# 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

## MULTIPLE OUTPUTS

- What if we have a vector of *d*-outputs rather than a single one, i.e. what if observations **X**, **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

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

which is easily seen to factorise in the different outputs, so that we need to solve $\boxed{d \text{ independent ML}}$ problems, giving

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

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

## 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

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

known as ridge regression, with solution

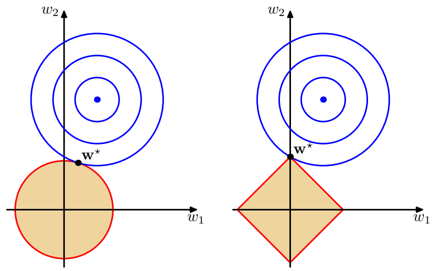
$$\mathbf{w}_{RR} = (\lambda \mathbf{I} + \mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{t}$$

## REGULARISED MAXIMUM LIKELIHOOD

- A more general form of the penalty term is

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

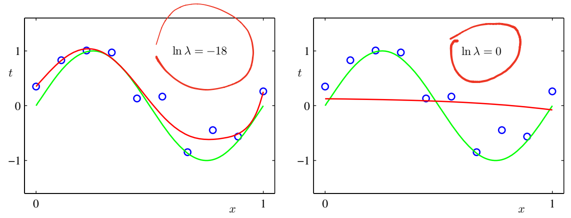
- $q = 2$ is the ridge regression, while $q = 1$ is the lasso regression.
- Lasso regression has the property that it produces sparse models as some coefficients tend to be set to zero. However, it has no analytic solution.



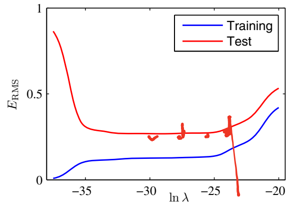
## EXAMPLE: REGULARISED ML

$\lambda$ - HYPERPARAMETER

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



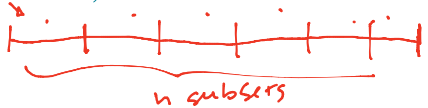
- If we compute the RMSE on a test set, we can see how the error changes with $\lambda$



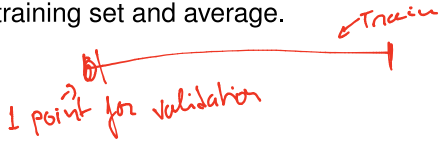
## TRAIN, VALIDATION, AND TEST DATA

- The regularisation coefficient $\lambda$ is a method parameter. But how can we set it?
- Ideally, we should divide our data in a train set, a test set, and a validation set, which can be used to set method's parameters.
- Often, we do not have all such data, hence we can resort to cross-validation

## TRAIN, VALIDATION, AND TEST DATA



- The regularisation coefficient $\lambda$ is a method parameter. But how can we set it?
- Ideally, we should divide our data in a train set, a test set, and a validation set, which can be used to set method's parameters.
- Often, we do not have all such data, hence we can resort to cross-validation
- *n*-fold cross-validation: split data set in *n* blocks, use in turn each block for validation and the rest for training, average the error on the *n* runs.
- leave one out cross-validation: validate in tuns on a single data point left out from the training set and average.



## EXPECTED LOSS

$p(x,t) = p(t|x)\,p(x)$

- If we have a model $p(\mathbf{x}, t)$ of input-output, one way to make a prediction (choose $t^*$ given $\mathbf{x}^*$) is by minimising an expected loss functional

$$\mathbb{E}[L] = \iint \{\underset{\uparrow}{y(\mathbf{x})} - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t. \qquad (1.87)$$

- The solution for the square loss functional is the conditional expectation

$$y(\mathbf{x}) = \frac{\int t p(\mathbf{x}, t)\,\mathrm{d}t}{p(\mathbf{x})} = \int t \boxed{p(t|\mathbf{x})}\,\mathrm{d}t = \mathbb{E}_t[t|\mathbf{x}] \qquad (1.89)$$

- This can be seen by summing and subtracting $\mathbb{E}[t|\mathbf{x}]$ inside the integral, getting

CONST (wrt $y$)

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}\,dt + \boxed{\int \{\mathbb{E}[t|\mathbf{x}] - t\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}\,dt} \qquad (1.90)$$

NOISE OF OBS.

## BIAS VARIANCE DECOMPOSITION

- If we do not have the full model, but only observe a dataset $\mathcal{D}$, then we can try to find the best approximant to the true conditional expectation, $y(\mathbf{x}, \mathcal{D})$.
- To test a method, we can try to generate many datasets and take the average $\mathbb{E}_{\mathcal{D}}$ w.r.t. the dataset. After some computations, calling $h(\mathbf{x})$ the true conditional expectation:

$$\boxed{\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}} \tag{3.41}$$

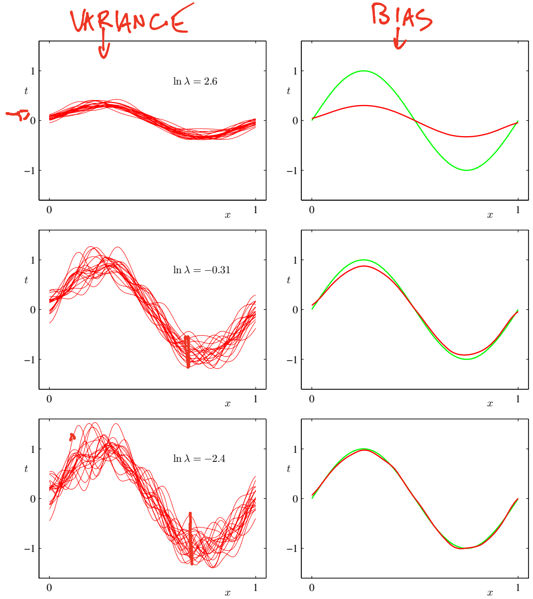
where

$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{3.42}$$

$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x} \tag{3.43}$$

$$\text{noise} = \int \{\underbrace{h(\mathbf{x})}_{E[t|x]} - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t \tag{3.44}$$

## Example: Bias variance decomposition

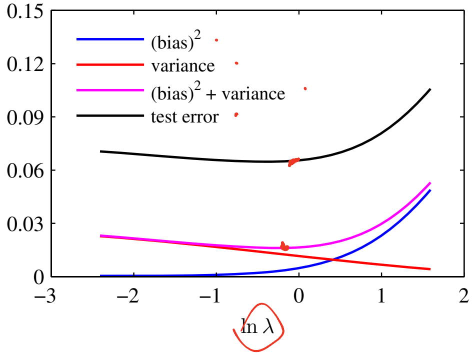


left: solutions for individual datasets

right: averages over datasets

## EXAMPLE: BIAS VARIANCE DECOMPOSITION

- For the sine example, we can compute bias and variance as a function of the regularisation coefficient. The trade off is evident.



## Outline

## THE BAYESIAN APPROACH

$\varepsilon \sim \mathcal{N}(0, \beta^{-1})$

$t = y(x) + \varepsilon$

- Regularisation works by biasing
- One way to bias estimators is to have prior beliefs and being Bayesian
- Let's assume the regression weights have a Gaussian prior $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \alpha \mathbf{I})$ and that the bias is zero
- The posterior is given by Bayes theorem: $\mathcal{N}(y, \beta^{-1})$

$$p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \alpha, \beta)p(\mathbf{w}|\alpha)}{\boxed{p(\mathbf{t}|\mathbf{X}, \alpha, \beta)}}$$

$$p(t|\alpha, \beta) = \int p(t|w, \alpha, \beta) \cdot p(w|\alpha) \, dw$$

## THE POSTERIOR DISTRIBUTION

- Hence, the log posterior is

$$\log p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \boxed{-\frac{\beta}{2}\sum_{j=1}^{N}[t_j - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x_j})]^2 - \alpha\mathbf{w}^T\mathbf{w} + const}$$

- As it is a quadratic function in **w**, it is the log of a Gaussian:

$$\boxed{p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m_N}, \mathbf{S_N})}$$

  with mean and variance

$$\boxed{\mathbf{m_N} = \beta\mathbf{S_N}\boldsymbol{\Phi}^T\mathbf{t}}$$

$$\boxed{\mathbf{S_N}^{-1} = \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^T\boldsymbol{\Phi}}$$

As $P(W|X,t)$ is GAUSSIAN

$W_N = W_{MAP}$

LIKE REG. ML CON

$\lambda = \beta/\alpha$

- Alternatively: use the formula for the product of two gaussians.

## THE POSTERIOR DISTRIBUTION

- In general, we can take a general Gaussian prior

$$p(\mathbf{w}|\mathbf{m_0}, \mathbf{S_0}) = \mathcal{N}(\mathbf{w}|\mathbf{m_0}, \mathbf{S_0})$$

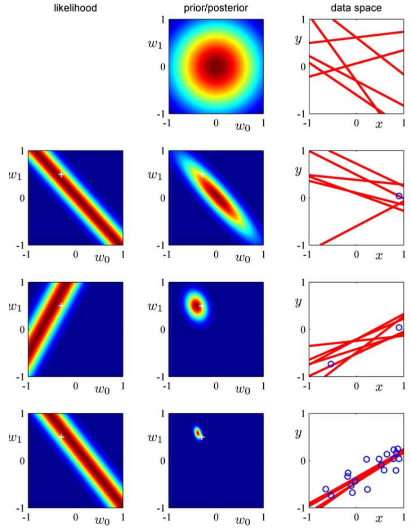
- This will result in a Gaussian posterior $p(\mathbf{w}|\mathbf{X}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(\mathbf{w}|\mathbf{m_N}, \mathbf{S_N})$ with

$$\mathbf{m_N} = \mathbf{S_N}[\mathbf{S_0}^{-1}\mathbf{m_0} + \beta\mathbf{\Phi}^T\mathbf{t}]$$

$$\mathbf{S_N}^{-1} = \mathbf{S_0}^{-1} + \beta\mathbf{\Phi}^T\mathbf{\Phi}$$

## POSTERIOR UPDATE

$y = w_0 + w_1 x$



$p(w)$

$p(w|t^{(1)})$

$p(w|t^{(2)})$

$p(w|t^{(N)})$

## THE PREDICTIVE DISTRIBUTION

- Given the posterior, one can find the MAP estimate. However, in a fully Bayesian treatment, one makes predictions by integrating out the parameters via their posterior distribution.

predictive distribution!

$$\boxed{p(t|\mathbf{t},\alpha,\beta) = \int p(t|\mathbf{t},\mathbf{w},\alpha,\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)d\mathbf{w}}$$

- The predictive distribution is still a Gaussian

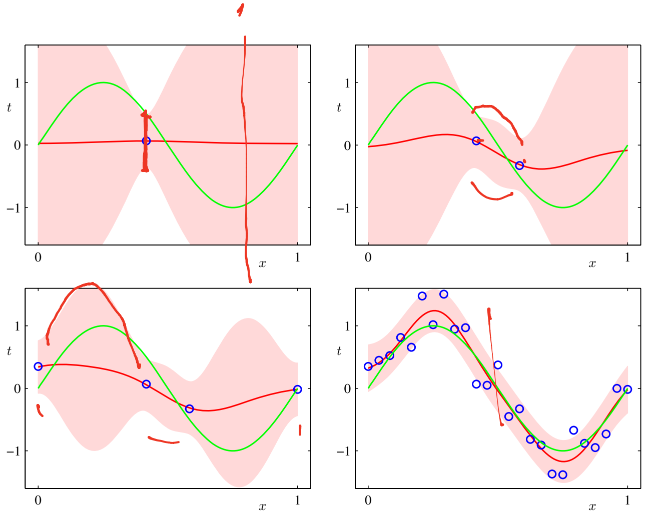
ND
$$p(t|\mathbf{t},\alpha,\beta) = \mathcal{N}(t|\underbrace{\mathbf{m_N}^T\boldsymbol{\phi}(\mathbf{x})}_{},\underbrace{\sigma_N^2(\mathbf{x})}_{})$$

with mean $\mathbf{m_N}^T\boldsymbol{\phi}(\mathbf{x})$ and variance

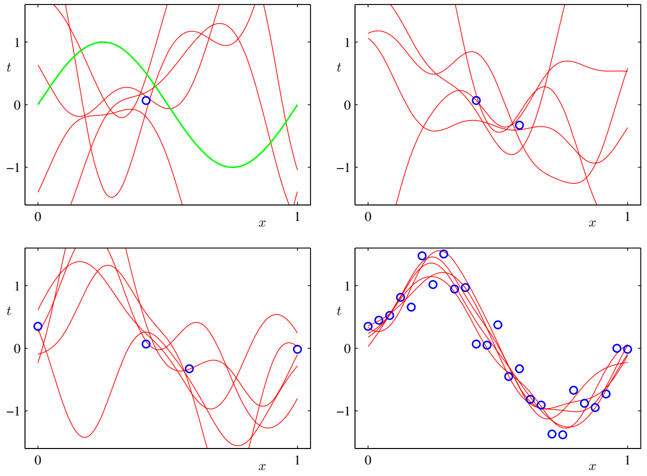
$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boxed{\boldsymbol{\phi}(\mathbf{x})^T\mathbf{S_N}\boldsymbol{\phi}(\mathbf{x})}$$

- It can be shown that $\sigma_{N+1}^2(\mathbf{x}) \leq \sigma_N^2(\mathbf{x})$ and $\sigma_N^2(\mathbf{x}) \to \cancel{0}$ $\frac{1}{\beta}$

## EXAMPLE

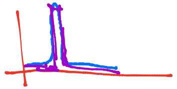


## EXAMPLE



## MARGINAL LIKELIHOOD

- The marginal likelihood $p(\mathbf{t}|\alpha, \beta)$, appearing at the denominator in Bayes theorem, can be used to identify good $\alpha$ and $\beta$, known as hyperparameters.
- Intuitively, we can place a prior distribution over $\alpha$ and $\beta$, compute their posterior, and use this in a fully Bayesian treatment of the regression:



$$\rightsquigarrow \boxed{p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta) p(\alpha, \beta)}$$

- If we assume the posterior is peaked around the mode, then we can take the MAP as an approximation of the full posterior for $\alpha$ and $\beta$. If the flat is prior, this will boil down to the ML solution.

$\rightsquigarrow$ Estimate $\alpha, \beta$ maximising $p(t|\alpha, \beta)$ (marg. likelihood)

## MARGINAL LIKELIHOOD

- Hence we need to optimise the marginal likelihood, which can be computed as:

$$\log p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\log\underset{\uparrow}{\alpha} + \frac{N}{2}\log\underset{\uparrow}{\beta} - E(\bigcirc\!\!\!\!\mathbf{m_N}) - \frac{1}{2}\log|\mathbf{S_N}^{-1}| - \frac{N}{2}\log 2\pi$$

with

$$E(\mathbf{m_N}) = \frac{\beta}{2}\|\mathbf{t}-\mathbf{\Phi m_N}\|^2 + \frac{\alpha}{2}\mathbf{m_N}^T\mathbf{m_N}$$

- This optimisation problem can be solved with any optimisation routine, or with specialised methods, see Bishop.