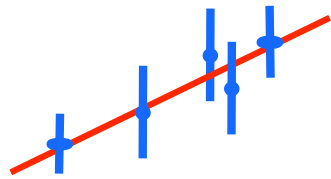


Review: Data Space Metric



Data $X_i \pm \sigma_i$ $i = 1 \dots N$

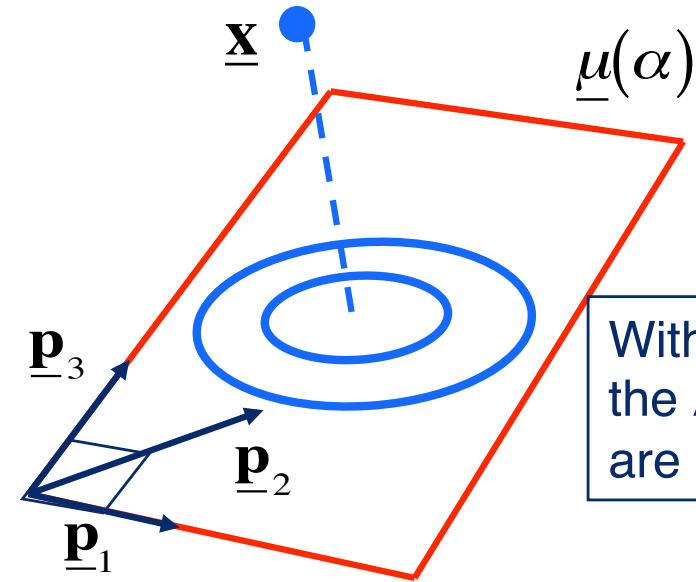
Data space metric: $g_{ij} = \frac{\delta_{ij}}{\sigma_i^2}$

Distance from data to model:

$$\|\underline{\mathbf{x}} - \underline{\mu}(\alpha)\|^2 = \sum_{i=1}^N \left(\frac{X_i - \mu_i(\alpha)}{\sigma_i} \right)^2 = \chi^2.$$

For **linear models** (scaling patterns), the model surface is a flat M -dimensional hyper-plane spanned by M vectors $\underline{\mathbf{p}}_k$.

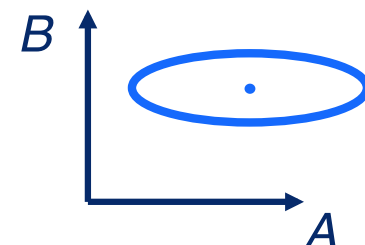
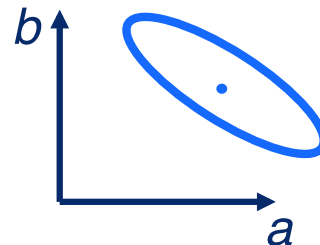
With this metric, the χ^2 contours are **circular**.



Non-orthogonal vs Orthogonal patterns:

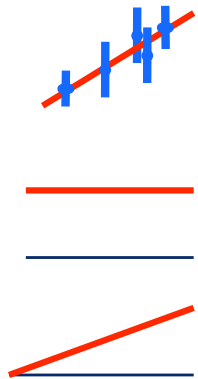
$$\underline{\mu} = b \underline{\mathbf{p}}_1 + a \underline{\mathbf{p}}_2$$

$$\underline{\mu} = B \underline{\mathbf{p}}_1 + A \underline{\mathbf{p}}_3$$



Scaling Orthogonal Patterns

Scaling **non-orthogonal patterns**:



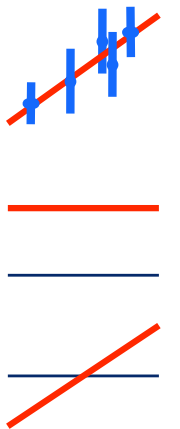
$$y = b + a x$$

$$= b \underline{\mathbf{p}}_1 + a \underline{\mathbf{p}}_2$$

$$\underline{\mathbf{p}}_1 = \{1, 1, \dots, 1\}$$

$$\underline{\mathbf{p}}_2 = \{x_1, x_2, \dots, x_N\}$$

Scaling **orthogonal patterns**:



$$y = B + A (x - \hat{x})$$

$$= B \underline{\mathbf{p}}_1 + A \underline{\mathbf{p}}_3$$

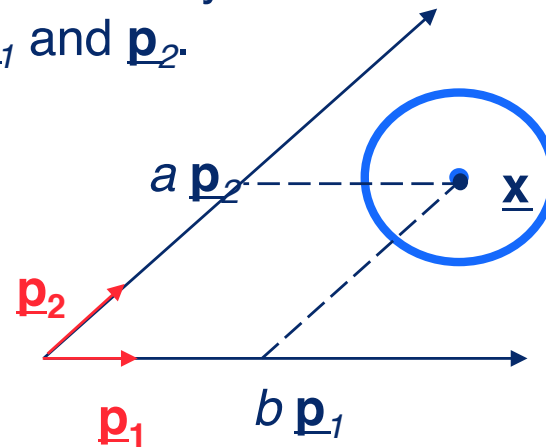
$$\underline{\mathbf{p}}_1 = \{1, 1, \dots, 1\}$$

$$\underline{\mathbf{p}}_3 = \{(x_1 - \hat{x}), (x_2 - \hat{x}), \dots, (x_N - \hat{x})\}$$

$$= \underline{\mathbf{p}}_2 - \hat{x} \underline{\mathbf{p}}_1$$

Model surface is the plane spanned by vectors

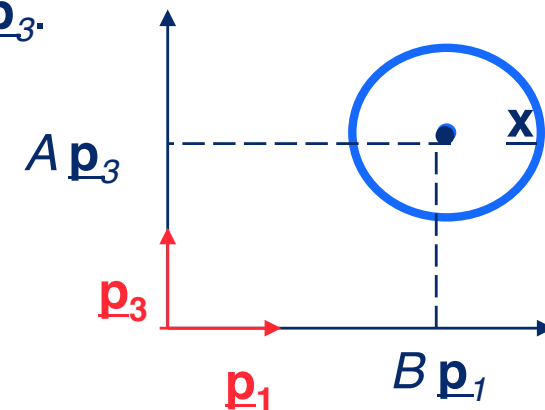
$\underline{\mathbf{p}}_1$ and $\underline{\mathbf{p}}_2$.



Data vector $\underline{\mathbf{x}}$ may be above or below the model plane.

Model plane also spanned by orthogonal vectors

$\underline{\mathbf{p}}_1$ and $\underline{\mathbf{p}}_3$.



Scaling Orthogonal Patterns

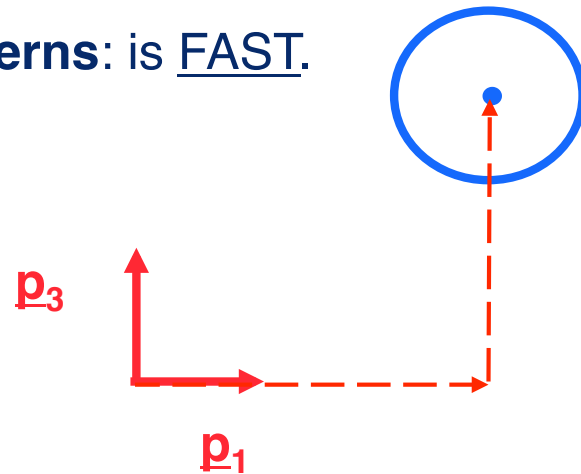
Scaling **non-orthogonal patterns** can be SLOW.

Reach $\eta\sigma$ after 1 step,
 $\eta\sigma \cos\theta$ after 2 steps,
 $\eta\sigma (\cos\theta)^n$ after $n + 1$ steps.

Scaling **orthogonal patterns**: is FAST.

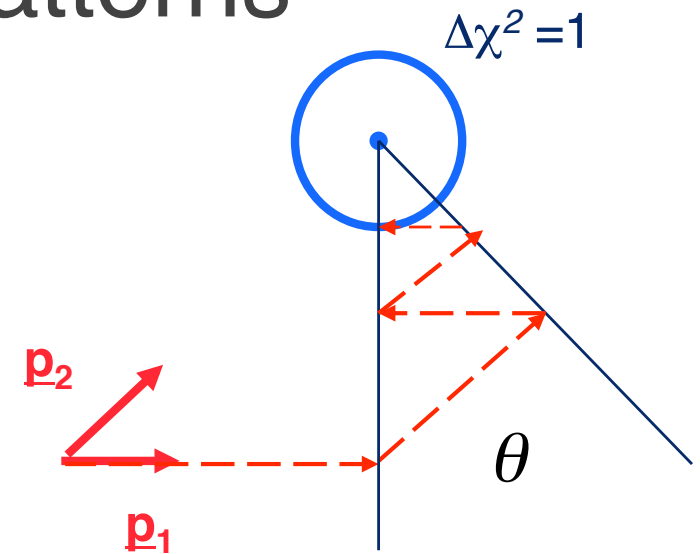
$$\theta = 90^\circ \quad \cos\theta = 0$$

2 steps only !



Scaling **ortho-normal patterns**:

$$\underline{p} \Rightarrow \frac{\underline{p}}{\|\underline{p}\|} \quad \|\underline{p}\|^2 = \underline{p} \bullet \underline{p} = \sum_i \frac{p_i^2}{\sigma_i^2}$$



Iterated Optimal Scaling

$$\hat{\underline{u}} = \sum_k^M \hat{\alpha}_k \underline{p}_k = \sum_k^M \hat{\beta}_k \frac{\underline{p}_k}{\|\underline{p}_k\|}$$

$$\hat{\alpha}_k = \frac{\underline{X} \bullet \underline{p}_k}{\underline{p}_k \bullet \underline{p}_k} \quad \text{Var}[\hat{\alpha}_k] = \frac{1}{\underline{p}_k \bullet \underline{p}_k}$$

$$\hat{\beta}_k = \underline{X} \bullet \frac{\underline{p}_k}{\|\underline{p}_k\|} \quad \text{Var}[\hat{\beta}_k] = 1$$

How to construct Orthogonal Patterns

- **1. Diagonalise Hessian Matrix**
- **2. Graham-Schmidt Process**
- **3. Differences between successive χ^2 fits**

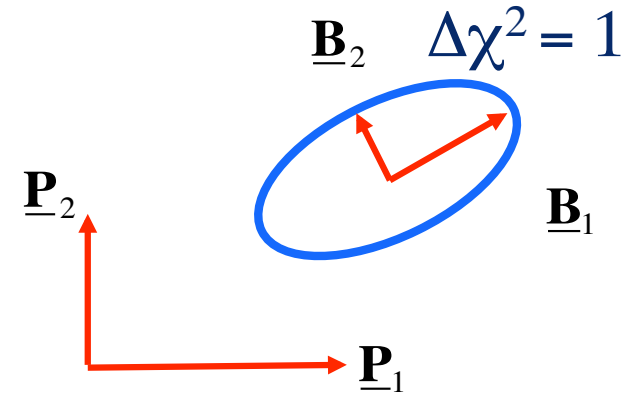
1. Diagonalise Hessian Matrix

- Quadratic approximation to χ^2 surface:

$$\underline{\mu}(\alpha) = \underline{\mu}(\hat{\alpha}) + \sum_i \underline{P}_i \Delta\alpha_i + \dots \quad \Delta\alpha_i \equiv \alpha - \hat{\alpha}$$

$$\chi^2(\alpha) = \chi^2(\hat{\alpha}) + \sum_{i,j} \Delta\alpha_i H_{ij} \Delta\alpha_j + \dots$$

$$H_{ij} \equiv \frac{1}{2} \frac{\partial^2 \chi^2}{\partial \alpha_i \partial \alpha_j} \Big|_{\alpha=\hat{\alpha}}$$



- Orthogonal basis vectors \underline{B}_j are the eigenvectors of H_{ij} along the principal axes of the χ^2 contours.**

$$\underline{\mu}(\beta) = \underline{\mu}(\hat{\beta}) + \sum_j \underline{B}_j \Delta\beta_j + \dots$$

$$\chi^2(\beta) = \chi^2(\hat{\beta}) + \sum_{i,j} \left(\frac{\Delta\beta_j}{\sigma(\beta_i)} \right)^2 + \dots$$

$$\frac{1}{2} \frac{\partial^2 \chi^2}{\partial \beta_i \partial \beta_j} \Big|_{\beta=\hat{\beta}} = \frac{\delta_{ij}}{\sigma^2(\beta_i)}$$

Hessian Matrix for Non-Linear Models

Model and derivatives: $\underline{\mu}(\alpha) \quad \frac{\partial \underline{\mu}}{\partial \alpha_k} \equiv \underline{\mathbf{P}}_k \quad \frac{\partial^2 \underline{\mu}}{\partial \alpha_k \partial \alpha_j} \equiv \underline{\mathbf{C}}_{kj}$

M Gradient vectors: $\underline{\mathbf{P}}_k$ $M(M-1) / 2$ Curvature vectors: $\underline{\mathbf{C}}_{jk}$

Badness-of-fit: $\chi^2 = \sum_{i=1}^N \left(\frac{x_i - \mu_i}{\sigma_i} \right)^2 = \|\underline{\mathbf{x}} - \underline{\mu}\|^2$

$$\frac{\partial \chi^2}{\partial \alpha_k} = -2 \sum_{i=1}^N \frac{x_i - \mu_i}{\sigma_i^2} \frac{\partial \mu_i}{\partial \alpha_k} = -2(\underline{\mathbf{x}} - \underline{\mu}) \cdot \underline{\mathbf{P}}_k$$

Hessian Matrix: $H_{jk} = \frac{1}{2} \frac{\partial^2 \chi^2}{\partial \alpha_j \partial \alpha_k} = \underline{\mathbf{P}}_j \cdot \underline{\mathbf{P}}_k - (\underline{\mathbf{x}} - \underline{\mu}) \cdot \underline{\mathbf{C}}_{jk}$

Best fit: $0 = \frac{\partial \chi^2}{\partial \alpha_k} = -2(\underline{\mathbf{x}} - \underline{\mu}) \cdot \underline{\mathbf{P}}_k \Rightarrow \underline{\mu} \cdot \underline{\mathbf{P}}_k = \underline{\mathbf{x}} \cdot \underline{\mathbf{P}}_k$

Parameter Error Bars: $\text{Cov}[\alpha_j, \alpha_k] = H^{-1}_{jk}$

Linear Model: $\underline{\mathbf{C}}_{jk} = 0 \quad H_{jk} = \underline{\mathbf{P}}_j \cdot \underline{\mathbf{P}}_k$

2. Gram-Schmidt Orthogonalization

The Gram-Schmidt process:

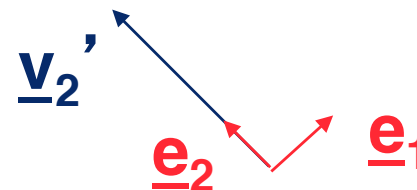
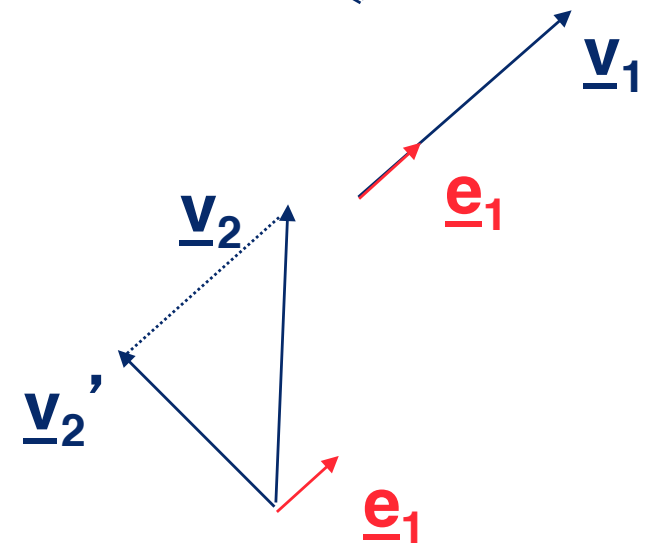
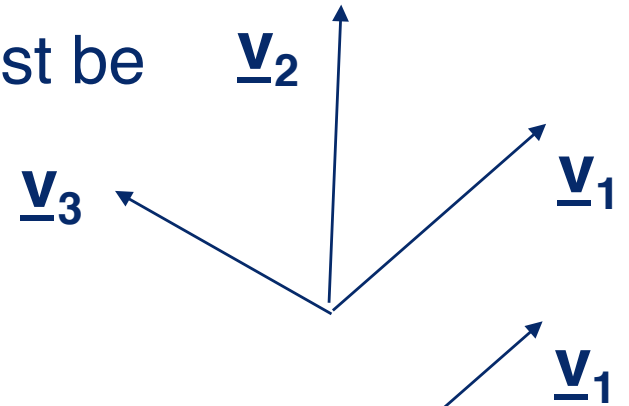
- 1. Start with M vectors $\underline{\mathbf{v}}_i$, $i = 1 \dots M$. They must be independent, i.e. no two of them parallel.
- 2. Normalize vector $\underline{\mathbf{v}}_1$:

$$\underline{\mathbf{e}}_1 \equiv \frac{\underline{\mathbf{v}}_1}{\|\underline{\mathbf{v}}_1\|}$$

- 3. Make $\underline{\mathbf{v}}_2'$ perpendicular to $\underline{\mathbf{e}}_1$:
 - i.e. subtract component of $\underline{\mathbf{v}}_2$ in direction of $\underline{\mathbf{e}}_1$

$$\underline{\mathbf{v}}_2' = \underline{\mathbf{v}}_2 - (\underline{\mathbf{v}}_2 \cdot \underline{\mathbf{e}}_1) \underline{\mathbf{e}}_1$$

- 4. Normalize $\underline{\mathbf{v}}_2'$: $\underline{\mathbf{e}}_2 \equiv \frac{\underline{\mathbf{v}}_2'}{\|\underline{\mathbf{v}}_2'\|}$



2. Gram-Schmidt Orthogonalization

- 5. Make $\underline{\mathbf{v}}_3'$ perpendicular to $\underline{\mathbf{e}}_1$:

$$\underline{\mathbf{v}}_3' = \underline{\mathbf{v}}_3 - (\underline{\mathbf{v}}_3 \cdot \underline{\mathbf{e}}_1) \underline{\mathbf{e}}_1$$

- 6. Make $\underline{\mathbf{v}}_3''$ perpendicular to $\underline{\mathbf{e}}_2$:

$$\underline{\mathbf{v}}_3'' = \underline{\mathbf{v}}_3' - (\underline{\mathbf{v}}_3' \cdot \underline{\mathbf{e}}_2) \underline{\mathbf{e}}_2$$

– Note: $\underline{\mathbf{v}}_3''$ is perpendicular to $\underline{\mathbf{e}}_1$ AND $\underline{\mathbf{e}}_2$.

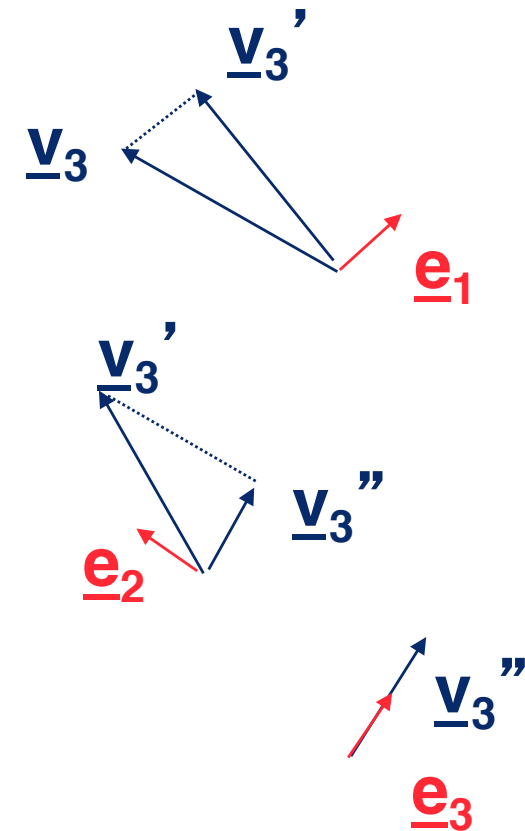
- 7. Normalize $\underline{\mathbf{v}}_3''$:

$$\underline{\mathbf{e}}_3 \equiv \frac{\underline{\mathbf{v}}_3''}{\left\| \underline{\mathbf{v}}_3'' \right\|}$$

... and so on, make $\underline{\mathbf{v}}_4$ perpendicular to $\underline{\mathbf{e}}_1$, $\underline{\mathbf{e}}_2$, $\underline{\mathbf{e}}_3$ and normalise to get $\underline{\mathbf{e}}_4$

Repeat up to $\underline{\mathbf{v}}_M$ to get **complete ortho-normal basis** $\underline{\mathbf{e}}_1, \underline{\mathbf{e}}_2, \dots, \underline{\mathbf{e}}_M$.

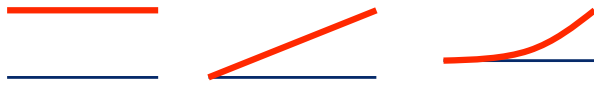
Easy to code ! (Try it !)



Orthogonal Polynomials

non-orthogonal polynomial:

$$y = A + Bx + Cx^2$$



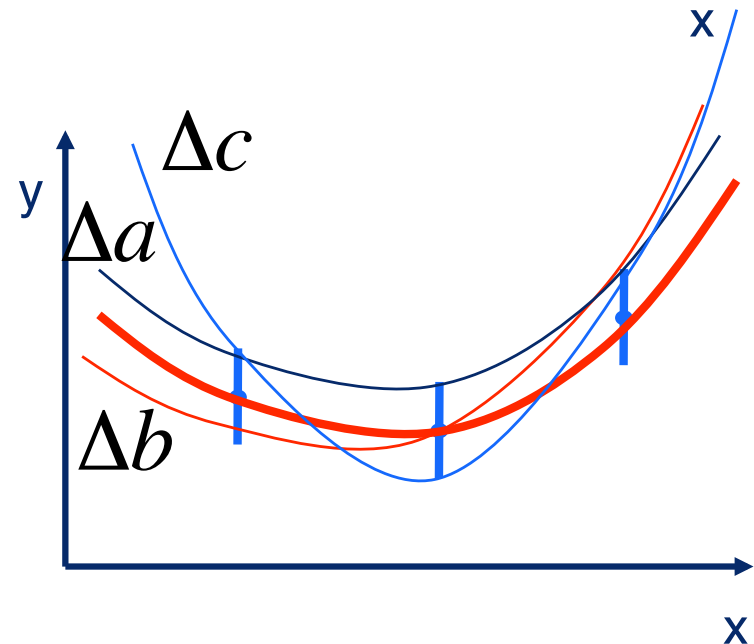
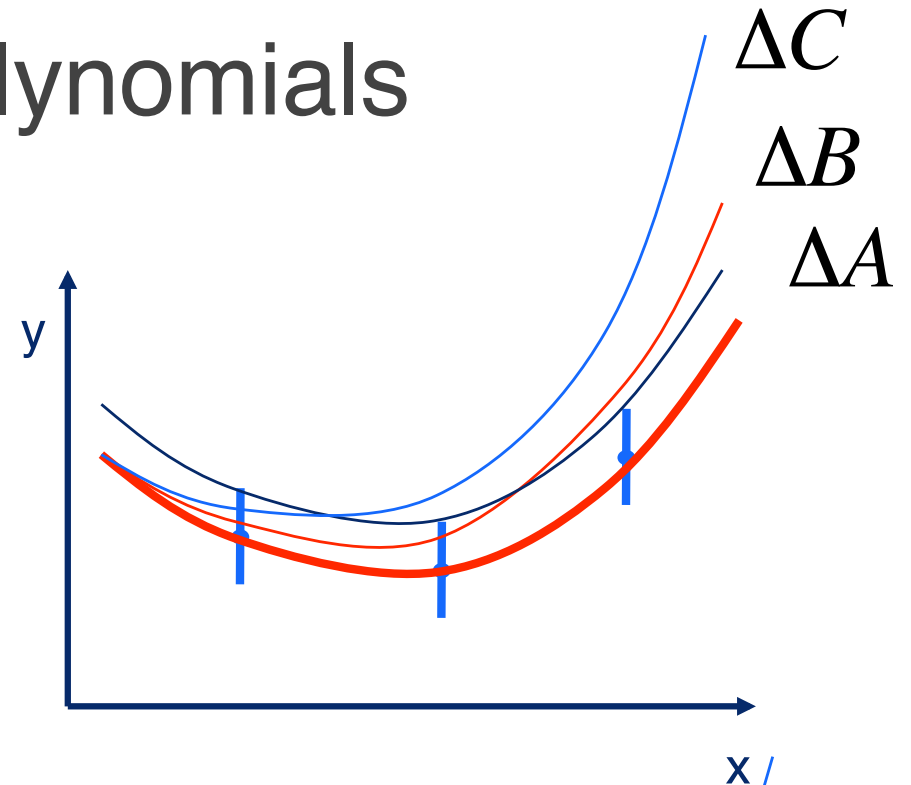
orthogonal polynomials:

$$P_i \cdot P_j \equiv \sum_{k=1}^N \frac{P_i(x_k) P_j(x_k)}{\sigma_k^2} = \frac{\delta_{ij}}{\text{Var}[\alpha_i]}$$

$$y = a P_0(x) + b P_1(x) + c P_2(x)$$



Note: every dataset has its own $1/\sigma^2$ weights, defines its own orthogonal polynomials.



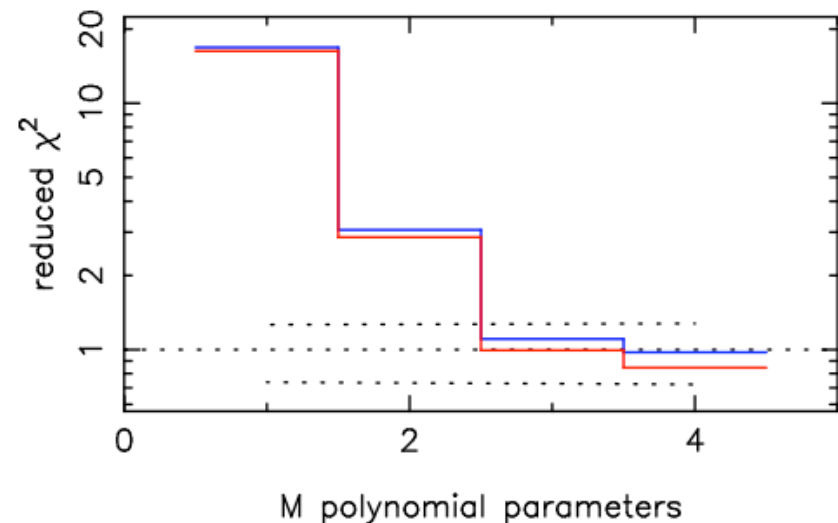
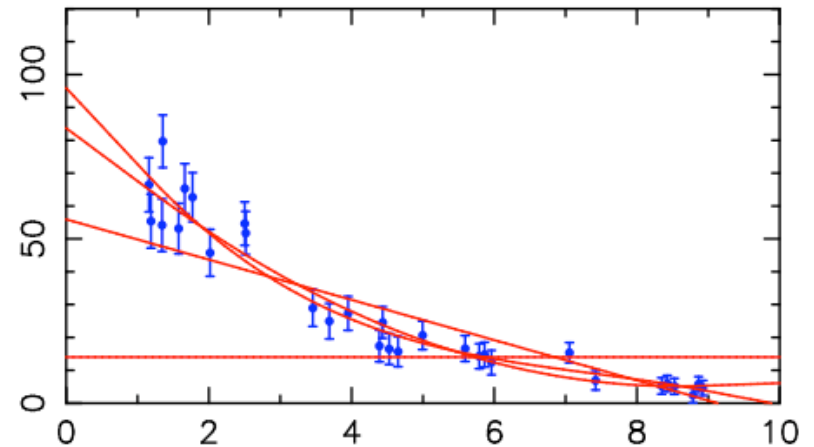
3. Differences between successive χ^2 fits

- Fit: $A + Bx + Cx^2$
 - A, B, C are not independent
 - $1, x, x^2$ are not orthogonal
- If $P_k(x)$ is a polynomial of degree k fitted to the data, then $P_k(x) - P_{k-1}(x)$ are **orthogonal**:
- $a P_0(x) + b [P_1(x) - P_0(x)] + c [P_2(x) - P_1(x)]$
 - a, b, c are independent



Note: every dataset has its own $1/\sigma^2$ weights, defines its own orthogonal polynomials.

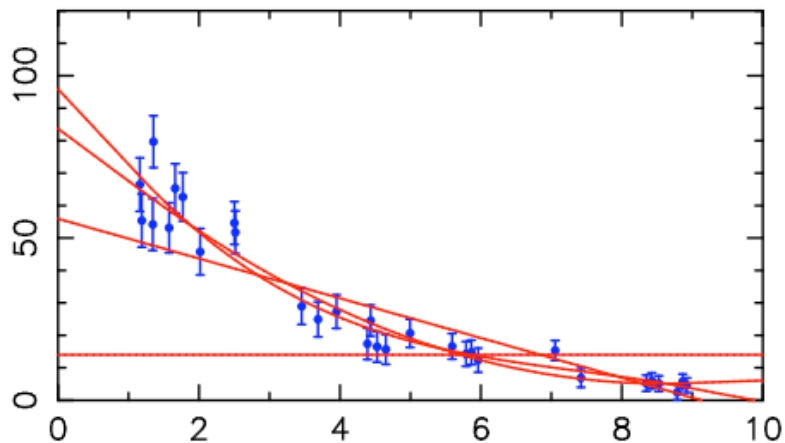
Polynomial Fit $N = 30$ $M = 1 \dots 4$



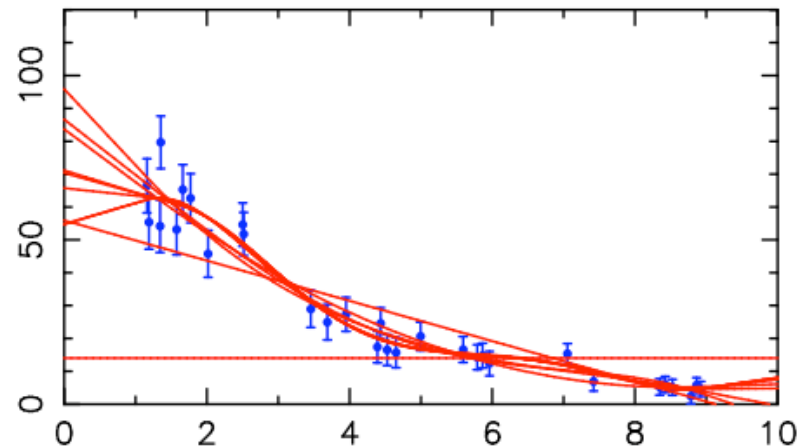
Fake data, true model is an exponential.

Polynomial Fits

Polynomial Fit $N = 30$ $M = 1 \dots 4$



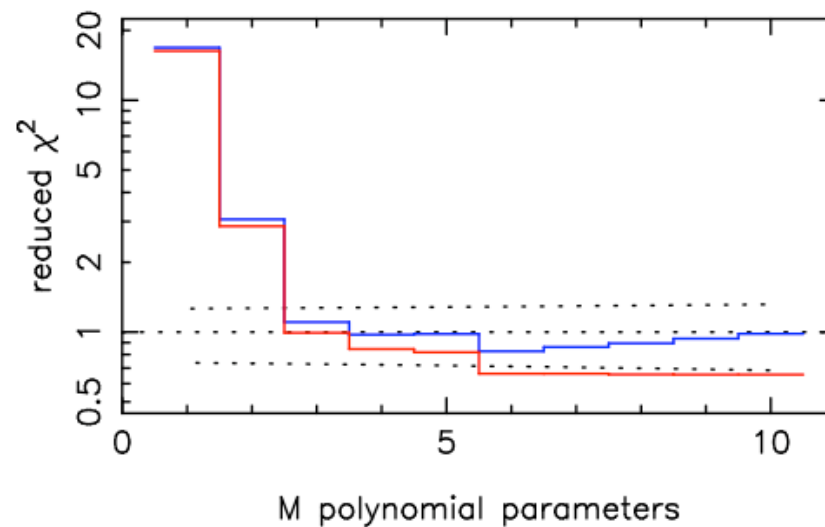
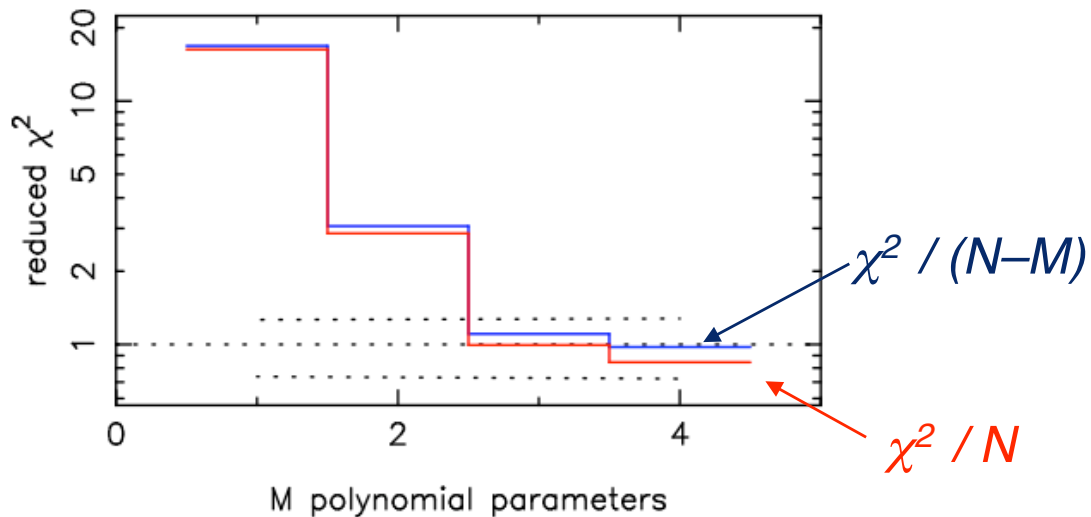
Polynomial Fit $N = 30$ $M = 1 \dots 10$



Badness-of-Fit Statistic

$$\chi^2 / N \rightarrow 0 \text{ as } M \rightarrow N$$

$$\frac{\chi^2}{N - M} \approx 1 \pm \left(\frac{2}{N - M} \right)^{1/2}$$



How many parameters to use ?

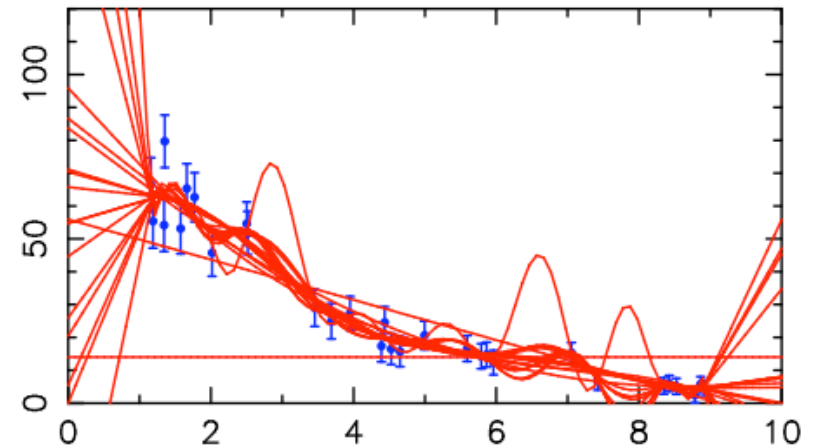
Fit $N = 30$ points with
 $M = 1, 2, \dots, 20$ polynomial
coefficients.

Higher $M =$ more flexible model.
Lower χ^2 , but less satisfactory fit.

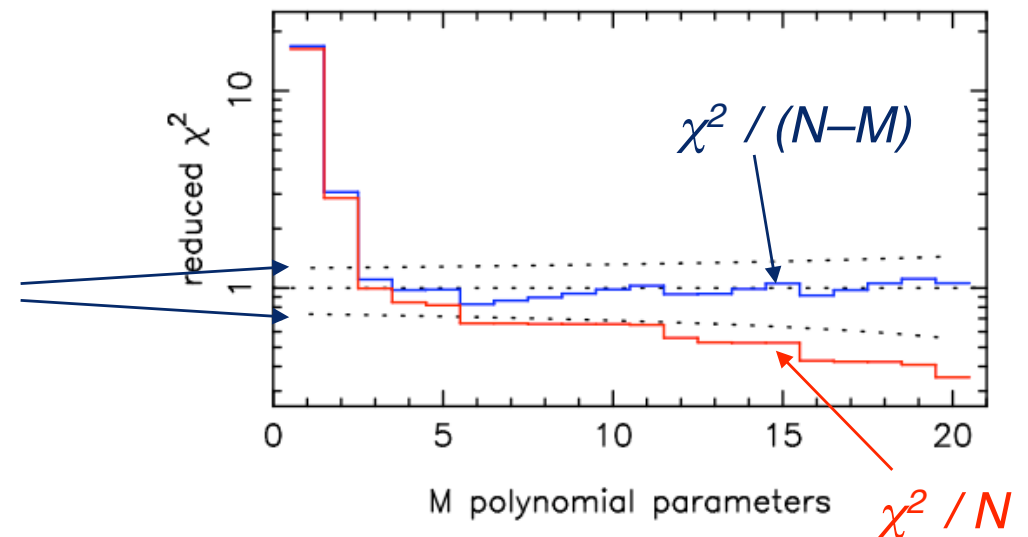
χ^2_{\min} rejects $M = 1, 2$.
accepts $M = 3, 4, \dots$

$$\frac{\chi^2}{N - M} \approx 1 \pm \sqrt{\frac{2}{N - M}}$$

Polynomial Fit $N = 30$ $M = 1 \dots 20$



Note "flailing" in data gaps
and beyond ends for high M



How many parameters to use ?

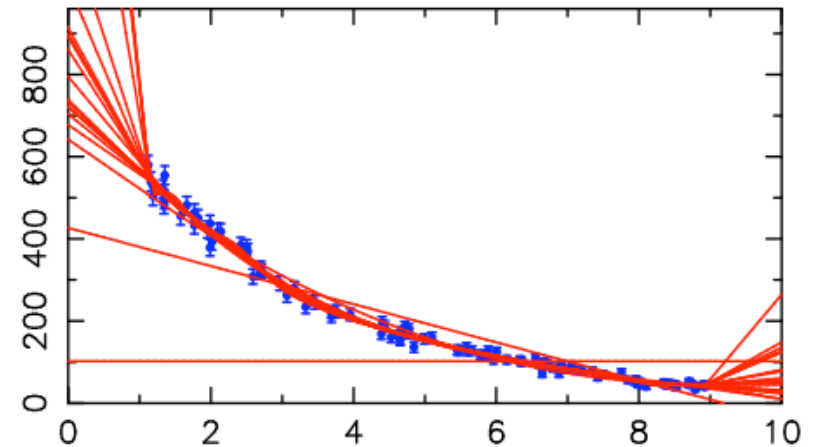
Fit $N = 100$ points with
 $M = 1, 2, \dots, 20$ polynomial
coefficients.

More data points and smaller
error bars than before.

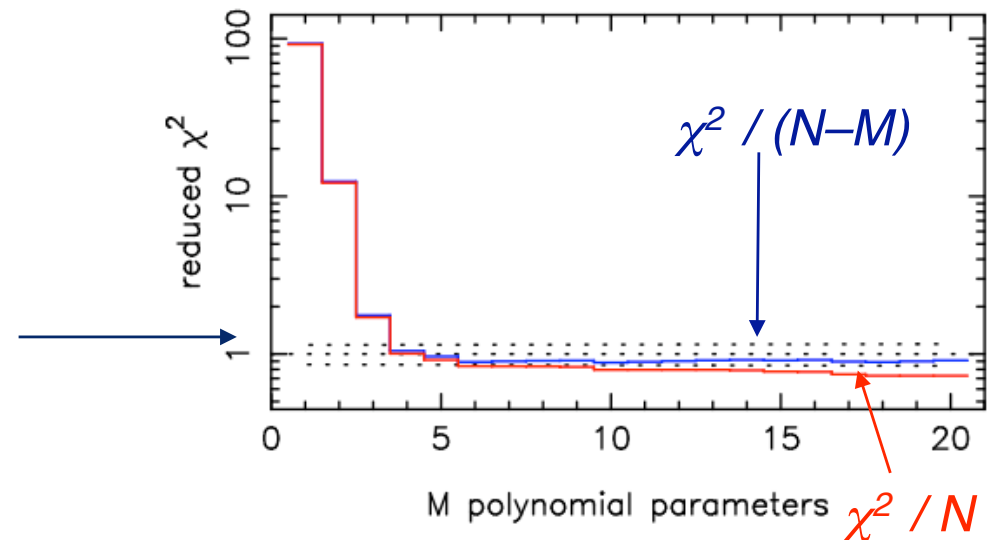
χ^2_{\min} rejects $M = 1, 2, 3$.
accepts $M = 4, 5, \dots$

$$\frac{\chi^2}{N - M} \approx 1 \pm \sqrt{\frac{2}{N - M}}$$

Polynomial Fit $N = 100$ $M = 1 \dots 20$



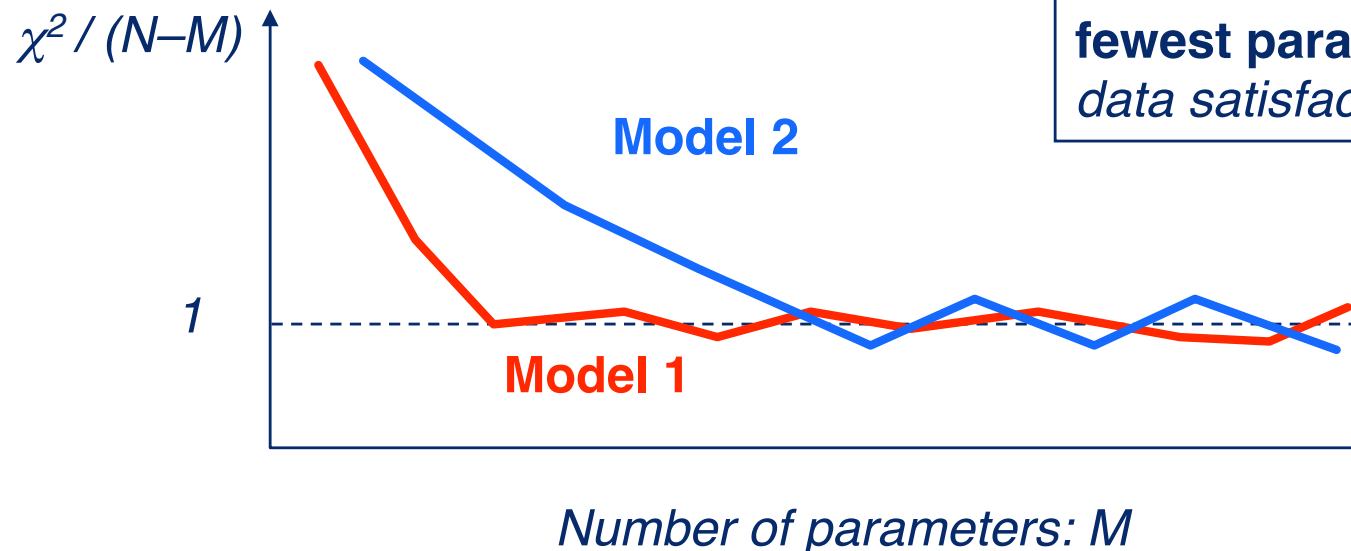
Note flailing beyond the range
of the data for high M



Occam's Razor - "Keep it Simple"

- William of Occam (ca. 1286–1347):
"It is futile to do with more, what can be done with fewer"
or: "Keep it simple, stupid!"
- Fit 2 different models, 1 and 2, to the same data.
- Each may be embellished with M parameters, e.g. increasing numbers of polynomial coefficients.
- **Prefer the simpler model.**

*Choose the **simplest model**, i.e. the model that needs the **fewest parameters** to fit the data satisfactorily.*



Information Criteria: AIC, AICc, BIC

- Each parameter improves the fit: $-2 \ln(L)$ decreases.
- Include a penalty for each new parameter.
- Does the reduction in $-2 \ln(L)$ offset the penalty?

Datapoints: N Parameters: M Likelihood: L

Akaike Information Criterion: $AIC \equiv -2 \ln(L) + 2 M$

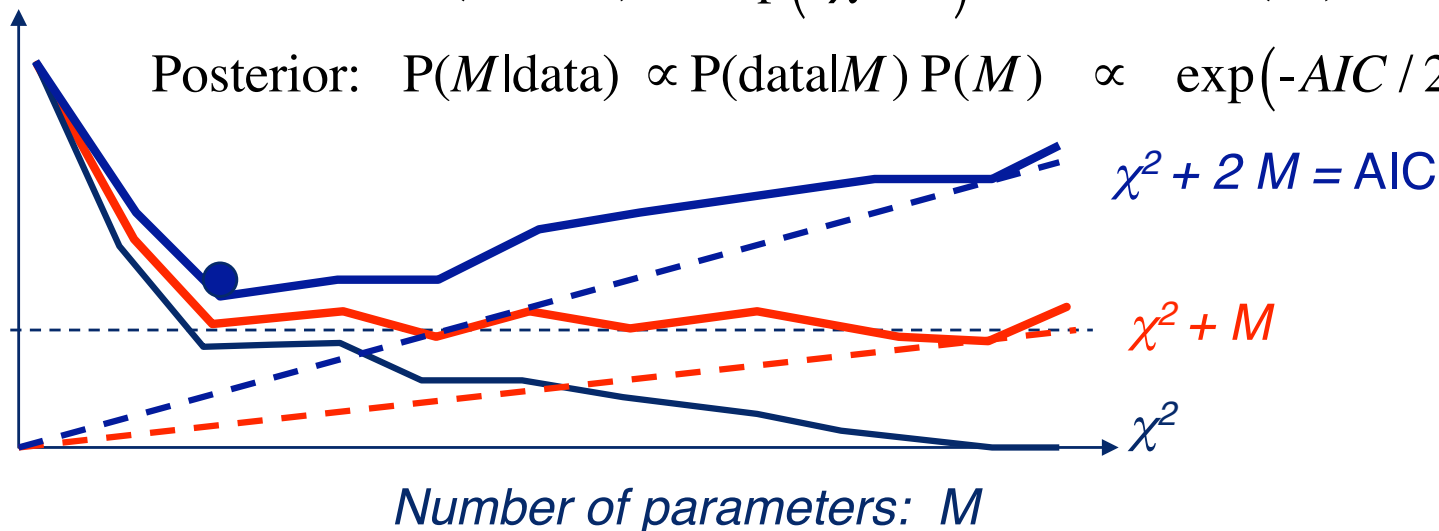
Corrected AIC: $AICc \equiv -2 \ln(L) + 2M / \left(1 - \frac{M-1}{N}\right)$

Bayesian Information Criterion: $BIC \equiv -2 \ln(L) + \ln(N) M$

Minimise the AIC
(or **AICc** or **BIC**) to choose the **simplest model** (needing the **fewest parameters**) that fits the data.

Likelihood: $P(\text{data}|M) \propto \exp(-\chi^2 / 2)$ Prior: $P(M) \propto \exp(-M)$

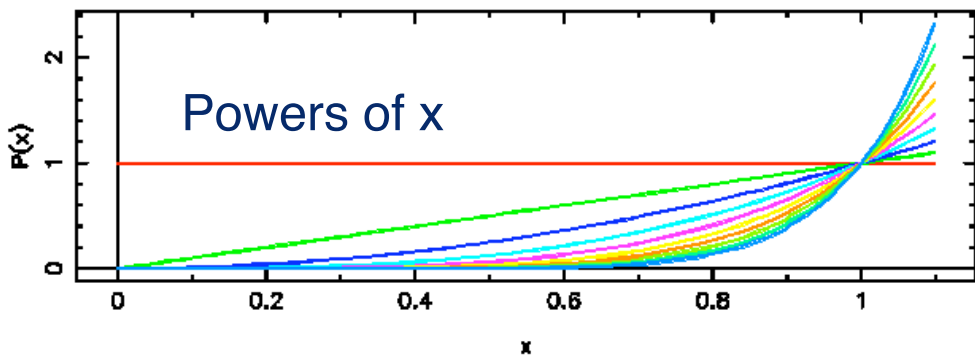
Posterior: $P(M|\text{data}) \propto P(\text{data}|M) P(M) \propto \exp(-AIC / 2)$



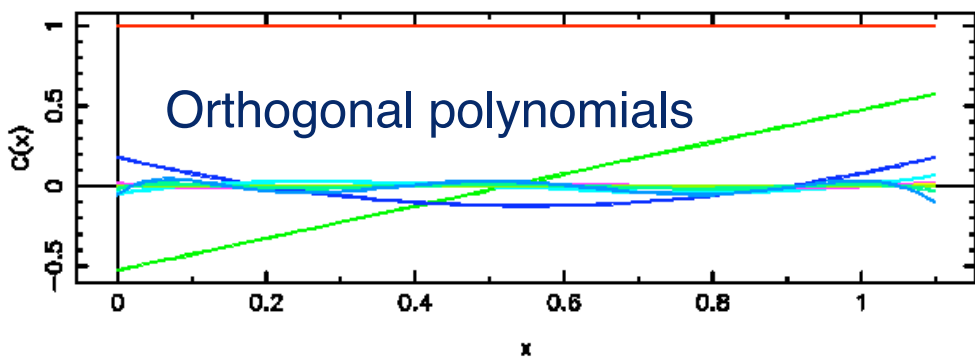
BIC may be better than AIC.

Comparison of AIC, AICc, BIC

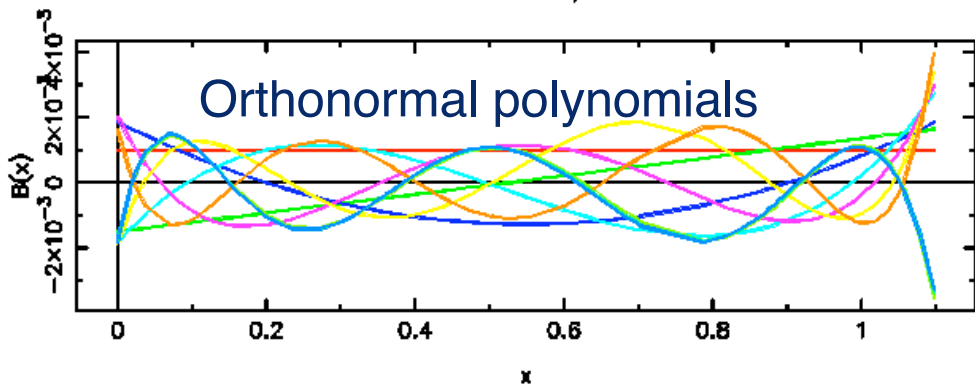
10 original patterns



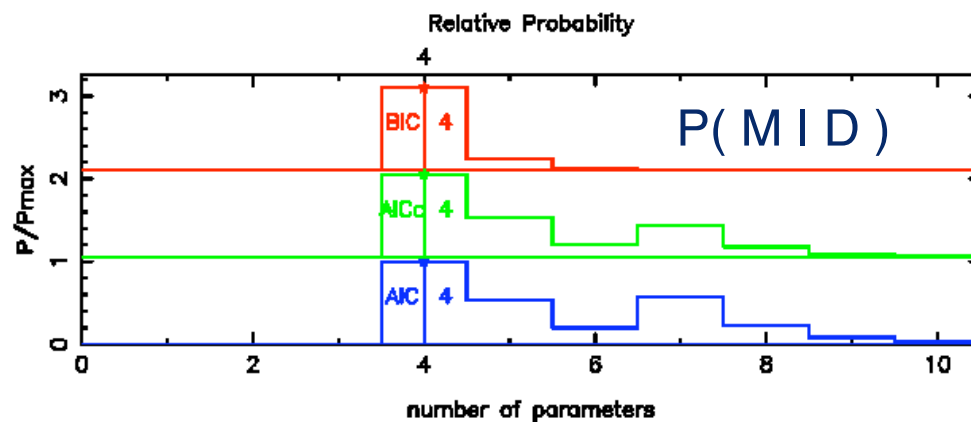
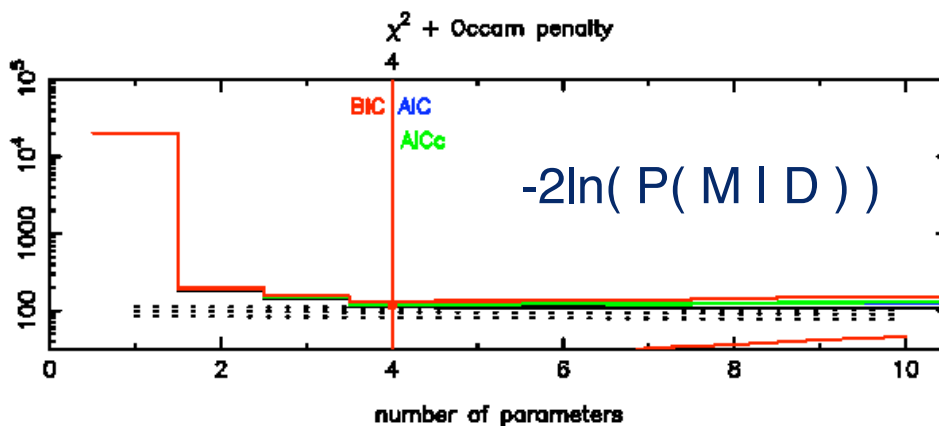
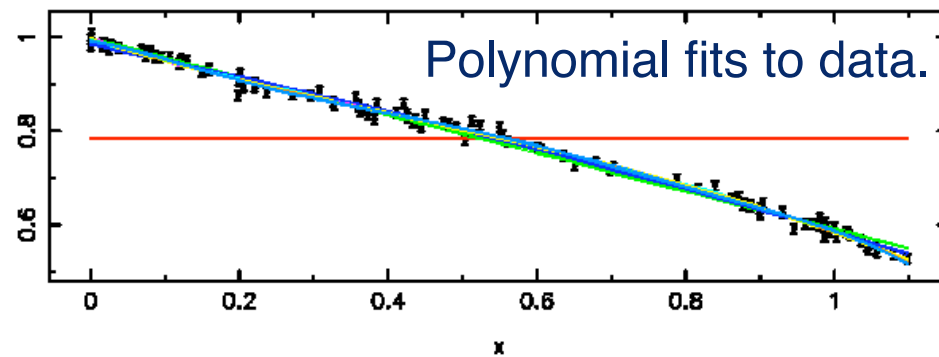
10 orthogonal patterns



10 orthonormal patterns

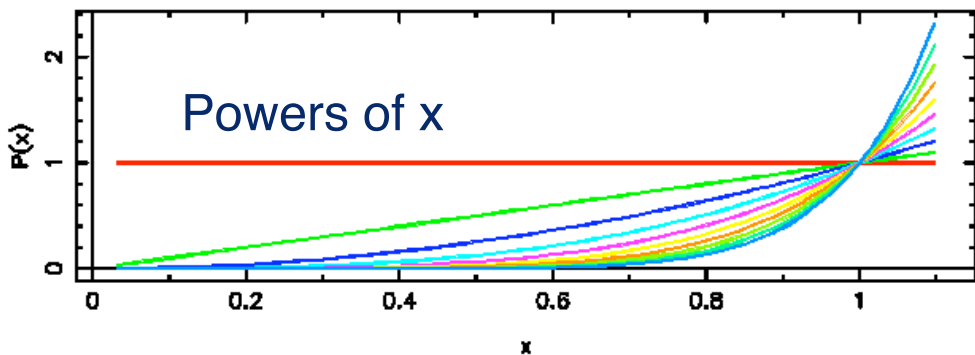


Poly 4 Data 100 σ 0.0100 Patterns 10

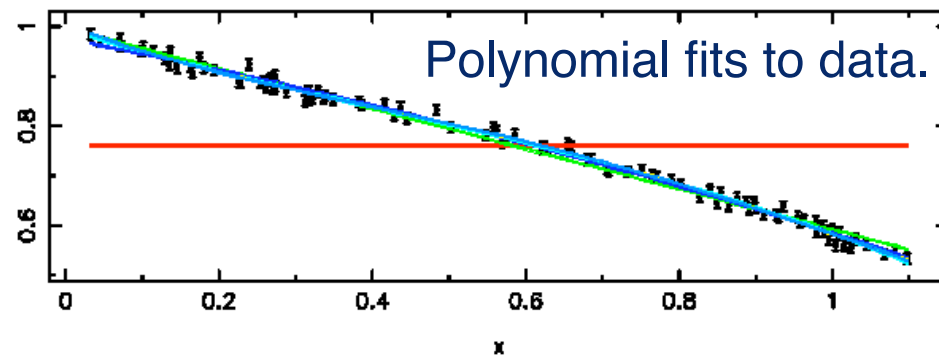


Comparison of AIC, AICc, BIC

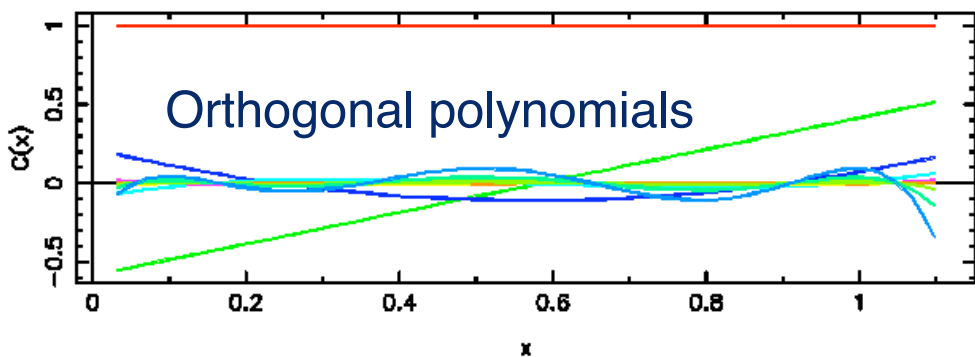
10 original patterns



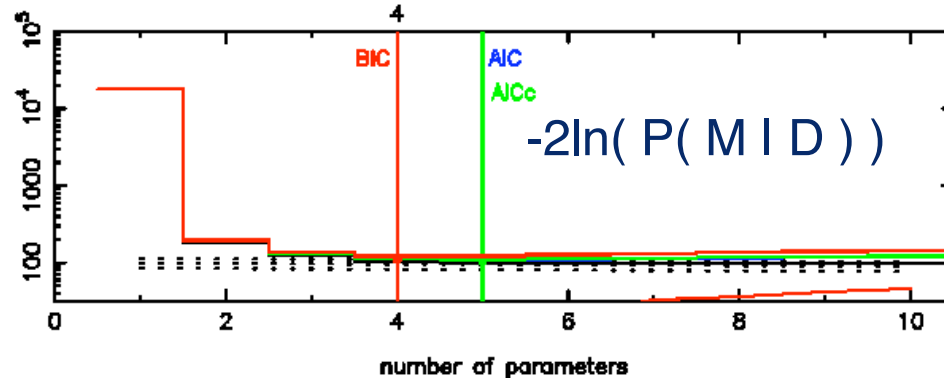
Poly 4 Data 100 σ 0.0100 Patterns 10



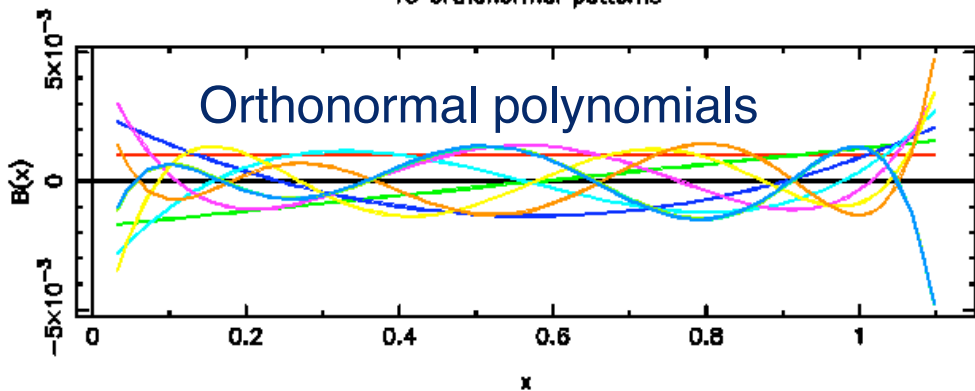
10 orthogonal patterns



$\chi^2 + \text{Occam penalty}$



10 orthonormal patterns



Relative Probability

