigma_1} & \cdots & \frac{X_{M-1}(x_1)}{\sigma_1} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \frac{X_0(x_{N-1})}{\sigma_{N-1}} & \frac{X_1(x_{N-1})}{\sigma_{N-1}} & \cdots & \frac{X_{M-1}(x_{N-1})}{\sigma_{N-1}} \end{array} \right) \end{array}$$

$$A_{ij} = \frac{X_j(x_i)}{\sigma_i} \qquad b_i = \frac{y_i}{\sigma_i}$$

The system is then solved in least squares sense:

$$b = Aa$$

Fit parameters

Weighted observations

Weighted design matrix

Goodness of fit

Define the reduced quantity (*"goodness of fit"*)

$$\chi_\nu^2 = \frac{\chi^2}{\nu}$$

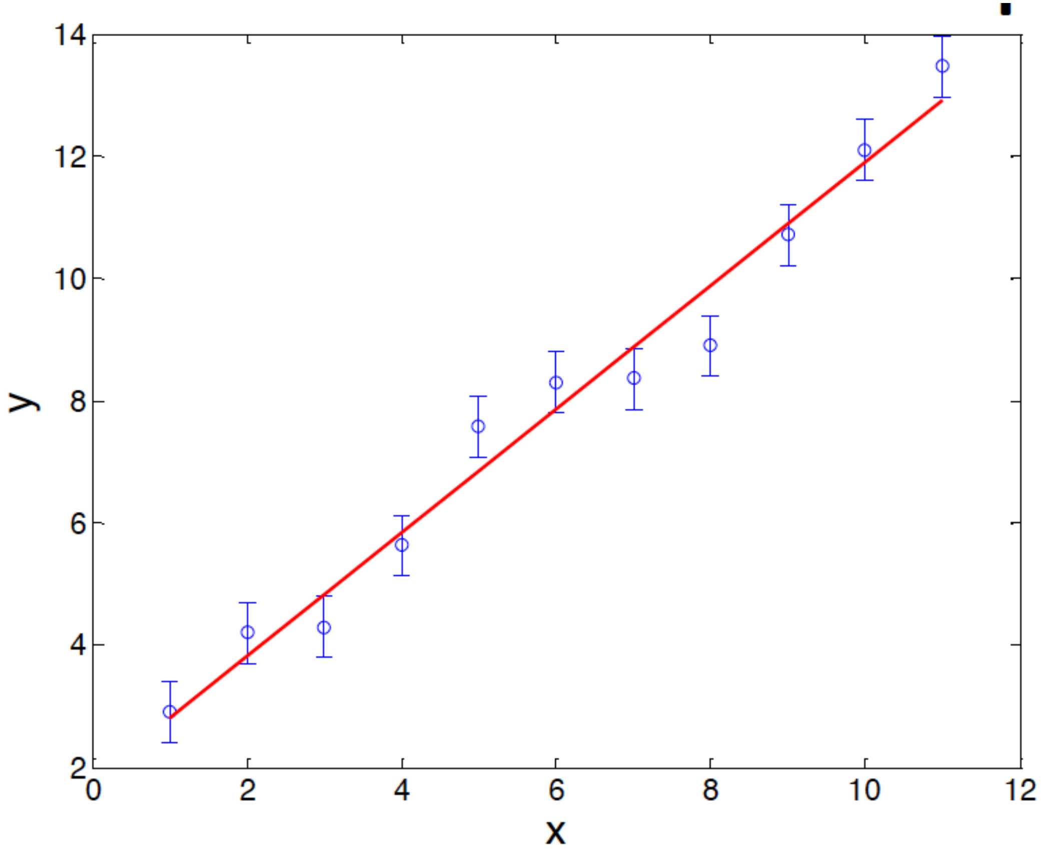
where ν is the number of degrees of freedom (number of data points minus number of fit parameters)

$$\chi_\nu^2 << 1 \qquad \chi_\nu^2 \sim 1 \qquad \chi_\nu^2 >> 1$$

$$\xrightarrow{\hspace{10em}} \chi_\nu^2$$

probably you are overfitting the data	the fit is good!	the fit does not reproduce well the data

Example



$$a = 1.0857$$

$$b = -0.0107$$

$$\nu = N - 2$$

$$\chi^2 = \sum_i \frac{(y_i - (ax_i + b))^2}{\sigma_i^2}$$

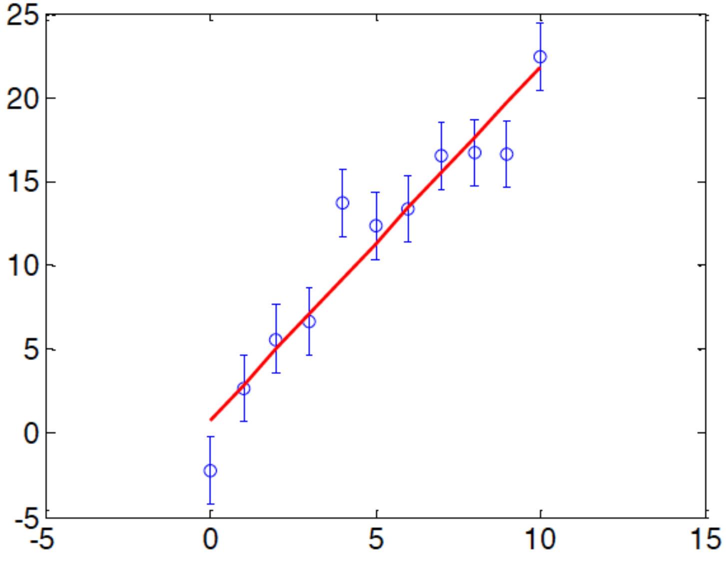
$$\chi^2 = 11.2$$

$$\chi^2_\nu = \frac{\chi^2}{\nu} = 1.2$$

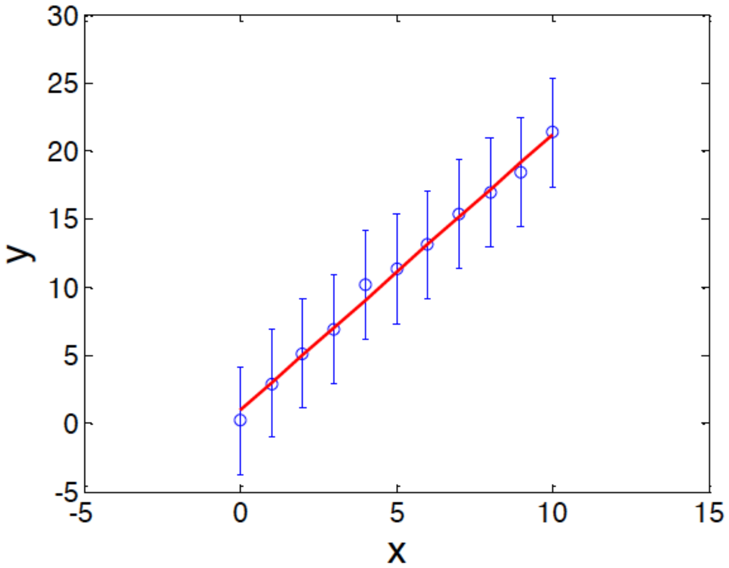
↑

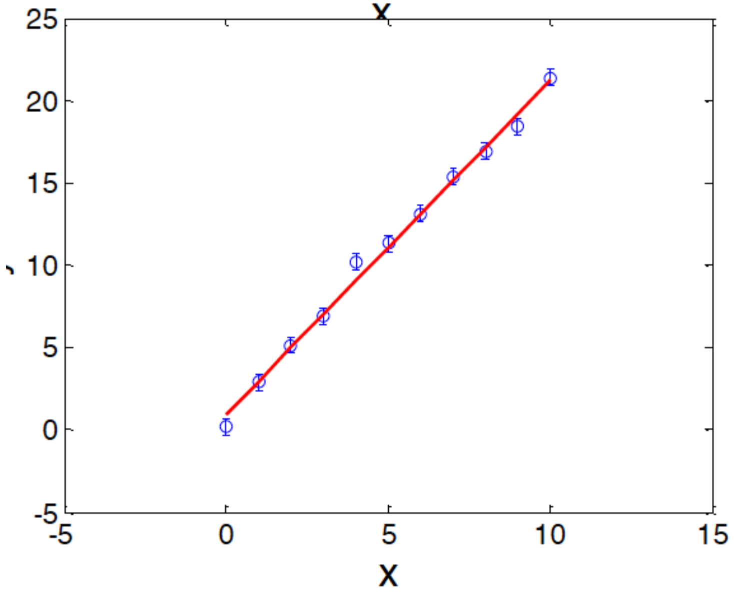
GOOD FIT

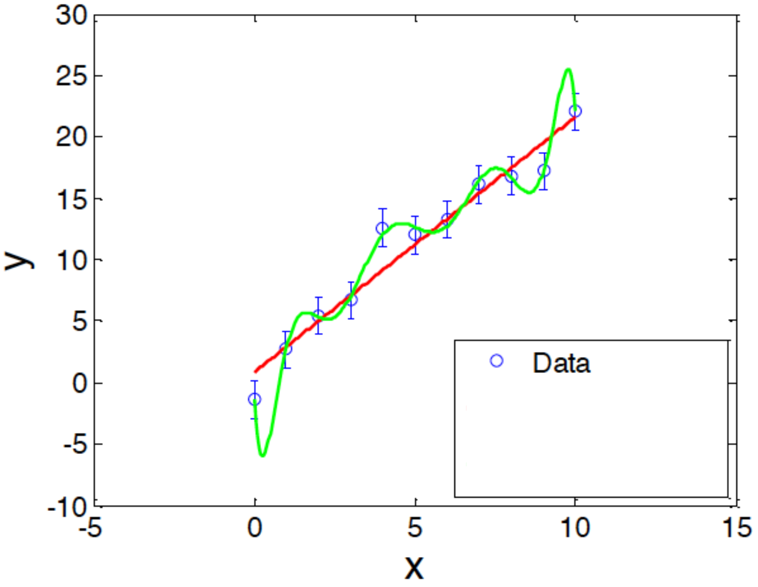
Data overfitting



$$\chi_{\nu}^{2} << 1$$







Data underfitting

$$\chi_\nu^2 >> 1$$