Regression

Dr. Francis Colas

21.10.2011

Introduction

Regression:

- find the relationship between variables,
- find the best curve to fit data,
- predict the value for a new data point;

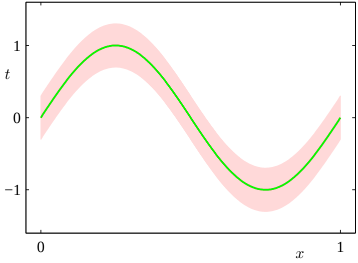
Characteristics:

- supervised learning,
- evaluate using goodness of fit,
- problem of overfitting.

Polynomial curve fitting

Example:

- synthetic data,
- generated based on a sine function (green),
- Gaussian noise (red).

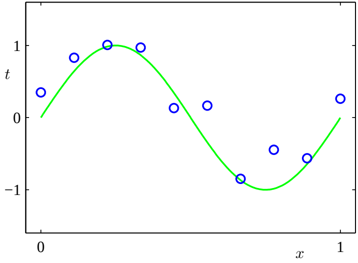


Courtesy C. Bishop, PRML.

Polynomial curve fitting

Example:

- synthetic data,
- generated based on a sine function (green),
- Gaussian noise (red).



Courtesy C. Bishop, PRML.

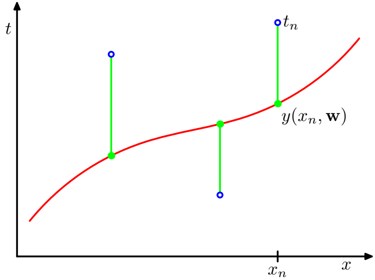
Problem statement

Aim:

- from points (x_n, t_n),
- fit model: $y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j . x^j$,

$\Leftrightarrow$ find weight vector: $\mathbf{w}$,

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



Solution

Mathematically:

- Looking for: $\arg\min_{\mathbf{w}} E(\mathbf{w})$

$\Rightarrow$ solve: $\frac{dE}{d\mathbf{w}} = \mathbf{0}$

$\Rightarrow$ solve:

$$\begin{pmatrix} \sum_{n=1}^{N} x_n^0 (\sum_{j=0}^{M} w_j x_n^j - t_n) \\ \vdots \\ \sum_{n=1}^{N} x_n^k (\sum_{j=0}^{M} w_j x_n^j - t_n) \\ \vdots \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \end{pmatrix}$$

Autonomous Systems Lab

Solution

Let:

$$\mathbf{\Phi} = \begin{pmatrix} x_1^0 & \cdots & x_1^j & \cdots & x_1^M \\ \vdots & \ddots & \vdots & & \vdots \\ x_n^0 & \cdots & x_n^j & \cdots & x_n^M \\ \vdots & & \vdots & \ddots & \vdots \\ x_N^0 & \cdots & x_N^j & \cdots & x_N^M \end{pmatrix}$$

N rows by $M+1$ columns

$$\mathbf{\Phi} = \left(\phi_{n,j} = x_n^j\right)$$

Then:

$$\frac{dE}{d\mathbf{w}} = \mathbf{w}^{\mathrm{T}}\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi} - \mathbf{t}^{\mathrm{T}}\mathbf{\Phi} = 0$$

Finally:

$$\mathbf{w} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}$$

Where $(\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}} = \mathbf{\Phi}^{+}$ is the pseudoinverse of $\mathbf{\Phi}$

ETH Zürich

Autonomous Systems Lab

Solution

Let:

$$\boldsymbol{\Phi} = \left(\phi_{n,j} = x_n^j\right)$$

Then:

$$\frac{dE}{d\mathbf{w}} = \mathbf{w}^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} - \mathbf{t}^{\mathrm{T}}\boldsymbol{\Phi} = 0$$

Finally:

$$\mathbf{w} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}$$

Where $(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}} = \boldsymbol{\Phi}^{+}$ is the pseudoinverse of $\boldsymbol{\Phi}$

ETH Zürich

Polynomial curve fitting

Solve:

- model: $y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j . x^j$,
- minimize SSE: $E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, \mathbf{w}) - t_n)^2$,
- let: $\mathbf{\Phi} = (x_n^j)_{n,j}$,
- solution:

$$\mathbf{w} = \mathbf{\Phi}^{+} \mathbf{t}$$

Summary:

- model: linear combination of monomials,
- equation: linear system,
- solution: linear algebra.

Autonomous Systems Lab

Linear regression

Definition:

- model: linear combination of basis functions,
- minimize SSE,
- equation: linear system,
- solution: linear algebra.

Basis functions:

- set of functions $\boldsymbol{\phi} = (\phi_j)_j$,
- define the model: $y(x, \mathbf{w}) = \sum_{j=0}^{M} w_j \phi_j(x) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(x)$.

ETH Zürich

Solution

Build the $N \times M + 1$ design matrix:

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0(x_1) & \cdots & \phi_j(x_1) & \cdots & \phi_M(x_1) \\ \vdots & \ddots & \vdots & & \vdots \\ \phi_0(x_n) & \cdots & \phi_j(x_n) & \cdots & \phi_M(x_n) \\ \vdots & & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \cdots & \phi_j(x_N) & \cdots & \phi_M(x_N) \end{pmatrix}$$

Compute the pseudo-inverse and the weights:

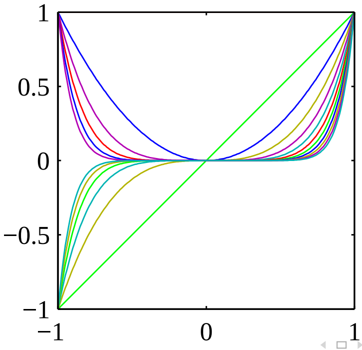
$$\mathbf{w} = \mathbf{\Phi}^{+}\mathbf{t}$$

Autonomous Systems Lab

Basis functions

Different choices:

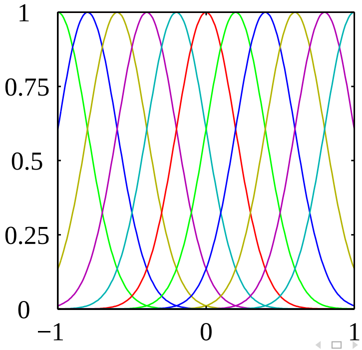
- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...



Basis functions

Different choices:

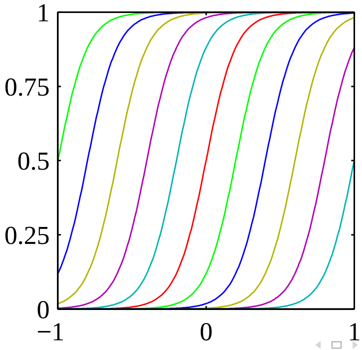
- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...



Basis functions

Different choices:

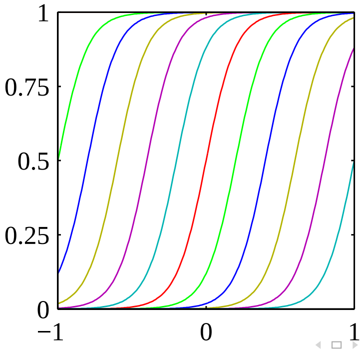
- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...



Basis functions

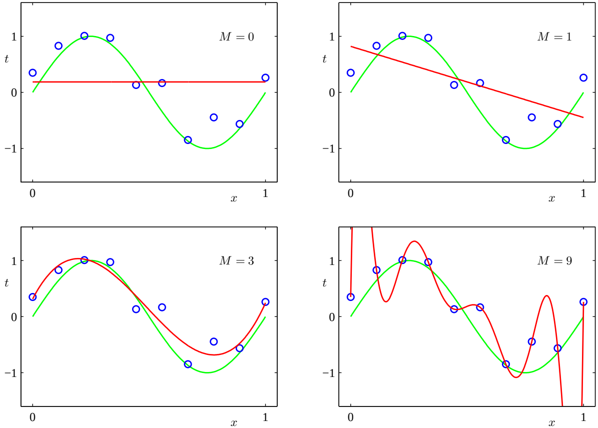
Different choices:

- monomials (polynomial curve fitting),
- Gaussian functions,
- sigmoidal functions
- ...



Overfitting

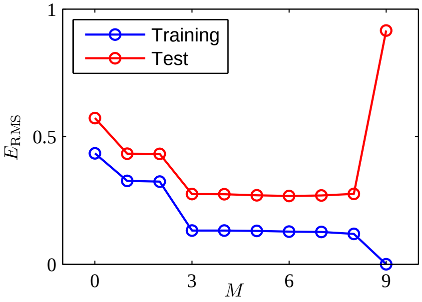
Result depends on the choice and number of basis functions.
With monomials:



Validation

Comparing models:

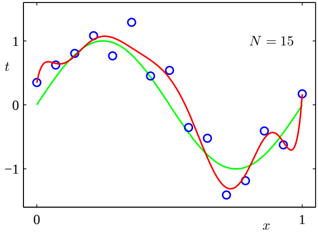
- training set: involved in fitting each model,
- test set: to compare the models:

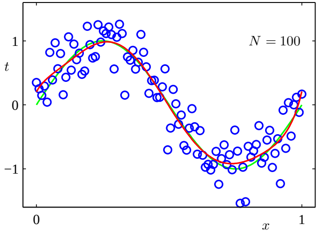


But you need a large enough dataset.

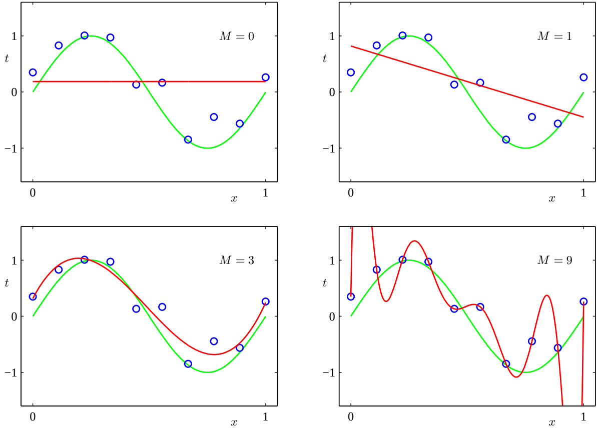
Overfitting

Trade-off between complexity of the model and the data





Regularization



Regularization

Weight vector:

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
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			17.36	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

Overfitting $\rightarrow$ large values!

Regularization

Regularization:

- penalize high values,
- change the error function:

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

- still linear algebra:

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

Choice of λ:

- too large: simple model,
- too small: overfitting,
- trade-off between data and model.

Summary

Linear regression:

- linear combination of basis functions,
- find weight minimizing error,
- linear system of equation,
- overfitting;

Regularization:

- penalize big weights,
- change the error,
- still linear system,
- choice of regularization parameter.

Reasoning

Regression:

- estimate parameters,
- based on data,
- given a model;

Last week:

- estimate state,
- based on data,
- given a model.

Probabilistic reasoning applied:

- regression,
- machine learning.

Probabilistic formulation

We have:

- x values,
- predictions y for x given $\mathbf{w}$,
- t values that should be close to y,

We want:

- $\mathbf{w}$.

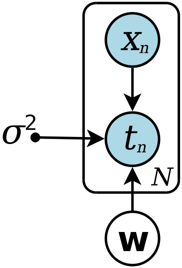
More formally:

- $t_n = y(x_n, \mathbf{w}) + \epsilon$
- $P(t_n | x_n, \mathbf{w}) = \mathcal{N}(y(x_n, \mathbf{w}), \sigma^2)$.

Probabilistic formulation

Assumptions:

- points x_n are all independent,
- values t_n are independent from everything given x_n and $\mathbf{w}$,
- Gaussian noise with identical variance.



$$P(\mathbf{w}, x_1, t_1, \ldots, x_N, t_N) = P(\mathbf{w}) \prod_{n=1}^{N} P(x_n) P(t_n | x_n, \mathbf{w})$$

Inference

Maximum likelihood estimation (MLE):

$$\arg\max_{\mathbf{w}} P(\mathbf{t}|\mathbf{x}, \mathbf{w})$$

Product $\rightarrow$ taking log:

$$\begin{aligned}
&\ln P(\mathbf{t}|\mathbf{x}, \mathbf{w}) \\
&= \sum_{n=1}^{N} \ln P(t_n|x_n, \mathbf{w}) \\
&= \sum_{n=1}^{N} \left(-\frac{1}{2}\left(\ln\left(2\pi\sigma^2\right)\right) - \frac{1}{2\sigma^2}\left(t_n - y\left(x_n, \mathbf{w}\right)\right)^2\right) \\
&= -\frac{N}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{\sigma^2}\frac{1}{2}\sum_{n=1}^{N}\left(y\left(x_n, \mathbf{w}\right) - t_n\right)^2 \\
&= \beta - \frac{1}{\sigma^2}E(\mathbf{w})
\end{aligned}$$

Maximum likelihood $\Leftrightarrow$ regression!

Inference

Maximum likelihood estimation (MLE):

$$\arg\max_{\mathbf{w}} P(\mathbf{t}|\mathbf{x}, \mathbf{w})$$

Product $\rightarrow$ taking log:

$$\begin{aligned}
&\ln P(\mathbf{t}|\mathbf{x}, \mathbf{w}) \\
&= \sum_{n=1}^{N} \ln P(t_n|x_n, \mathbf{w}) \\
&= \sum_{n=1}^{N} \left(-\frac{1}{2}\left(\ln\left(2\pi\sigma^2\right)\right) - \frac{1}{2\sigma^2}\left(t_n - y\left(x_n, \mathbf{w}\right)\right)^2\right) \\
&= -\frac{N}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{\sigma^2}\frac{1}{2}\sum_{n=1}^{N}\left(y\left(x_n, \mathbf{w}\right) - t_n\right)^2 \\
&= \beta - \frac{1}{\sigma^2}E(\mathbf{w})
\end{aligned}$$

Maximum likelihood $\Leftrightarrow$ regression!

Inference

Maximum likelihood estimation (MLE):

$$\arg\max_{\mathbf{w}} P(\mathbf{t}|\mathbf{x},\mathbf{w})$$

Product $\rightarrow$ taking log:

$$\begin{aligned}
&\ln P(\mathbf{t}|\mathbf{x},\mathbf{w})\\
&= \textstyle\sum_{n=1}^{N} \ln P(t_n|x_n,\mathbf{w})\\
&= \textstyle\sum_{n=1}^{N} \left(-\frac{1}{2}\left(\ln\left(2\pi\sigma^2\right)\right) - \frac{1}{2\sigma^2}\left(t_n - y\left(x_n,\mathbf{w}\right)\right)^2\right)\\
&= -\frac{N}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{\sigma^2}\frac{1}{2}\textstyle\sum_{n=1}^{N}\left(y\left(x_n,\mathbf{w}\right) - t_n\right)^2\\
&= \beta - \frac{1}{\sigma^2}E(\mathbf{w})
\end{aligned}$$

Maximum likelihood $\Leftrightarrow$ regression!

Inference

Maximum a posteriori (MAP):

$$\arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$

Assuming Gaussian prior over $\mathbf{w}$:

$$\begin{aligned}
&\ln P(\mathbf{w}|\mathbf{x}, \mathbf{t}) \\
&= \alpha + \ln P((\mathbf{t}|\mathbf{x}, \mathbf{w}) + \ln P(\mathbf{w}) \\
&= \alpha + \beta - \frac{1}{\sigma^2} E(\mathbf{w}) + \gamma - \frac{1}{2\sigma_{\mathbf{w}}^2} \mathbf{w}^{\mathrm{T}} \mathbf{w} \\
&= \delta - \frac{1}{\sigma^2} \left(E(\mathbf{w}) + \frac{\sigma^2}{2\sigma_{\mathbf{w}}^2} \|\mathbf{w}\|^2 \right)
\end{aligned}$$

Maximum a posteriori $\Leftrightarrow$ regularization!

Inference

Maximum a posteriori (MAP):

$$\arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$

Assuming Gaussian prior over **w**:

$$\begin{aligned}
&\ln P(\mathbf{w}|\mathbf{x}, \mathbf{t}) \\
&= \alpha + \ln P((\mathbf{t}|\mathbf{x}, \mathbf{w}) + \ln P(\mathbf{w}) \\
&= \alpha + \beta - \tfrac{1}{\sigma^2} E(\mathbf{w}) + \gamma - \tfrac{1}{2\sigma_{\mathbf{w}}^2} \mathbf{w}^{\mathrm{T}}\mathbf{w} \\
&= \delta - \tfrac{1}{\sigma^2} \left(E(\mathbf{w}) + \tfrac{\sigma^2}{2\sigma_{\mathbf{w}}^2} \|\mathbf{w}\|^2 \right)
\end{aligned}$$

Maximum a posteriori $\Leftrightarrow$ regularization!

Inference

Maximum a posteriori (MAP):

$$\arg\max_{\mathbf{w}} P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$

Assuming Gaussian prior over $\mathbf{w}$:

$$\begin{aligned}
&\ln P(\mathbf{w}|\mathbf{x}, \mathbf{t}) \\
&= \alpha + \ln P((\mathbf{t}|\mathbf{x}, \mathbf{w}) + \ln P(\mathbf{w}) \\
&= \alpha + \beta - \tfrac{1}{\sigma^2} E(\mathbf{w}) + \gamma - \tfrac{1}{2\sigma_{\mathbf{w}}^2} \mathbf{w}^{\mathrm{T}} \mathbf{w} \\
&= \delta - \tfrac{1}{\sigma^2} \left(E(\mathbf{w}) + \tfrac{\sigma^2}{2\sigma_{\mathbf{w}}^2} \|\mathbf{w}\|^2 \right)
\end{aligned}$$

Maximum a posteriori $\Leftrightarrow$ regularization!

Example

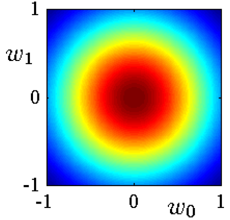
Fitting a line incrementally:

$P(t_n|x_n, \mathbf{w})$ $\qquad$ $P(\mathbf{w}|...)$ $\quad$ data space

Example

Fitting a line incrementally:

$P(t_n|x_n, \mathbf{w})$ $P(\mathbf{w}|...)$ data space

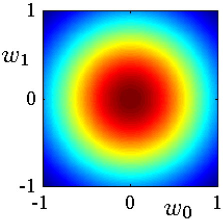


Example

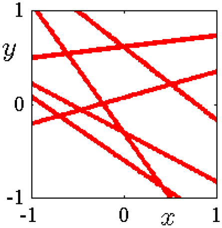
Fitting a line incrementally:

$P(t_n|x_n, \mathbf{w})$

$P(\mathbf{w}|...)$



data space

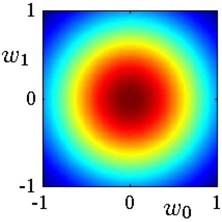


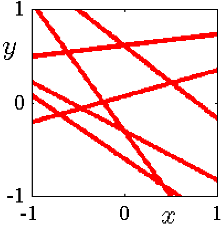
Autonomous Systems Lab

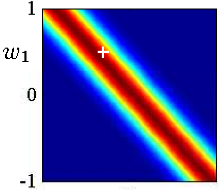
Example

Fitting a line incrementally:

$P(t_n|x_n, \mathbf{w})$ $P(\mathbf{w}|...)$ data space

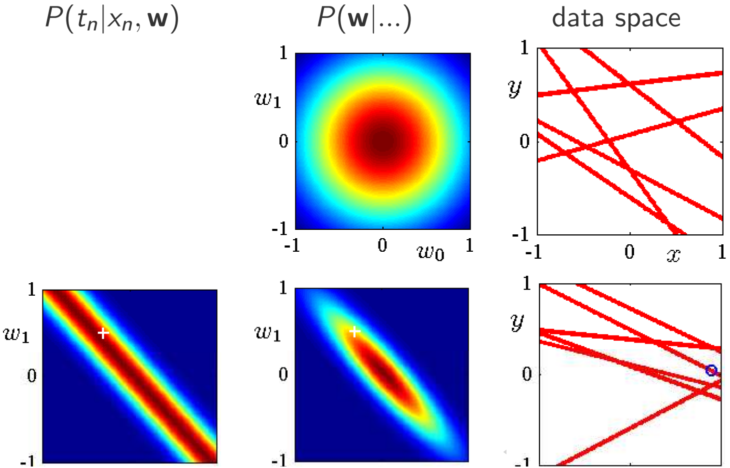






Example

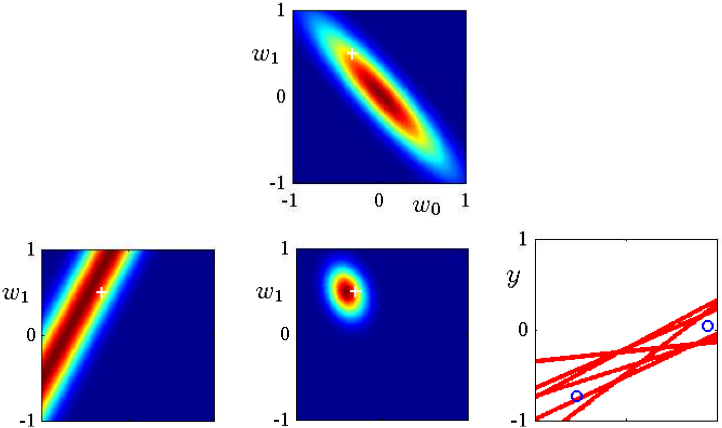
Fitting a line incrementally:



Example

Fitting a line incrementally:

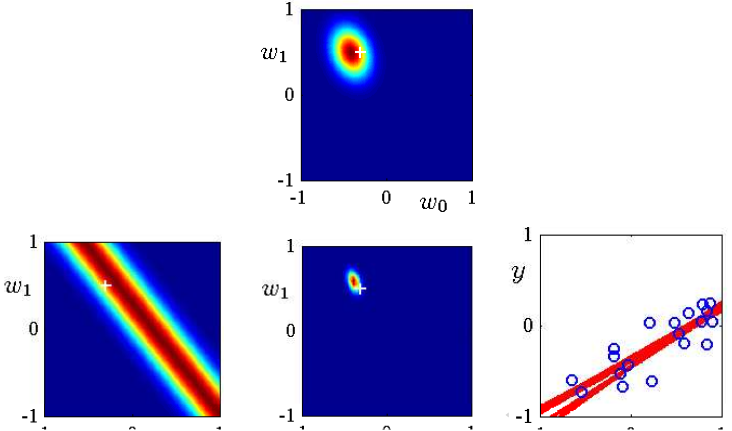
$P(t_n|x_n, \mathbf{w})$ $P(\mathbf{w}|\ldots)$ data space



Example

Fitting a line incrementally:

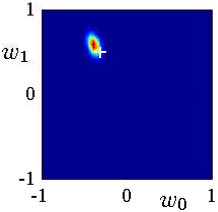
$P(t_n|x_n, \mathbf{w})$ $P(\mathbf{w}|\ldots)$ data space

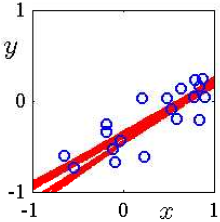


Example

Fitting a line incrementally:

$P(t_n|x_n, \mathbf{w})$ $P(\mathbf{w}|\ldots)$ data space





Prediction

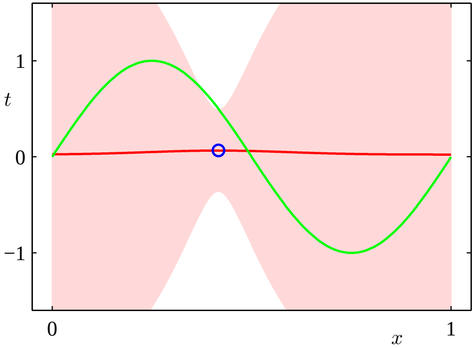
Predict the value for a new point:

$$P(\tilde{t}|\tilde{x}, \mathbf{x}, \mathbf{t}) = \int_{\mathbf{w}} P(\tilde{t}|\tilde{x}, \mathbf{w}) P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$

Prediction

Predict the value for a new point:

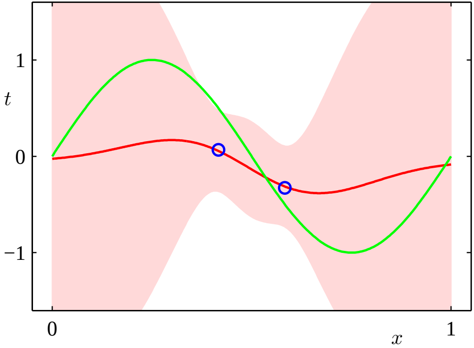
$$P(\tilde{t}|\tilde{x}, \mathbf{x}, \mathbf{t}) = \int_{\mathbf{w}} P(\tilde{t}|\tilde{x}, \mathbf{w}) P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$



Prediction

Predict the value for a new point:

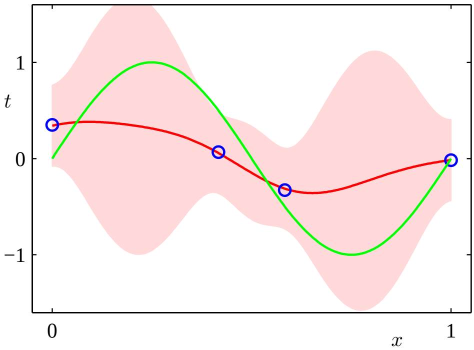
$$P(\tilde{t}|\tilde{x}, \mathbf{x}, \mathbf{t}) = \int_{\mathbf{w}} P(\tilde{t}|\tilde{x}, \mathbf{w}) P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$



Prediction

Predict the value for a new point:

$$P(\tilde{t}|\tilde{x}, \mathbf{x}, \mathbf{t}) = \int_{\mathbf{w}} P(\tilde{t}|\tilde{x}, \mathbf{w}) P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$

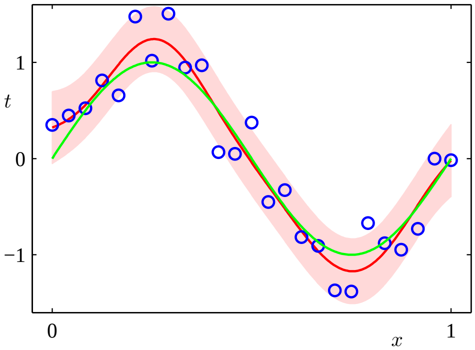


Autonomous Systems Lab

Prediction

Predict the value for a new point:

$$P(\tilde{t}|\tilde{x}, \mathbf{x}, \mathbf{t}) = \int_{\mathbf{w}} P(\tilde{t}|\tilde{x}, \mathbf{w}) P(\mathbf{w}|\mathbf{x}, \mathbf{t})$$



ETH Zürich

Summary

Linear regression:

- linear combination of basis functions,
- linear system of equation,
- overfitting;

Regularization:

- penalize big weights,
- still linear system,
- choice of regularization parameter;

Probabilistic formulation:

- Gaussian noise,
- MLE is regression,
- Gaussian prior,
- MAP is regularization.