Statistical Machine Learning

Christian Walder

College of Engineering and Computer Science
The Australian National University

Canberra
Semester One, 2020.

(Many figures from C. M. Bishop, "Pattern Recognition and Machine Learning")

Outlines

Part III

Linear Regression

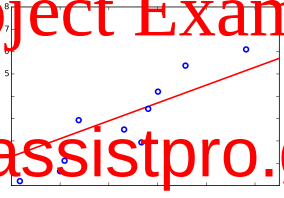
$N = 10$

$\mathbf{x} \equiv (x_1, \dots, x_N)^T$

$\mathbf{t}$

x_i

t_i



- Predictor $y(x, \mathbf{w})$?
- Performance measure?
- Optimal solution w^*?
- Recall: projection, inverse

Probabilities, Losses

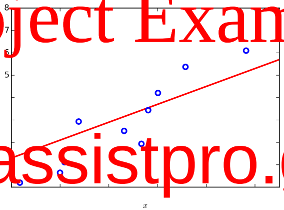
- Ga
- Ba
- Exp

Linear Curve Fitting - Least Squares

$$N = 10$$

$$\mathbf{x} \equiv (x_1, \ldots, x_N)^T$$

$$\mathbf{t} \equiv (t_1, \ldots, t_N)^T$$



$y(x,$

$$X \equiv [\mathbf{x} \quad 1]$$

$$w^* = (X^T X)^{-1} X^T \mathbf{t}$$

We assume

$$t = \underbrace{y(\mathbf{x}, \mathbf{w})}_{\text{deterministic}} + \underbrace{\epsilon}_{\text{Gaussian noise}}$$

Curve fitting - revisited

- *a priori* belief about the parameter $\mathbf{w}$ captured in the prior probability $p(\mathbf{w})$
- obs
- cal
  obs

$$\overline{}$$
$$p(\mathcal{D})$$

- $p(\mathcal{D}\,|\,\mathbf{w})$ as a function of $\mathbf{w}$: likelihood fun
- likelihood expresses how probable the data ar different values of $\mathbf{w}$ — it is not a probability d respect $\mathbf{w}$ (but it is with respect to $\mathcal{D}$; prove it)

- Consider the linear regression problem, where we have random variables $\mathbf{x}_n$ and t_n.
- We
- We

For a given θ the density defines the probab observing $t_n|\mathbf{x}_n$.

- We are interested in finding θ that maxi probability (called the likelihood) of the data.

Likelihood function $p(\mathcal{D} \mid \mathbf{w})$

Frequentist Approach

-
- v
  some 'estimator'
- error bars on the estimated $\mathbf{w}$ obtained from the distribution of possible data sets $\mathcal{D}$

Bayesian Approach

-
- par
  fro
  distr

Frequentist Estimator - Maximum Likelihood

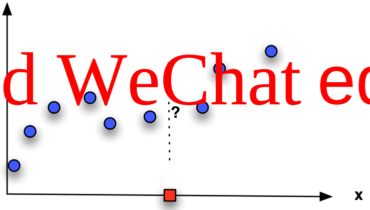
- choose $\mathbf{w}$ for which the likelihood $p(\mathcal{D} \mid \mathbf{w})$ (the probability of the observed data) is maximal
- the most common heuristic for learning a single fixed $\mathbf{w}$
- equ
  fun
- log i
- maximising the likelihood $\iff$ minimi
- Example: Fair-looking coin is tossed three tim
  landing on heads.
- Maximum likelihood estimate of the probability of landing heads will give 1.

Bayesian Approach

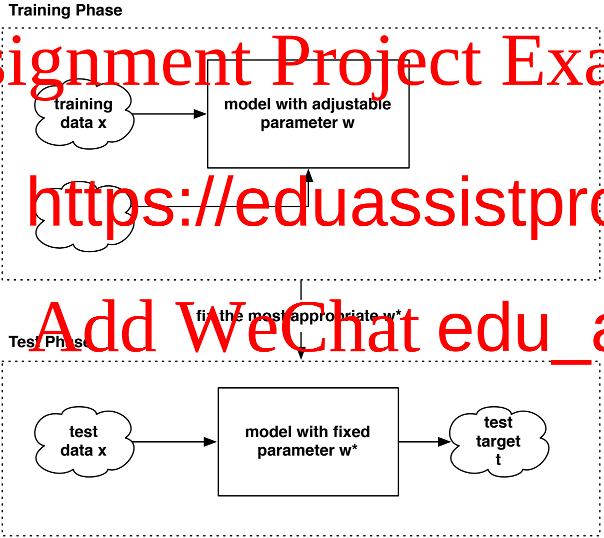
- including prior knowledge easy (via prior $\mathbf{w}$)
- subjective choice of prior, allows better results by incorporating domain knowledge
- sometimes choice of prior motivated by convinient mathematical form
- prio
- nee
  -
  - advances in approximation schemes (Vari Expectation Propagation)
- there is no true $\mathbf{w}$.

Regression

- Given a training data set of N observations $\{\mathbf{x}_n\}$ and target values t_n.
- Goal : Learn to predict the value of one ore more target values t given a new value of the input .
- Exa

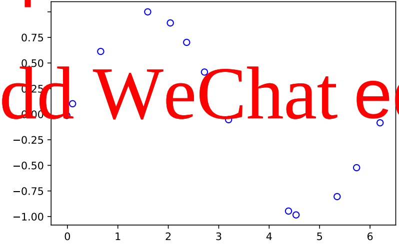


Supervised Learning: (non-Bayesian) Point Estimate



Why Linear Regression?

- Analytic solution when minimising sum of squared errors
- Well understood statistical behaviour
- Efficient algorithms exist for convex losses and regularizers
- But



Linear Basis Function Models

- Linear combination of fixed nonlinear basis functions

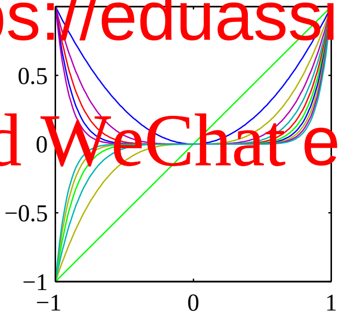
- parameter $\mathbf{w} = (w_0, \ldots, w_{M-1})$
- basis functions $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \ldots, \phi_{M-}$
- convention $\phi_0(\mathbf{x}) = 1$
- w_0 is the bias parameter

Polynomial Basis Functions

- Scalar input variable x

$$\phi_j(x) = x^j$$

- Limitation : Polynomials are global functions of the input variable x so the learned function will extrapolate poorly

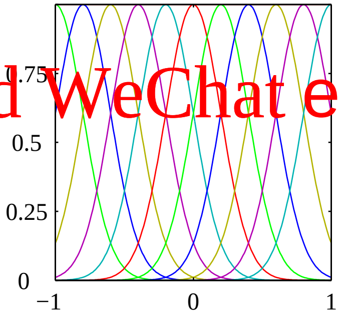


'Gaussian' Basis Functions

- Scalar input variable x

$$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

- Not a probability distribution.
- No normalisation required, taken care of by the model par
- We



Sigmoidal Basis Functions

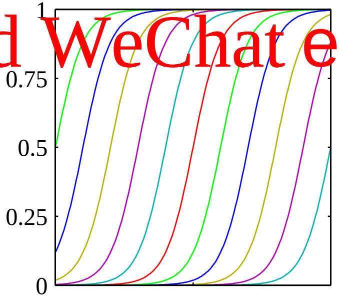
- Scalar input variable x

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right)$$

where $\sigma(a)$ is the logistic sigmoid function defined by

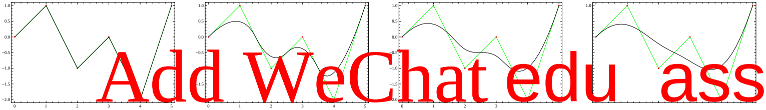
$$\frac{1}{\quad}$$

- $\sigma(a$ … tan



Other Basis Functions

- Fourier Basis : each basis function represents a specific frequency and has infinite spatial extent.
- Wavelets : localised in both space and frequency (also mutually orthogonal to simplify appliciation).
- Spli
inp
sm



Linear Splines | Quadratic Splines | Cubic Splines | Splines

Approximate the points $\{(0,0),(1,1),(2,-1),(3,0),(4,-2),(5,1)\}$ by different splines.

Maximum Likelihood and Least Squares

- No special assumption about the basis functions $\phi_j(\mathbf{x})$. In the simplest case, one can think of $\phi_j(\mathbf{x}) = x_j$, or $\phi(\mathbf{x}) = \mathbf{x}$.
- Assume target t is given by

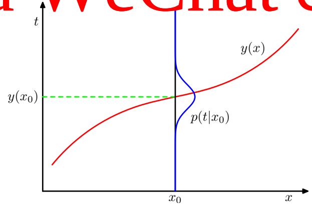
$$t = \underline{y(\mathbf{x}, \mathbf{w})} + \epsilon$$

wh

pre

- Thus

$$p(t \mid \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(\mathbf{x}, \mathbf{w}$$



- Likelihood of one target t given the data $\mathbf{x}$ was

$$p(t \mid \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t \mid y(\mathbf{x}, \mathbf{w}), \beta^{-1})$$

- Now, a set of inputs $\mathbf{X}$ with corresponding target values $\mathbf{t}$.
- Ass
(i.i.
sa

$$p(\mathbf{t} \mid \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(\mathbf{x}_n$$

$$= \prod_{n=1}^{N} \mathcal{N}(t_n \mid \mathbf{w}^T \phi \quad n$$

- From now on drop the conditioning variable $\mathbf{X}$ from the notation, as with supervised learning we do not seek to model the distribution of the input data.

- Consider the logarithm of the likelihood $p(\mathbf{t}\,|\,\mathbf{w},\beta)$ (the logarithm is a monotone function!)

$$\ln p(\mathbf{t}\,|\,\mathbf{w},\beta) = \sum^{N} \ln \mathcal{N}(t_n\,|\,\mathbf{w}^T\phi(\mathbf{x}_n),\beta^{-1})$$

$$= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta$$

where the sum-of-squares error function is

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^T\phi(x_n)\}^2.$$

- $\arg\max_{\mathbf{w}} \ln p(\mathbf{t}\,|\,\mathbf{w},\beta) \to \arg\min_{\mathbf{w}} E_D(\mathbf{w})$

- Goal: Find a more compact representation.
- Rewrite the error function

N

$-$ $-$

wh

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \dots & \phi \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \dots & \phi \\ \vdots & \vdots & \ddots & \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \dots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix}$$

- The log likelihood is now

$$\ln p(\mathbf{t} \mid \mathbf{w}, \beta) = \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w})$$
$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta \frac{1}{2} (\mathbf{t} - \mathbf{\Phi w})^T (\mathbf{t} - \mathbf{\Phi w})$$

- Fin
- The

$$\nabla_{\mathbf{w}} \ln \quad (\; | \quad , \beta) = \beta \mathbf{\Phi} \; (\qquad \mathbf{\Phi} \quad).$$

Setting the gradient to zero gives

$$0 = \mathbf{\Phi}^T \mathbf{t} - \mathbf{\Phi}^T \mathbf{\Phi w},$$

- which results in

$$\mathbf{w}_{ML} = (\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{t} = \mathbf{\Phi}^\dagger \mathbf{t}$$

where $\mathbf{\Phi}^\dagger$ is the Moore-Penrose pseudo-inverse of the matrix $\mathbf{\Phi}$.

Maximum Likelihood and Least Squares

- The log likelihood with the optimal $\mathbf{w}_{ML}$ is now

$$\ln p(\mathbf{t} \,|\, \mathbf{w}_{ML}, \beta)$$
$$= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta\frac{1}{2}(\mathbf{t} - \mathbf{\Phi}\mathbf{w}_{ML})^T(\mathbf{t} - \mathbf{\Phi}\mathbf{w}_{ML})$$

- Fin

results in

$$\frac{1}{\beta_{ML}} = \frac{1}{N}(\mathbf{t} - \mathbf{\Phi}\mathbf{w}_{ML})^T(\mathbf{t} -$$

- Note: We can first find the maximum likelihood for $\mathbf{w}$ as this does not depend on β. Then we can use $\mathbf{w}_{ML}$ to find the maximum likelihood solution for β.
- Could we have chosen optimisation wrt β first, and then wrt to $\mathbf{w}$?

Sequential Learning - Stochastic Gradient Descent

- For large data sets, calculating the maximum likelihood parameters $\mathbf{w}_{ML}$ and β_{ML} may be costly.
- For
- Us
- If the
  then

  1. initialise $\mathbf{w}^{(0)}$ to some starting value
  2. update the parameter vector at iteration

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta$$

where E_n is the error function after presenting the nth data set, and η is the learning rate.

Sequential Learning - Stochastic Gradient Descent

- For t des

- The value for the learning rate must be chosen ca too large learning rate may prevent the algorith converging. A too small learning rate does follo too slowly.

Regularized Least Squares

- Add regularisation in order to prevent overfitting

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

wit

- Sim

$$E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

- Maximum likelihood solution

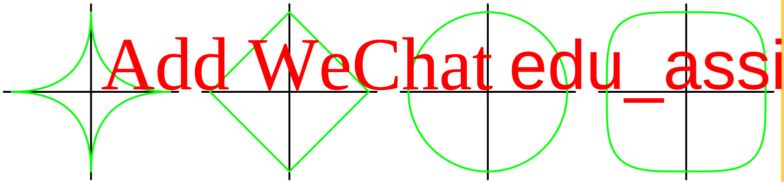
$$\mathbf{w} = \left(\lambda \mathbf{I} + \boldsymbol{\Phi}^T\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^T\mathbf{t}$$

Regularized Least Squares

- More general regulariser

$$E_W(\mathbf{w}) = \frac{1}{2}\sum_{j=1}^{M}|w_j|^q$$

- $q =$



$q = 0.5$ $q = 1$ $q = 2$ $q = 4$

Lagrangian Dual View of the Regulariser

- By the Lagrange multiplier method, minimization of the regularized error function

$$\sum^{N} \qquad \sum^{M}$$

- is eq
sum-of-squares error,

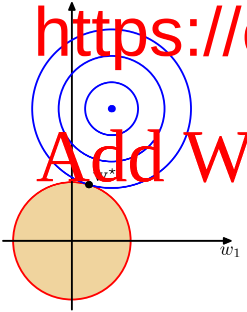
$$\frac{1}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^T\phi(\mathbf{x}_n))^2 \quad \text{subject to} \quad \sum_{j=1}$$

- This yields the figures on the next slide.

Comparison of Quadratic and Lasso Regulariser

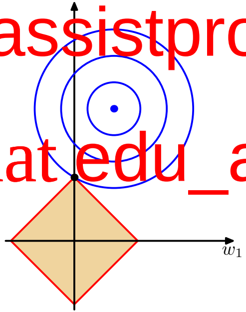
Quadratic regulariser

$$\frac{1}{2}\sum_{j=1}^{M} w_j^2$$



Lasso regulariser

$$\frac{1}{2}\sum_{j=1}^{M} |w_j|$$



Multiple Outputs

- More than 1 target variable per data point.
- $\mathbf{y}$ becomes a vector instead of a scalar. Each dimension can be treated with a different set of basis functions (and that
dim
- Her

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \boldsymbol{\phi}(\mathbf{x})$$

where $\mathbf{y}$ is a K-dimensional column vecto
matrix of model parameters, and
$\boldsymbol{\phi}(\mathbf{x}) = \Big(\phi_0(\mathbf{x}), \dots, \phi_{M-1}(\mathbf{x})\Big)$, with $\phi_0(\mathbf{x}) = 1$, as before.

- Define target matrix $\mathbf{T}$ containing the target vector $\mathbf{t}_n^T$ in the n^{th} row.

- Suppose the conditional distribution of the target vector is an isotropic Gaussian of the form

- Then

$$\ln p(\mathbf{T} \mid \mathbf{X}, \mathbf{W}, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n \mid \mathbf{W}^T \phi(\mathbf{x}$$

$$= \frac{NK}{2} \ln \left(\frac{\beta}{2\pi} \right) - \frac{\beta}{2}$$

Multiple Outputs

- Maximisation with respect to $\mathbf{W}$ results in

$$\mathbf{W}_{ML} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{T}.$$

- For

- The
decouples.
- Holds also for a general Gaussian noise distrib
arbitrary covariance matrix.

- Why? $\mathbf{W}$ defines the mean of the Gaussian noise distribution. And the maximum likelihood solution for the mean of a multivariate Gaussian is independent of the covariance.

Loss Function for Regression

- Ov
  and
- Re
  corr
- Frequentists viewpoint of the model complex
  bias-variance trade-off.

Loss Function for Regression

- Choose an estimator $y(\mathbf{x})$ to estimate the target value t for each input $\mathbf{x}$.
- Choose a loss function $L(t, y(\mathbf{x}))$ which measures the difference between the target t and the estimate $y(\mathbf{x})$.
- The
- Common choice: Squared Loss

$$L(t, y(\mathbf{x})) = \{y(\mathbf{x}) - t\}^2$$

- Expected loss for squared loss function

$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 \, p(\mathbf{x}, t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t.$$

Loss Function for Regression

- Expected loss for squared loss function

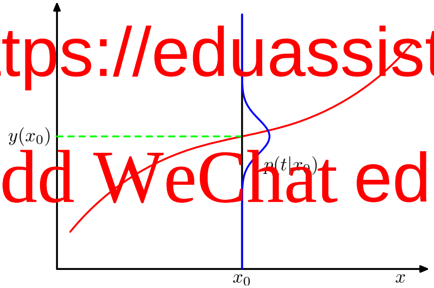
$$2$$

- Min

$$y(\mathbf{x}) = \frac{t\, p(\mathbf{x}, t)\, \mathrm{d}t}{p(\mathbf{x})} = \quad t\, p(t \mid \mathbf{x}$$

(calculus of variations is not required to derive th
we may work point-wise by fixing an $\mathbf{x}$
stationarity to solve for $y(\mathbf{x})$ — why is that sufficient?).

- The regression function which minimises the expected squared loss, is given by the mean of the conditional distribution $p(t \mid \mathbf{x})$.



Analysing the Squared Loss (1)

- Analyse the expected loss

$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 \, p(\mathbf{x}, t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t.$$

- Re

$$= \{y(\mathbf{x}) - \mathbb{E}[t \,|\, \mathbf{x}]\}^2 + \qquad {}^2$$

$$+ 2\{y(\mathbf{x}) - \mathbb{E}[t \,|\, \mathbf{x}]\}\{$$

- Claim

$$\iint \{y(\mathbf{x}) - \mathbb{E}[t \,|\, \mathbf{x}]\}\{\mathbb{E}[t \,|\, \mathbf{x}] - t\} \, p(\mathbf{x}, t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}t = 0.$$

- Claim

$$\iint \{y(\mathbf{x}) - \mathbb{E}\left[t \,|\, \mathbf{x}\right]\}\{\mathbb{E}\left[t \,|\, \mathbf{x}\right] - t\}\, p(\mathbf{x}, t)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t = 0$$

- Se
dep

- Calculate the integral over t

$$\begin{aligned}
\int \{\mathbb{E}\left[t \,|\, \mathbf{x}\right] - t\}\, p(\mathbf{x}, t)\, \mathrm{d}t &= \mathbb{E}\left[t \,|\, \mathbf{x}\right] p(\mathbf{x}) - p(\mathbf{x}) \qquad \frac{\qquad}{p(\mathbf{x})}\, \mathrm{d}t \\
&= \mathbb{E}\left[t \,|\, \mathbf{x}\right] p(\mathbf{x}) - p(\mathbf{x})\mathbb{E}\left[t \,|\, \mathbf{x}\right] \\
&= 0
\end{aligned}$$

- The expected loss is now

$\mathbb{E}$

- Min
already).
- Second term represents the intrinsic variabilit
target data (can be regarded as noise). Indepe
choice $y(\mathbf{x})$, can not be reduced by learning a be

The Bias-Variance Decomposition (1)

- Consider again squared loss for which the optimal prediction is given by the conditional expectation $h(x)$

- Sin (fini $\mathcal{D}$
- $\mathcal{D}$ is a finite sample from the unknown joint
- Notate the dependency of the learned function by $y(\mathbf{x}; \mathcal{D})$.
- Evaluate performance of algorithm by taking the expectation $\mathbb{E}_{\mathcal{D}}[L]$ over all data sets $\mathcal{D}$

The Bias-Variance Decomposition (2)

- Taking the expectation over data sets $\mathcal{D}$, using Eqn 1, and interchanging the order of expectations for the first term:

$$2$$

- Again, add and subtract the expectation

$$\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 = \{y(\mathbf{x};\mathcal{D}) -$$
$$+ \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};$$

and show that the mixed term vanishes under the expectation $\mathbb{E}_{\mathcal{D}}$.

The Bias-Variance Decomposition (3)

- Expected loss $\mathbb{E}_{\mathcal{D}}[L]$ over all data sets $\mathcal{D}$

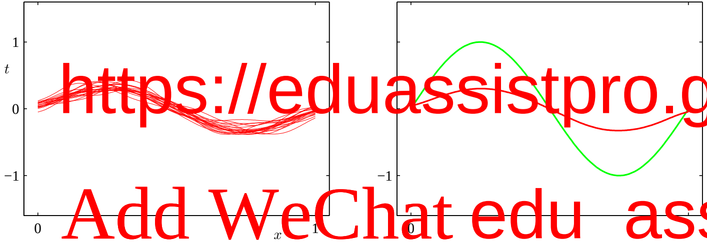
$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}.$$

where

$$\text{noise} = \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\, \mathrm{d}\mathbf{x}$$

- $(\text{bias})^2$: How accurate is a model across differe sets? (How much does the average prediction over all data sets differ from the desired regression function ?)
- variance : How sensitive is the model to small changes in the training set? (How much do solutions for individual data sets vary around their average ?

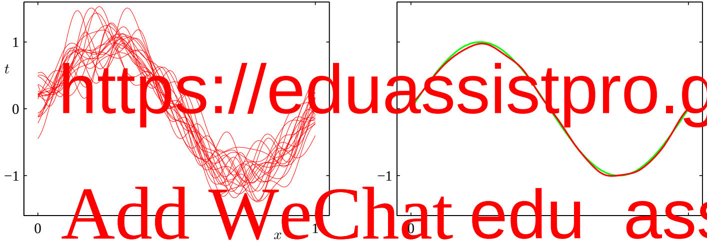
- Simple models have low variance and high bias.



Left: Result of fitting the model to 100 data s
Right: Average of the 100 fits in red, the sinusoidal function from where the data were created in green.

The Bias-Variance Decomposition

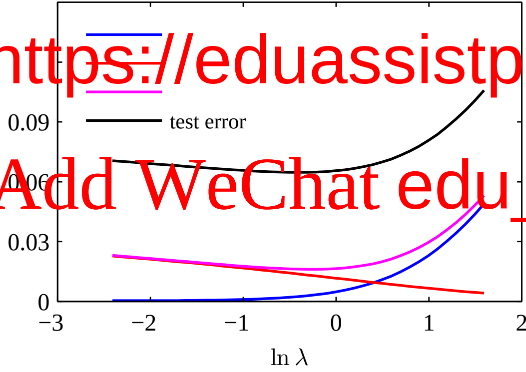
Complex models have high variance and low bias



Left: Result of fitting the model to 100 data s
Right: Average of the 100 fits in red, the sinusoidal function from where the data were created in green.

The Bias-Variance Decomposition

- Dependence of bias and variance on the model complexity
- Squared bias, variance, their sum, and test data
- The minimum for $(\text{bias})^2 + \text{variance}$ occurs close to the value that gives the minimum error



Unbiased Estimators

- You may have encountered *unbiased estimators*
- Why guarantee zero bias? To quote the pioneer of Bayesian inference, Edwin Jaynes, from his book *Pro*

The Bias-Variance Decomposition

- Tradeoff between bias and variance
  - simple models have low variance and high bias
  - complex models have high variance and low bias
- The
  mo
- Exp

$$\text{expected loss} = (\text{bias})^2 + \text{varia}$$

- The noise comes from the data, and can not be re
  from the expected loss.
- To analyse the bias-variance decomposition : many data sets needed, which are not always available.