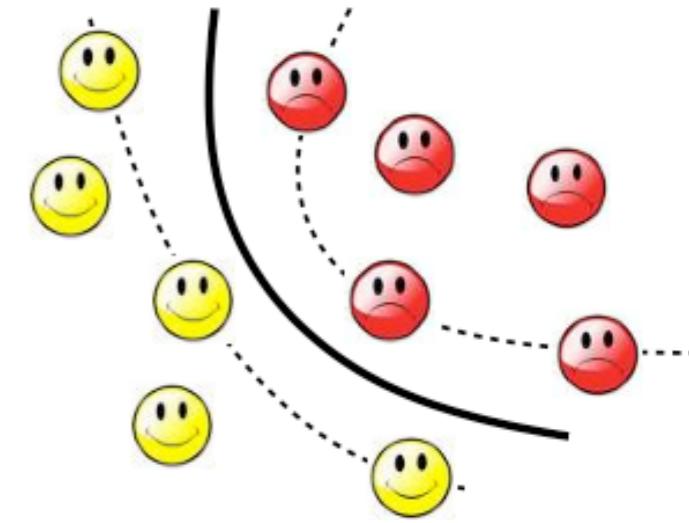




THE UNIVERSITY OF  
**SYDNEY**



# Data Mining and Knowledge Discovery (COMP 5318)

## Linear Regression

Roman Marchant  
23 April 2018



THE UNIVERSITY OF  
SYDNEY

Roman Marchant

[roman.merchant@sydney.edu.au](mailto:roman.merchant@sydney.edu.au)

Rm 542, Building J12, School of IT

Availability: Monday 1400-1600 (Appointment only)



THE UNIVERSITY OF  
SYDNEY

# 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

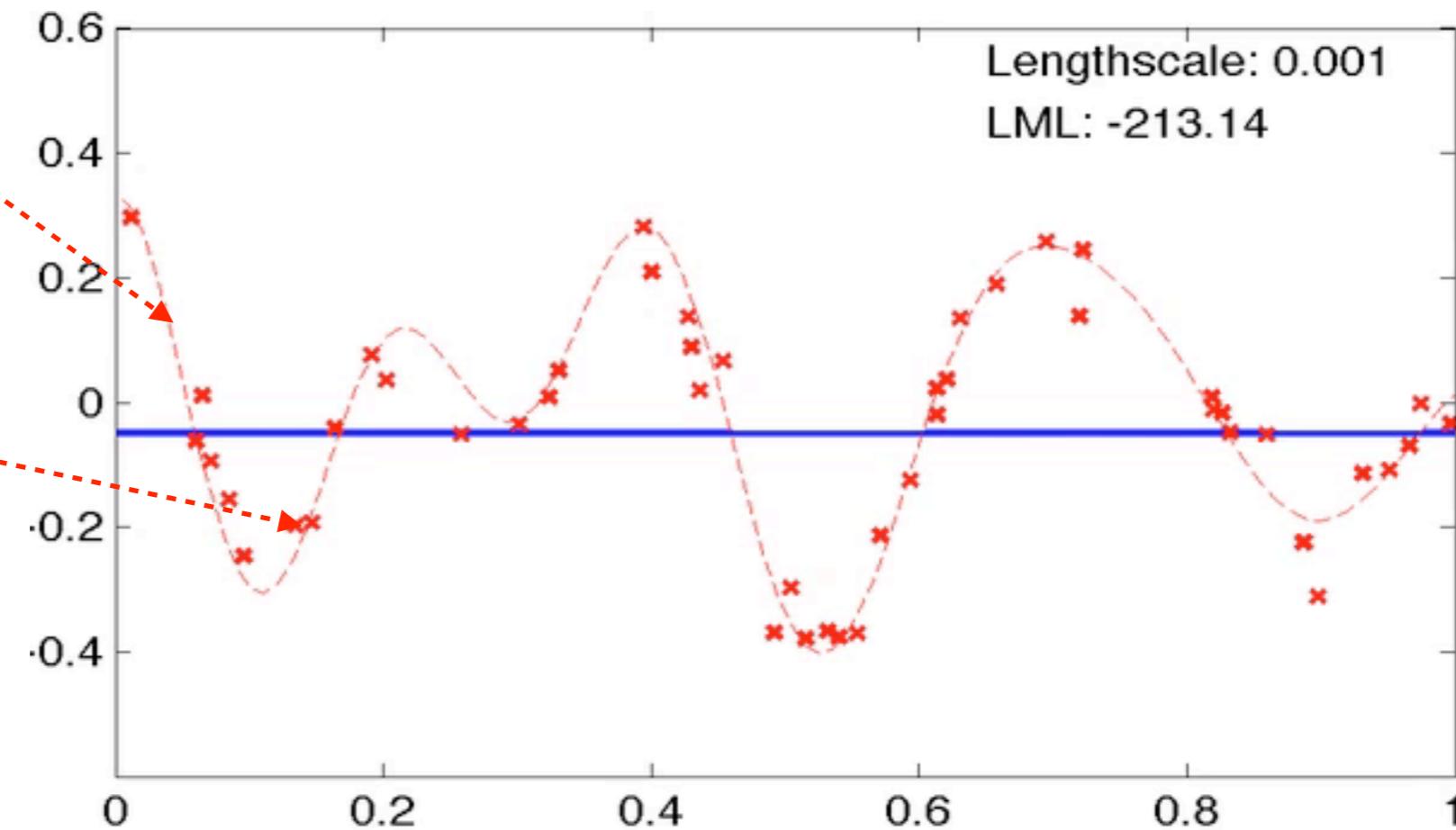


THE UNIVERSITY OF  
SYDNEY

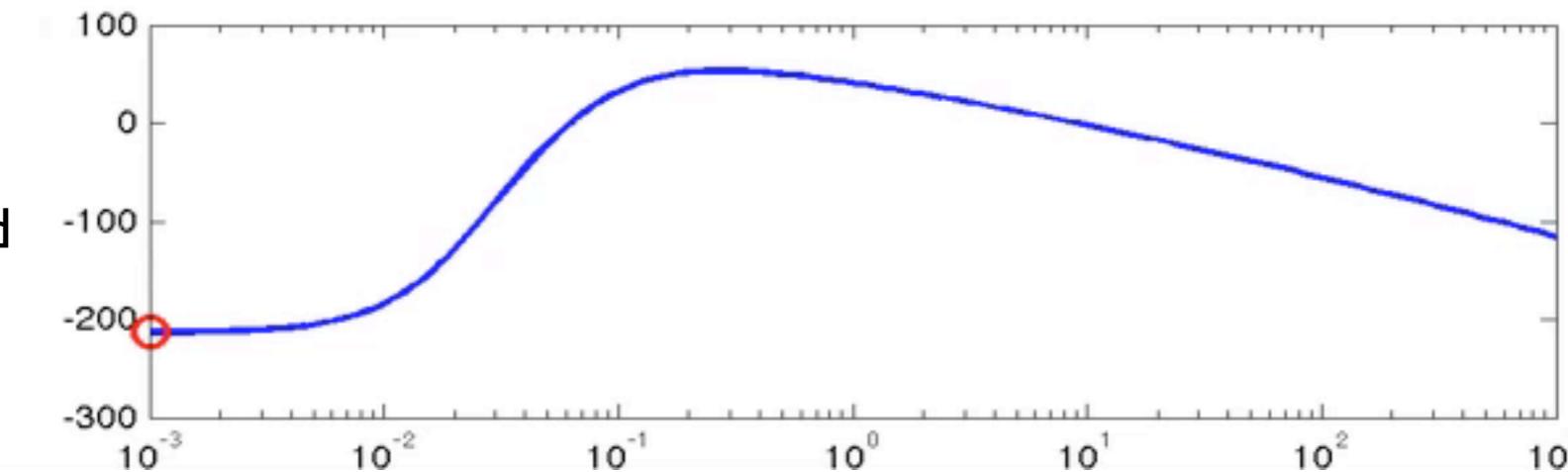
# Learning as Optimisation

Unknown Function  
 $f(x)$

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



THE UNIVERSITY OF  
SYDNEY

# 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  $N$ :  $\{(\mathbf{x}_n, t_n)\}_{n=1,\dots,N}$



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

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



THE UNIVERSITY OF  
SYDNEY

# Probabilistic Linear Regression

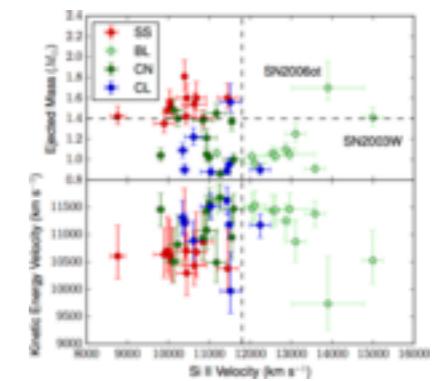
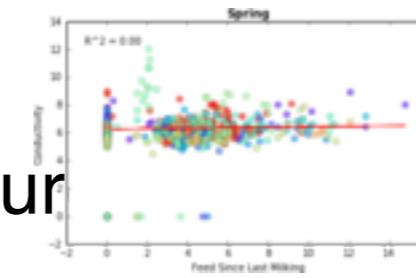
## Academia

Robotics: Autonomous Navigation, Agriculture, 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.

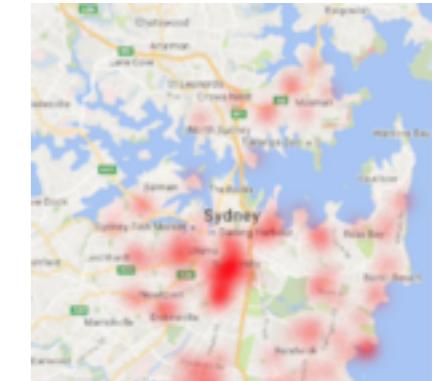
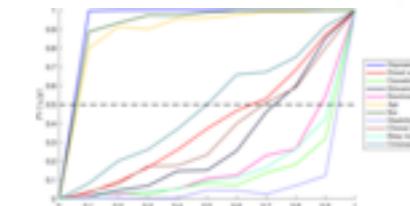


## Industry

Retail: Amazon, Facebook, Google, etc.

Consultancy: Mining, Energy Generation and Distribution.

Banks: Loan estimation, Risk assessment.





THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

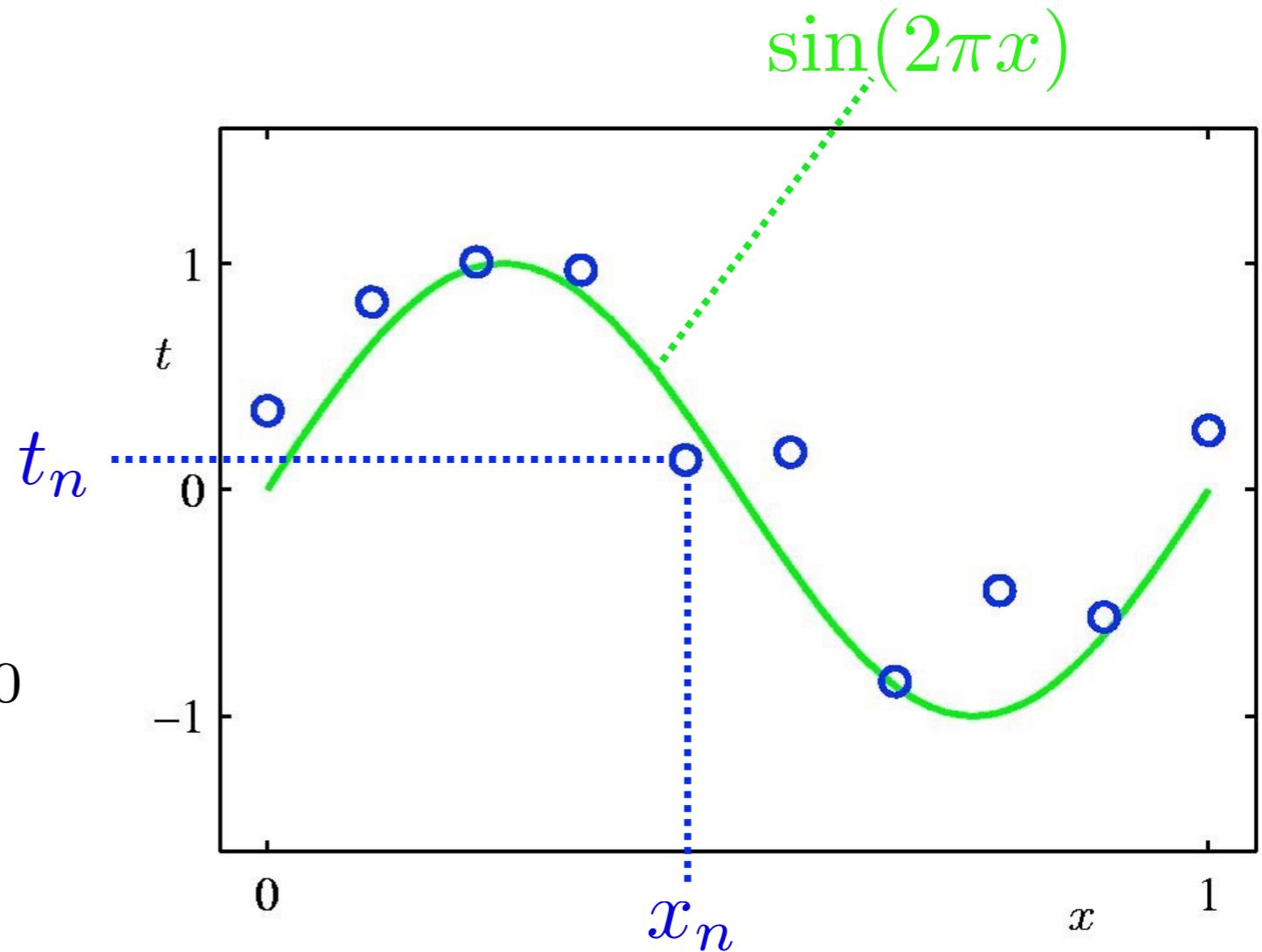
$$t = f(x) = \sin(2\pi x)$$

$$N = 10$$

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

Polynomial fit:

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

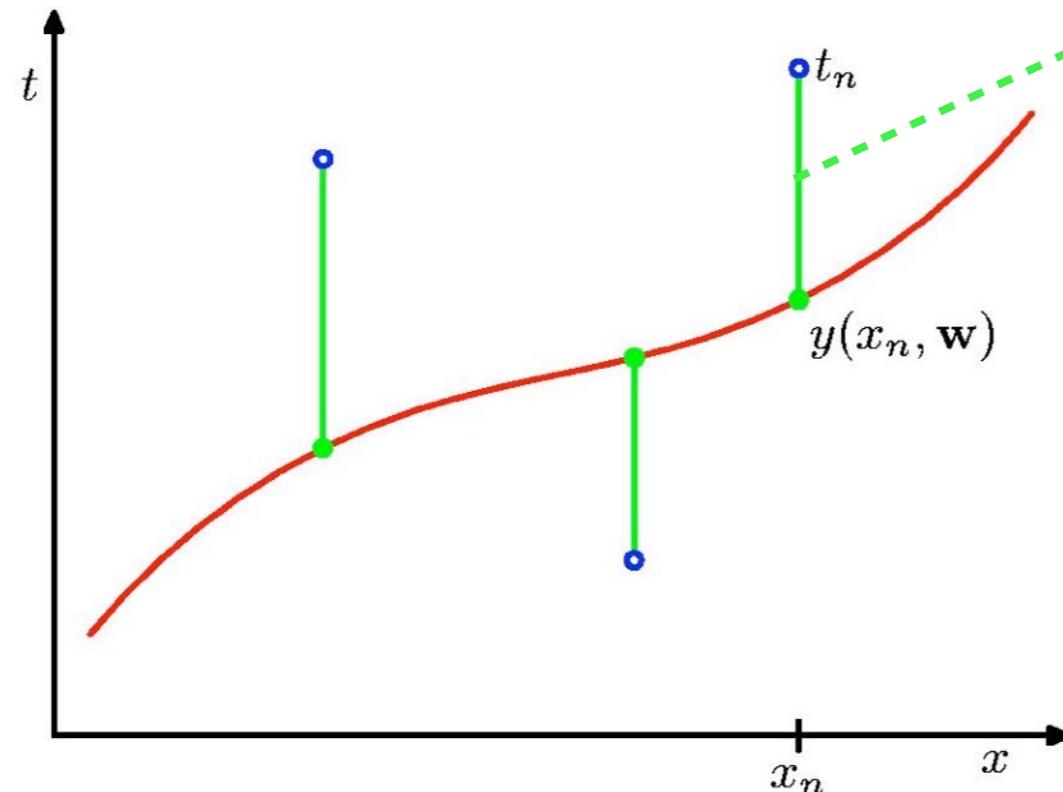




# Example: Polynomial Curve Fitting

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

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



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

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

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

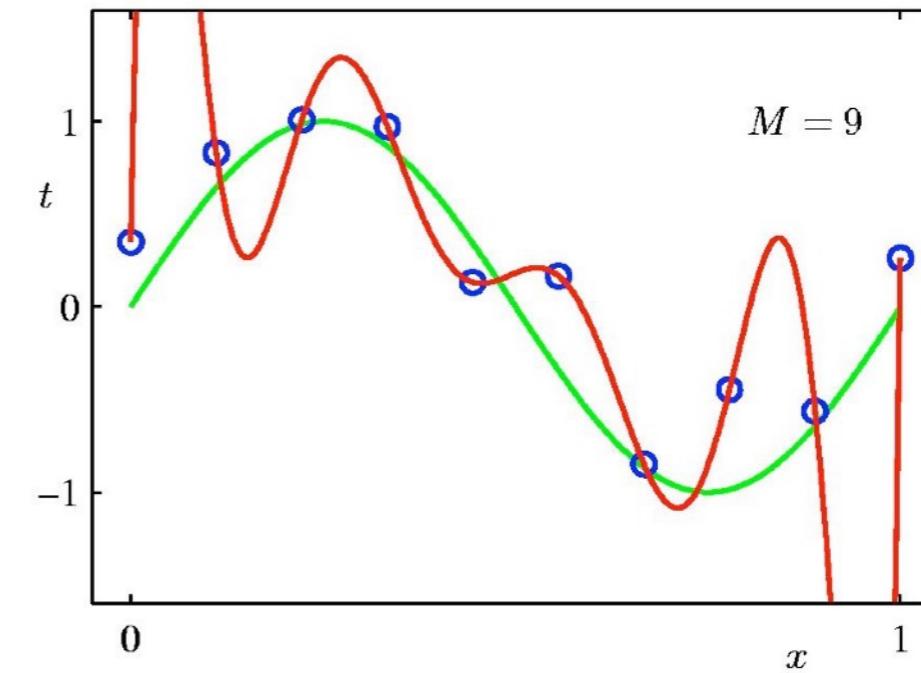
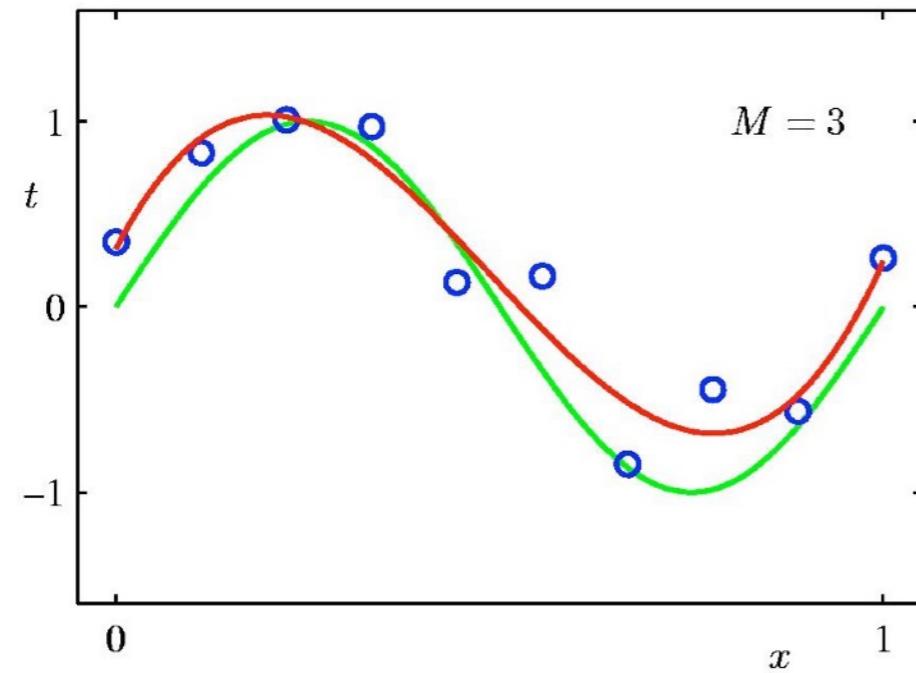
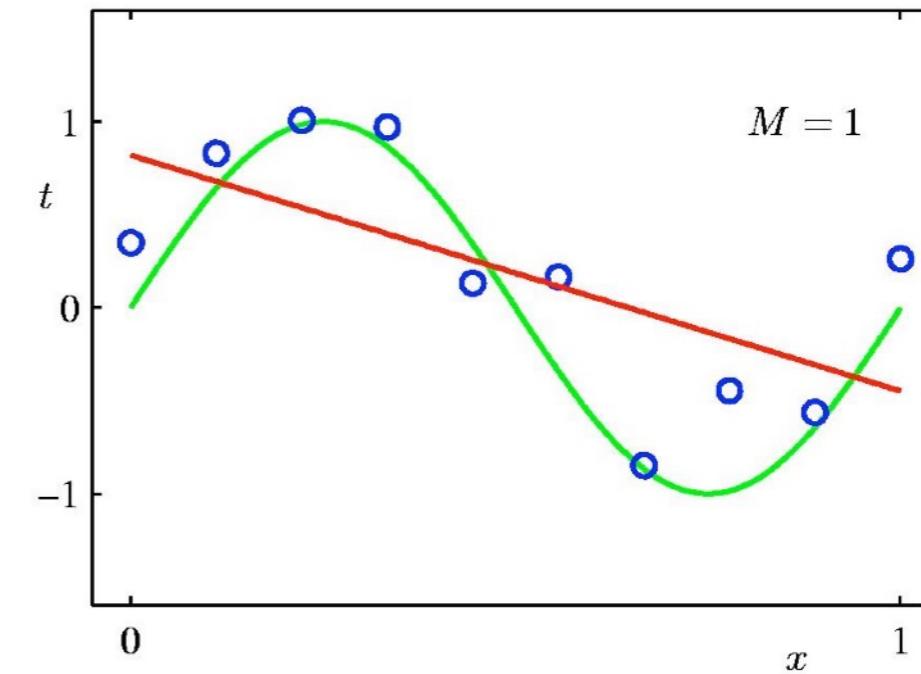
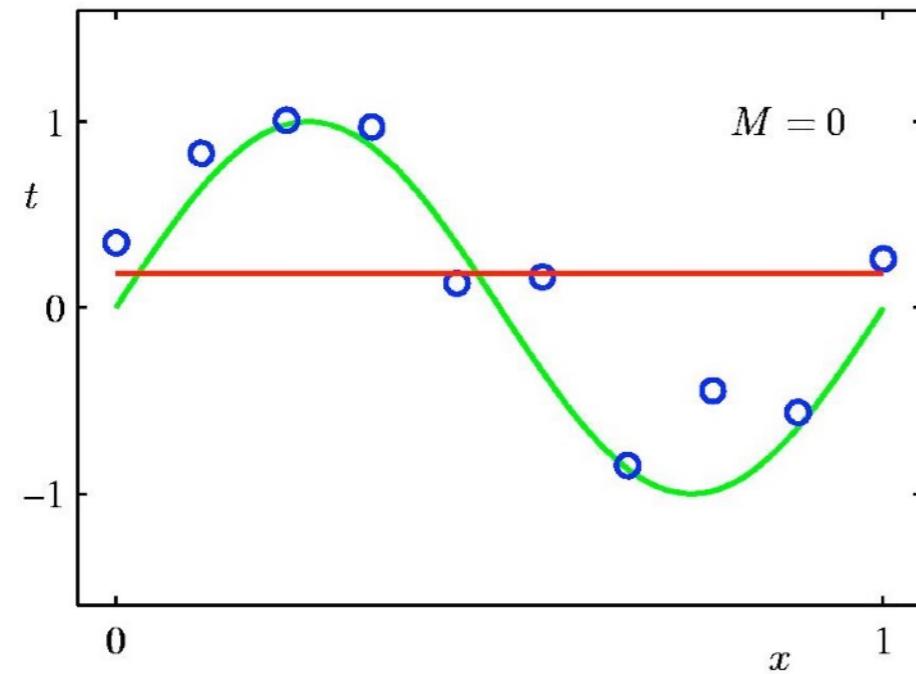
Best fit:  $\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} E(\mathbf{w})$



THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

Degree of Polynomial: M



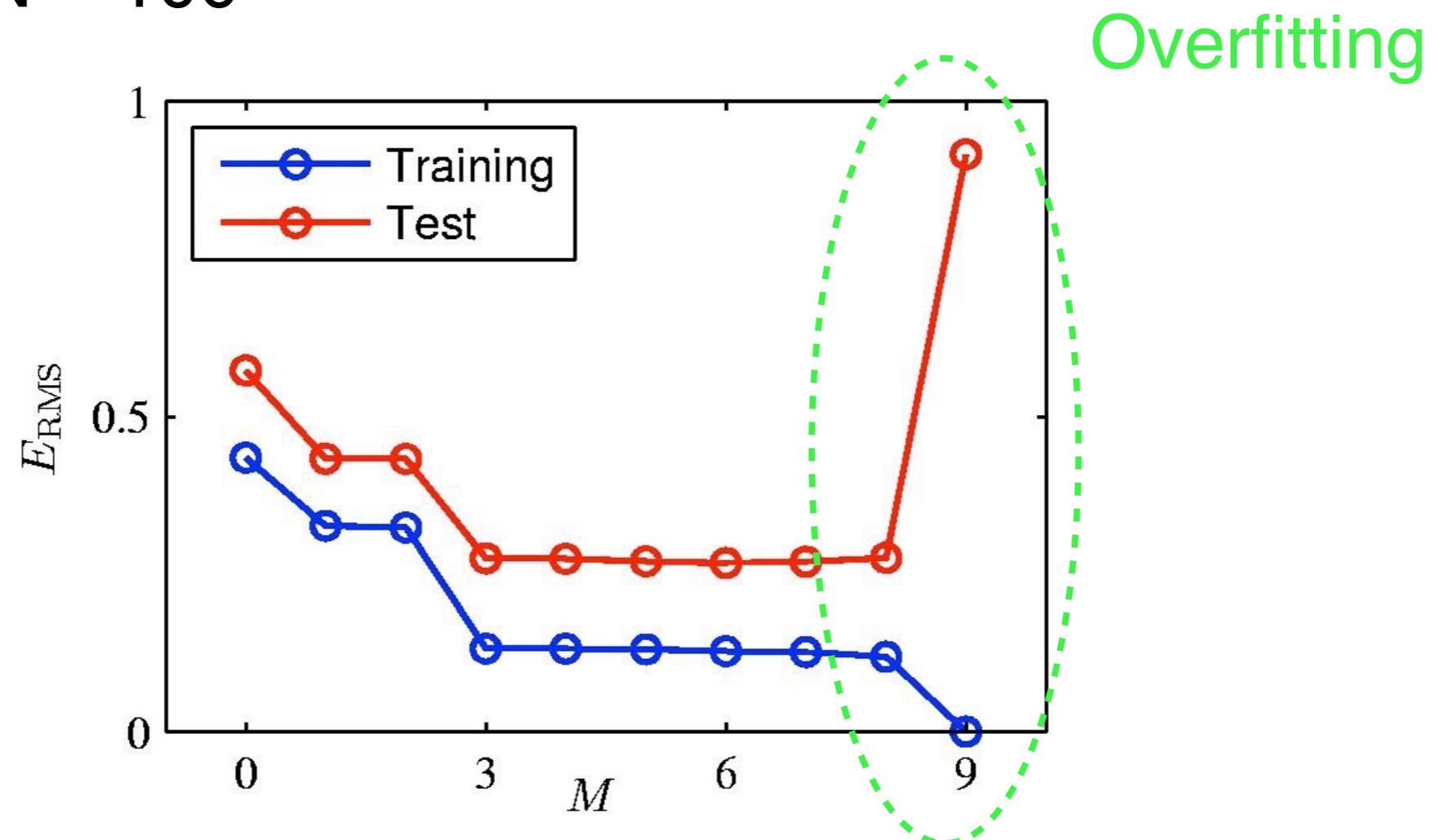


THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

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

Test set:  $N = 100$





THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

## Polynomial Coefficients

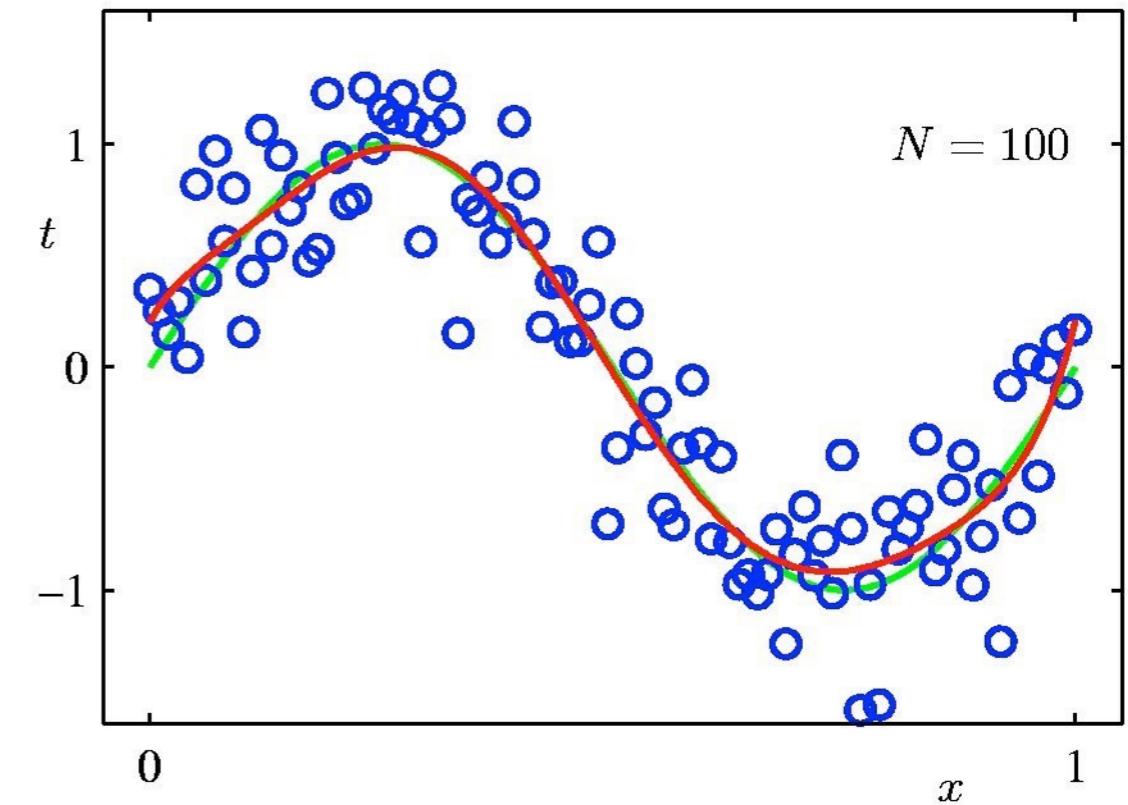
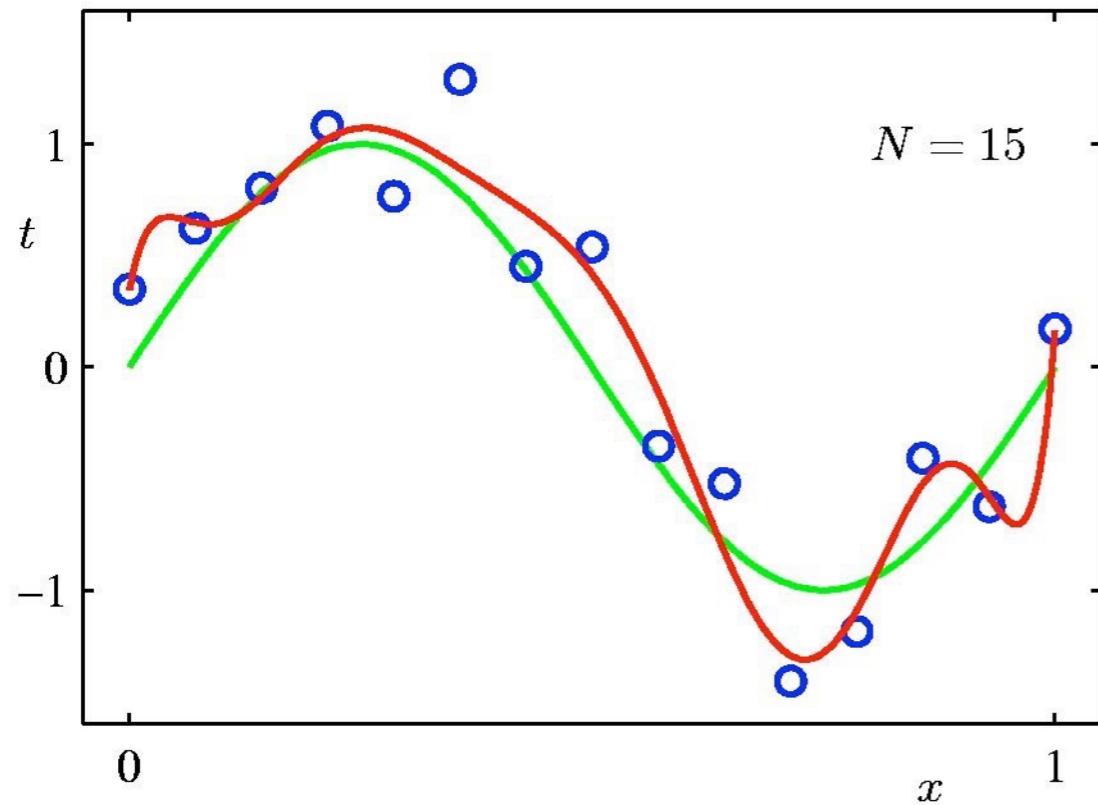
	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^*$	0.19	0.82	0.31	0.35
$w_1^*$		-1.27	7.99	232.37
$w_2^*$			-25.43	-5321.83
$w_3^*$			17.37	48568.31
$w_4^*$				-231639.30
$w_5^*$				640042.26
$w_6^*$				-1061800.52
$w_7^*$				1042400.18
$w_8^*$				-557682.99
$w_9^*$				125201.43



THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

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





THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

Regularisation:

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

Penalise large coefficient

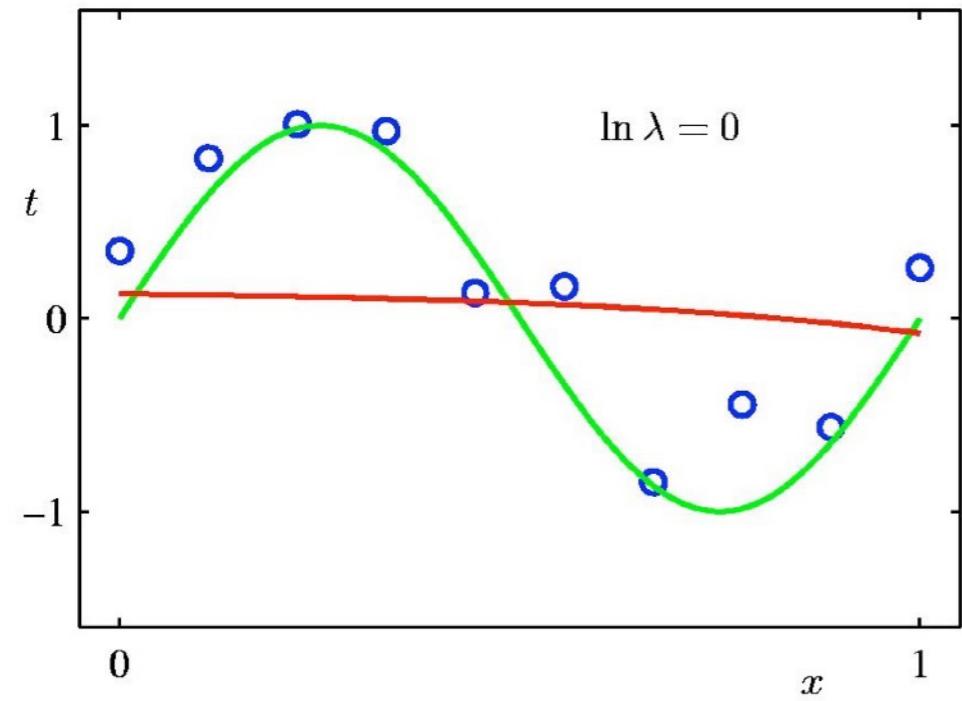
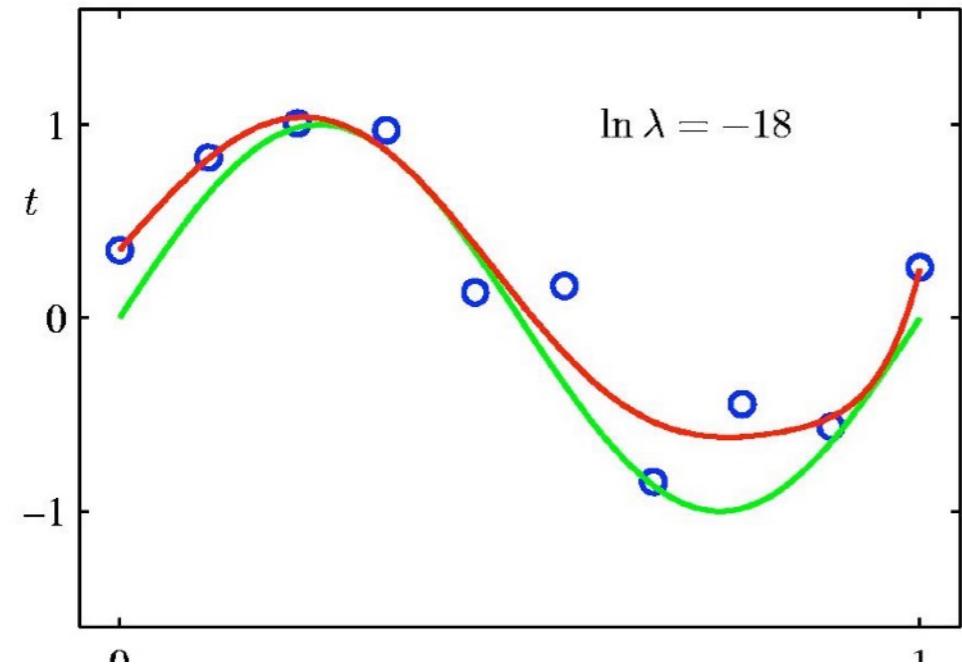
$$\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$$

$\lambda$  encodes the relative importance of the



# Example: Polynomial Curve Fitting

Regularisation:



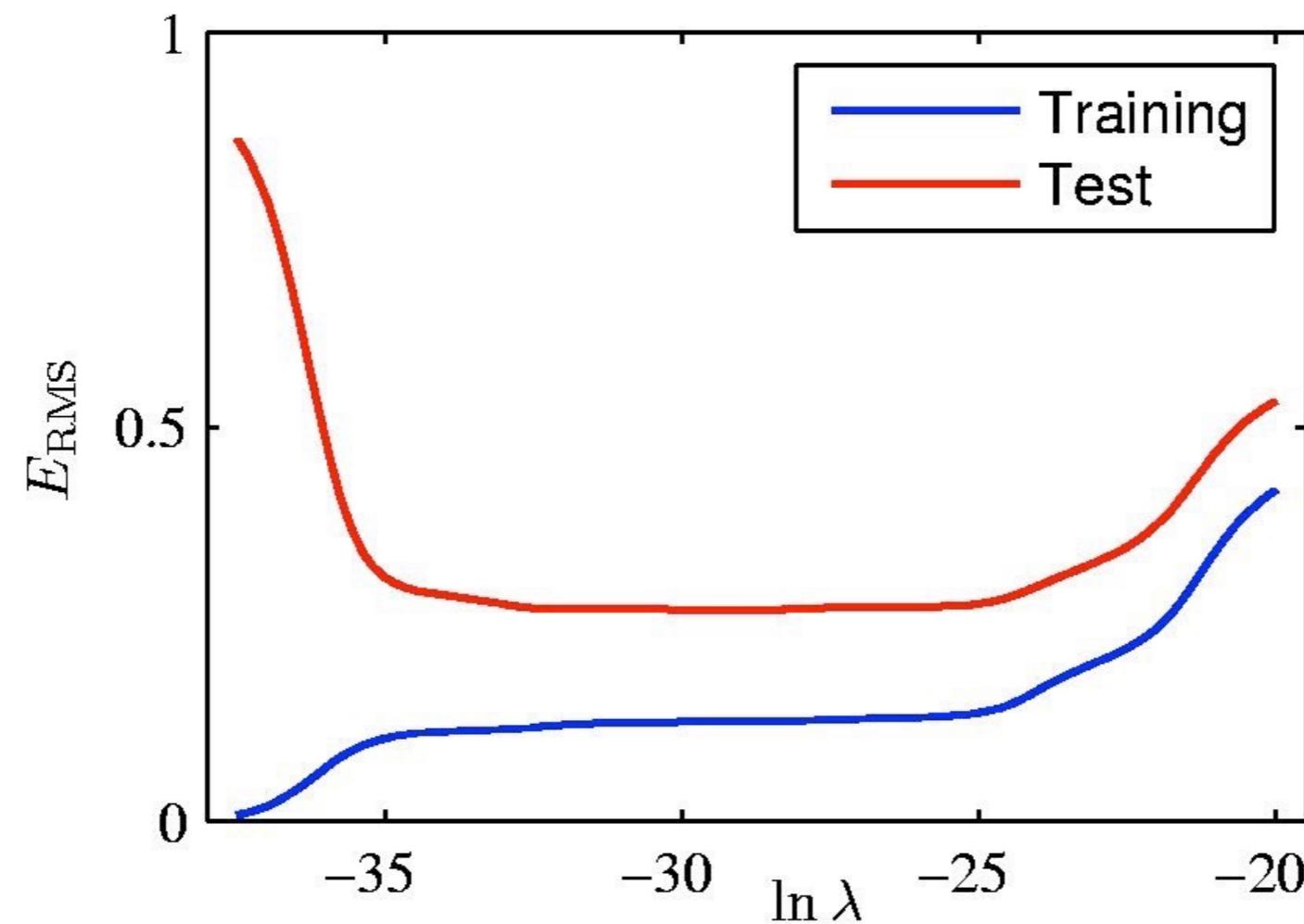
	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
$w_0^*$	0.35	0.35	0.13
$w_1^*$	232.37	4.74	-0.05
$w_2^*$	-5321.83	-0.77	-0.06
$w_3^*$	48568.31	-31.97	-0.05
$w_4^*$	-231639.30	-3.89	-0.03
$w_5^*$	640042.26	55.28	-0.02
$w_6^*$	-1061800.52	41.32	-0.01
$w_7^*$	1042400.18	-45.95	-0.00
$w_8^*$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



THE UNIVERSITY OF  
SYDNEY

# Example: Polynomial Curve Fitting

Regularisation ( $M = 9$ )





THE UNIVERSITY OF  
SYDNEY

# Linear Basis Function Models

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

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

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



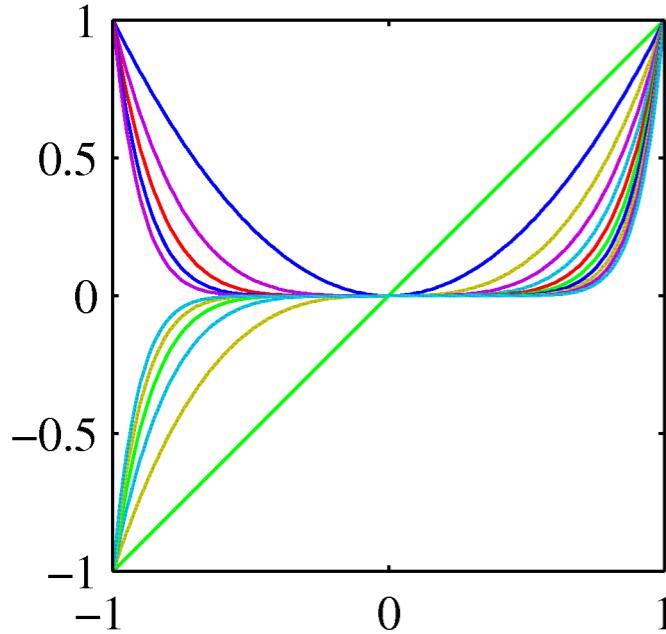
THE UNIVERSITY OF  
SYDNEY

# Linear Basis Function Models

Examples of Basis Functions in 1D:

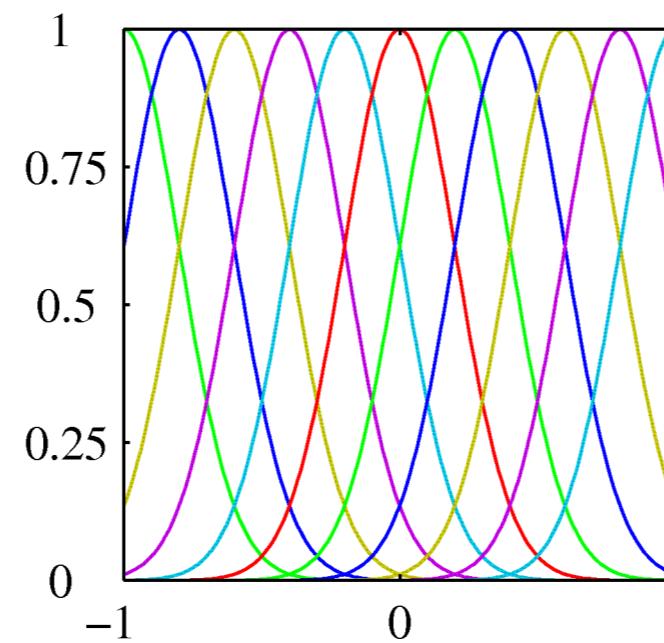
Polynomial  
Basis Functions

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



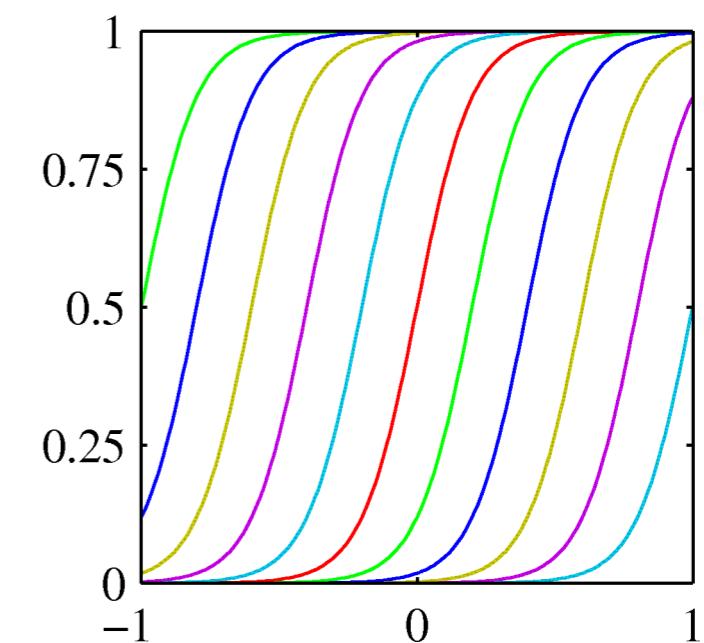
Gaussian  
Basis Functions

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



Sigmoidal  
Basis Functions

$$\phi_j(x) = \frac{1}{1 + \exp \left\{ -\frac{x - \mu_j}{s} \right\}}$$





THE UNIVERSITY OF  
SYDNEY

# Modelling Noisy Observations

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

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

equivalently,

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

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

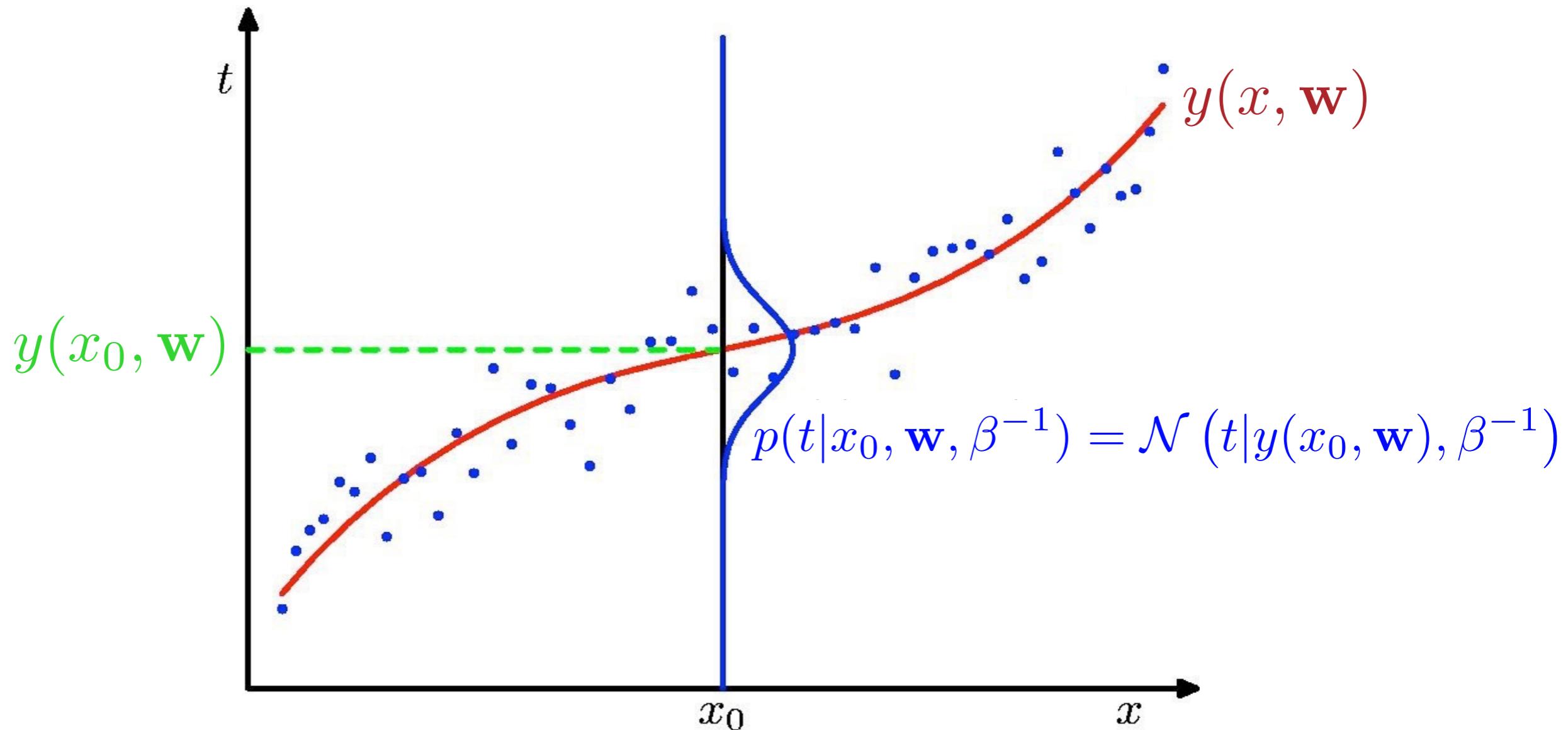
The expression of the likelihood of the iid data given the model is:

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



# Modelling Noisy Observations

THE UNIVERSITY OF  
SYDNEY





THE UNIVERSITY OF  
SYDNEY

$$\begin{aligned} p(\boldsymbol{\tau} | \mathbf{X}, \mathbf{w}, \beta^{-1}) &= \prod_{n=1}^N \mathcal{N}(t_n | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \prod_{n=1}^N \sqrt{\frac{\beta}{2\pi}} \exp -\frac{\beta(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n))^2}{2} \\ &= \left( \frac{\beta}{2\pi} \right)^{N/2} \prod_{n=1}^N \exp -\frac{\beta(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n))^2}{2} \end{aligned}$$

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

$$\begin{aligned} \ln p(\boldsymbol{\tau} | \mathbf{X}, \mathbf{w}, \beta^{-1}) &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln 2\pi - \beta \sum_{n=1}^N \frac{(t_n - \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_n))^2}{2} \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln 2\pi - \beta E(\mathbf{w}) \end{aligned}$$



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

$$\Rightarrow \frac{\partial \ln p(\boldsymbol{\tau} | \mathbf{w}, \beta^{-1})}{\partial \mathbf{w}} \Big|_{\mathbf{w}_{\text{ML}}} = 0$$

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

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

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

where

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



THE UNIVERSITY OF  
SYDNEY

# Maximum Likelihood

We can also maximise the log likelihood with respect to the noise precision parameter  $\beta$ .

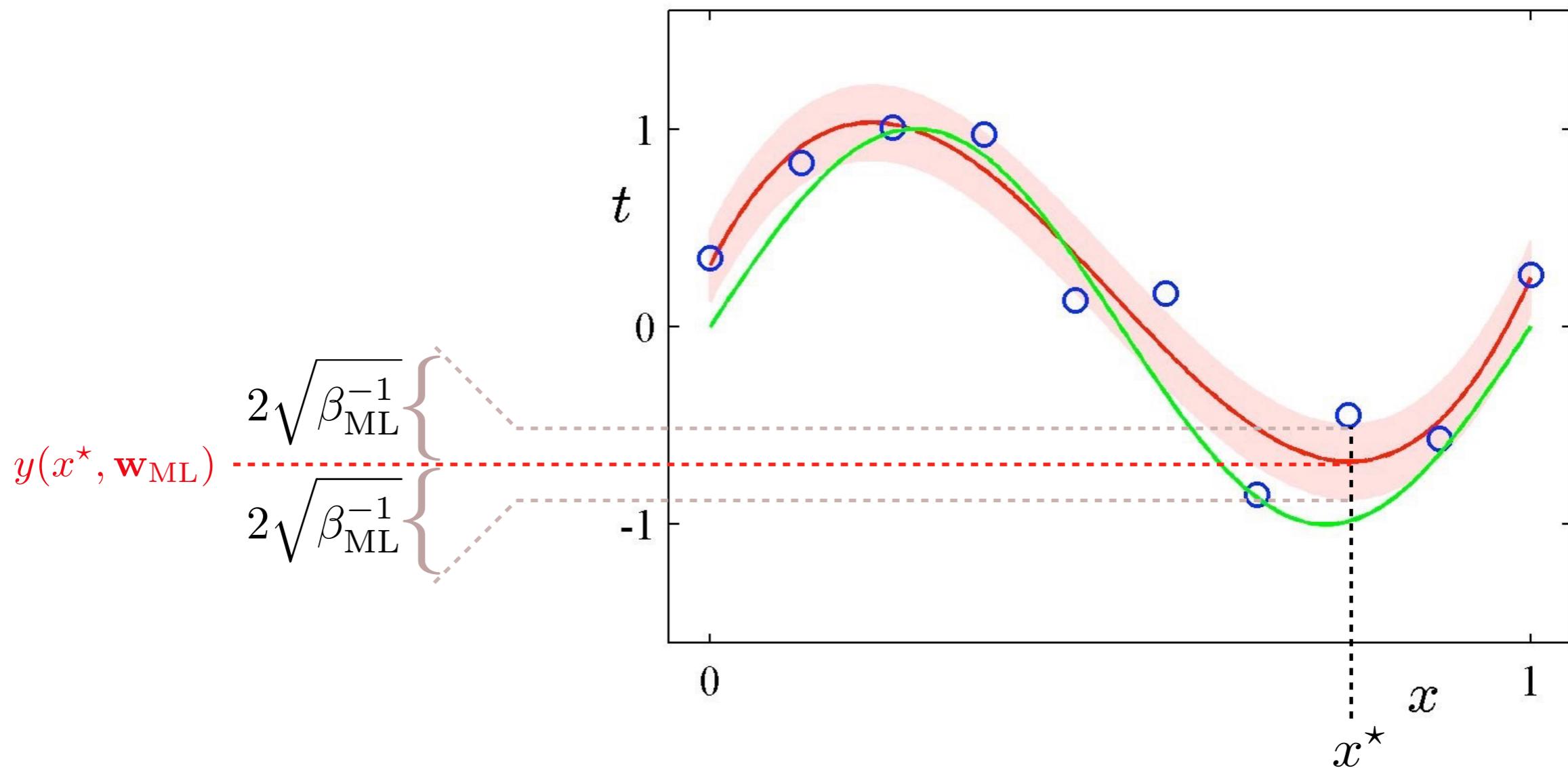
$$\frac{1}{\beta_{\text{ML}}} = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{w}_{\text{ML}}^T \boldsymbol{\phi}(\mathbf{x}_n))^2$$



THE UNIVERSITY OF  
SYDNEY

# Predictive Distribution

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





THE UNIVERSITY OF  
SYDNEY

# Sequential Learning

Deal with Large Datasets

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

Stochastic Gradient Descent

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

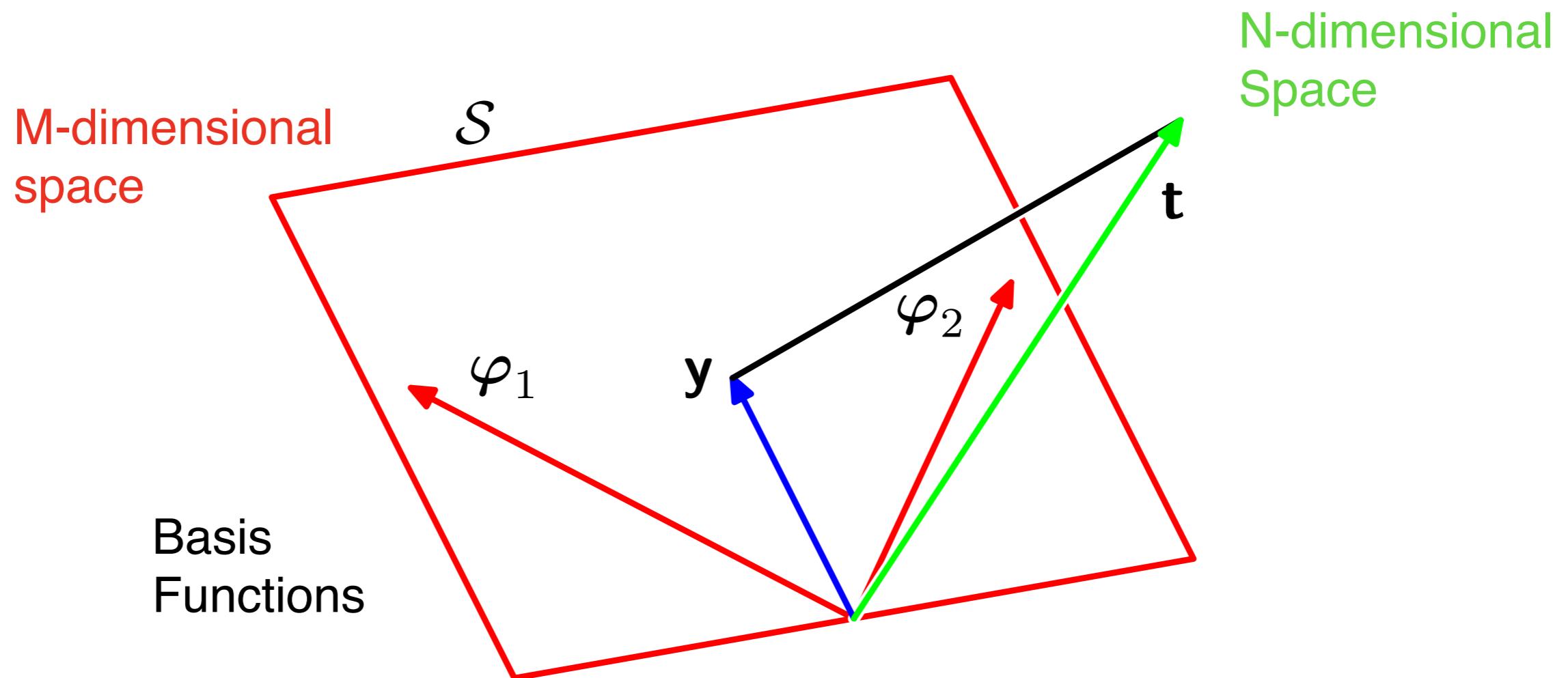
$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)T} \boldsymbol{\phi}_n) \boldsymbol{\phi}_n$$

Least Mean Square algorithm (LMS)



THE UNIVERSITY OF  
SYDNEY

# Geometry of Least Squares





THE UNIVERSITY OF  
SYDNEY

# 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:  $E(\mathbf{w}) = E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$   
Data Term      Regularisation Term

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

Following previous maximisation procedure:

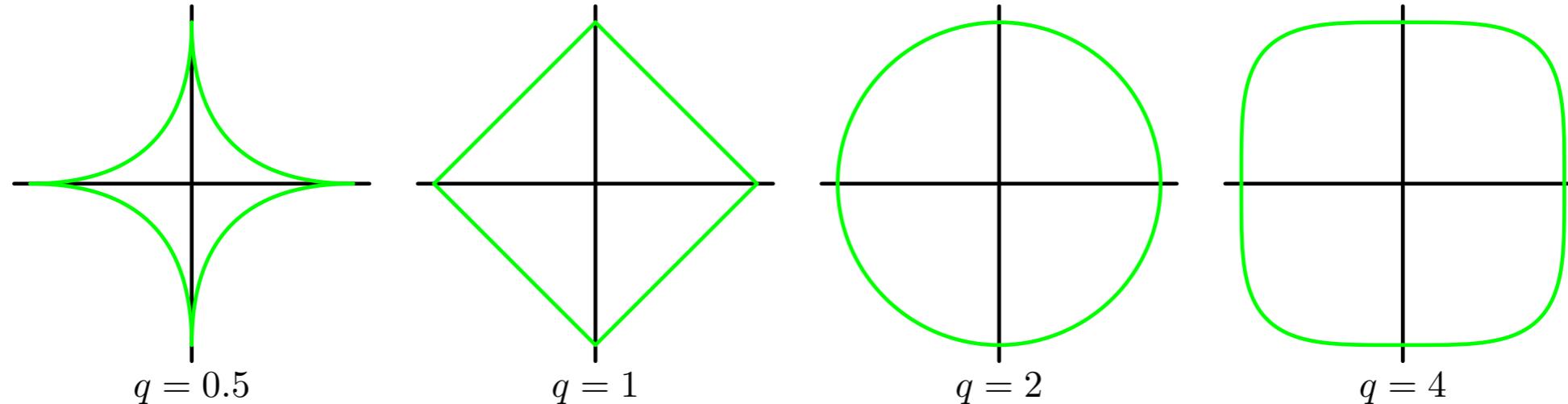
$$\mathbf{w} = (\lambda \mathbf{I} + \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{\tau}$$



THE UNIVERSITY OF  
SYDNEY

# Generalised Regulariser

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





THE UNIVERSITY OF  
SYDNEY

# Multiple Outputs

$K > 1$  target variables.

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

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

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

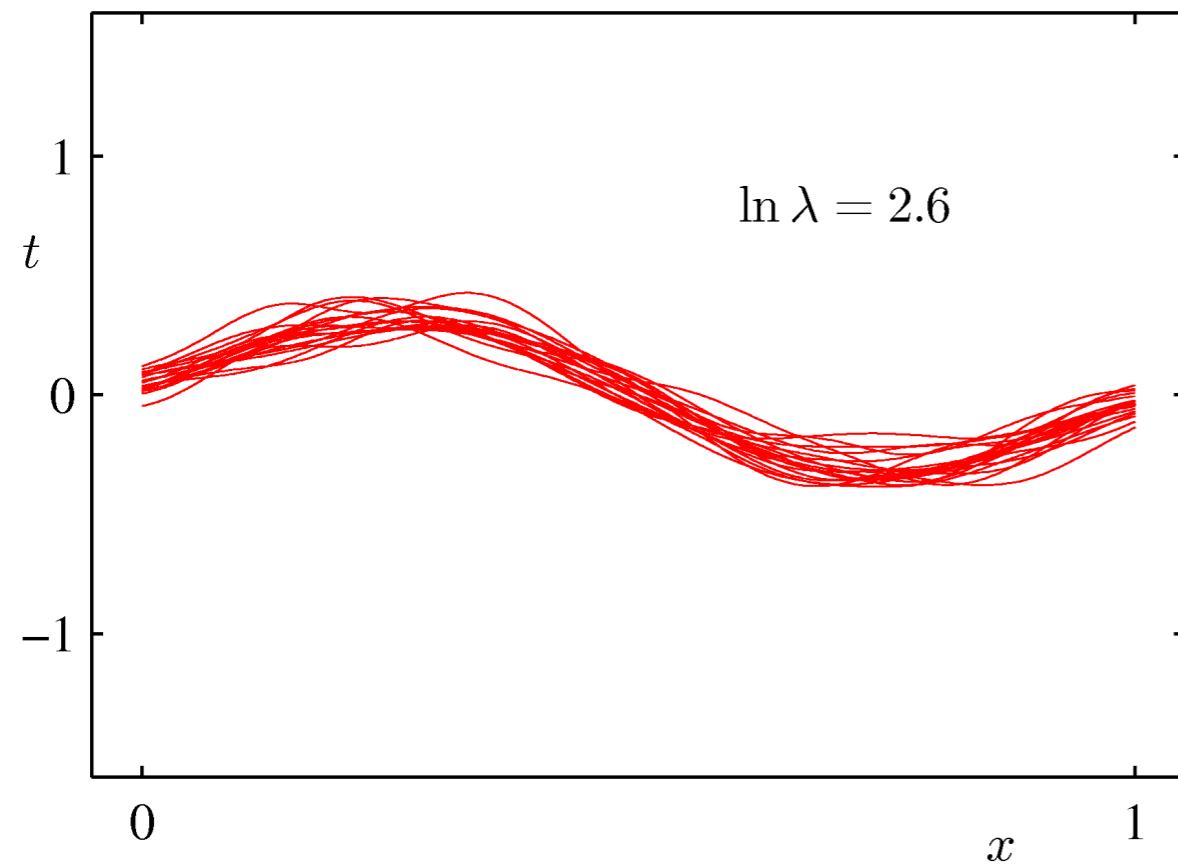
$$\mathbf{W}_{\text{ML}} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{T}.$$



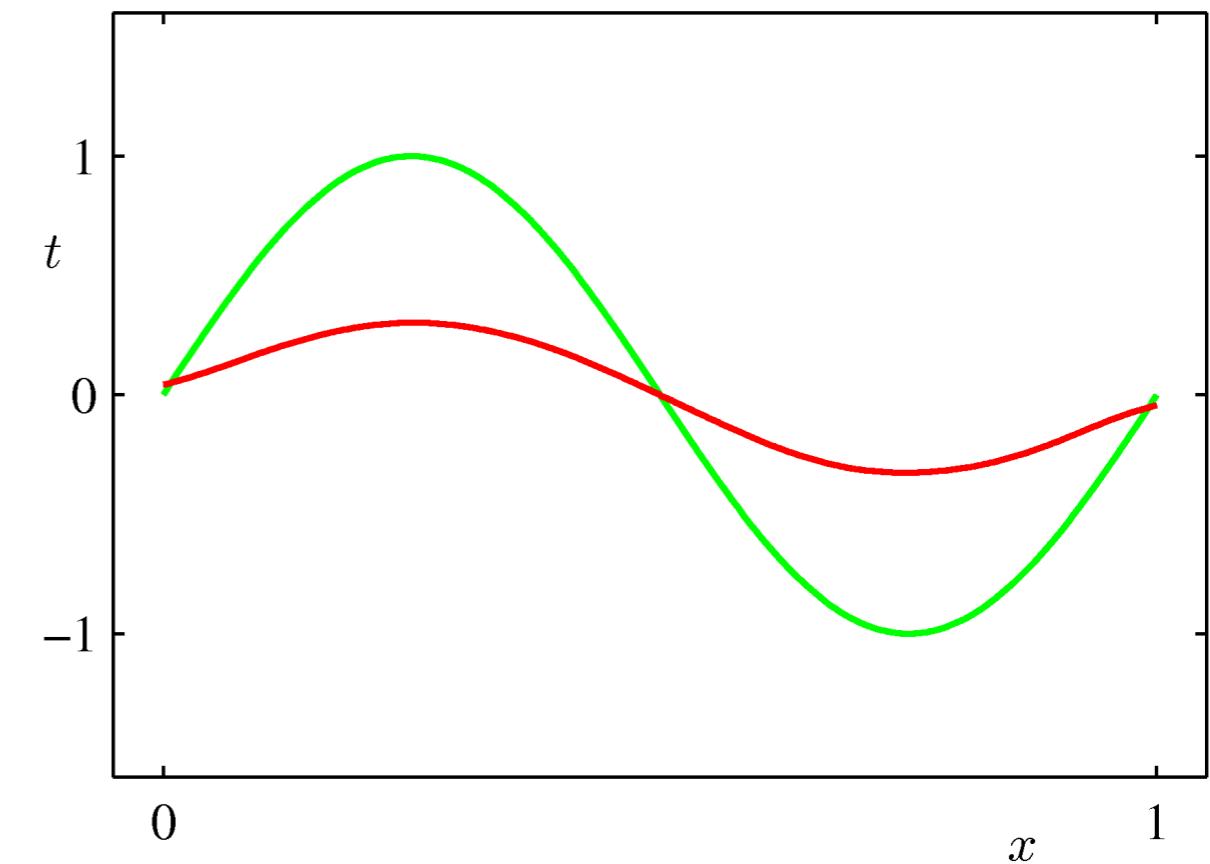
THE UNIVERSITY OF  
SYDNEY

# Bias-Variance Visualisation

20 datasets with varying regularisation parameter.



Result of fitting the model to  
each dataset.



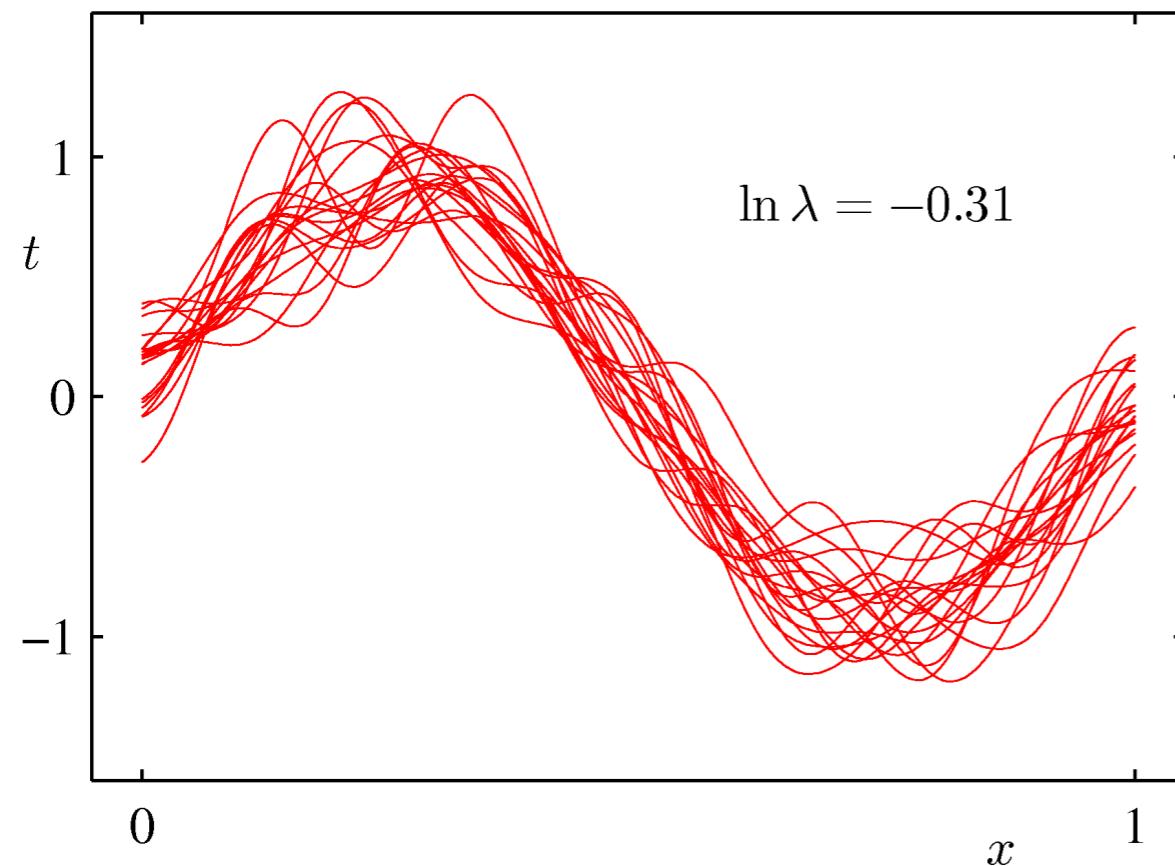
Average of the fits.



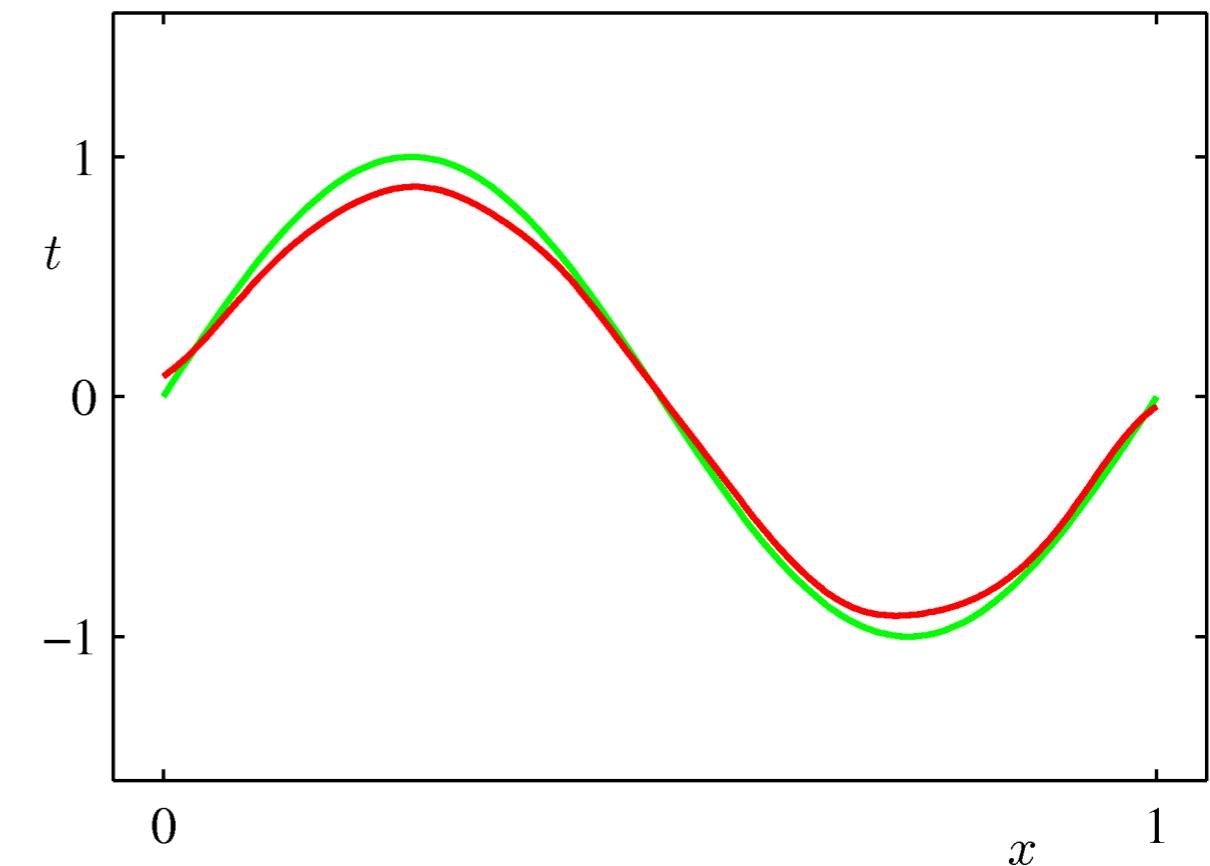
THE UNIVERSITY OF  
SYDNEY

# Bias-Variance Visualisation

20 datasets with varying regularisation parameter.



Result of fitting the model to  
each dataset.



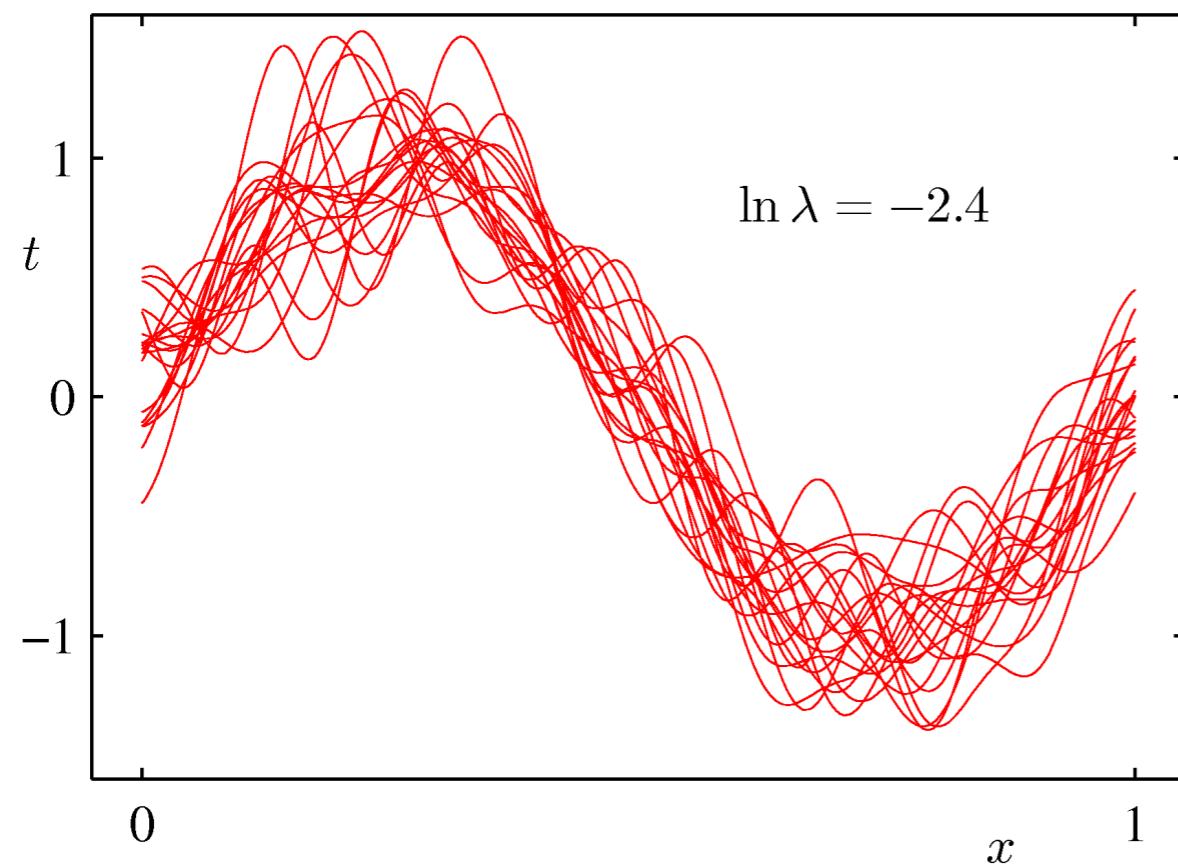
Average of the fits.



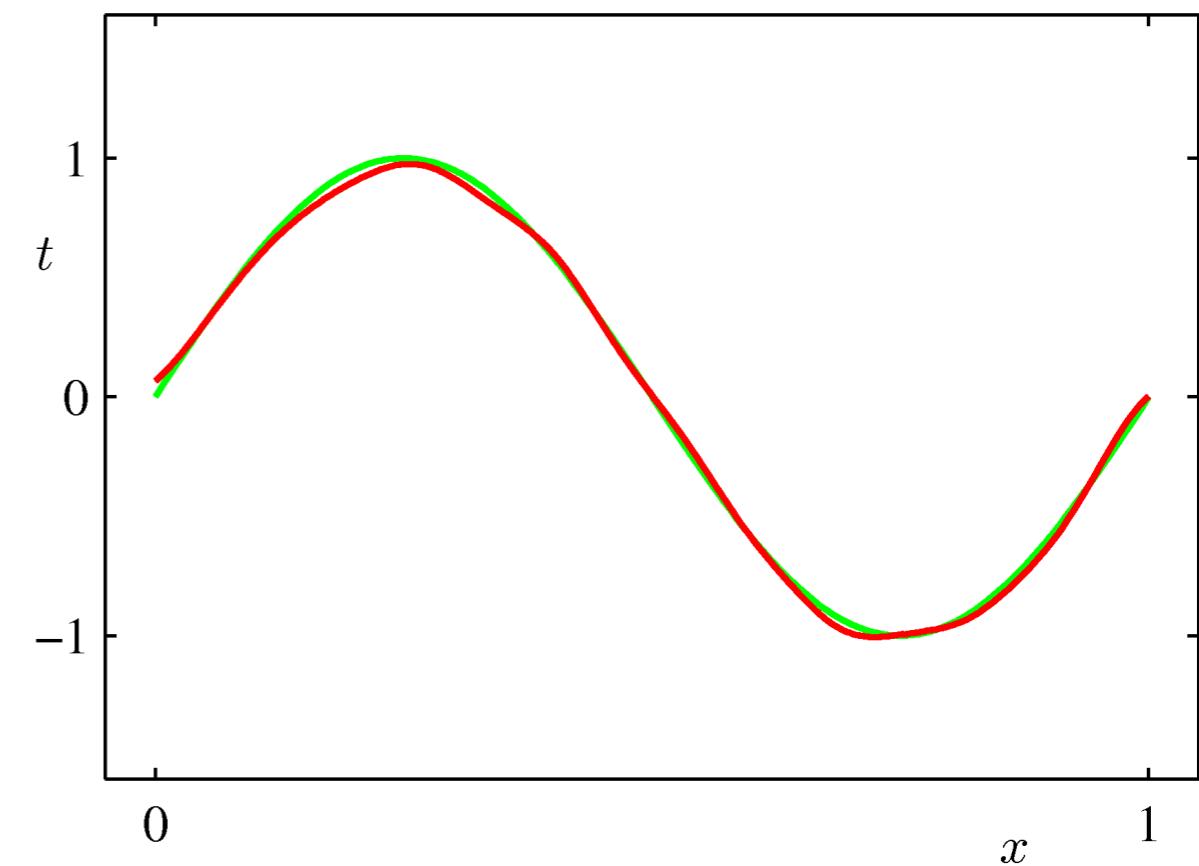
THE UNIVERSITY OF  
SYDNEY

# Bias-Variance Visualisation

20 datasets with varying regularisation parameter.



Result of fitting the model to each dataset.



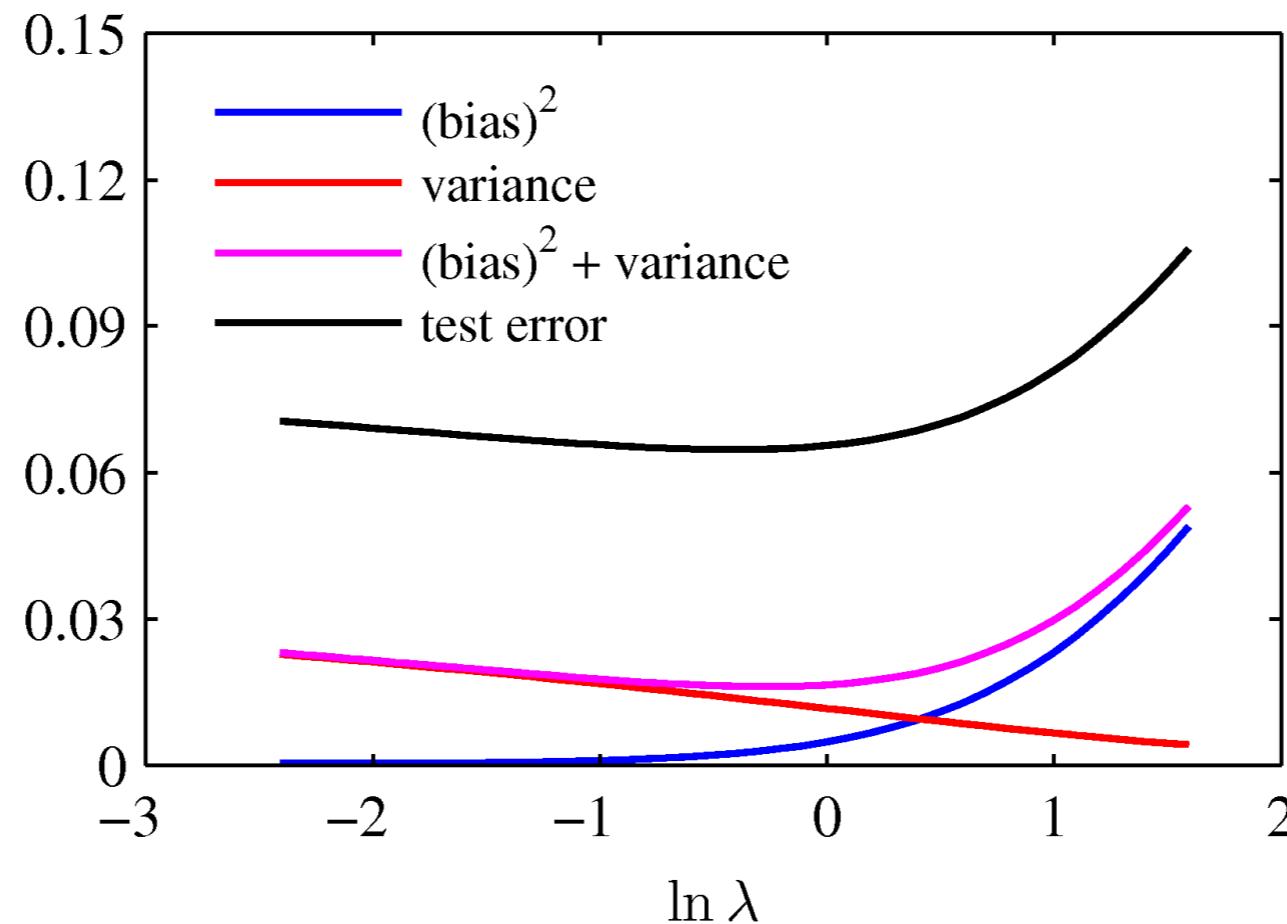
Average of the fits.



# The Bias-Variance Trade Off

THE UNIVERSITY OF  
SYDNEY

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





THE UNIVERSITY OF  
**SYDNEY**

# Example Real Applications

# Probabilistic Crime Model

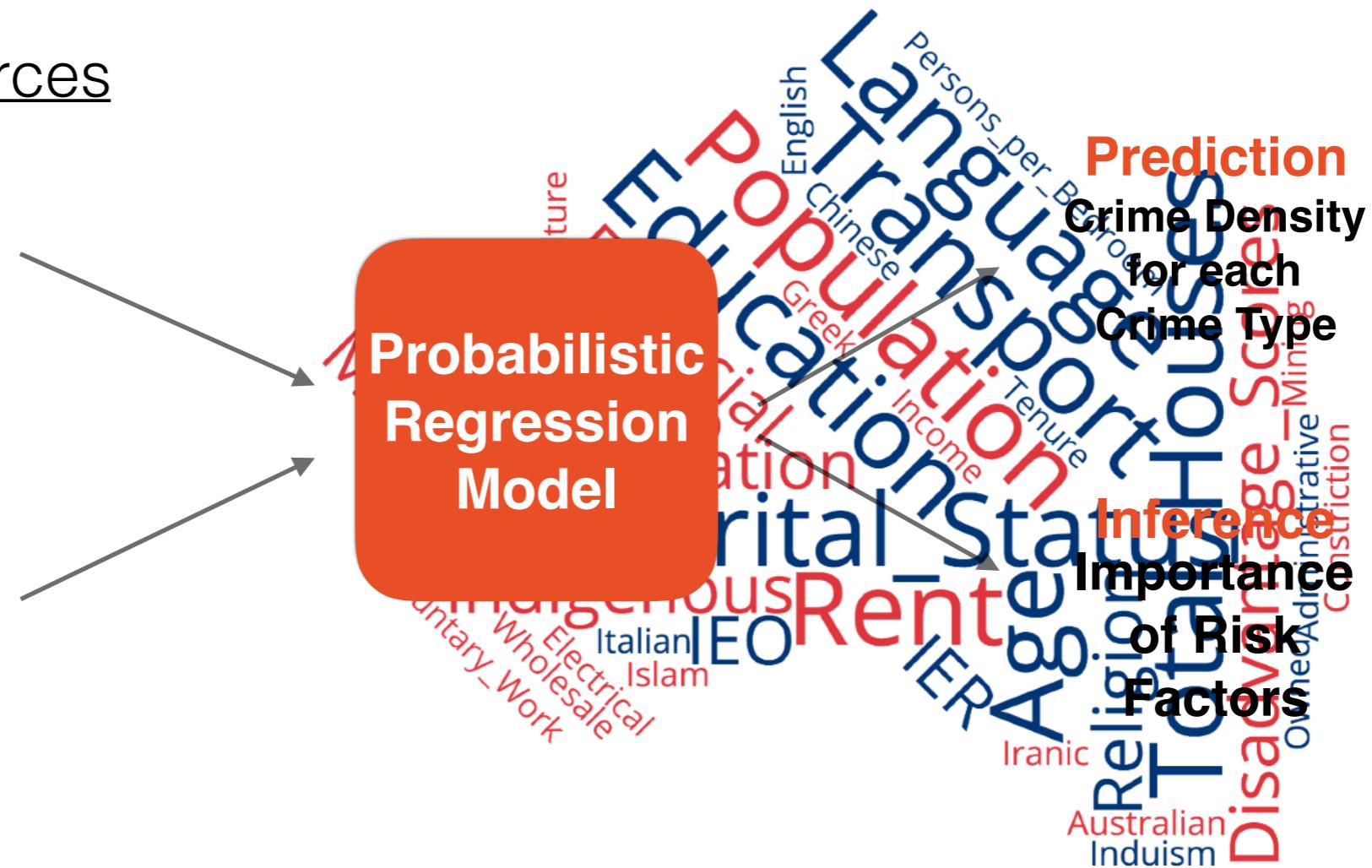


THE UNIVERSITY OF  
**SYDNEY**

# Information Sources

# Historic Space-Time Criminal Records

# Demographic Information

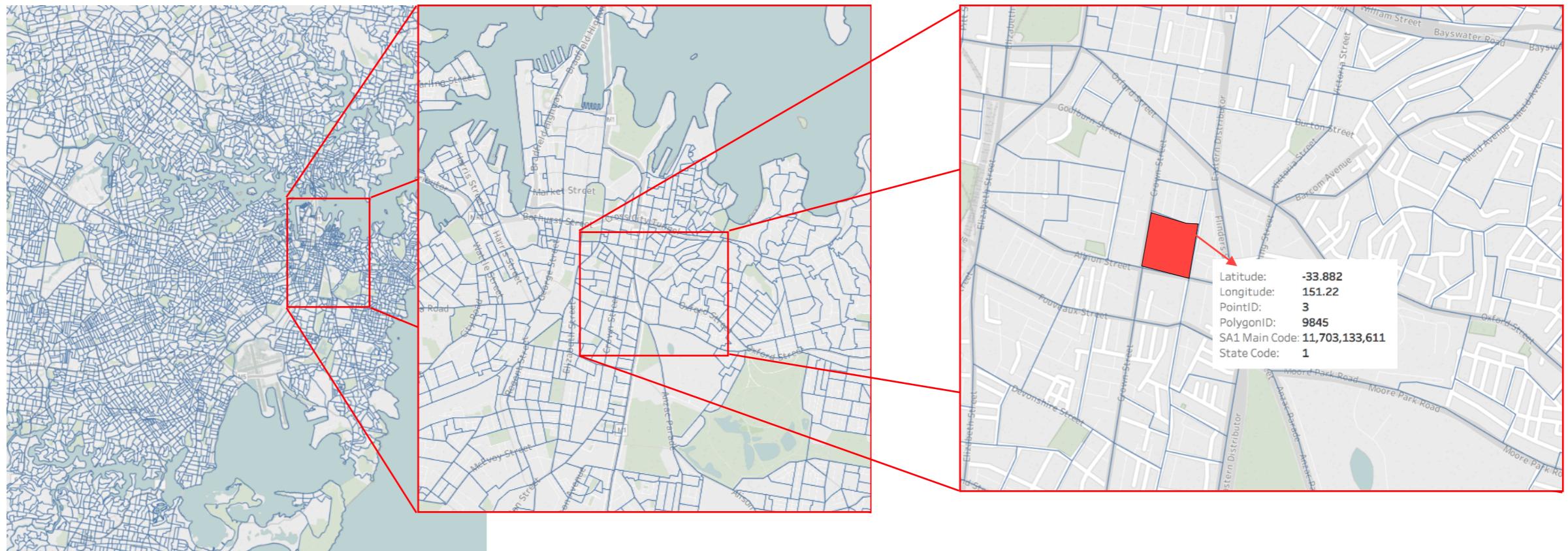




THE UNIVERSITY OF  
SYDNEY

# Probabilistic Crime Model

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

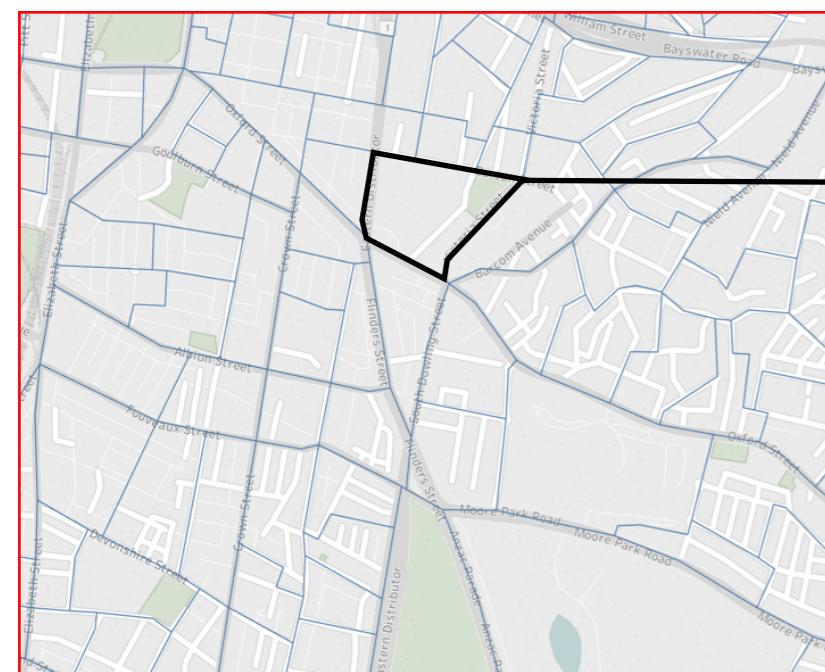




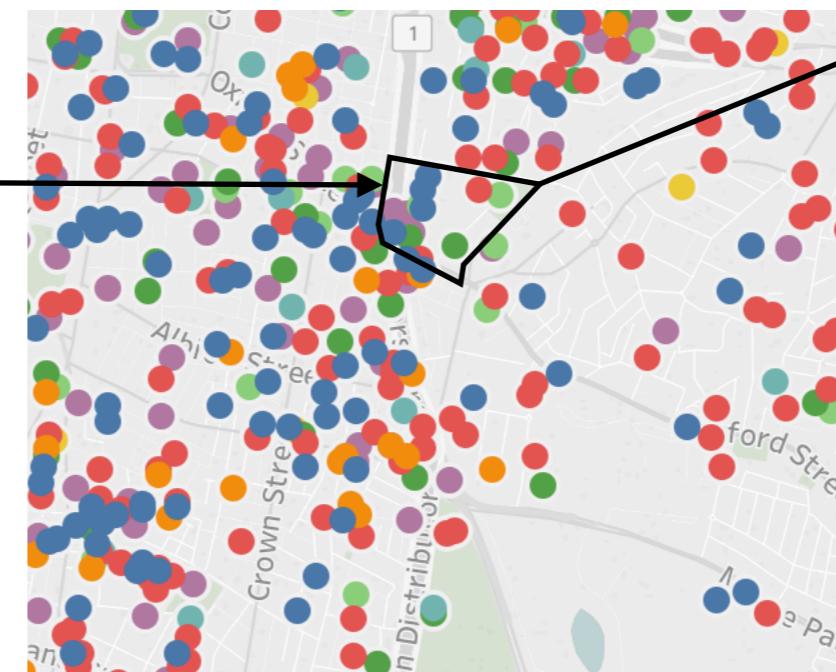
THE UNIVERSITY OF  
SYDNEY

# Probabilistic Crime Model

## 2. Accumulate Crime Occurrence per Geographic Unit



COPS/BOCSAR  
Unit Record Incident Data



Example Simulated Data

Latitude: -33.882  
Longitude: 151.22  
PointID: 3  
PolygonID: 9845  
SA1 Main Code: 11,703,133,611  
State Code: 1

Number of Assaults  
Number of Robberies  
Number of MVT  
...



THE UNIVERSITY OF  
SYDNEY

# Probabilistic Crime Model

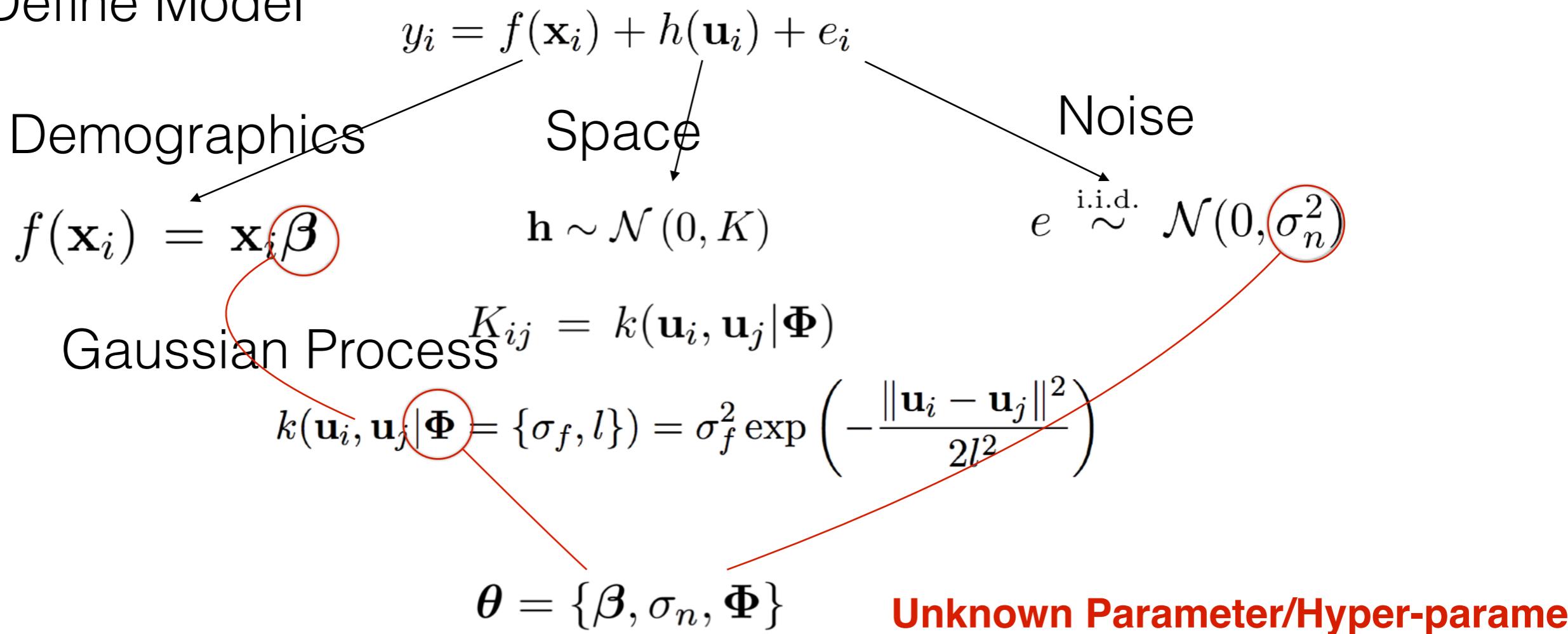
## 4. Ensemble Database

Geographic Area		Demographic																		Crime Counts									
Spatial Coordinates																													
1134623	309	2.6.47249191	0	2838	2765	0	0	0	1179	4333	61.4886731	216.828479	71.197411	64.7249191	48.5436893	142.394822	177.993528	113.268608	55.0161812	48.5436893	0	1175	1136	1186	1196	10			
1134524	435	1.2.29885057	391	2961	1987	0	6.89655172	6.89655172	1178	2400	52.8735632	59.7701149	27.5862059	45.9770115	280.45977	206.896552	110.344828	124.137931	66.6666667	18.3908046	6.89655172	1109	1085	969	1206	10			
1135115	601	2.3.22778702	430	2166	1729	9.98336106	4.99158053	14.0750416	793	2084	81.530782	139.767055	33.2778702	88.1863561	131.447587	184.69218	114.808652	114.808652	68.2196339	26.6222963	16.6389351	1073	1062	992	1111	9			
1132806	306	1.3.26797386	440	1468	1392	13.0718954	0	13.0718954	570	2167	55.5555556	75.1633987	45.751634	68.627451	183.006536	209.150327	124.183007	114.379085	55.5555556	65.3594771	0	1015	981	957	1078	8			
1137514	562	1.1.77935959	580	1531	1339	0	0	0	460	2167	56.9395108	85.4092708	110.849395	120.569395	140.569395	149.466192	108.540925	76.5124555	26.6903915	10.6761566	968	938	997	987	5				
1135422	242	1.4.1322314	0	2374	2611	16.5289256	12.3966942	28.9256198	780	3100	24.7933884	95.0413232	103.305785	115.702479	103.305785	99.1735537	202.479339	173.553719	53.7190083	12.396694	16.5289256	1167	1152	1181	1071	8			
1132022	335	2.5.97014925	486	1183	1241	8.95522388	20.8955224	29.8507463	590	3033	56.7164179	89.5522388	59.7014925	95.5223881	110.447761	158.208955	155.223881	107.452687	77.6119403	80.5970145	8.95522388	1012	1019	992	985	5			
1133304	256	1.3.90625	600	3156	2698	0	0	0	1841	2600	23.4375	39.0625	23.4375	62.5	417.96875	210.9375	85.9375	97.85625	23.4375	15.625	0	1137	1099	982	1231	10			
1133229	380	1.2.63157895	520	2568	2137	7.89473584	0	7.89473684	1025	2084	52.8735632	59.7701149	27.5862059	45.9770115	280.45977	206.896552	110.344828	124.137931	66.6666667	18.3908046	6.89655172	1118	1098	995	1175	10			
1152625	388	1.2.57711959	700	2524	2349	0	0	0	843	1550	56.710309	56.8785289	59.2783505	142.783505	144.020619	90.206185	146.937085	150.927835	56.7010309	15.4639175	0	1158	1138	1163	1090	8			
1153843	702	5.7.12250712	350	1490	1219	8.54700855	17.0700855	17.0700855	814	2160	82.62102862	56.980057	31.3390313	92.5925026	267.806268	179.487179	108.262108	76.5230769	117.264099	39.8860394	14.2450142	977	972	872	1051	7			
1138817	612	2.3.26797386	369	2416	1114	4.90196078	0	4.90196078	818	2817	55.5555556	80.0653595	42.4816601	83.3333333	184.6405253	130.718954	129.084967	107.843137	70.2614379	80.8816565	37.5816993	1090	1081	1016	1139	9			
1137921	496	4.8.06451613	330	1112	0	6.0483871	6.0483871	554	2200	64.516129	102.822581	32.2506465	96.7741935	211.693548	131.048387	139.112903	112.903226	22.5461911	148.541114	84.5806366	58.3196972	29.1771184	95.9151194	2					
1138825	377	1.2.65251989	360	2365	1312	7.95755968	0	7.95755968	897	2074	92.8318693	84.8803636	7.95755968	55.7029178	188.328912	225.464191	148.541114	84.5806366	58.3196972	29.1771184	95.9151194	2	1040	954	1150	10			
1152879	288	1.3.47211959	260	1750	1100	10.4166667	20.8333333	20.8333333	720	1896	62.5	83.3333333	41.6666667	79.8611111	218.771	177.1777	163.194444	52.0833333	69.4444444	62.5	15667	1008	974	1017	6				
1132801	464	3.1.5517241	495	2170	1937	8.62068966	12.9310345	21.5517241	812	2990	88.362059	56.0344828	36.6317931	20.206185	185.344828	252.155172	105.603426	90.5172414	36.637931	15.0862069	23.7058696	1084	1047	102	1117	9			
1134010	431	1.2.32018561	490	2100	1739	0	6.9605684	6.9605684	985	3033	23.2018561	39.4431555	11.604081	97.4477954	64.037123	169.373555	60.324826	39.4431555	10.8816703	13.9211137	1076	1056	97	1166	10				
1132810	326	3.5.70342205	440	1652	1698	5.70342205	11.0468441	11.0468441	676	2817	85.5513308	127.3764	51.3307985	57.0342205	146.38783	184.41064	142.4585551	106.453878	57.0342205	11.4068441	998	970	978	1026	7				
1141425	304	1.3.28947368	524	2895	2229	0	0	0	1447	2700	52.6315789	30.4736842	13.738421	42.7631579	142.210526	230.263158	78.9473684	65.7897437	20.652632	9.86842105	0	1149	1129	10	1218	10			
1136948	441	2.4.53514739	435	2111	2062	0	0	0	686	2000	61.2244898	140.589569	56.6893424	65.7596372	95.2380952	163.265306	131.519274	145.124717	90.7029478	31.7460317	18.1405896	1094	1100	1102	1068	8			
1153416	439	1.2.77940433	663	2271	2380	0	0	0	834	2000	54.6666667	79.8611111	21.7416667	40.4655554	109.339408	104.783599	168.731240	20.511339	63.7813212	13.667426	1169	1141	1187	1105	9				
1152833	417	2.4.79616307	340	1449	1289	7.1942446	0	7.1942446	876	2167	86.3303933	67.8361	23.980	40.7673861	266.18705	117.509595	122.302158	115.107914	79.1366908	26.3778969	1045	1056	961	1069	8				
1137902	537	1.1.86219739	425	1641	1567	0	0	0	547	2817	33.51595531	76.723	81.856853	68.9013035	158.286778	139.564804	150.837989	100.5586569	102.420857	67.0391061	7.44878957	1051	1033	1049	109	7			
138846	365	1.2.73972603	363	2550	1492	0	8.21917808	8.21917808	895	2500	57.5324666	10.9041	57.7494205	52.0547945	183.561644	123.287671	123.287671	10.728037	77.7028037	67.7570903	11.667426	1059	1054	952	1134	9			
141334	352	2.1.84059059	520	2595	2150	0	0	0																					



# Probabilistic Crime Model

## 5. Define Model





## 6. Learn Model

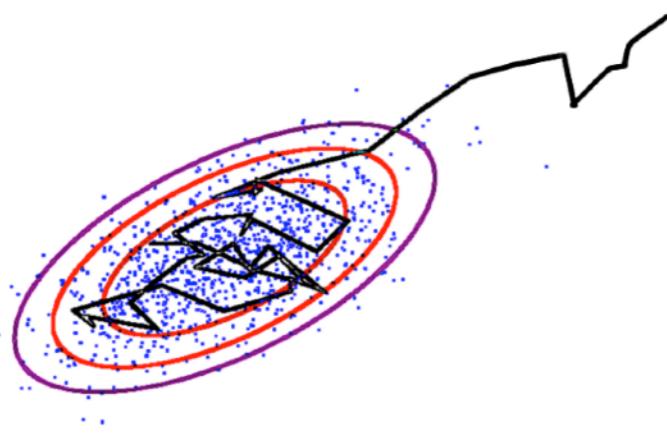
Data  $\mathcal{D} = (X, \mathbf{y})$

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

### Bayes Theorem

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

Posterior      Likelihood      Prior



Conduct sampling algorithm  
**Markov Chain Monte Carlo**

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

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

---

#### Algorithm 1 Metropolis-Hastings MCMC

---

```
1: Initialise  $\boldsymbol{\theta}^{[0]}$ .
2: for  $k = 1$  to  $M - 1$  do
3:   Sample  $u \sim U(0, 1)$ .
4:   Sample  $\boldsymbol{\theta}^p \sim q(\boldsymbol{\theta}^p|\boldsymbol{\theta}^{[k]})$ 
5:   if  $u < A = \min \left\{ 1, \frac{p(\mathcal{D}|\boldsymbol{\theta}^p)p(\boldsymbol{\theta}^p)}{p(\mathcal{D}|\boldsymbol{\theta}^{[k]})p(\boldsymbol{\theta}^{[k]})} \right\}$  then
6:      $\boldsymbol{\theta}^{[k+1]} = \boldsymbol{\theta}^*$ 
7:   else
8:      $\boldsymbol{\theta}^{[k+1]} = \boldsymbol{\theta}^{[k]}$ 
9:   end if
10: end for
```

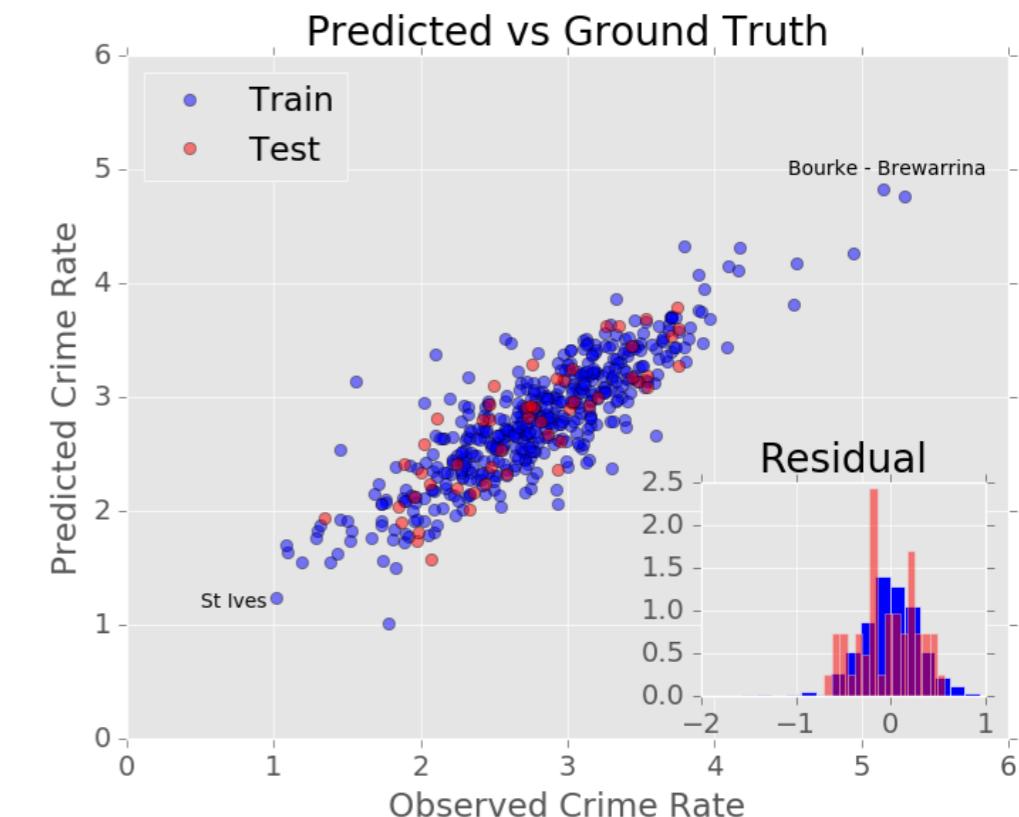
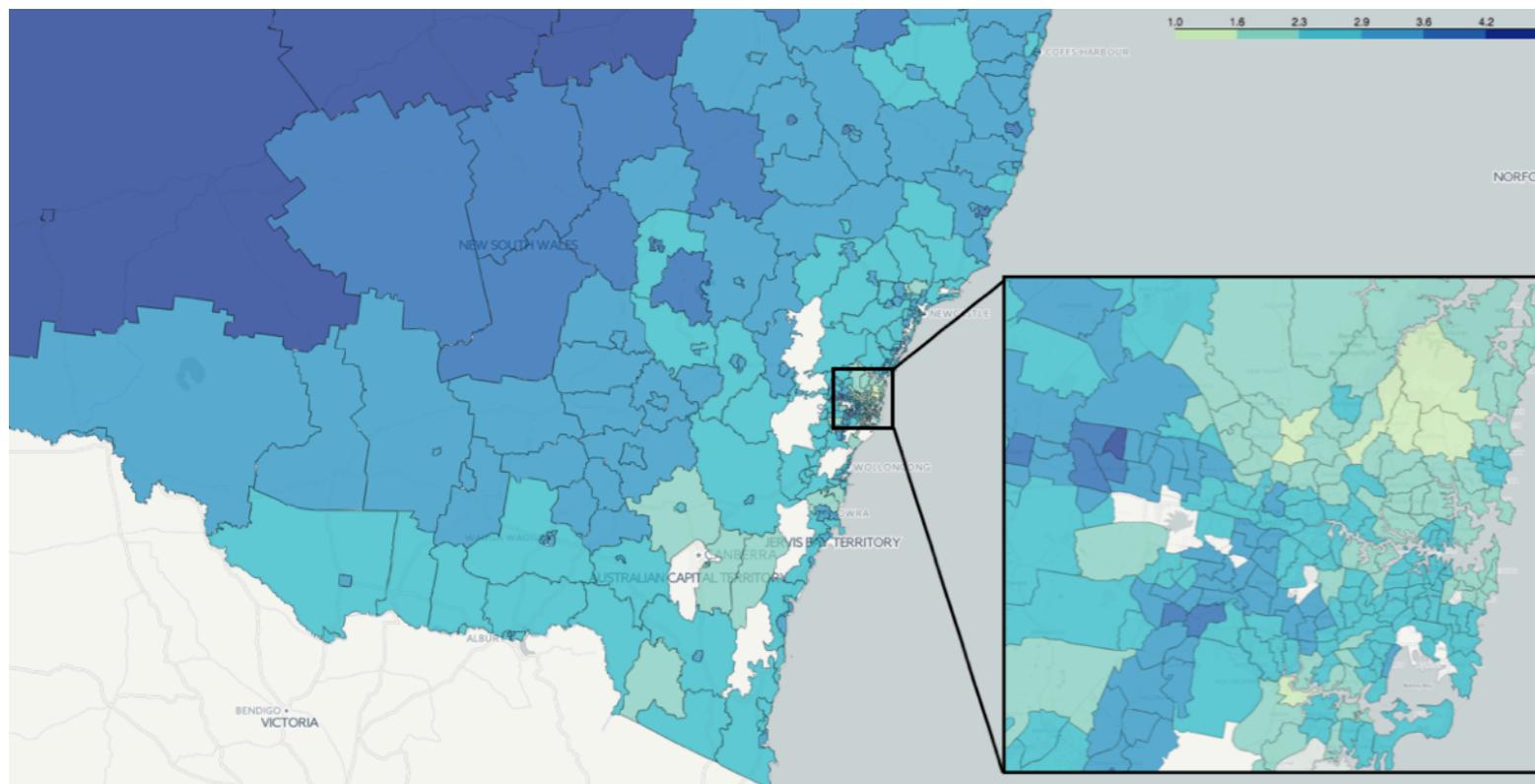
---



THE UNIVERSITY OF  
SYDNEY

# Probabilistic Crime Model

## 7. Evaluate Results, Spatial DV Assault Rates and Predictions





# Probabilistic Crime Model

## 7. Inference on Factors

