Computational Statistics
Linear Regression

Luca Bortolussi

Department of Mathematics and Geosciences
University of Trieste

Office 238, third floor, H2bis
luca@dmi.units.it

Trieste, Winter Semester 2015/2016

Outline

FITTING A STRAIGHT LINE

- Consider training data $(x_n, y_n)_{n=1,\ldots,N}$. We want to find the best linear fit to this data, i.e. the best straight line $y(x) = w_1 \cdot x + w_0$
- Let's take a curve fitting approach, and find the coefficients $\mathbf{w} = (w_0, w_1)$ that minimise sum-of-squares error

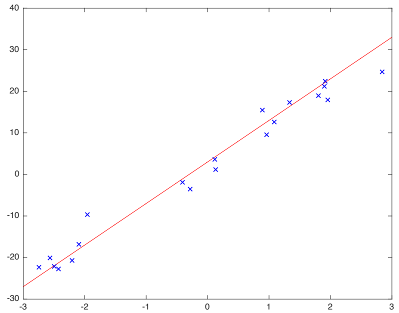
$$E(\mathbf{w}) = \sum_{n=1}^{N} [y_n - y(x_n)]^2$$

FITTING A STRAIGHT LINE

- Consider training data $(x_n, y_n)_{n=1,\dots,N}$. We want to find the best linear fit to this data, i.e. the best straight line $y(x) = w_1 \cdot x + w_0$
- Let's take a curve fitting approach, and find the coefficients $\mathbf{w} = (w_0, w_1)$ that minimise sum-of-squares error

$$E(\mathbf{w}) = \sum_{n=1}^{N} [y_n - y(x_n)]^2$$

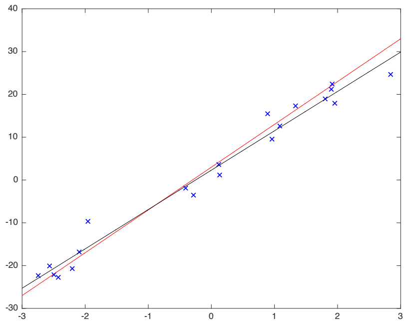
$\nabla_w E(w) = 0$



FITTING A STRAIGHT LINE

- Consider training data $(x_n, y_n)_{n=1,\dots,N}$. We want to find the best linear fit to this data, i.e. the best straight line $y(x) = w_1 \cdot x + w_0$
- Let's take a curve fitting approach, and find the coefficients $\mathbf{w} = (w_0, w_1)$ that minimise sum-of-squares error

$$E(\mathbf{w}) = \sum_{n=1}^{N} [y_n - y(x_n)]^2$$



Set $\nabla_w E(w) = 0$

$$\frac{\partial}{\partial w_0} E(w) = \sum_{n=1}^{N} -2\left[y_n - w_0 - w_1 x_n\right] = 0$$

$$\frac{\partial}{\partial w_1} E(w) = \sum_{n=1}^{N} -2\left[y_n - w_0 - w_1 x_n\right] x_n = 0$$

$\langle y \rangle = \frac{1}{N}\sum_{n=1}^{N} y_n \quad \langle x \rangle = \frac{1}{N}\sum_{n=1}^{N} x_n$

and similarly $\langle x^2 \rangle$, $\langle xy \rangle$

Now divide by N the equations above:

$$\langle y \rangle - w_0 - w_1 \langle x \rangle = 0$$

$$\langle xy \rangle - w_0 \langle x \rangle - w_1 \langle x^2 \rangle = 0$$

$$\Rightarrow \begin{cases} w_1 = \dfrac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} \\[2ex] w_0 = \langle y \rangle - w_1 \langle x \rangle \end{cases}$$

FITTING A STRAIGHT LINE

- Consider training data $(x_n, y_n)_{n=1,\dots,N}$. We want to find the best linear fit to this data, i.e. the best straight line $y(x) = w_1 \cdot x + w_0$
- Let's take a curve fitting approach, and find the coefficients $\mathbf{w} = (w_0, w_1)$ that minimise sum-of-squares error

$$E(\mathbf{w}) = \sum_{n=1}^{N} [y_n - y(x_n)]^2$$

$w_0 = 3$ $w_1 = 10$

1 DATASET

N	w_0	w_1
5	5.3812	8.1856
10	2.9735	9.6608
20	3.5493	9.6204
50	3.2084	9.9253
100	2.8327	9.8894
1000	3.0451	9.9464
10000	2.9937	10.0147
100000	3.0084	9.9992

GENERALISED BASIS FUNCTIONS

$\in \mathbb{R}^n$

- Suppose our inputs are real vectors, and outputs are real numbers, and we have observations $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$.
- We consider a set of M basis functions $\phi_j : \mathbb{R}^n \to \mathbb{R}$, and write $\boldsymbol{\phi}(\mathbf{x}) = (\phi_0(\mathbf{x}), \ldots, \phi_{M-1}(\mathbf{x}))$. By convention, $\boxed{\phi_0 \equiv 1.}$
- We consider the linear model

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

- $y(\mathbf{x}, \mathbf{w})$ is linear in the parameters $\mathbf{w}$, but can be non-linear in the input state $\mathbf{x}$.

GENERALISED BASIS FUNCTIONS

In 1-d, Poly basis is
$d > N$

Basis functions can, and usually are, non-linear functions of the inputs. Examples are

$w_0 + w_1 x + \dots + w_d x^d$.

- Polynomials up to degree d. In 1 dimension, $1, x, x^2, \dots, x^d$
- Gaussian basis functions: $\phi_j = \exp\{-\frac{(x-\mu_j)^2}{2s^2}\}$, where μ_j is the location and s is the lengthscale of the Gaussian.
- Sigmoid functions $\phi_j = \sigma\left(\frac{x-\mu_j}{s}\right)$, with $\sigma(a) = \frac{1}{1+\exp(-a)}$

LOGIT FUNCTION

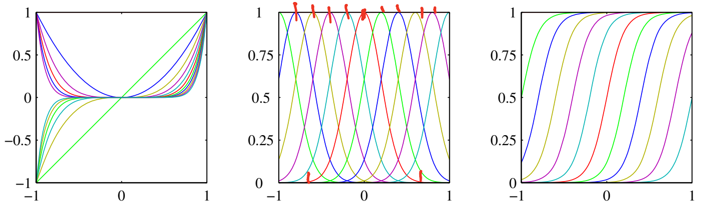
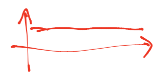


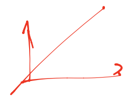
Figure 3.1 Examples of basis functions, showing polynomials on the left, Gaussians of the form (3.4) in the centre, and sigmoidal of the form (3.5) on the right.

$\phi_0(x) = 1$



$\phi_1(x) = x$

$\phi_2(x) = x^2$



$w_0\phi_0(x) + w_1\phi_1(x) =$

$\boxed{w_0 + w_1 x}$

$f(x)$ ← TARGET FUNCTION

$y = f(x) + \varepsilon$ ← ADDITIVE NOISE

ε RANDOM VARIABLE

$\varepsilon \sim \mathcal{N}(0, \sigma^2)$

MAXIMUM LIKELIHOOD REGRESSION

[β - PRECISION]

- Assume Gaussian noise: $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Hence $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y(\mathbf{x}, \mathbf{w}), \beta^{-1})$

MAXIMUM LIKELIHOOD REGRESSION

- Assume Gaussian noise: $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Hence $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y(\mathbf{x}, \mathbf{w}), \beta^{-1})$ i.i.d. INDIPENDENT IDENTICALLY DISTR.
- Given observations $\mathbf{X}, \mathbf{t}$: $(\mathbf{x_i}, t_i)_{i=1,\ldots,N}$, the likelihood is then

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(\cancel{y_i}\, t_i | \underbrace{\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_i})}, \underbrace{\beta^{-1}})$$

MAXIMUM LIKELIHOOD REGRESSION

- Assume Gaussian noise: $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Hence $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y(\mathbf{x}, \mathbf{w}), \beta^{-1})$
- Given observations $\mathbf{X}, \mathbf{t}$: $(\mathbf{x_i}, t_i)_{i=1,\dots,N}$, the likelihood is then

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(y_i|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_i}), \beta^{-1})$$

$\frac{1}{\sqrt{2\pi\beta^{-1}}}$

giving a log-likelihood of

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

$$= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta \boxed{E_D(\mathbf{w})} \qquad (3.11)$$

where the sum-of-squares error function is defined by

$$E_D(\mathbf{w}) = \boxed{\frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)\}^2.} \qquad (3.12)$$

MAXIMUM LIKELIHOOD REGRESSION

- Assume Gaussian noise: $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Hence $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y(\mathbf{x}, \mathbf{w}), \beta^{-1})$

MAXIMUM LIKELIHOOD REGRESSION

- Assume Gaussian noise: $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Hence $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y(\mathbf{x}, \mathbf{w}), \beta^{-1})$
- Given observations $\mathbf{X}, \mathbf{t}$: $(\mathbf{x_i}, t_i)_{i=1,\dots,N}$, the likelihood is then

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(y_i|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_i}), \beta^{-1})$$

MAXIMUM LIKELIHOOD REGRESSION

- Assume Gaussian noise: $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- Hence $p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(y(\mathbf{x}, \mathbf{w}), \beta^{-1})$
- Given observations $\mathbf{X}, \mathbf{t}$: $(\mathbf{x_i}, t_i)_{i=1,\dots,N}$, the likelihood is then

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{i=1}^{N} \mathcal{N}(y_i|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x_i}), \beta^{-1})$$

giving a log likelihood of

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w}) \end{aligned} \tag{3.11}$$

where the sum-of-squares error function is defined by

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n)\}^2. \tag{3.12}$$

MAXIMUM LIKELIHOOD REGRESSION

- Compute the gradient w.r.t. $\mathbf{w}$ of the log-likelihood, set it to zero and solve for $\mathbf{w}$.

MAXIMUM LIKELIHOOD REGRESSION

- Compute the gradient w.r.t. **w** of the log-likelihood, set it to zero and solve for **w**.

Fare i conti

$\phi^T \phi w = \phi^T t$

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.15}$$

which are known as the *normal equations* for the least squares problem. Here $\boldsymbol{\Phi}$ is an $N \times M$ matrix, called the *design matrix*, whose elements are given by $\Phi_{nj} = \phi_j(\mathbf{x}_n)$, so that

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}. \tag{3.16}$$

MAXIMUM LIKELIHOOD REGRESSION

- Compute the gradient w.r.t. **w** of the log-likelihood, set it to zero and solve for **w**.

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.15}$$

which are known as the *normal equations* for the least squares problem. Here $\boldsymbol{\Phi}$ is an $N \times M$ matrix, called the *design matrix*, whose elements are given by $\Phi_{nj} = \phi_j(\mathbf{x}_n)$, so that

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}. \tag{3.16}$$

- Looking for the ML solution of the precision β, we get

MAXIMUM LIKELIHOOD REGRESSION

- Compute the gradient w.r.t. **w** of the log-likelihood, set it to zero and solve for **w**.

$$\mathbf{w}_{\mathrm{ML}} = \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.15}$$

which are known as the *normal equations* for the least squares problem. Here $\boldsymbol{\Phi}$ is an $N \times M$ matrix, called the *design matrix*, whose elements are given by $\Phi_{nj} = \phi_j(\mathbf{x}_n)$, so that

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}. \tag{3.16}$$

- Looking for the ML solution of the precision β, we get

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \tag{3.21}$$

MAXIMUM LIKELIHOOD REGRESSION: BIAS TERM

$\phi_0(x) = 1$

- The parameter w_0 is known also as bias term.

At this point, we can gain some insight into the role of the bias parameter w_0. If we make the bias parameter explicit, then the error function (3.12) becomes

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(\mathbf{x}_n)\}^2. \tag{3.18}$$

Setting the derivative with respect to w_0 equal to zero, and solving for w_0, we obtain

$$\boxed{w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\overline{\phi_j}} \tag{3.19}$$

where we have defined

$$\boxed{\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n,} \qquad \boxed{\overline{\phi_j} = \frac{1}{N}\sum_{n=1}^{N}\phi_j(\mathbf{x}_n).} \tag{3.20}$$

Thus the bias w_0 compensates for the difference between the averages (over the training set) of the target values and the weighted sum of the averages of the basis function values.

MULTIPLE OUTPUTS

- What if we have a vector of d-outputs rather than a single one, i.e. what if observations $\mathbf{X}, \mathbf{T}$ are $(\mathbf{x_i}, \mathbf{t_i})_{l=1,\dots,N}$?
- If we use separate weights for each output dimension, $\mathbf{W} = (w_{ij})$, then the model is

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

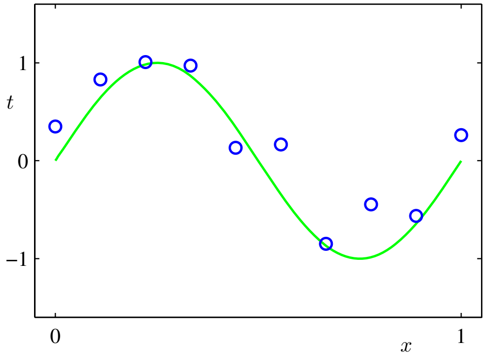
which is easily seen to factorise in the different outputs, so that we need to solve d independent ML problems, giving

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

- Generalise to the case in which some coefficients of $\mathbf{W}$ are shared among outputs (i.e., constrained to be equal).

AN EXAMPLE (BISHOP)

- As an example, consider data generated by the model $t = \boxed{sin(2\pi x)} + \epsilon$, from which we generate few observations:



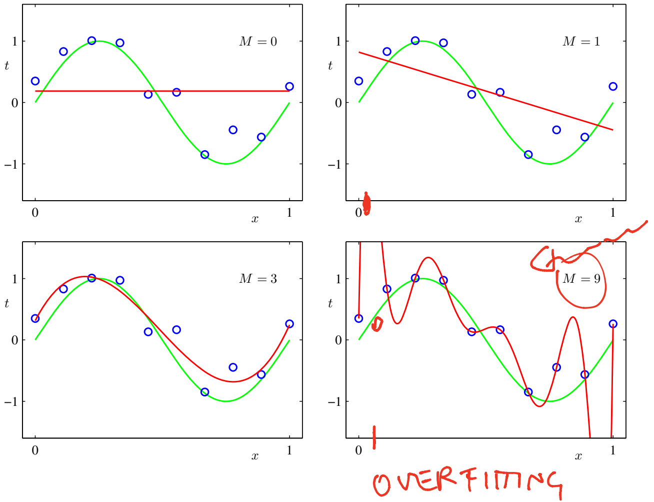
- We want to fit a polynomial model of degree M, where M is to be chosen:

$(1, x, \ldots, x^M)$

$$y(x, \mathbf{w}) = w_0 x^0 + w_1 x^1 + \ldots + w_M x^M$$

AN EXAMPLE (BISHOP)

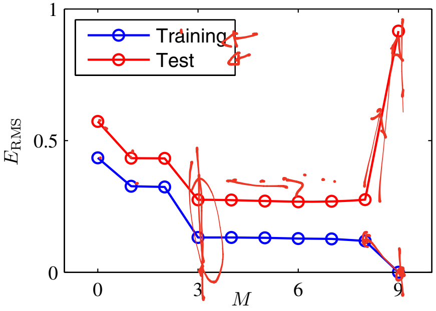
- Max likelihood solution for different M



AN EXAMPLE (BISHOP)

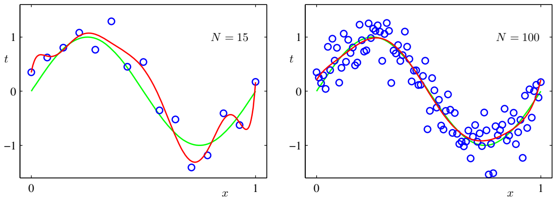
- For $M = 9$ we face the problem of overfitting: the model is too complex - ML explains noise rather than data.
- To validate a model, we need test data, different from the train data. Then we can compute the root mean square error on test (and train) data.

$$E_{RMS} = \sqrt{2E_D(\mathbf{w_{ML}})/N}$$



AN EXAMPLE (BISHOP)

- Overfitting depends also on how many observations: the more observations, the less overfitting:



- The fine-tuning of model to data reflects usually in large 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

REGULARISED MAXIMUM LIKELIHOOD

- One way to avoid overfitting is to penalise solutions with large values of coefficients **w**.
- This can be enforced by introducing a regularisation term on the error function to be minimised:

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

- $\lambda > 0$ is the regularisation coefficient, and governs how strong is the penalty.
- A common choice is

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

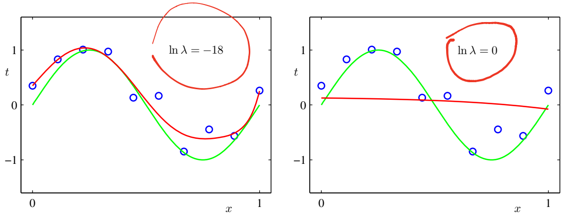
known as ridge regression, with solution

$$\mathbf{w}_{\mathbf{RR}} = (\lambda\mathbf{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$$

EXAMPLE: REGULARISED ML

λ - HYPERPARAMETER

- Let's consider the sine example, and fit the model of degree $M = 9$ by ridge regression, for different $\lambda' s$.



- If we compute the RMSE on a test set, we can see how the error changes with λ