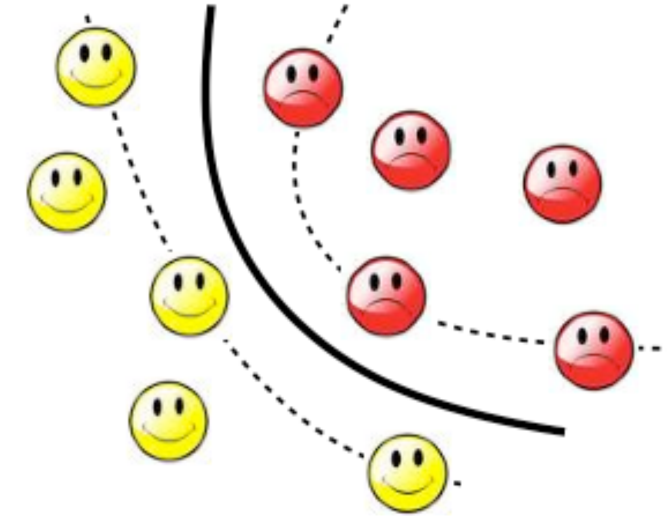
THE UNIVERSITY OF SYDNEY



Data Mining and Knowledge Discovery (COMP 5318)

Linear Regression

Roman Marchant

23 April 2018

Roman Marchant
roman.marchant@sydney.edu.au
Rm 542, Building J12, School of IT
Availability: Monday 1400-1600 (Appointment only)

Announcements

Assignment 1 is out (Due 07 May)

Build a classifier to classify apps from the Apps Market into a set of categories based on their descriptions.

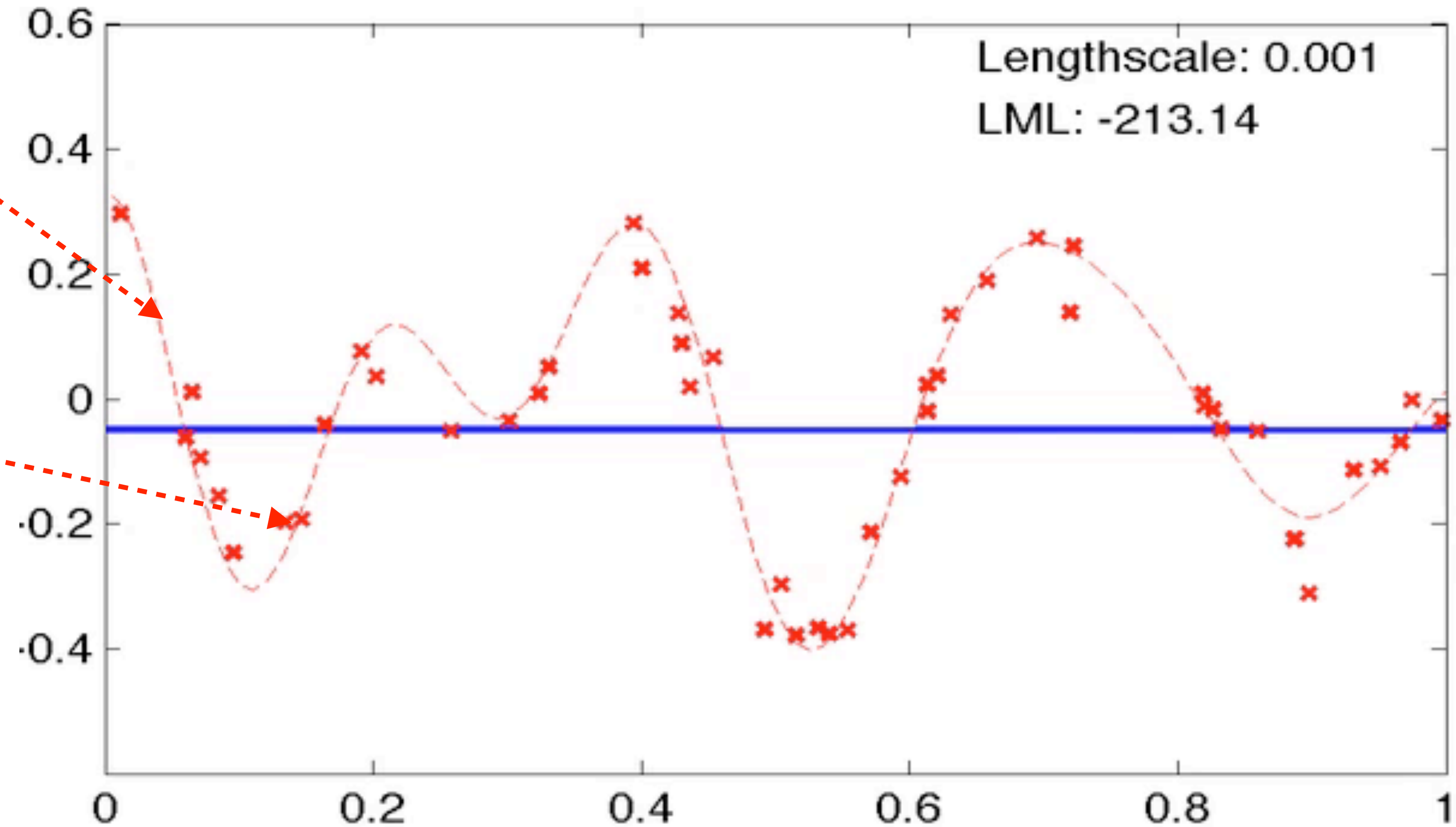
Implement own pre-processing and classification method.

Can you optimisation and linear algebra existing.packages

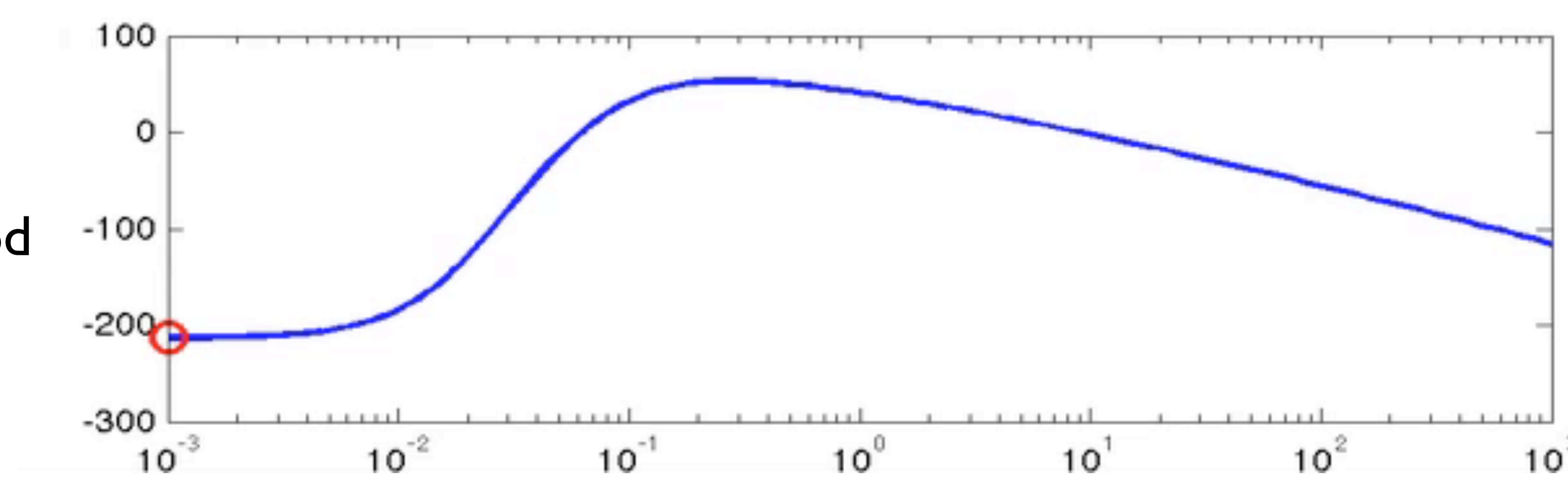
Learning as Optimisation

Unknown Function
$f(\mathbf{x})$

Noisy Samples
from f



Log Marginal Likelihood



THE UNIVERSITY OF SYDNEY

Linear Regression

C. Bishop, *Pattern Recognition and Machine Learning,* Chapter 3: Linear Models for Regression
Springer New York, 2006

Linear Regression

Goal: Predict the value of one or more continuous target variables t given the value of a D-dimensional vector x of input variables.

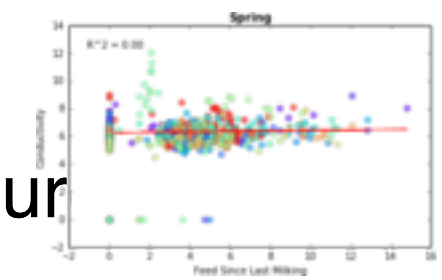
Given training data of size N: $\{(\mathbf{x}_n, t_n)\}_{n=1,\ldots,N}$

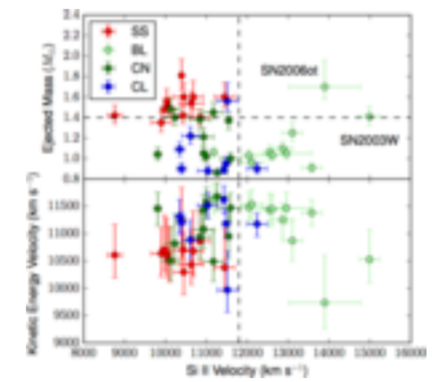
Deterministic: Construct function $y(\mathbf{x})$ to predict t.

Probabilistic: Find predictive distribution $p(t|\mathbf{x})$

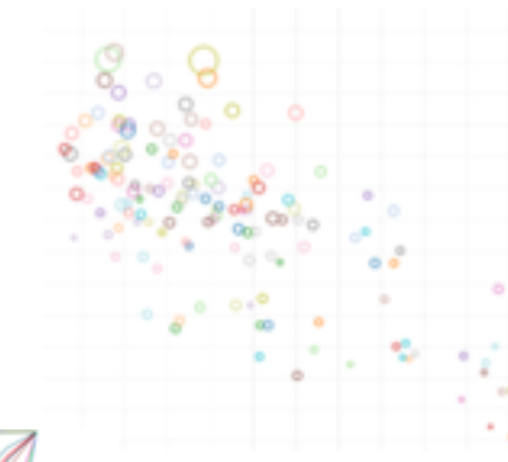
Probabilistic Linear Regression

Academia

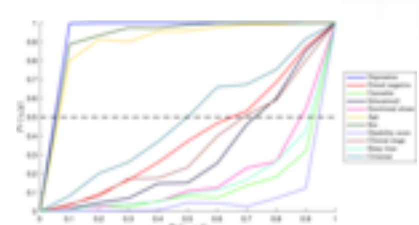


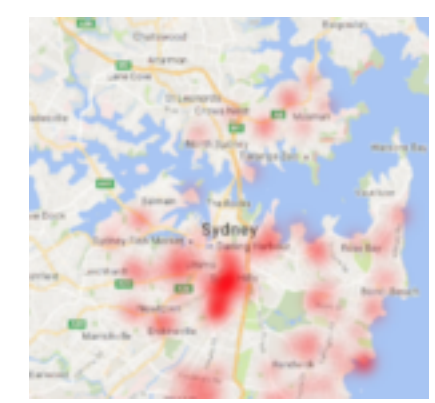


Robotics: Autonomous Navitation, Agricultur
Environmental Monitoring.
Biology: Cancer Research, Metabolic inference, Brain and Mind Centre, Milk Production.
Astronomy: Light Curve Modelling.
Social Sciences: Criminology, Subnational/International Conflict, Linguistics, Aged Care Facilities.



Industry





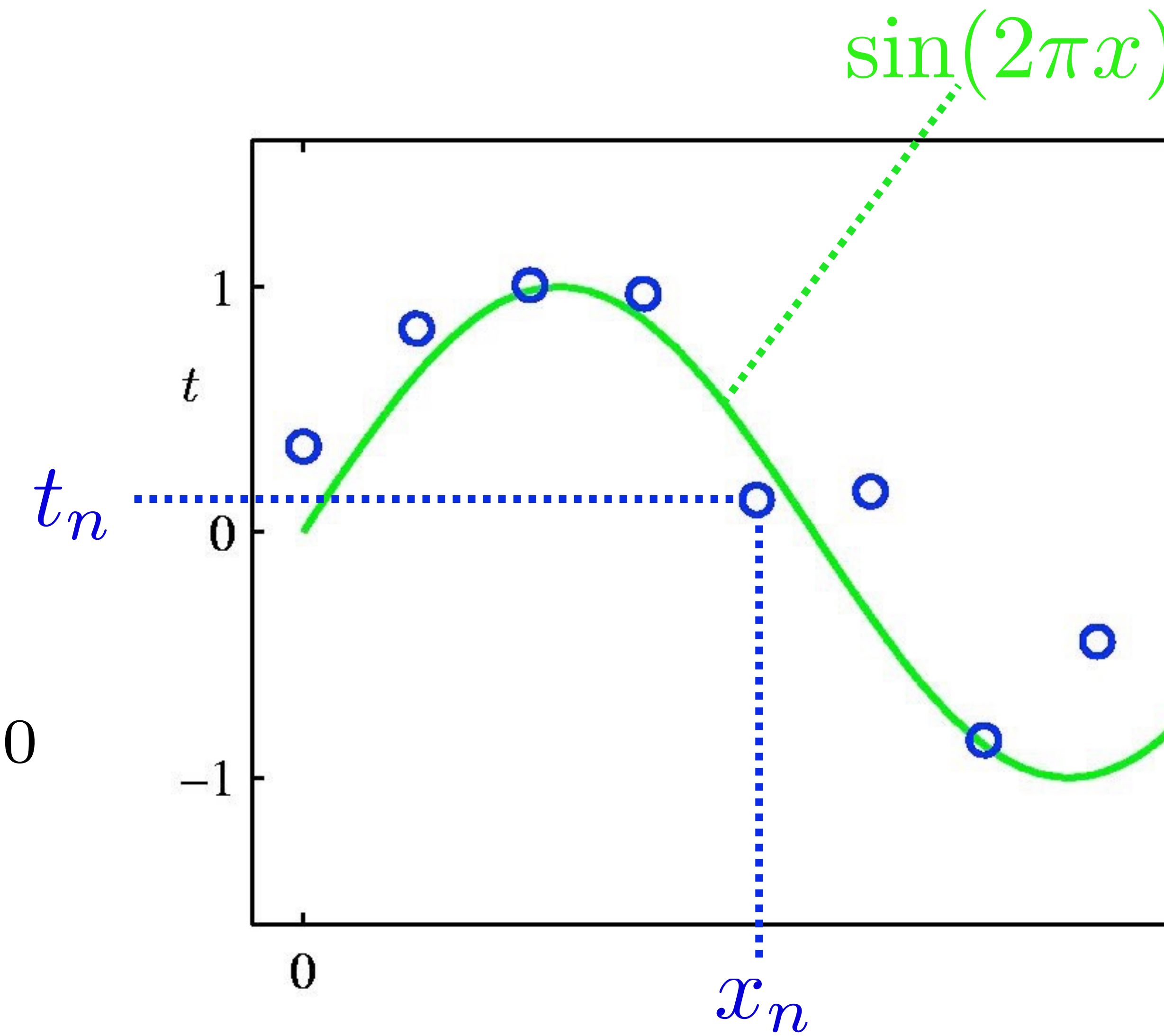
Retail: Amazon, Facebook, Google, etc.
Consultancy: Mining, Energy Generation and Distribution.
Banks: Loan estimation, Risk assessment.

Example: Polynomial Curve Fitting

$$t = f(x) = \sin(2\pi x)$$

$$N = 10$$

$$\mathcal{D} = \{(x_n, t_n)\}_{n=1,\ldots,10}$$



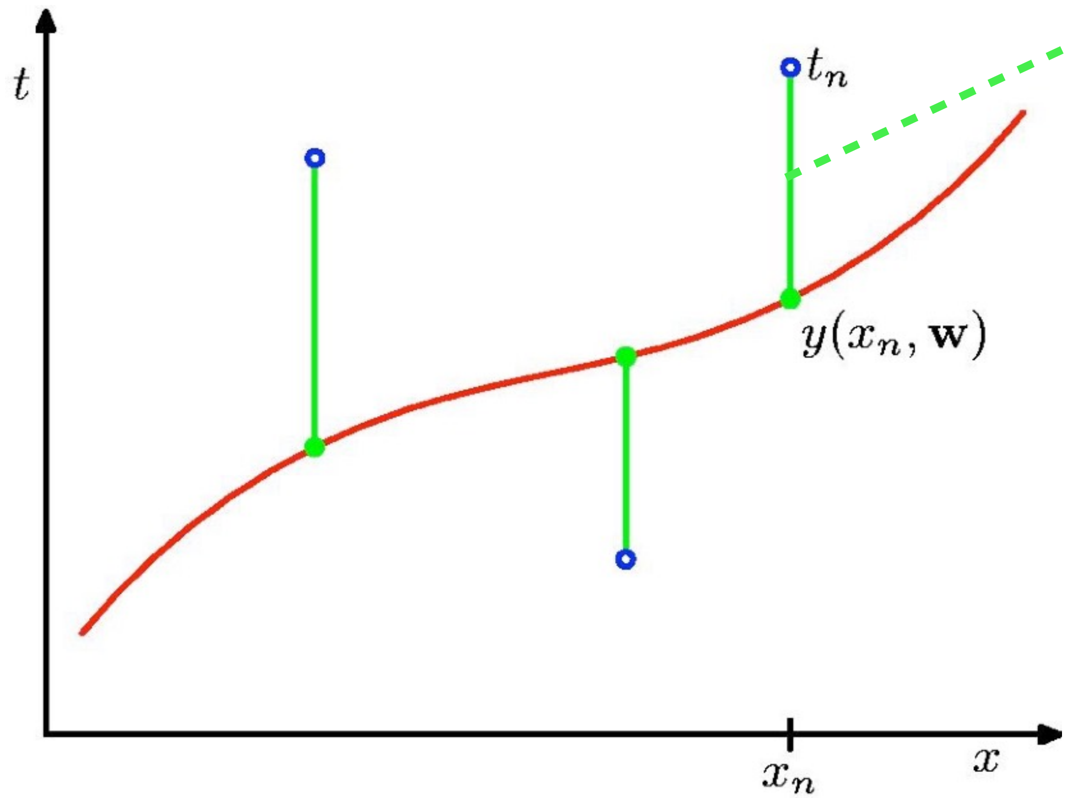
Polynomial fit:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Example: Polynomial Curve Fitting

Sum-of-Squares Error Function $E(\mathbf{w})$

Polynomial Fit: $y(x, \mathbf{w})$



$$E_n = y(x_n, \mathbf{w}) - t_n$$

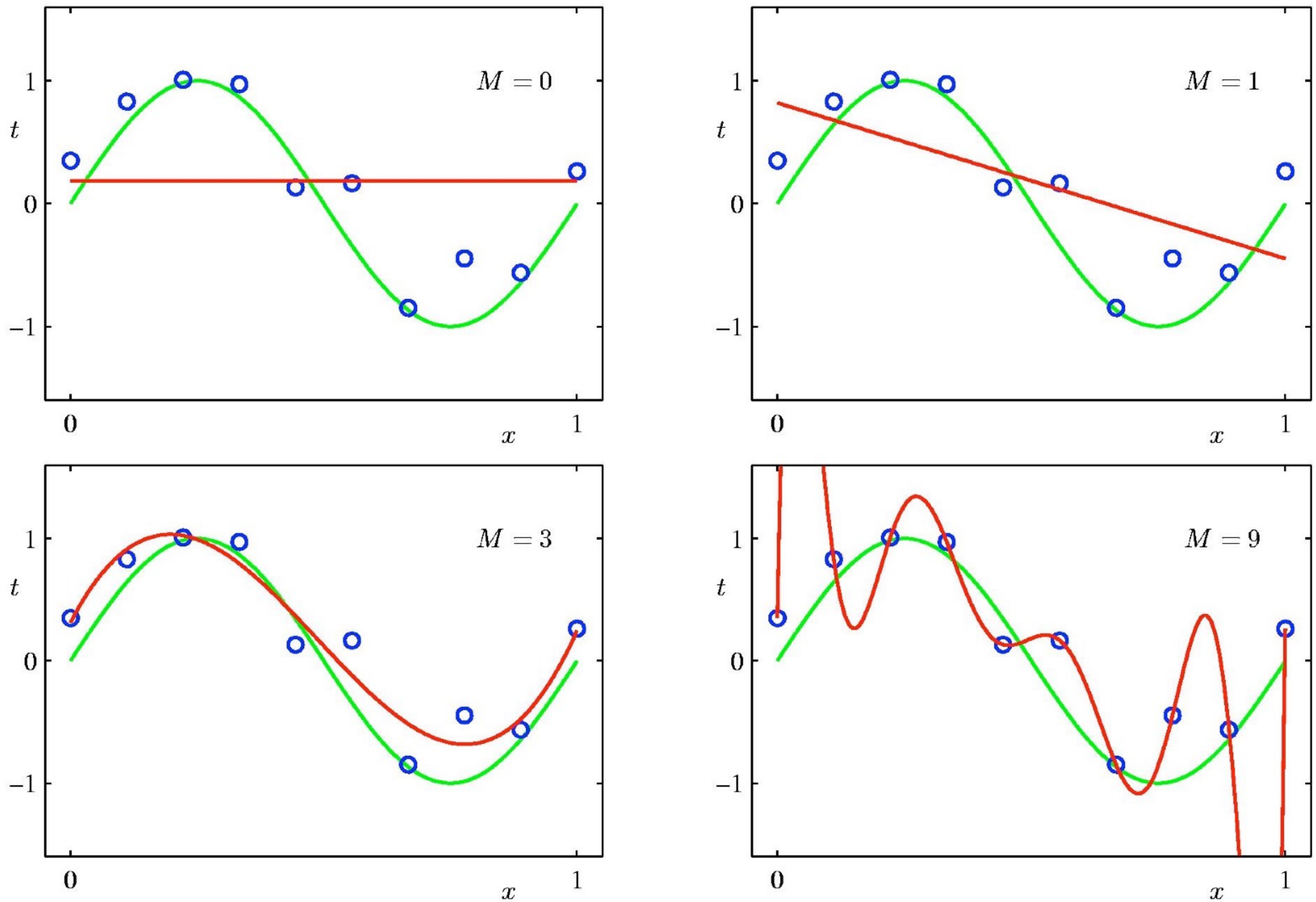
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} E_n^2$$

$$\boxed{E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2}$$

Best fit: $$\mathbf{w}^\star = \underset{\mathbf{w}}{\operatorname{argmin}}\ E(\mathbf{w})$$

Example: Polynomial Curve Fitting

Degree of Polynomial: M

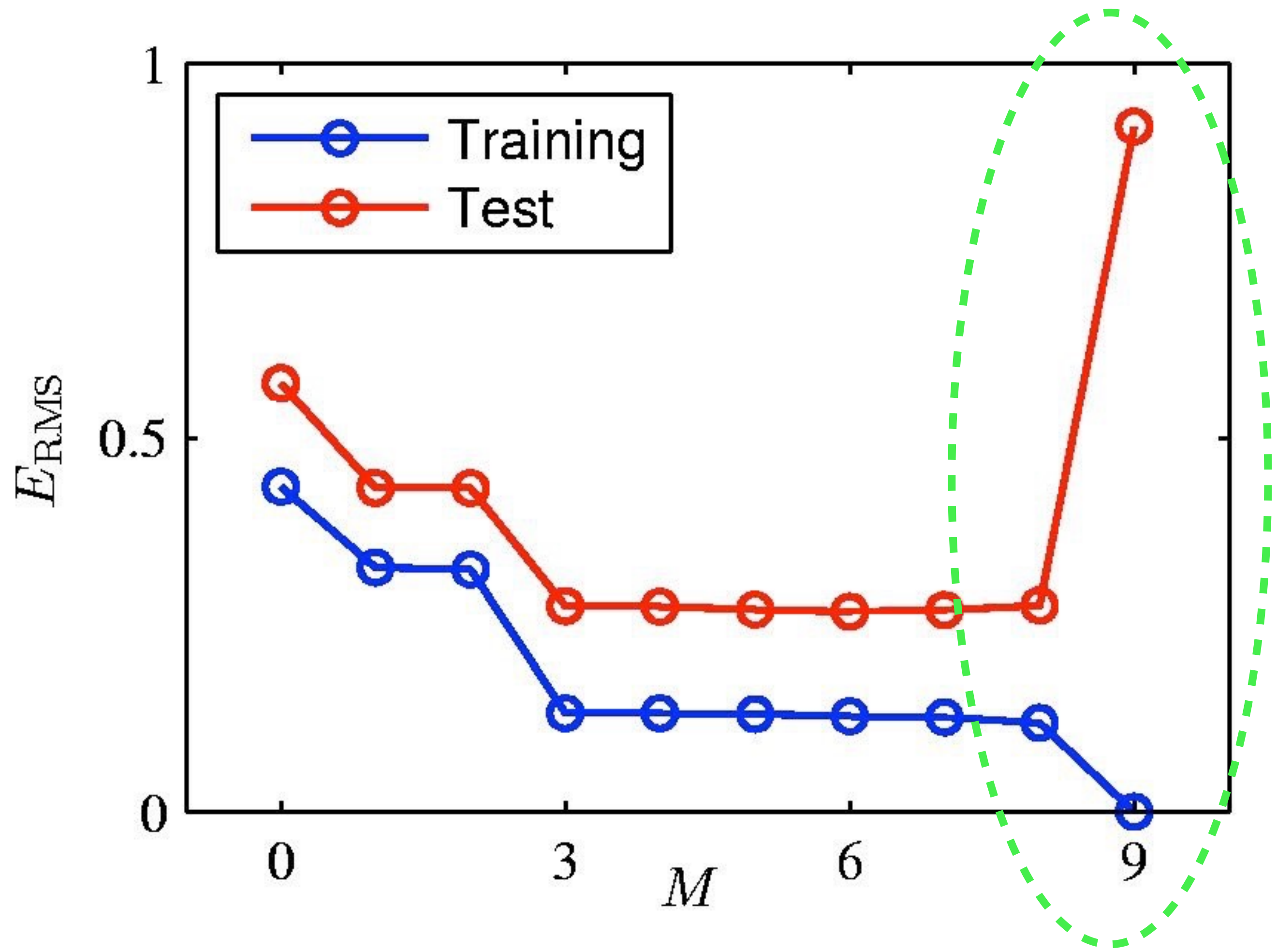


Example: Polynomial Curve Fitting

Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

Test set: N = 100

Overfitting



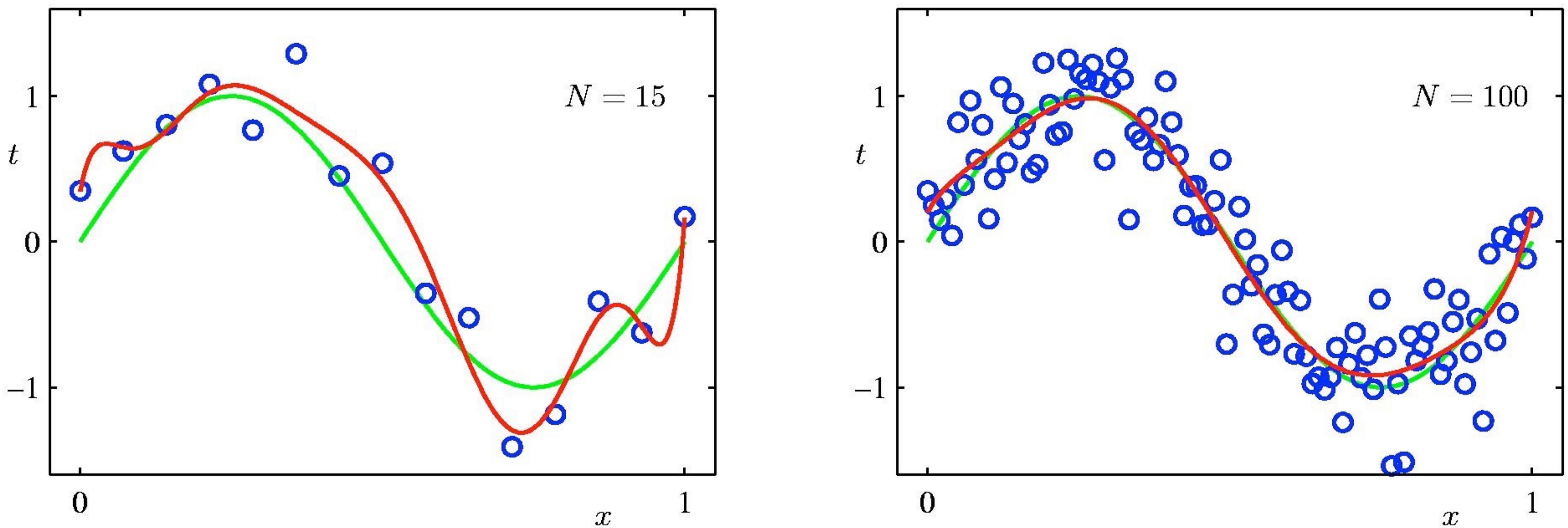
Example: Polynomial Curve Fitting

Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Example: Polynomial Curve Fitting

Behaviour with dataset size ($M = 9$):



Example: Polynomial Curve Fitting

Regularisation:

$$\widetilde{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,\mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

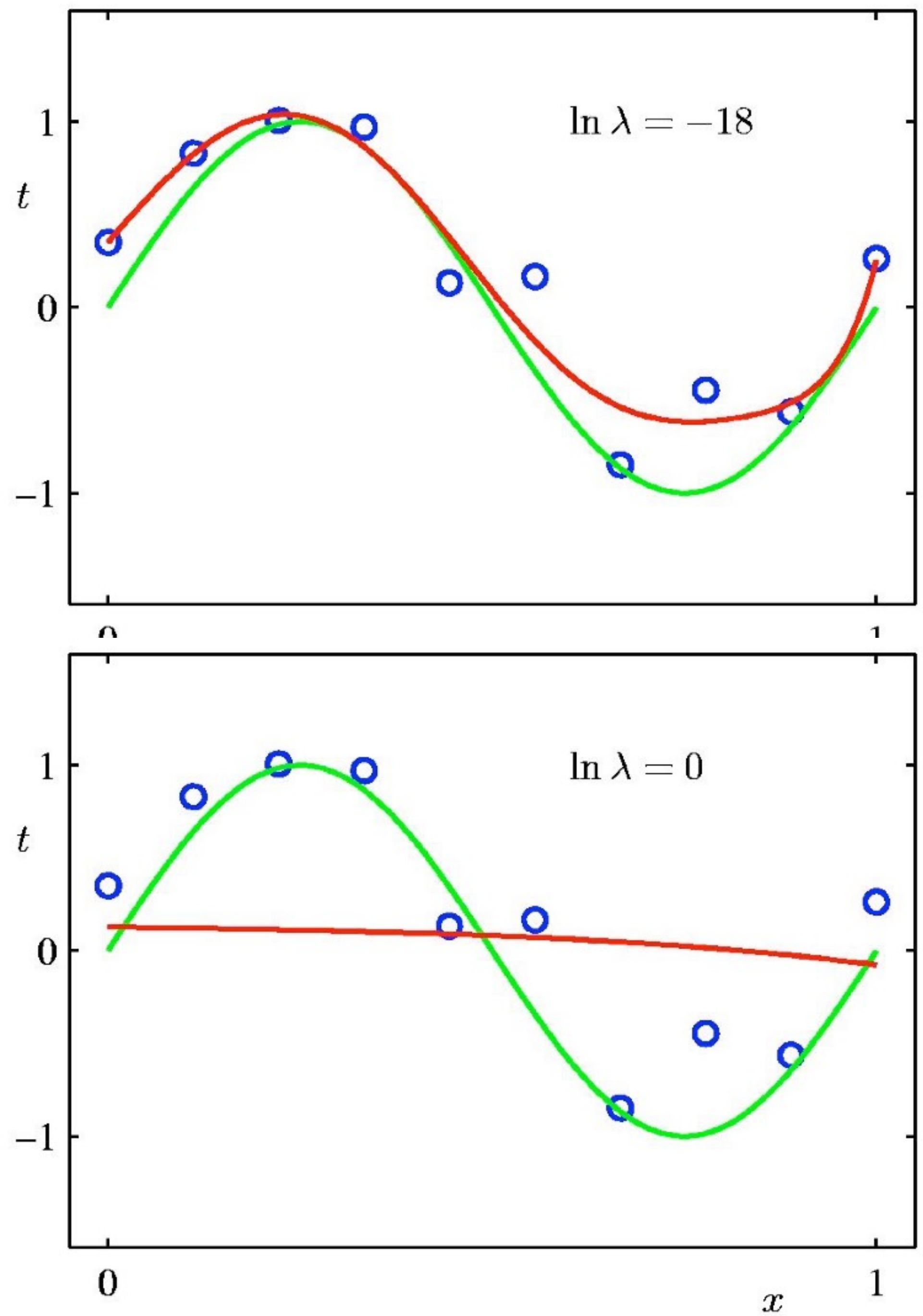
Penalise large coefficient

$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^{\mathrm{T}}\mathbf{w} = w_0^2 + w_1^2 + \ldots + w_M^2$$

λ encodes the relative importance of the

Example: Polynomial Curve Fitting

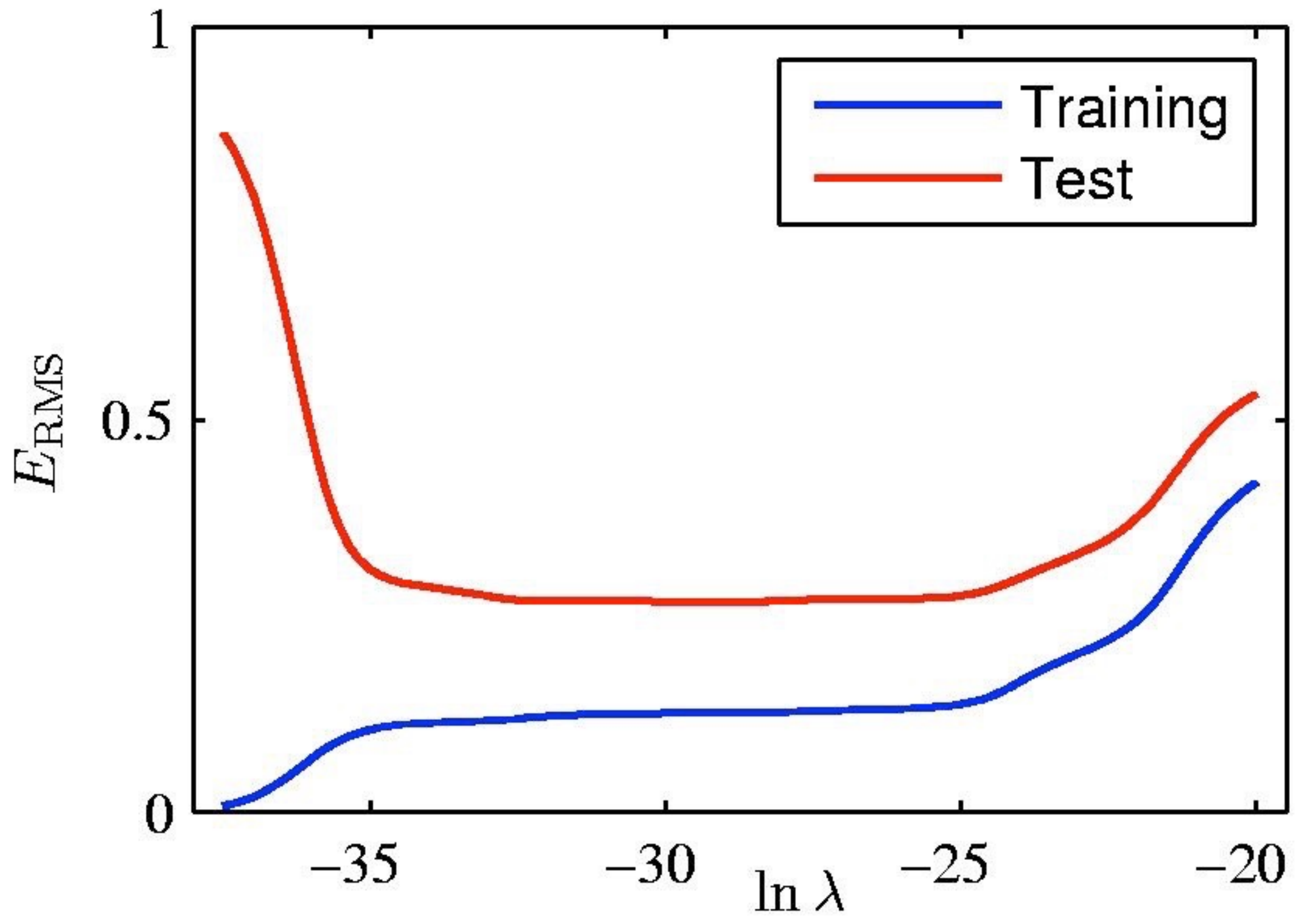
Regularisation:



	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^\star$	0.35	0.35	0.13
$w_1^\star$	232.37	4.74	-0.05
$w_2^\star$	-5321.83	-0.77	-0.06
$w_3^\star$	48568.31	-31.97	-0.05
$w_4^\star$	-231639.30	-3.89	-0.03
$w_5^\star$	640042.26	55.28	-0.02
$w_6^\star$	-1061800.52	41.32	-0.01
$w_7^\star$	1042400.18	-45.95	-0.00
$w_8^\star$	-557682.99	-91.53	0.00
$w_9^\star$	125201.43	72.68	0.01

Example: Polynomial Curve Fitting

Regularisation ($M = 9$)



Linear Basis Function Models

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x})$$

$$\mathbf{w} = (w_0, ..., w_{M-1})^{\mathrm{T}} \qquad \boldsymbol{\phi} = (\phi_0, ..., \phi_{M-1})^{\mathrm{T}}$$

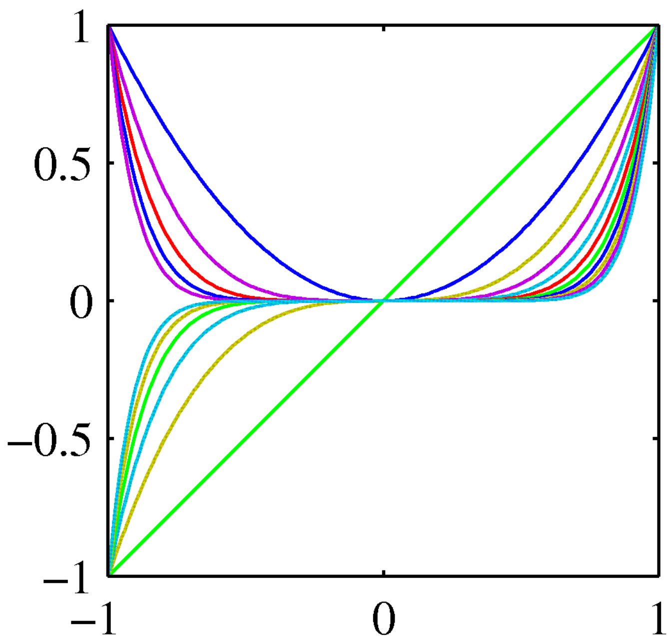
$\phi_j(\mathbf{x})$ are the *basis functions*. $\phi_0(\mathbf{x}) = 1$

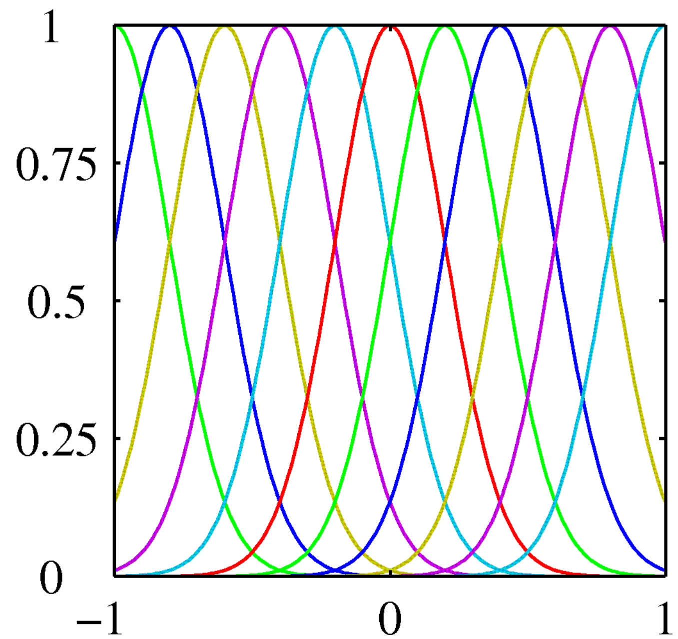
Linear Basis Function Models

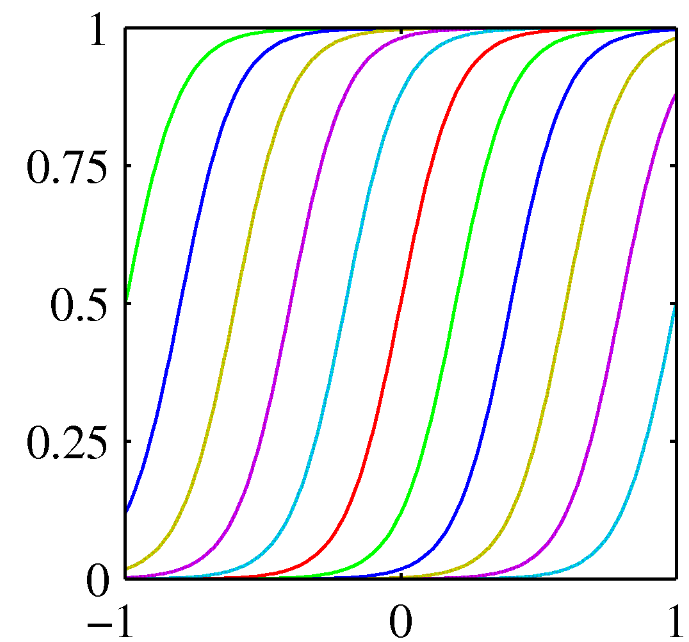
THE UNIVERSITY OF SYDNEY

Examples of Basis Functions in 1D:

Polynomial Basis Functions	Gaussian Basis Functions	Sigmoidal Basis Functions
$\phi_j(x) = x^j$	$\phi_j(x) = \exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$	$\phi_j(x) = \frac{1}{1+\exp\left\{-\frac{x-\mu_j}{s}\right\}}$







Modelling Noisy Observations

Lets assume observations from a deterministic function with added Gaussian noise.

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \text{with} \qquad \epsilon \sim \mathcal{N}(0, \beta^{-1})$$

equivalently,

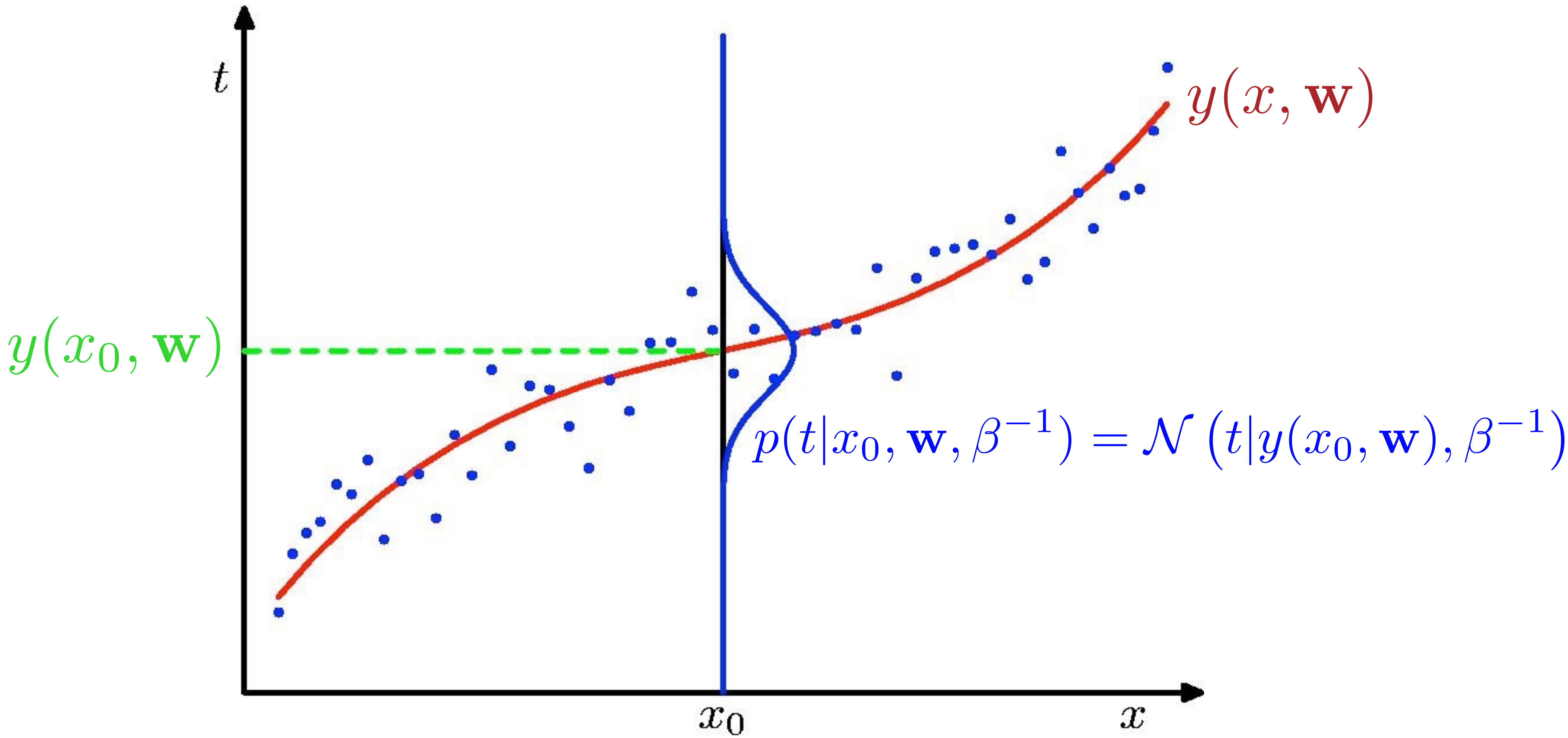
$$p(t|\mathbf{x}, \mathbf{w}, \beta^{-1}) = \mathcal{N}\left(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}\right)$$

Given the training data: $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ $\boldsymbol{\tau} = \{t_1, ..., t_N\}$

The expression of the likelihood of the iid data given the model is:

$$p(\boldsymbol{\tau}|\mathbf{X}, \mathbf{w}, \beta^{-1}) = \prod_{n=1}^{N} \mathcal{N}\left(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\right)$$

Modelling Noisy Observations



Maximum Likelihood

$$p(\boldsymbol{\tau}|\mathbf{X}, \mathbf{w}, \beta^{-1}) = \prod_{n=1}^{N} \mathcal{N}\left(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\right)$$

$$= \prod_{n=1}^{N} \sqrt{\frac{\beta}{2\pi}} \exp -\frac{\beta(t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n))^2}{2}$$

$$= \left(\frac{\beta}{2\pi}\right)^{N/2} \prod_{n=1}^{N} \exp -\frac{\beta(t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n))^2}{2}$$

$$\ln(\cdot) = \ln(\cdot)$$

$$\ln p(\boldsymbol{\tau}|\mathbf{X}, \mathbf{w}, \beta^{-1}) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln 2\pi - \beta\sum_{n=1}^{N}\frac{\left(t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2}{2}$$

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

Maximum Likelihood

THE UNIVERSITY OF SYDNEY

$$\mathbf{w}_{\mathrm{ML}} = \underset{\mathbf{w}}{\operatorname{argmax}} \; \ln p(\boldsymbol{\tau}|\mathbf{X}, \mathbf{w}, \beta^{-1})$$

$$\Rightarrow \qquad \left. \frac{\partial \ln p(\boldsymbol{\tau}|\mathbf{w}, \beta^{-1})}{\partial \mathbf{w}} \right|_{\mathbf{w}_{\mathrm{ML}}} = 0$$

$$\sum_{n=1}^{N} \left(t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)\right) \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = 0$$

Solving for $\mathbf{w}_{\mathrm{ML}}$

$$\boxed{\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\tau}}$$

where

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

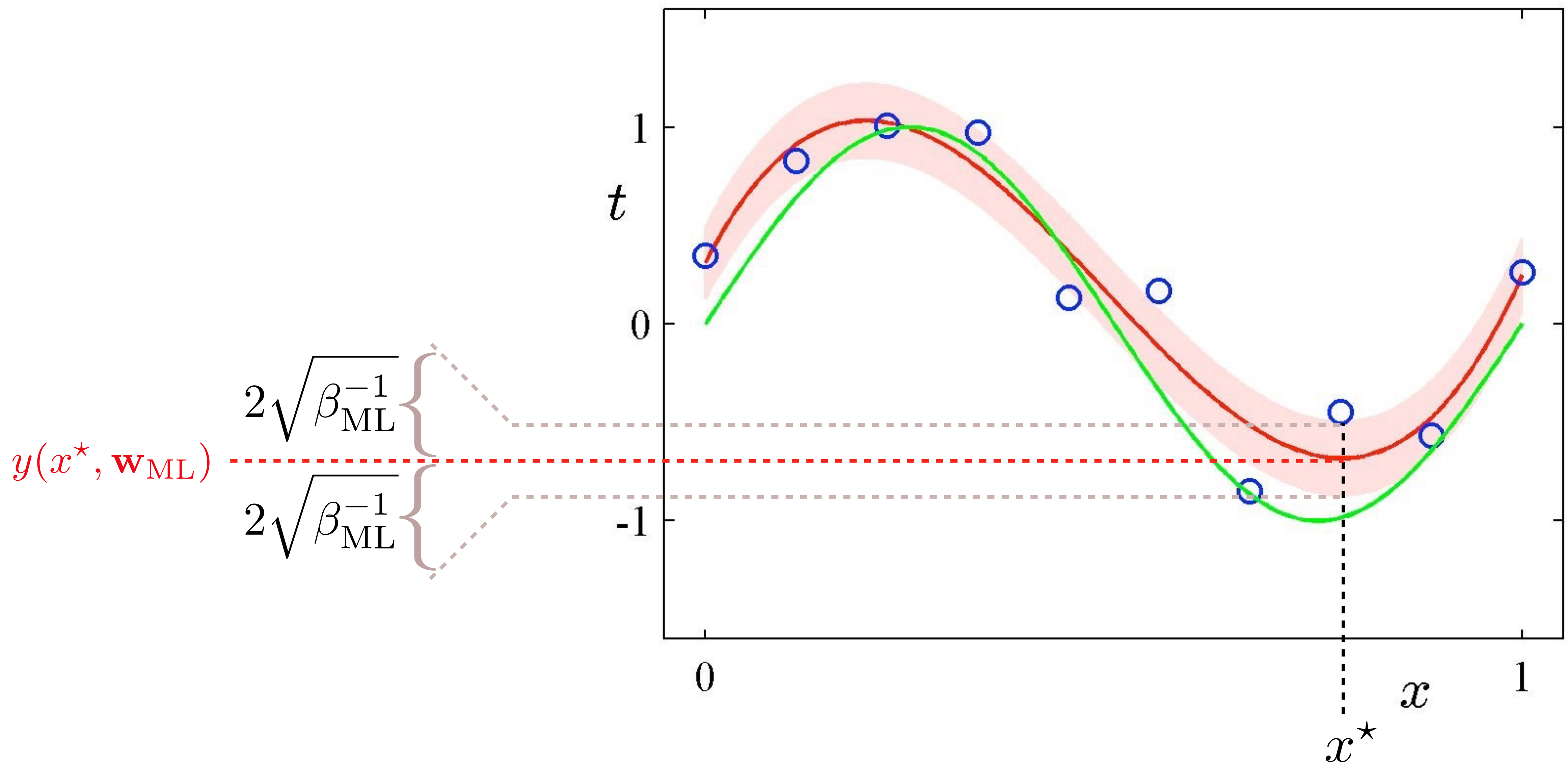
Maximum Likelihood

We can also maximise the log likelihood with respect to the noise precision parameter β .

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2$$

Predictive Distribution

$$p(t|x, \mathbf{w}_{\mathrm{ML}}, \beta_{\mathrm{ML}}) = \mathcal{N}(t|y(x, \mathbf{w}_{\mathrm{ML}}), \beta_{\mathrm{ML}}^{-1})$$



Sequential Learning

THE UNIVERSITY OF SYDNEY

Deal with Large Datasets

Two options: Incremental or Stochastic selection of data-points.

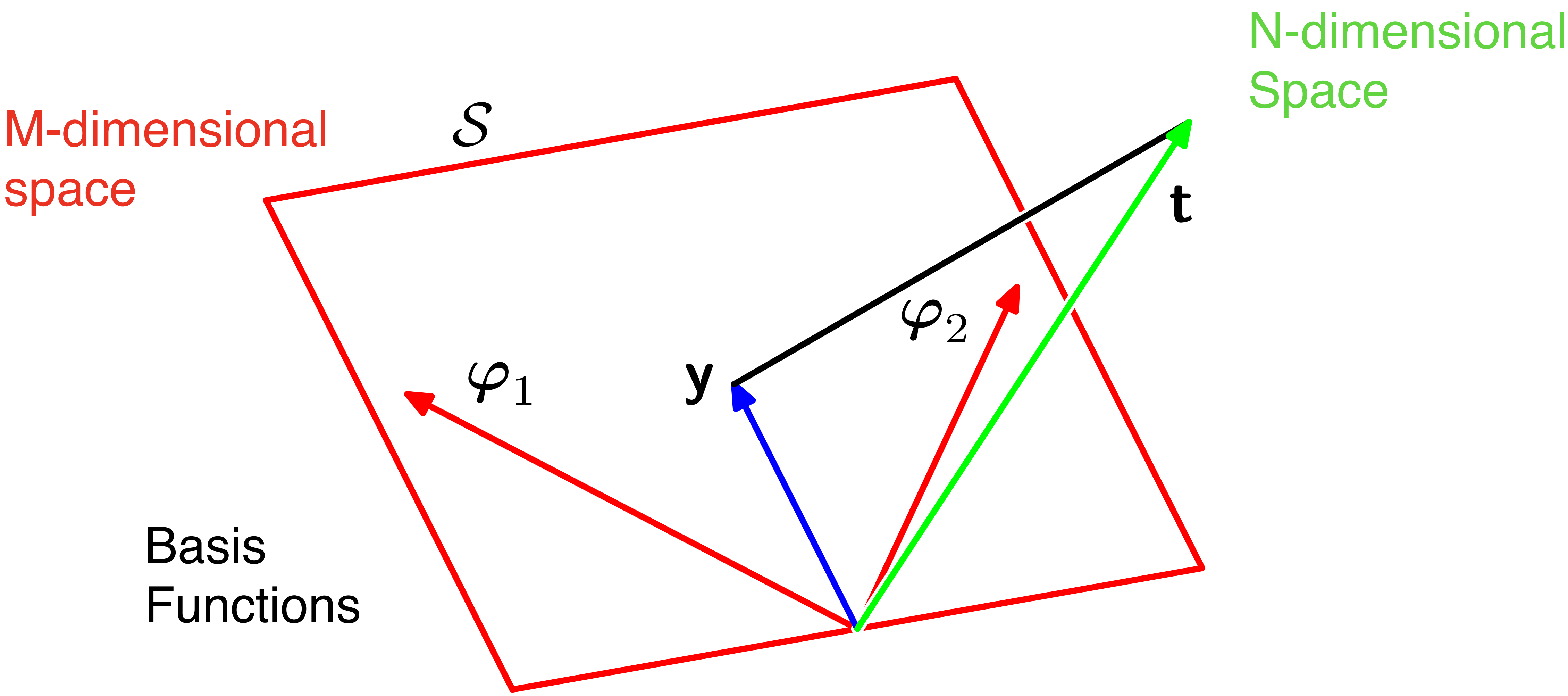
Stochastic Gradient Descent

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

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta (t_n - \mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}_n) \boldsymbol{\phi}_n$$

Least Mean Square algorithm (LMS)

Geometry of Least Squares



Regularised Least Squares

Are we dealing with overfitting? NO

The expression of the log likelihood:

$$\ln p(\boldsymbol{\tau}|\mathbf{w},\beta^{-1}) = \frac{N}{2}\ln\beta - \frac{N}{2}\ln 2\pi - \beta E(\mathbf{w})$$

Introduce regulariser: $$E(\mathbf{w}) = \underbrace{E_D(\mathbf{w})}_{\text{Data Term}} + \lambda \underbrace{E_W(\mathbf{w})}_{\text{Regularisation Term}}$$

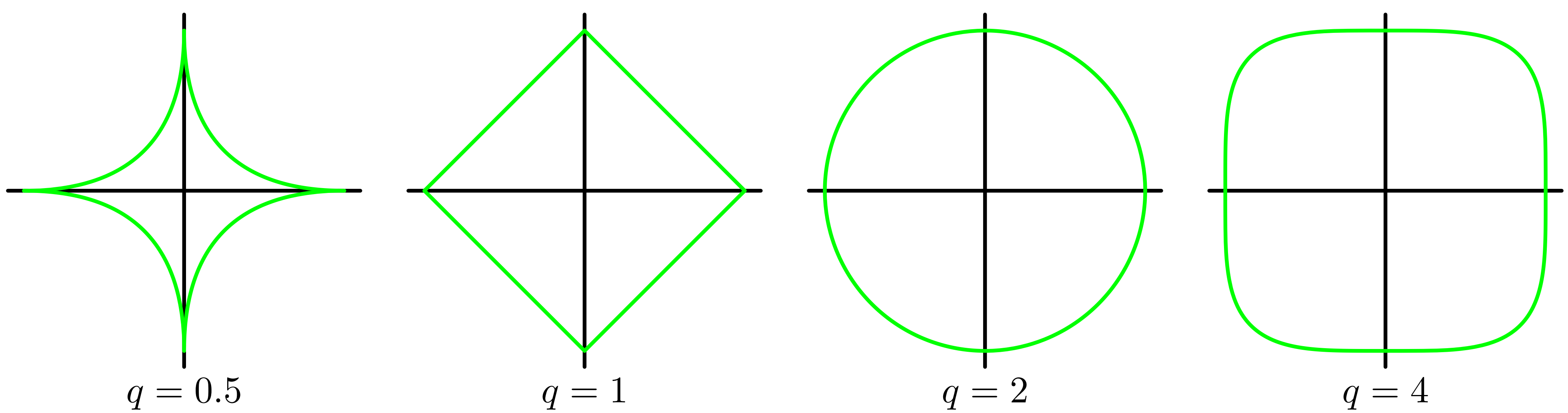
$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right)^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Following previous maximisation procedure:

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

Generalised Regulariser

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$



Multiple Outputs

K > 1 target variables.

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

$\mathbf{W}$ is a weight matrix $M \times K$

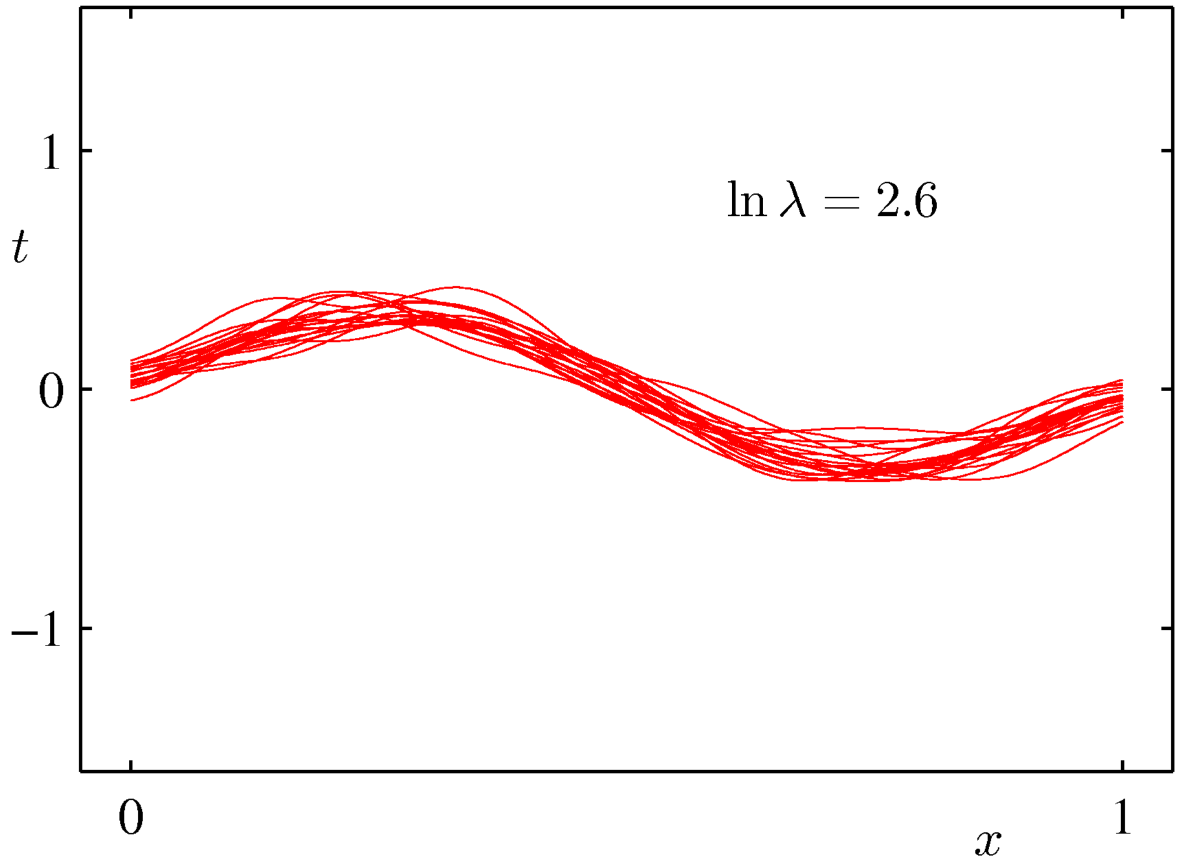
$\mathbf{T}$ is a target matrix $N \times K$

$$\begin{aligned} \ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n | \mathbf{W}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2} \ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2} \sum_{n=1}^{N} \left\| \mathbf{t}_n - \mathbf{W}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\|^2 \end{aligned}$$

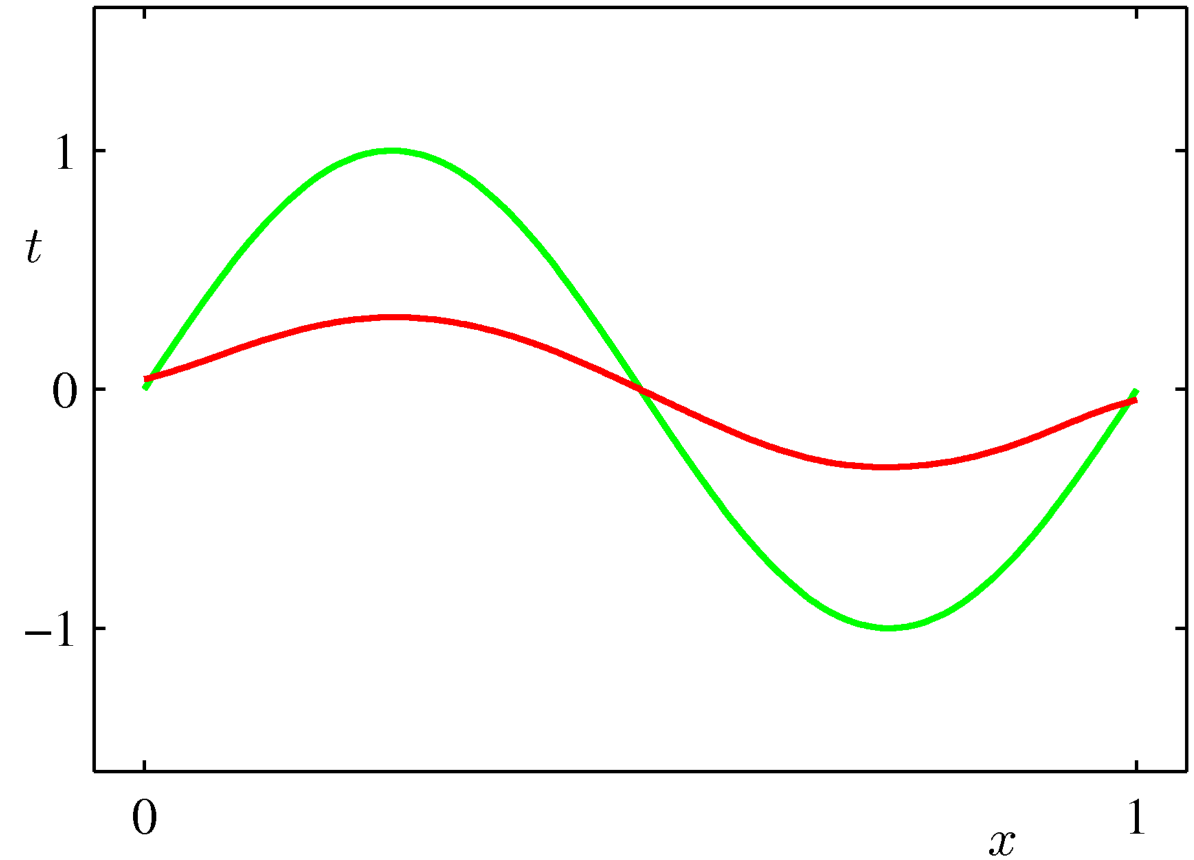
$$\mathbf{W}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{T}.$$

Bias-Variance Visualisation

20 datasets with varying regularisation parameter.



Result of fitting the model to each dataset.

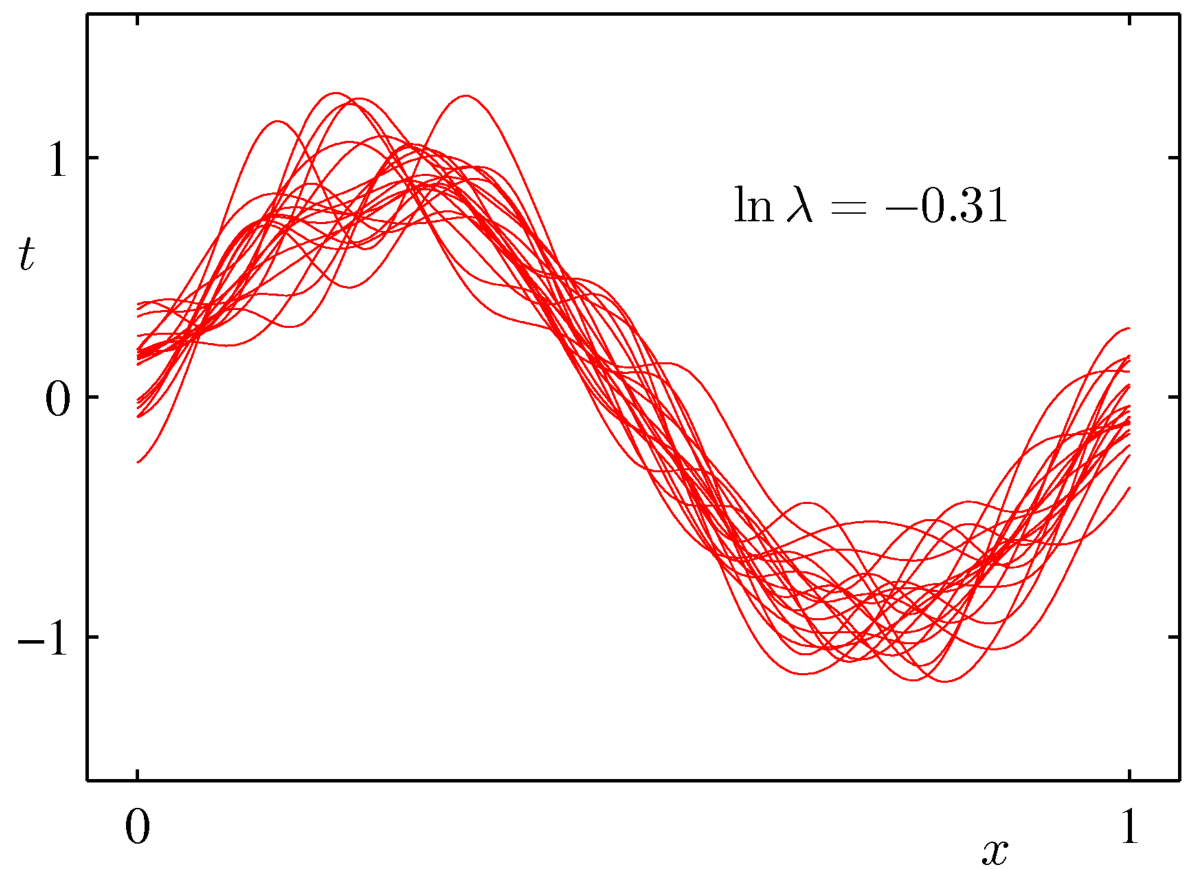


Average of the fits.

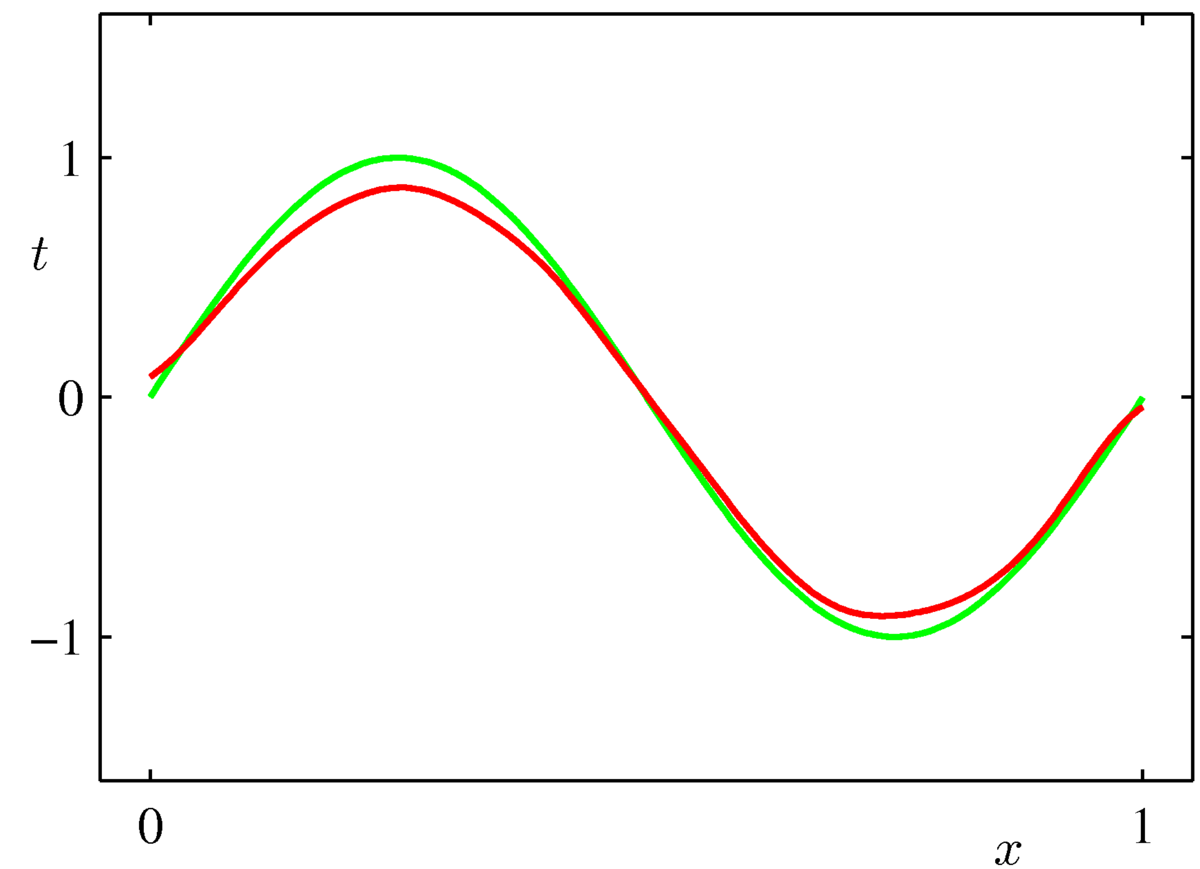
Bias-Variance Visualisation

THE UNIVERSITY OF SYDNEY

20 datasets with varying regularisation parameter.



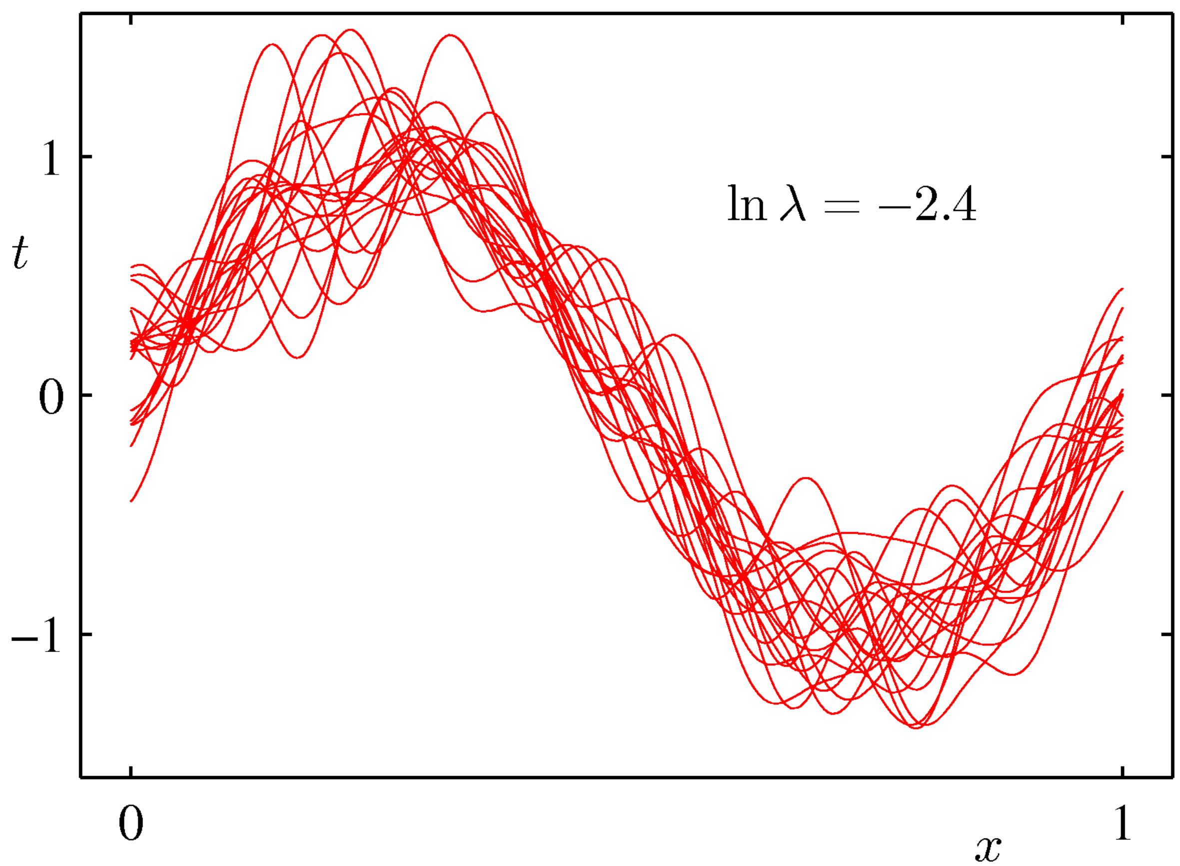
Result of fitting the model to each dataset.



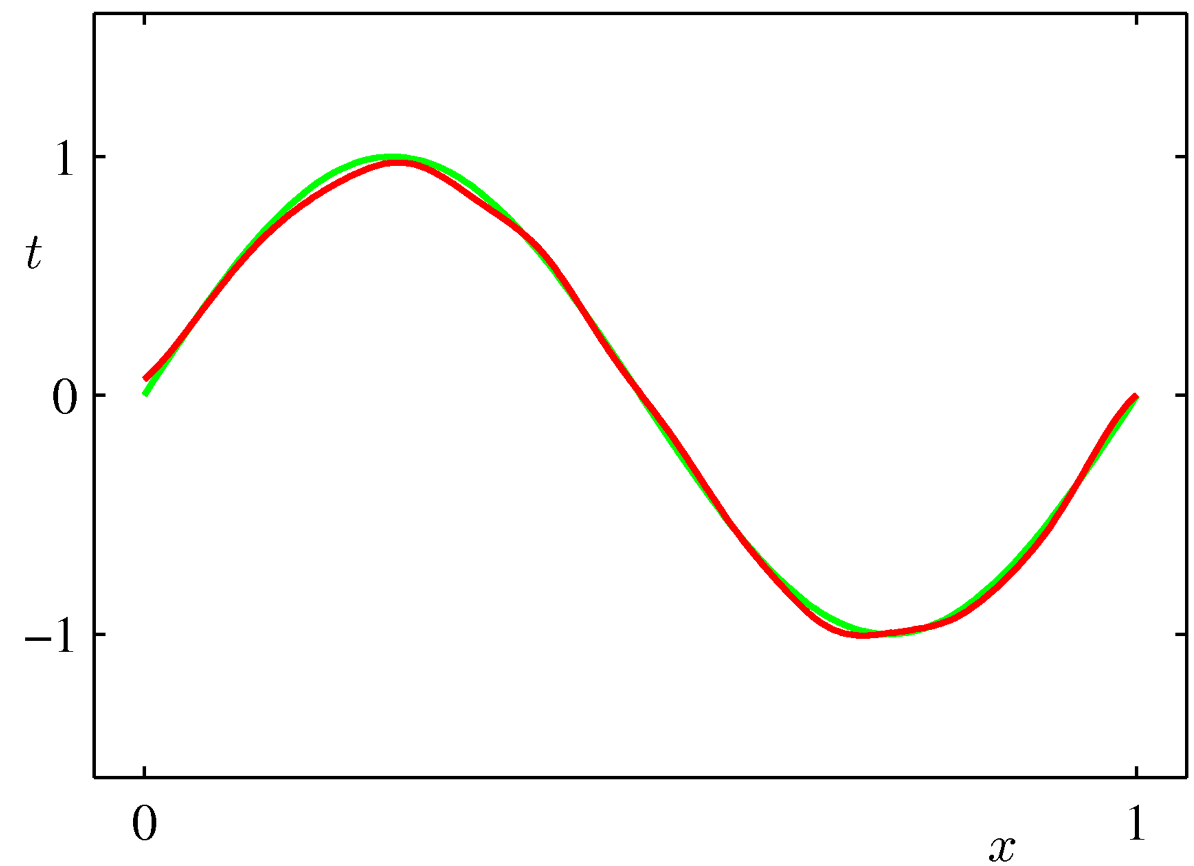
Average of the fits.

Bias-Variance Visualisation

20 datasets with varying regularisation parameter.



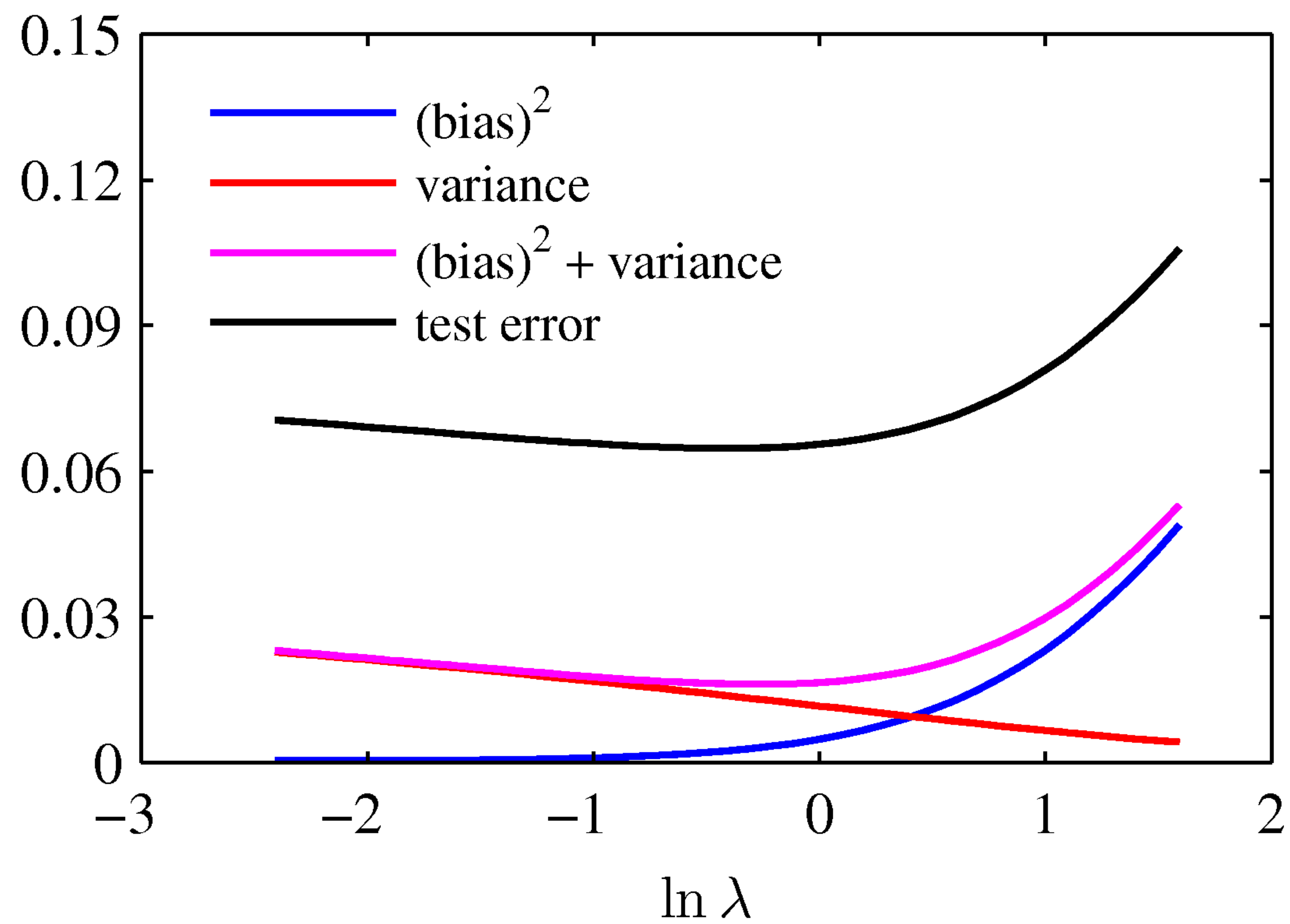
Result of fitting the model to each dataset.



Average of the fits.

The Bias-Variance Trade Off

From these plots, we note that an over-regularised model (large λ) will have a high bias, while an under-regularised model (small λ) will have a high variance.



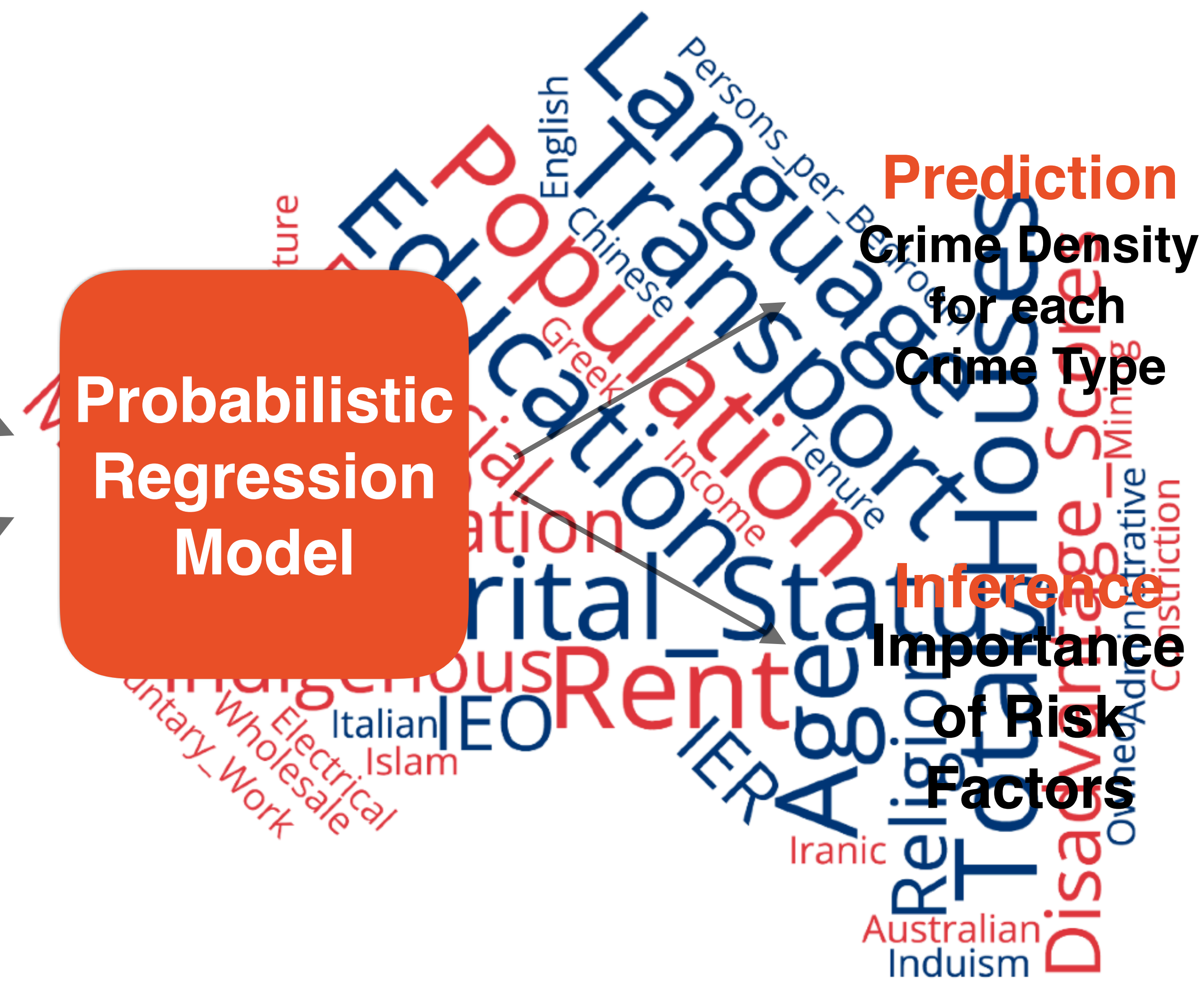
Example
Real Applications

Probabilistic Crime Model

Information Sources

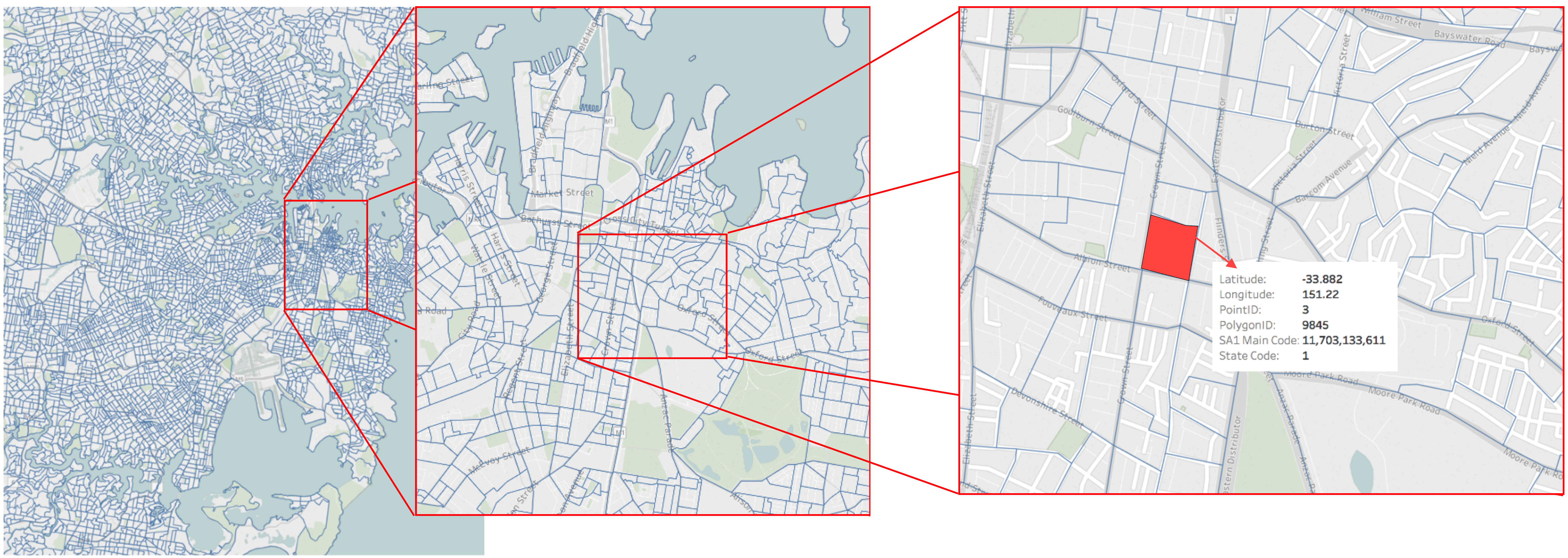
Historic Space-Time Criminal Records

Demographic Information



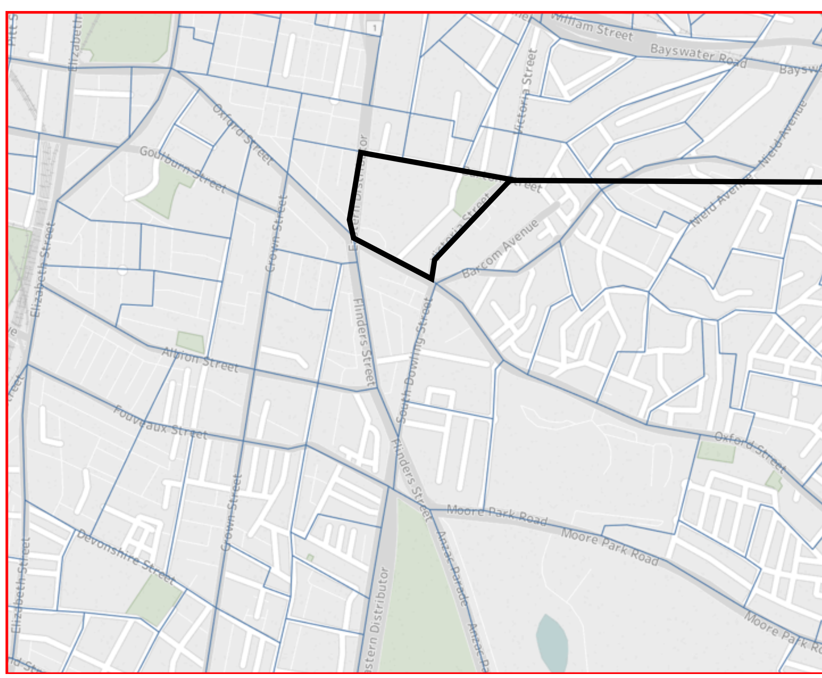
Probabilistic Crime Model

1. Discretise space into unit elements (SA1, SA2, Postcode, LGA)

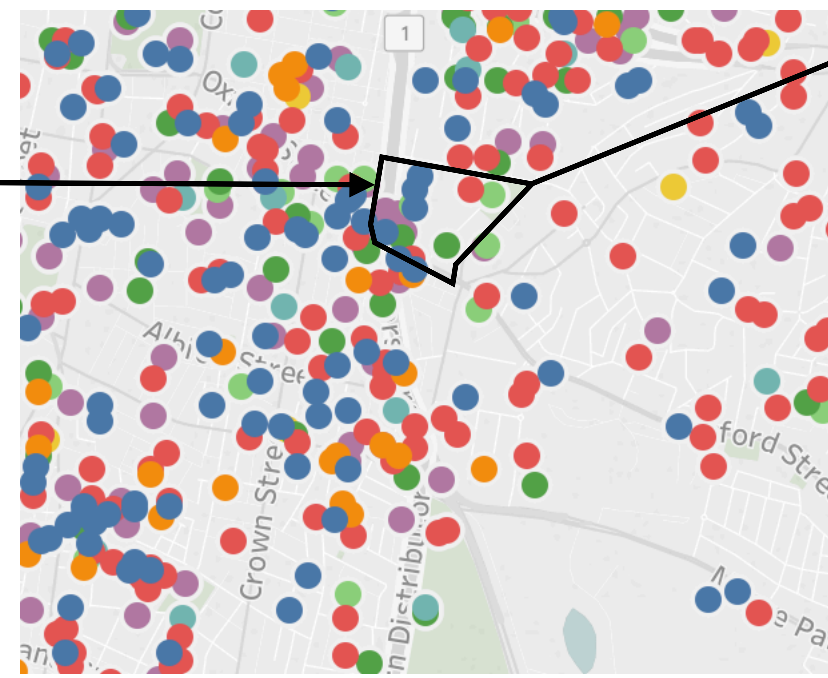


Probabilistic Crime Model

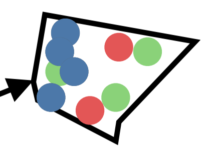
2. Accumulate Crime Occurrence per Geographic Unit



COPS/BOCSAR
Unit Record Incident Data



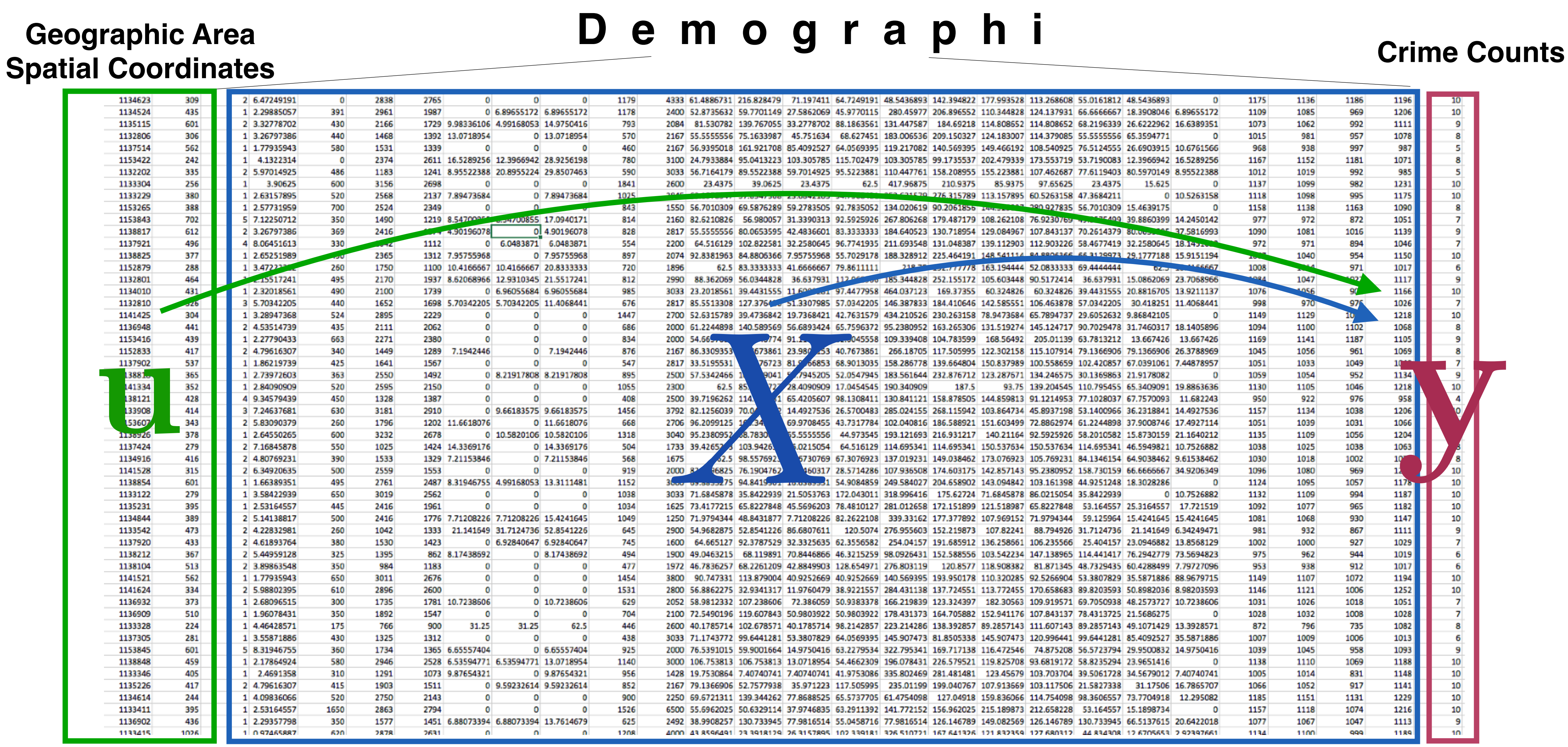
Example Simulated Data



Number of Assaults
Number of Robberies
Number of MVT
…

Probabilistic Crime Model

4. Ensemble Database



Probabilistic Crime Model

5. Define Model

$$y_i = f(\mathbf{x}_i) + h(\mathbf{u}_i) + e_i$$

Demographics

Space

Noise

$$f(\mathbf{x}_i) = \mathbf{x}_i \boldsymbol{\beta}$$

$$\mathbf{h} \sim \mathcal{N}(0, K)$$

$$e \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_n^2)$$

Gaussian Process

$$K_{ij} = k(\mathbf{u}_i, \mathbf{u}_j | \boldsymbol{\Phi})$$

$$k(\mathbf{u}_i, \mathbf{u}_j | \boldsymbol{\Phi} = \{\sigma_f, l\}) = \sigma_f^2 \exp\left(-\frac{\|\mathbf{u}_i - \mathbf{u}_j\|^2}{2l^2}\right)$$

$$\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma_n, \boldsymbol{\Phi}\}$$

Unknown Parameter/Hyper-parame

Probabilistic Crime Model

6. Learn Model

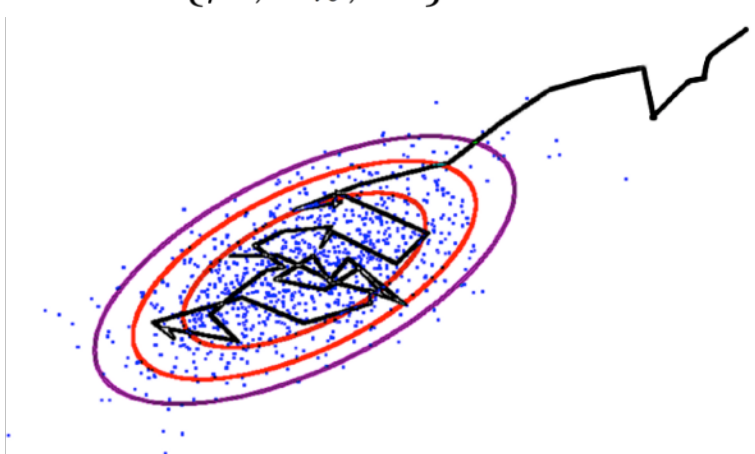
Data $\mathcal{D} = (X, \mathbf{y})$

$\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma_n, \boldsymbol{\Phi}\}$

Bayes Theorem

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

Posterior Likelihood Prior



Conduct sampling algorit
Markov **C**hain **M**onte **C**arl

Algorithm 1 Metropolis-Hastings MCMC

```
Initialise θ^[0].
for k = 1 to M − 1 do
    Sample u ∼ U(0, 1).
    Sample θ^p ∼ q(θ^p | θ^[k])
    if u < A = min{1, p(D|θ^p)p(θ^p) / (p(D|θ^[k])p(θ^[k]))} then
        θ^[k+1] = θ*
    else
        θ^[k+1] = θ^[k]
    end if
end for
```

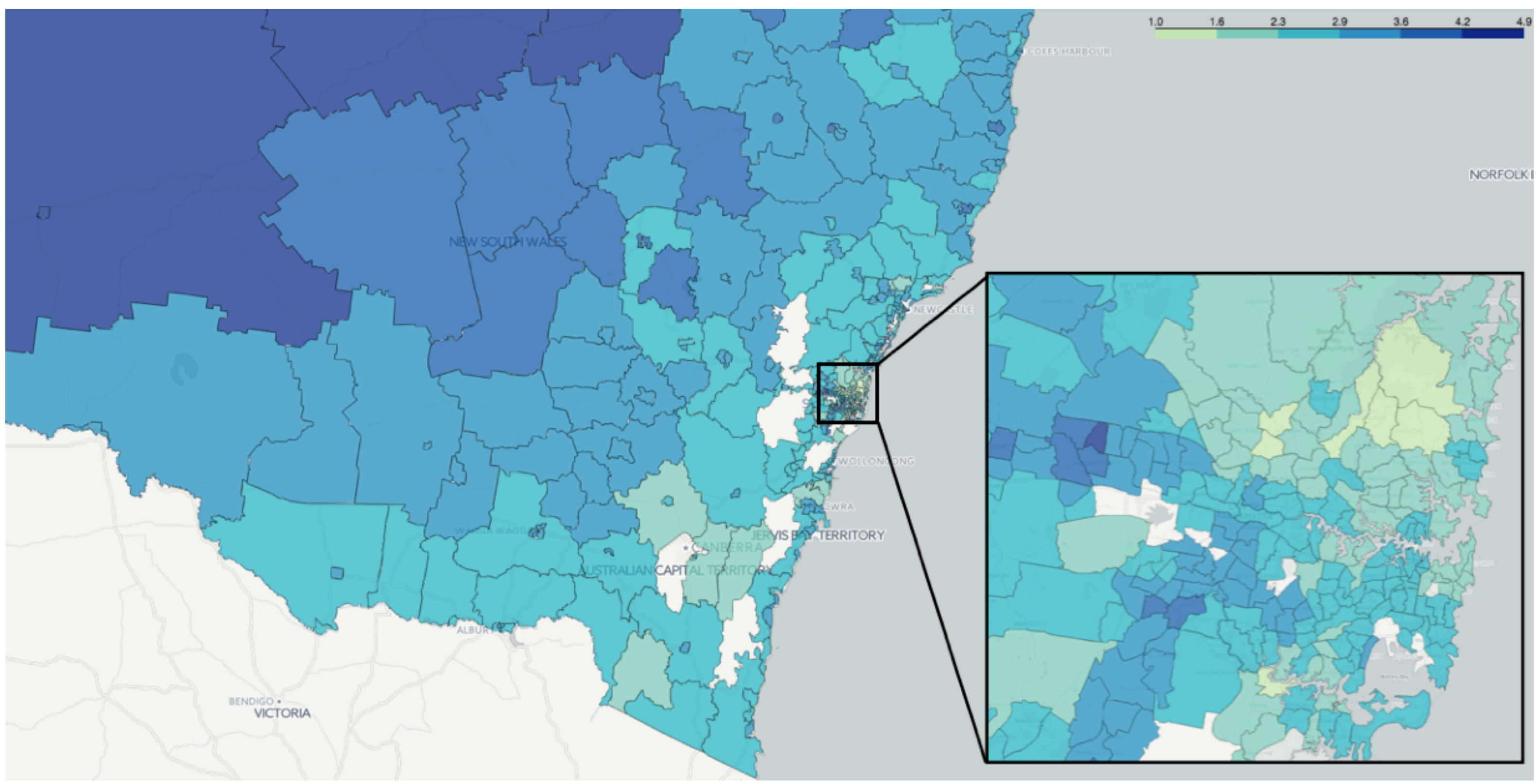
$$p(y^\star|\mathcal{D}) = \int_{\mathbb{R}^{|\boldsymbol{\theta}|}} p(y^\star|\mathcal{D}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\mathrm{d}\boldsymbol{\theta}$$

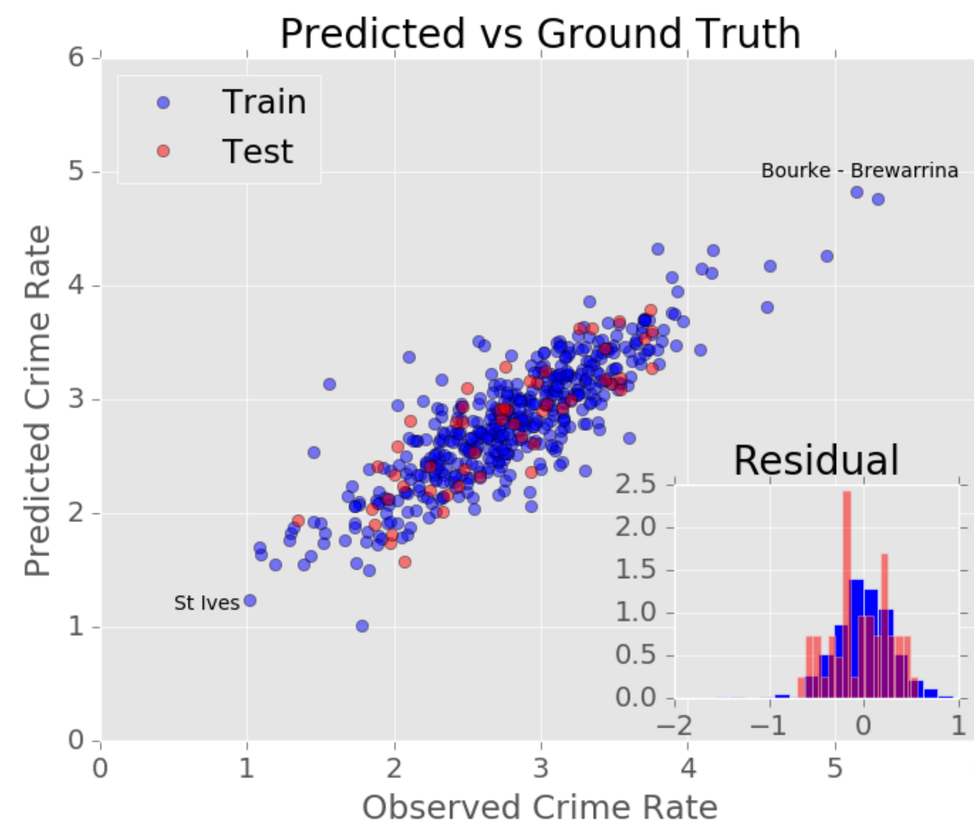
$$p(\theta_i|\mathcal{D}) = \int_{\mathbb{R}^{|\boldsymbol{\theta}|-1}} p(\theta_i|\mathcal{D}, \boldsymbol{\theta}_{\setminus i})p(\boldsymbol{\theta}_{\setminus i}|\mathcal{D})\mathrm{d}\boldsymbol{\theta}_{\setminus i}$$

Probabilistic Crime Model

THE UNIVERSITY OF SYDNEY

7. Evaluate Results, Spatial DV Assault Rates and Predictions





Probabilistic Crime Model

7. Inference on Factors