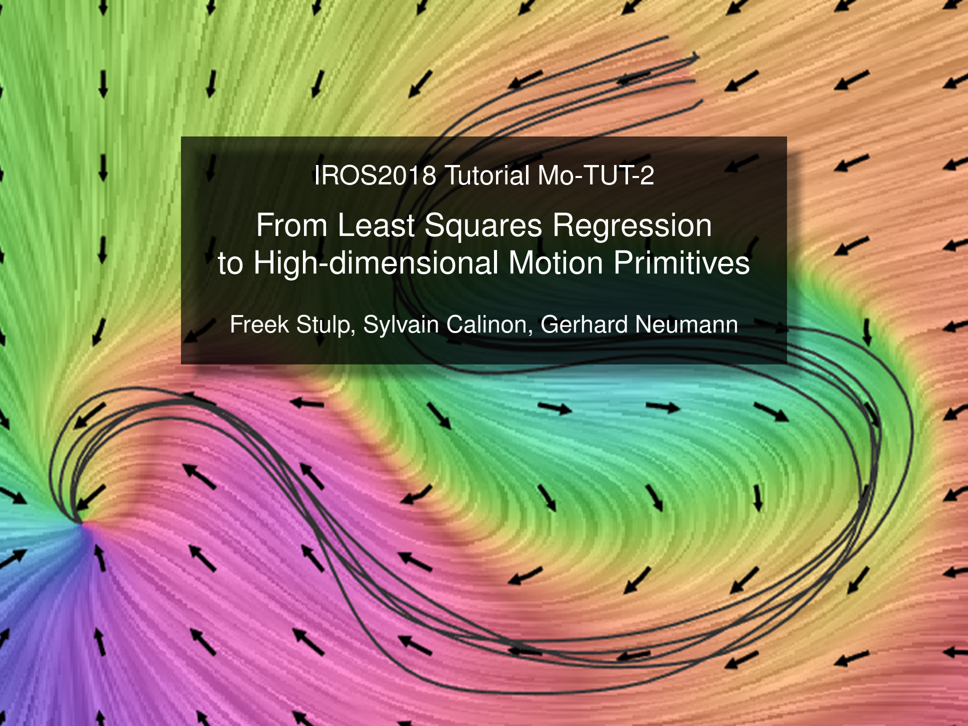


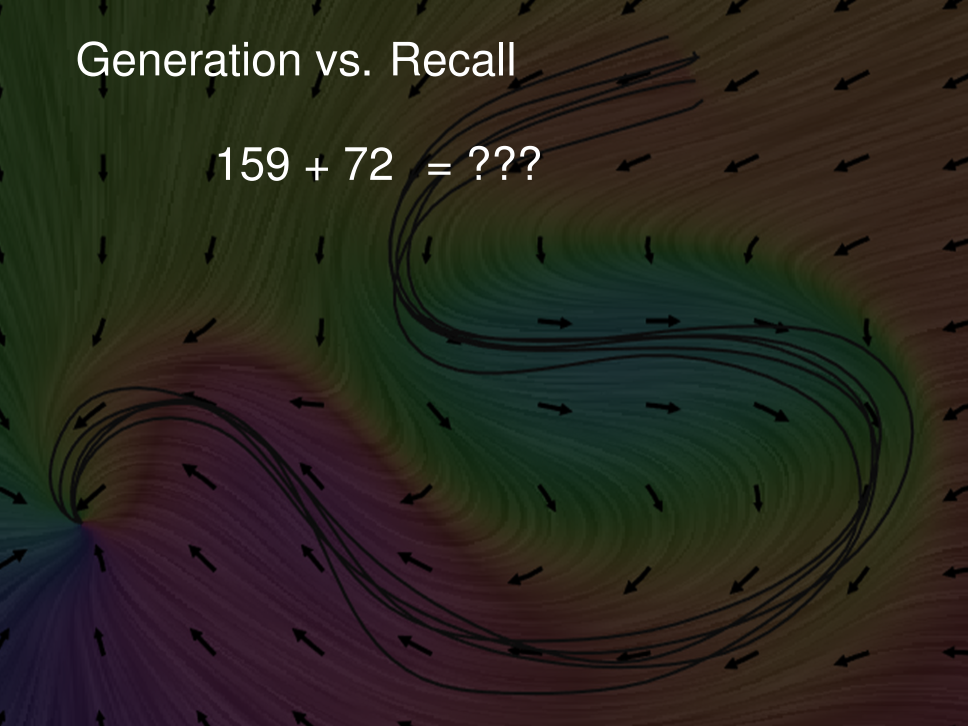
IROS2018 Tutorial Mo-TUT-2

From Least Squares Regression to High-dimensional Motion Primitives

Freek Stulp, Sylvain Calinon, Gerhard Neumann

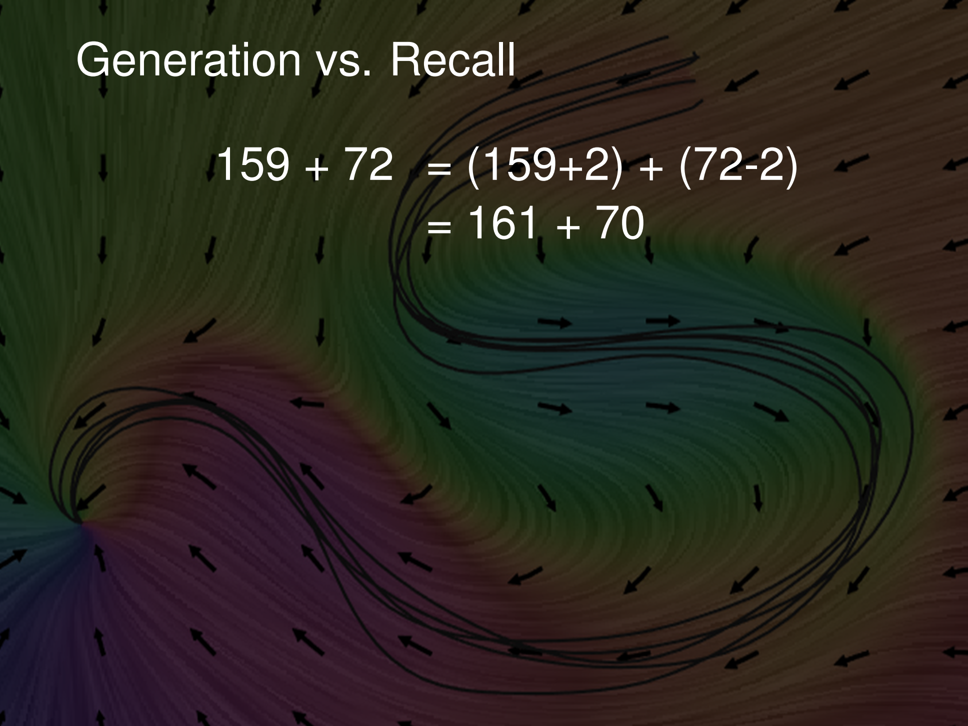
Generation vs. Recall

159 + 72 = ???

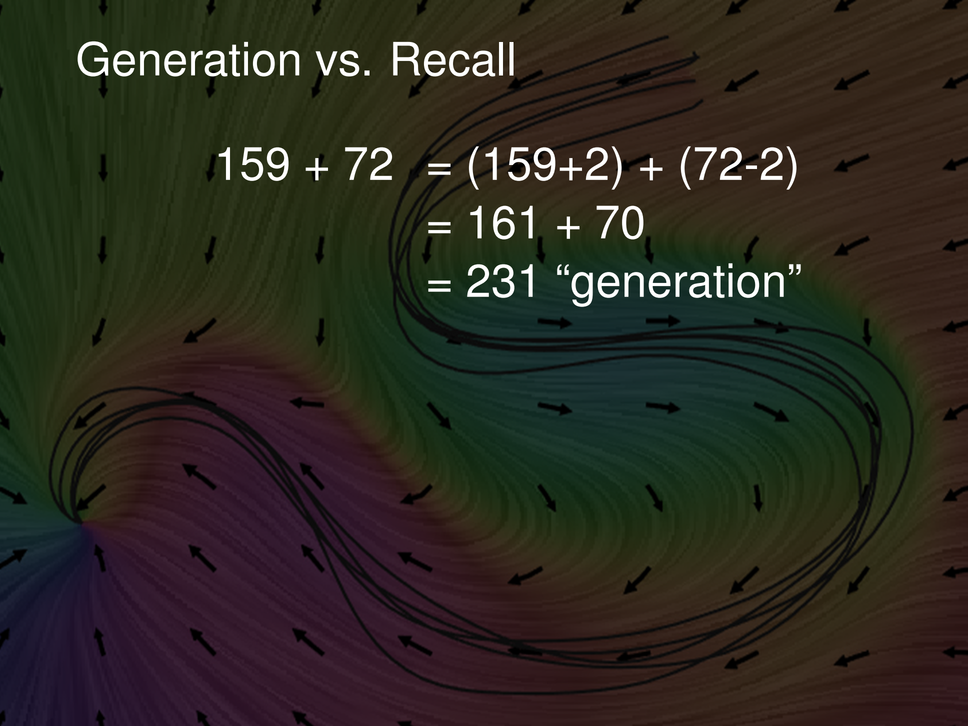


Generation vs. Recall

$$159 + 72 = (159+2) + (72-2)$$
$$= 161 + 70$$



Generation vs. Recall



$$
\begin{aligned}
159 + 72 &= (159+2) + (72-2) \\
&= 161 + 70 \\
&= 231 \text{ “generation”}
\end{aligned}
$$

Generation vs. Recall

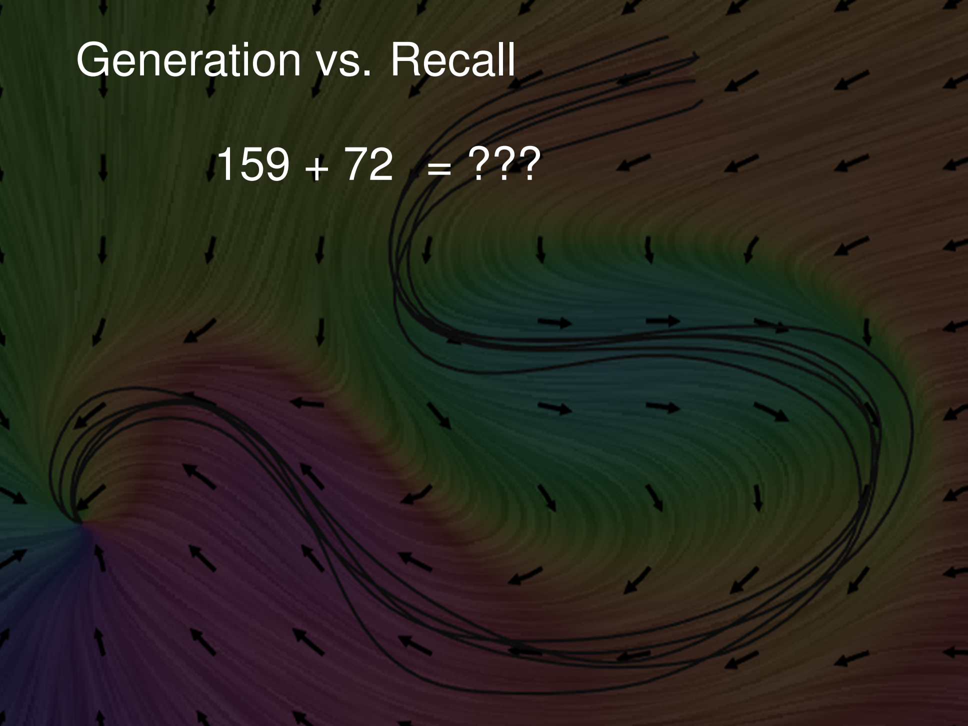
$$159 + 72 = (9+2) + (150+70)$$
$$= 11 + 220$$

Generation vs. Recall

159 + 72 = (9+2) + (150+70)
= 11 + 220
= 231 "generation"

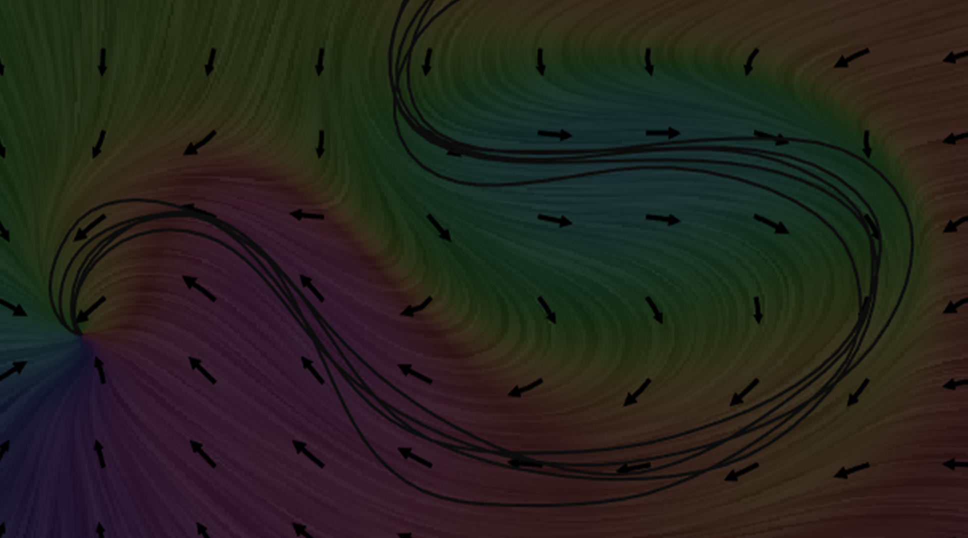
Generation vs. Recall

159 + 72 = ???



Generation vs. Recall

159 + 72 = 231 “recall”



Generation vs. Recall

159 + 72 = 231 “recall”

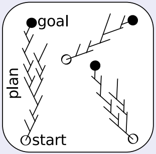
Distinction between these two strategies important in cognitive science, artificial intelligence, robotics, teaching

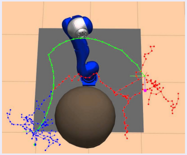
Generation vs. Recall

159 + 72 = 231 “recall”

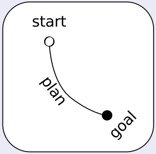
Distinction between these two strategies important in cognitive science, artificial intelligence, robotics, teaching

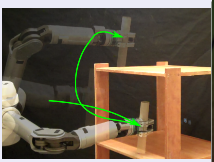
Motion Generation



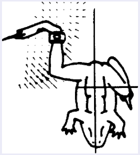


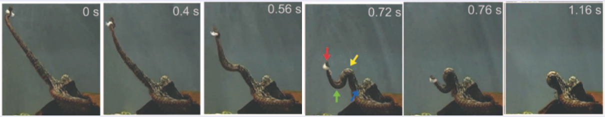
Motion Recall





Motion primitives in nature

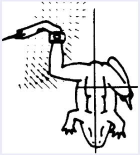


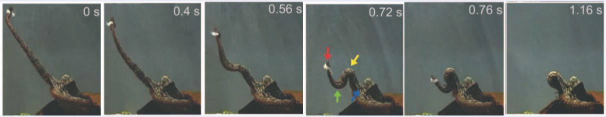


Giszter, S.; Mussa-Ivaldi, F. & Bizzi, E. Convergent force fields organized in the frog's spinal cord Journal of Neuroscience, 1993

Flash, T. & Hochner, B. Motor Primitives in Vertebrates and Invertebrates Current Opinion in Neurobiology, 2005

Motion primitives in nature





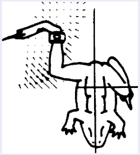
Giszter, S.; Mussa-Ivaldi, F. & Bizzi, E. Convergent force fields organized in the frog's spinal cord Journal of Neuroscience, 1993

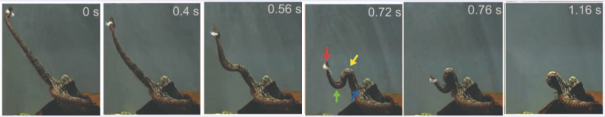
Flash, T. & Hochner, B. Motor Primitives in Vertebrates and Invertebrates Current Opinion in Neurobiology, 2005

Motion primitives for robots?

- Couple degrees of freedom to deal with high-dimensional systems
- Sequencing and superpositioning of MPs for more complex task
- Low-dimensional parameterization of MP enables learning
- MPs can be bootstrapped with demonstrations
- Direct mappings between task parameters and MP parameters

Motion primitives in nature

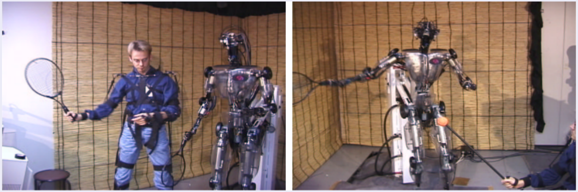




Giszter, S.; Mussa-Ivaldi, F. & Bizzi, E. Convergent force fields organized in the frog's spinal cord Journal of Neuroscience, 1993

Flash, T. & Hochner, B. Motor Primitives in Vertebrates and Invertebrates Current Opinion in Neurobiology, 2005

Motion primitives for robots!



Ijspeert, A. J.; Nakanishi, J. & Schaal, S. Movement imitation with nonlinear dynamical systems in humanoid robots. ICRA, 2002

Schedule

9:00	–	9:15	Introduction	
9:15	–	10:45	Regression Tutorial	Freek Stulp
10:45	–	11:00	Motion Primitives 1	Sylvain Calinon
11:00	–	11:30	Coffee Break	
11:30	–	12:15	Motion Primitives 1 (cont.)	Sylvain Calinon
12:15	–	13:15	Motion Primitives 2	Gerhard Neumann
13:15	–	13:30	Wrap up	

Regression Tutorial

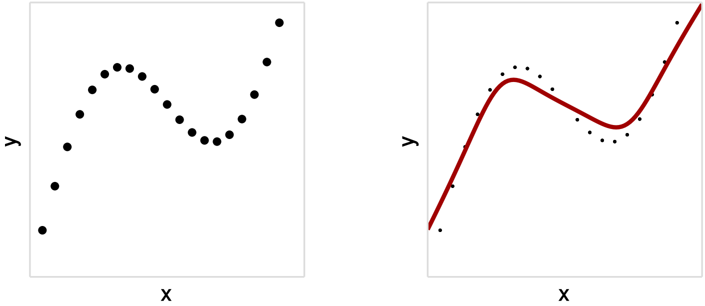
IROS'18 Tutorial

Freek Stulp
Institute of Robotics and Mechatronics, German Aerospace Center (DLR)
Autonomous Systems and Robotics, ENSTA-ParisTech

01.10.2018

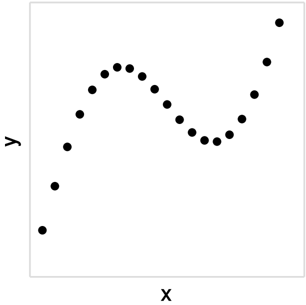
What Is Regression?

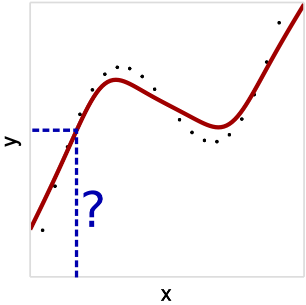
Estimating a relationship
between input variables
and continuous output variables
from data



What Is Regression?

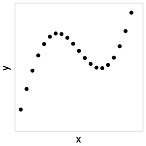
Estimating a relationship
between input variables
and continuous output variables
from data

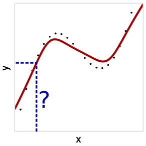




What Is Regression?

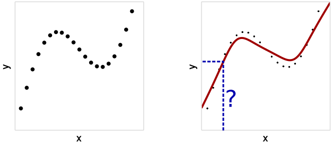
Estimating a relationship
between input variables
and continuous output variables
from data



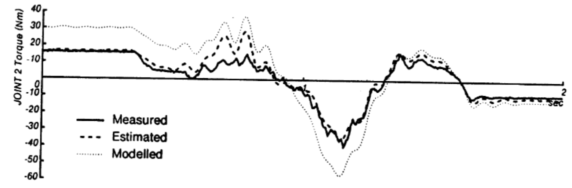


What Is Regression?

Estimating a relationship
between input variables
and continuous output variables
from data



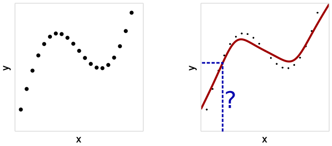
Application: Dynamic parameter estimation



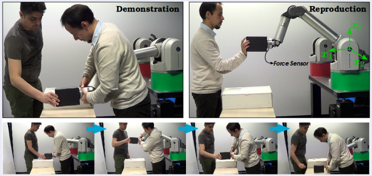
An, C.; Atkeson, C. and Hollerbach, J. (1985).
Estimation of inertial parameters of rigid body links of manipulators [404]
IEEE Conference on Decision and Control.

What Is Regression?

Estimating a relationship
between input variables
and continuous output variables
from data



Application: Programming by demonstration

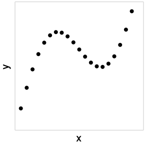


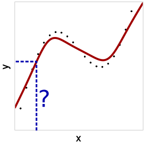
Rozo, L.; Calinon, S.; Caldwell, D. G.; Jimenez, P. and Torras, C. (2016).
Learning Physical Collaborative Robot Behaviors from Human Demonstrations
IEEE Trans. on Robotics.

Calinon, S.; Guenter, F. and Billard, A. (2007).
On Learning, Representing and Generalizing a Task in a Humanoid Robot [725]
IEEE Transactions on Systems, Man and Cybernetics.

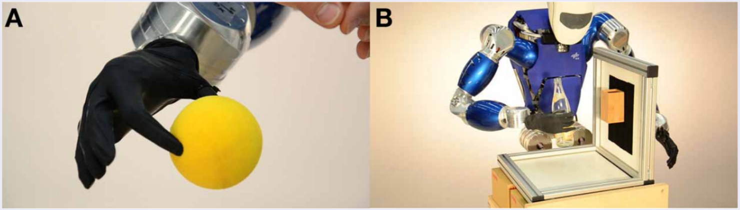
What Is Regression?

Estimating a relationship
between input variables
and continuous output variables
from data





Application: Biosignal Processing



Gijsberts, A., Bohra, R., Sierra Gonzlez, D., Werner, A., Nowak, M., Caputo, B., Roa, M. and Castellini, C. (2014)
Stable myoelectric control of a hand prosthesis using non-linear incremental learning
Frontiers in Neurorobotics

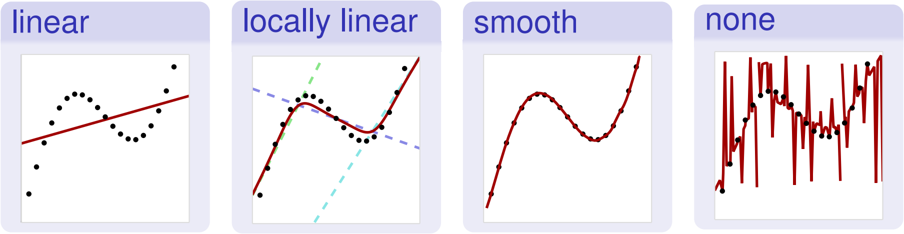
What Is Not Regression?

Training data

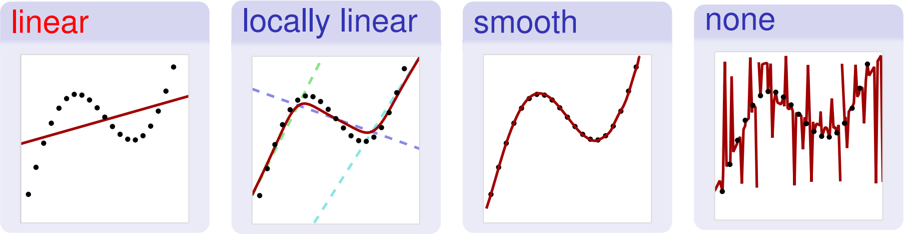
$$\{(\underbrace{\mathbf{x}_n}_{\text{input}}, \underbrace{\mathbf{y}_n}_{\text{target}})\}_{n=1}^{N} \qquad \forall n, \mathbf{x}_n \in X \wedge \mathbf{y}_n \in Y$$

Supervised Learning	targets available	
Regression	targets available	$Y \subseteq \mathbb{R}^M$
Classification	targets available	$Y \subseteq 1, \ldots K$
Reinforcement learning	no targets, only rewards	$r_n \subseteq \mathbb{R}$
Unsupervised learning	no targets at all	

Regression – Assumptions about the Function



Regression – Assumptions about the Function

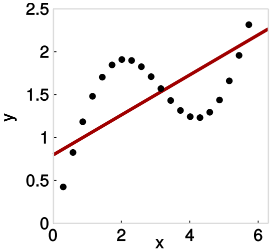


$\rightarrow$ Linear Least Squares

A.M. Legendre (1805).
Nouvelles méthodes pour la détermination des orbites des comtes [519]
Firmin Didot.

C.F. Gauss (1809).
Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum [943]

Linear Least Squares

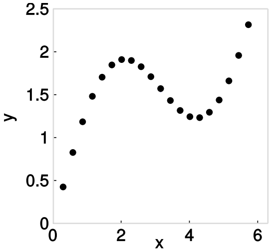


$f(\mathbf{x}) = \mathbf{a}^\intercal \mathbf{x}$

Linear Least Squares

Linear Least Squares

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$



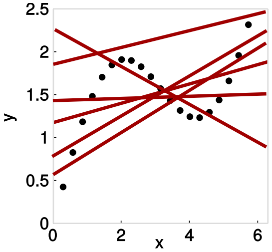
$f(\mathbf{x}) = \mathbf{a}^\mathsf{T}\mathbf{x}$

Linear Least Squares

- $\mathbf{X}$ is the $N \times D$ "design matrix"
- Each row is a D-dim. data point

Linear Least Squares

- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion



$$f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$$

Linear Least Squares

Linear Least Squares

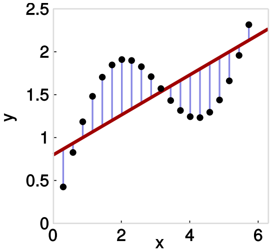
- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion

1 Define fitting criterion

Sum of squared residuals

$$S(\mathbf{a}) = \sum_{n=1}^{N} r_n^2 \tag{1}$$

$$= \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2 \tag{2}$$



$f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$

Linear Least Squares

Linear Least Squares

- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion

1 Define fitting criterion

Sum of squared residuals

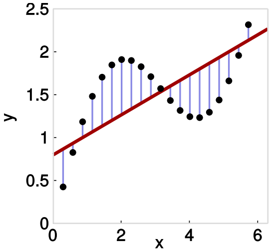
$$S(\mathbf{a}) = \sum_{n=1}^{N} r_n^2 \tag{1}$$

$$= \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2 \tag{2}$$

Applied to a linear model

$$S(\mathbf{a}) = \sum_{n=1}^{N} (y_n - \mathbf{a}^\intercal \mathbf{x}_n)^2 \tag{3}$$

$$= (\mathbf{y} - \mathbf{X}\mathbf{a})^\intercal (\mathbf{y} - \mathbf{X}\mathbf{a}), \tag{4}$$



$f(\mathbf{x}) = \mathbf{a}^\intercal \mathbf{x}$

Linear Least Squares

Linear Least Squares

- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion

2 Optimize **a** w.r.t. criterion

Minimize sum of squared residuals $S(\mathbf{a})$

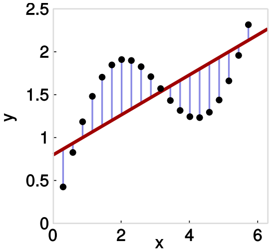
$$\mathbf{a}^* = \arg\min_{\mathbf{a}} S(\mathbf{a}) \tag{1}$$

$$= \arg\min_{\mathbf{a}} (\mathbf{y} - \mathbf{X}\mathbf{a})^\top (\mathbf{y} - \mathbf{X}\mathbf{a}) \tag{2}$$

Quadratic cost: when is its derivative 0?

$$S'(\mathbf{a}) = 2(\mathbf{a}(\mathbf{X}^\top \mathbf{X}) - \mathbf{X}^\top \mathbf{y}) \tag{3}$$

$$\mathbf{a}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{4}$$



$f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$

Linear Least Squares

Linear Least Squares

- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion

2 Optimize **a** w.r.t. criterion

Minimize sum of squared residuals $S(\mathbf{a})$

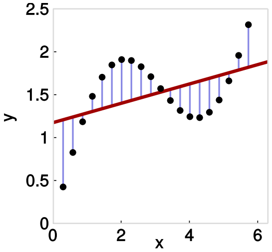
$$\mathbf{a}^* = \arg\min_{\mathbf{a}} S(\mathbf{a}) \quad (1)$$

$$= \arg\min_{\mathbf{a}} (\mathbf{y} - \mathbf{X}\mathbf{a})^\intercal(\mathbf{y} - \mathbf{X}\mathbf{a}) \quad (2)$$

Quadratic cost: when is its derivative 0?

$$S'(\mathbf{a}) = 2(\mathbf{a}(\mathbf{X}^\intercal\mathbf{X}) - \mathbf{X}^\intercal\mathbf{y}) \quad (3)$$

$$\mathbf{a}^* = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y}. \quad (4)$$



$f(\mathbf{x}) = \mathbf{a}^{*\intercal}\mathbf{x}$

Linear Least Squares

Linear Least Squares

- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion

2 Optimize **a** w.r.t. criterion

Minimize sum of squared residuals $S(\mathbf{a})$

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} S(\mathbf{a}) \tag{1}$$

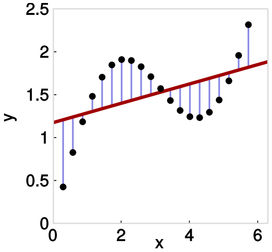
$$= \arg\min_{\mathbf{a}} (\mathbf{y} - \mathbf{X}\mathbf{a})^\intercal(\mathbf{y} - \mathbf{X}\mathbf{a}) \tag{2}$$

Quadratic cost: when is its derivative 0?

$$S'(\mathbf{a}) = 2(\mathbf{a}(\mathbf{X}^\intercal\mathbf{X}) - \mathbf{X}^\intercal\mathbf{y}) \tag{3}$$

$$\mathbf{a}^* = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y}. \tag{4}$$

Nice! Closed form solution to find $\mathbf{a}^*$

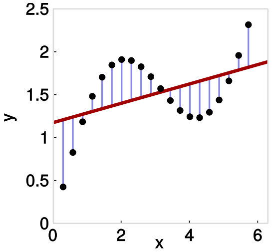


$f(\mathbf{x}) = \mathbf{a}^{*\intercal}\mathbf{x}$

Linear Least Squares

Linear Least Squares

- Which line fits the data best?
  1. Define fitting criterion
  2. Optimize **a** w.r.t. criterion



$f(\mathbf{x}) = \mathbf{a}^{*\mathsf{T}}\mathbf{x}$

Linear Least Squares

2 Optimize **a** w.r.t. criterion

Minimize sum of squared residuals $S(\mathbf{a})$

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} S(\mathbf{a}) \tag{1}$$

$$= \arg\min_{\mathbf{a}} (\mathbf{y} - \mathbf{X}\mathbf{a})^{\mathsf{T}}(\mathbf{y} - \mathbf{X}\mathbf{a}) \tag{2}$$

Quadratic cost: when is its derivative 0?

$$S'(\mathbf{a}) = 2(\mathbf{a}(\mathbf{X}^{\mathsf{T}}\mathbf{X}) - \mathbf{X}^{\mathsf{T}}\mathbf{y}) \tag{3}$$

$$\mathbf{a}^* = (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{y}. \tag{4}$$

Nice! Closed form solution to find $\mathbf{a}^*$

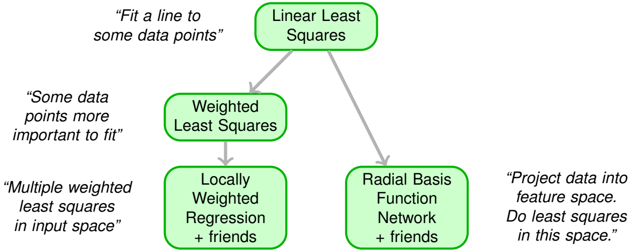
Offset trick

$$f(\mathbf{x}) = \mathbf{a}^{\mathsf{T}}\mathbf{x} + b$$

$$= \begin{bmatrix}\mathbf{a}\\ b\end{bmatrix}^{\mathsf{T}} \begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix}$$

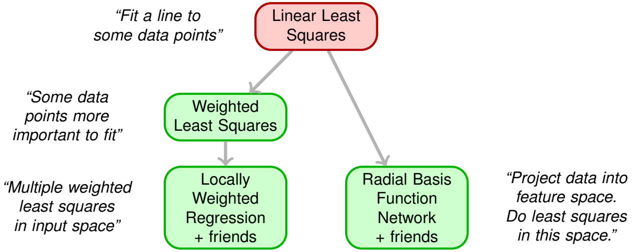
$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} & 1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D} & 1 \end{bmatrix}$$

Outline

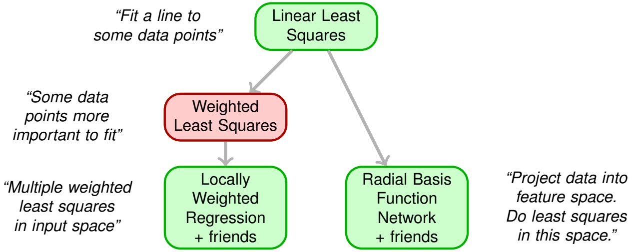


"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regression + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."

Outline


"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regression + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."

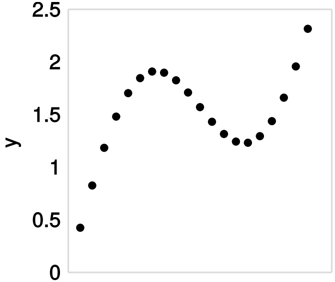
Outline



"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regression + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."

Weighted Linear Least Squares

Idea: more important to fit some points than others.

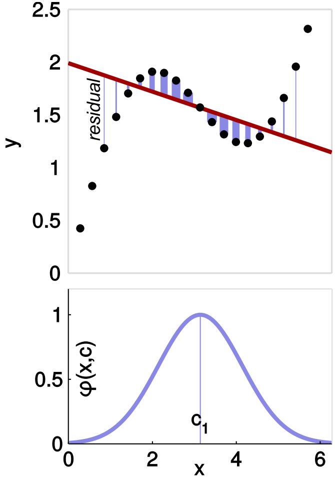


Weighted Linear Least Squares

Idea: more important to fit some points than others.

- Importance $\equiv$ Weight w_n
- Example weighting
  - manual
  - boxcar function
  - Gaussian function

$$
\begin{aligned}
w_n &= \phi(\mathbf{x}_n, \boldsymbol{\theta}) \\
&= \exp\left(-\tfrac{1}{2}(\mathbf{x}-\mathbf{c})^\intercal \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\mathbf{c})\right) \\
&\qquad \text{with } \boldsymbol{\theta} = (\mathbf{c}, \boldsymbol{\Sigma})
\end{aligned}
$$



Weighted Linear Least Squares

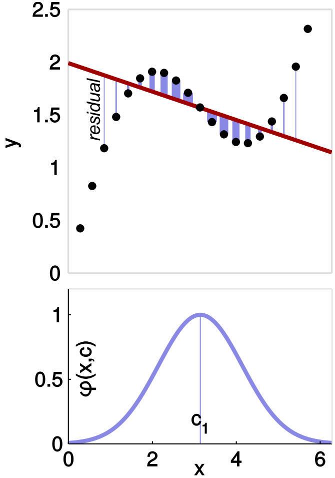
Idea: more important to fit some points than others.

1 Define fitting criterion

Weighted residuals:

$$S(\mathbf{a}) = \sum_{n=1}^{N} w_n (y_n - \mathbf{a}^\top \mathbf{x}_n)^2. \tag{5}$$

(6)



Weighted Linear Least Squares

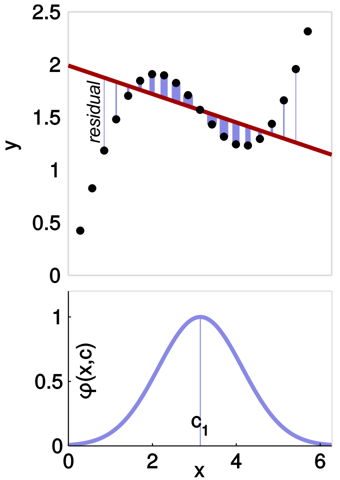
Idea: more important to fit some points than others.

❶ Define fitting criterion

Weighted residuals:

$$S(\mathbf{a}) = \sum_{n=1}^{N} w_n (y_n - \mathbf{a}^\top \mathbf{x}_n)^2. \qquad (5)$$

$$= (\mathbf{y} - \mathbf{Xa})^\top \mathbf{W} (\mathbf{y} - \mathbf{Xa}), \qquad (6)$$



Weighted Linear Least Squares

Idea: more important to fit some points than others.

1 Define fitting criterion

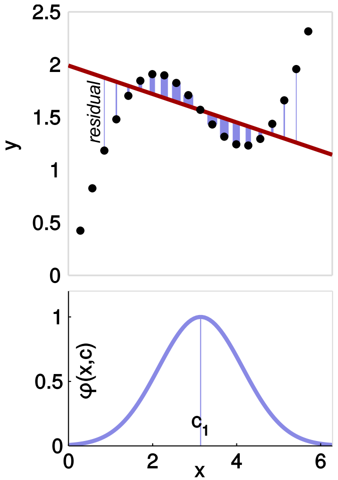
Weighted residuals:

$$S(\mathbf{a}) = \sum_{n=1}^{N} w_n (y_n - \mathbf{a}^\top \mathbf{x}_n)^2. \tag{5}$$

$$= (\mathbf{y} - \mathbf{X}\mathbf{a})^\top \mathbf{W} (\mathbf{y} - \mathbf{X}\mathbf{a}), \tag{6}$$

2 Optimize a w.r.t. criterion

$$\mathbf{a}^* = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{y}. \tag{7}$$



Weighted Linear Least Squares

Idea: more important to fit some points than others.

❶ Define fitting criterion

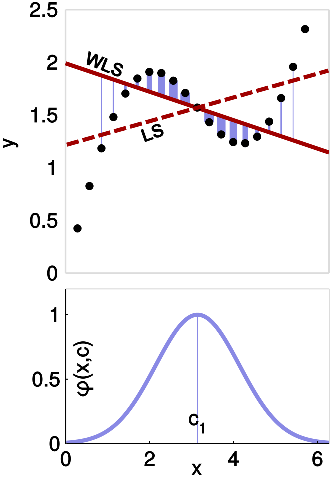
Weighted residuals:

$$S(\mathbf{a}) = \sum_{n=1}^{N} w_n (y_n - \mathbf{a}^\top \mathbf{x}_n)^2. \tag{5}$$

$$= (\mathbf{y} - \mathbf{X}\mathbf{a})^\top \mathbf{W} (\mathbf{y} - \mathbf{X}\mathbf{a}), \tag{6}$$

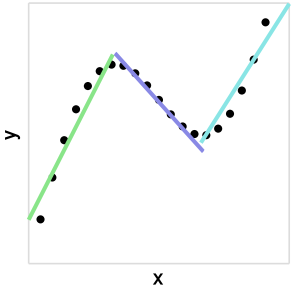
❷ Optimize **a** w.r.t. criterion

$$\mathbf{a}^* = (\mathbf{X}^\top \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W} \mathbf{y}. \tag{7}$$



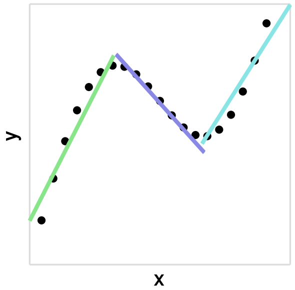
Locally Weighted Regressions

In robotics, functions usually non-linear. But often **locally** linear!



Locally Weighted Regressions

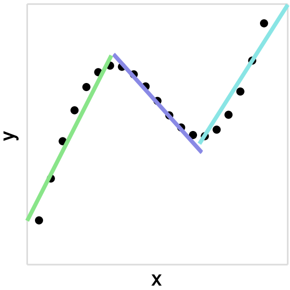
In robotics, functions usually non-linear. But often **locally** linear!



Idea: Do multiple, independent, locally weighted least sq. regressions

Locally Weighted Regressions

In robotics, functions usually non-linear. But often **locally** linear!



Idea: Do multiple, independent, locally weighted least sq. regressions

William S. Cleveland; Susan J. Devlin (1988).
Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting [4074]
Journal of the American Statistical Association.

Atkeson, C. G.; Moore, A. W. and Schaal, S. (1997).
Locally Weighted Learning for Control [2160]
Artificial Intelligence Review.

Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

$$
\begin{aligned}
&\texttt{for } e = 1 \dots E \\
&\quad \texttt{for } n = 1 \dots N \\
&\quad\quad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\intercal \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{W}_e \mathbf{y}.
\end{aligned}
\tag{8}
$$

Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

```
for e = 1 ... E
    for n = 1 ... N
        W_e^nn = g(x_n, c_e, Σ)
    a_e = (XᵀW_eX)^-1 XᵀW_e y.
```

$$\texttt{for } e = 1 \ldots E$$

$$\texttt{for } n = 1 \ldots N$$

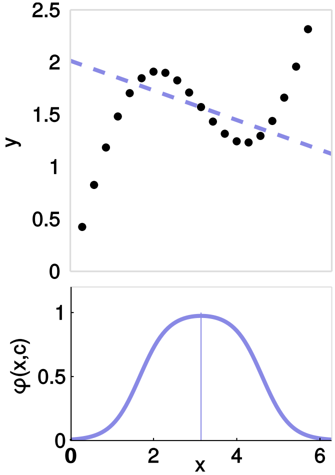
$$\mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma)$$

$$\mathbf{a}_e = (\mathbf{X}^\intercal \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{W}_e \mathbf{y}. \qquad (8)$$

Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

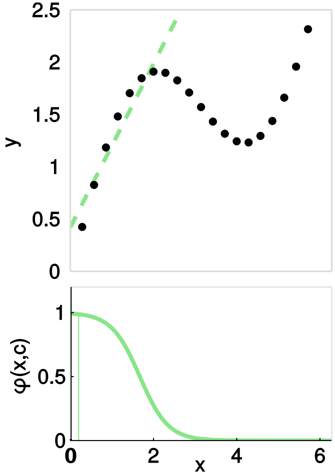
$$
\begin{aligned}
&\texttt{for } e = 1 \dots E \\
&\quad \texttt{for } n = 1 \dots N \\
&\quad\quad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\top \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_e \mathbf{y}. \qquad (8)
\end{aligned}
$$



Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

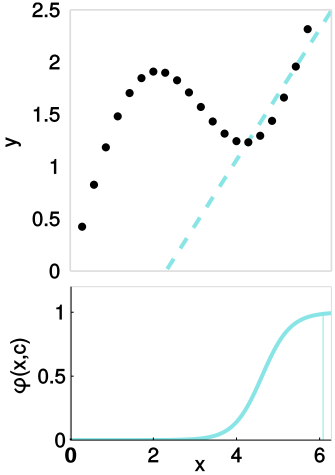
$$
\begin{aligned}
&\texttt{for } e = 1 \ldots E \\
&\quad \texttt{for } n = 1 \ldots N \\
&\quad\quad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\intercal \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{W}_e \mathbf{y}. \qquad (8)
\end{aligned}
$$



Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

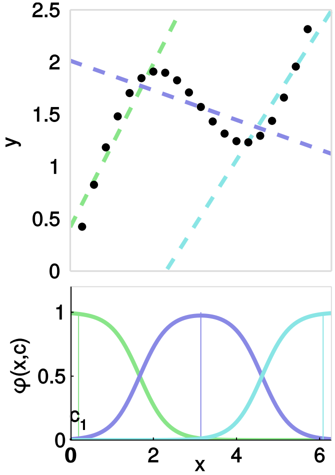
$$
\begin{aligned}
&\texttt{for } e = 1 \dots E \\
&\quad \texttt{for } n = 1 \dots N \\
&\qquad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\top \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_e \mathbf{y}. \qquad (8)
\end{aligned}
$$



Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

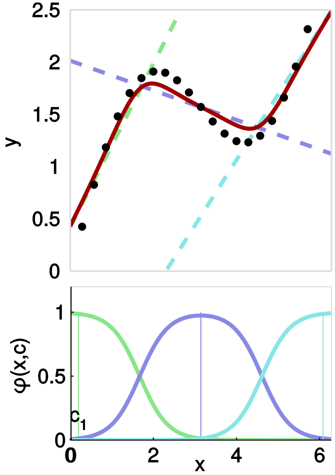
$$
\begin{aligned}
&\texttt{for } e = 1 \dots E \\
&\quad \texttt{for } n = 1 \dots N \\
&\qquad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\top \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_e \mathbf{y}. \qquad (8)
\end{aligned}
$$



Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

$$
\begin{aligned}
&\texttt{for } e = 1 \dots E \\
&\quad \texttt{for } n = 1 \dots N \\
&\qquad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\top \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{W}_e \mathbf{y}. \qquad (8)
\end{aligned}
$$



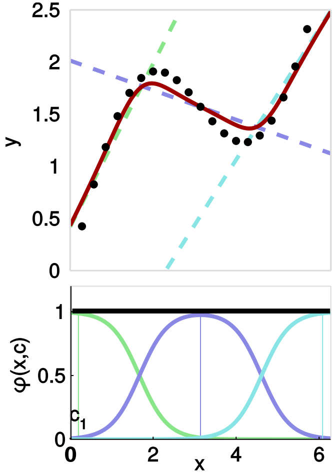
Resulting model

$$
f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \theta_e) \cdot (\mathbf{a}_e^\top \mathbf{x}). \qquad (9)
$$

Locally Weighted Regressions

- **Idea**: multiple, independent, locally weighted least squares regressions
  - Locally: radial weighting function with different centers ("receptive field")

$$
\begin{aligned}
&\texttt{for } e = 1 \dots E \\
&\quad \texttt{for } n = 1 \dots N \\
&\qquad \mathbf{W}_e^{nn} = g(\mathbf{x}_n, \mathbf{c}_e, \Sigma) \\
&\quad \mathbf{a}_e = (\mathbf{X}^\intercal \mathbf{W}_e \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{W}_e \mathbf{y}. \qquad (8)
\end{aligned}
$$



(ϕ must be normalized)

Resulting model

$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \theta_e) \cdot (\mathbf{a}_e^\intercal \mathbf{x}). \qquad (9)$$

Variations of Locally Weighted Regressions

Receptive Field Weighted Regression

- Incremental, not batch
- E, centers $\mathbf{c}_{1 \ldots E}$ and widths $\boldsymbol{\Sigma}_{1 \ldots E}$ determined automatically
- Disadvantage: many open parameters

Schaal, S. and Atkeson, C. G. (1997).
Receptive Field Weighted Regression [34]
Technical Report TR-H-209, ATR Human Information Processing Laboratories.

Variations of Locally Weighted Regressions

Receptive Field Weighted Regression

- Incremental, not batch
- E, centers $\mathbf{c}_{1\dots E}$ and widths $\boldsymbol{\Sigma}_{1\dots E}$ determined automatically
- Disadvantage: many open parameters

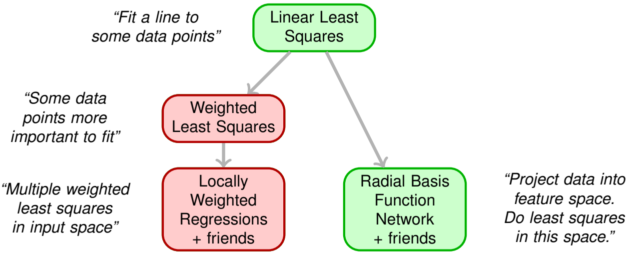
Schaal, S. and Atkeson, C. G. (1997).
Receptive Field Weighted Regression [34]
Technical Report TR-H-209, ATR Human Information Processing Laboratories.

Locally Weighted Projection Regression

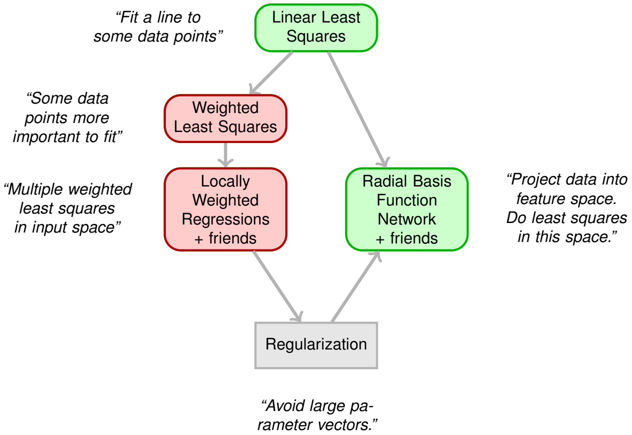
- As RFWR, but also performs dimensionality reduction within each receptive field

Vijayakumar, S. and Schaal, S. (2000).
Locally Weighted Projection Regression . . . [208]
International Conference on Machine Learning.

Outline

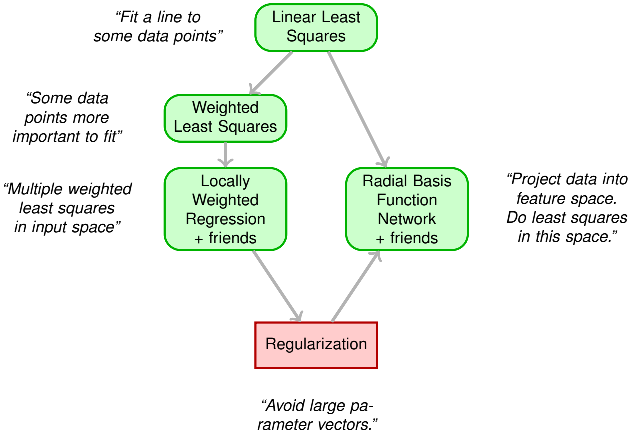

"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regressions + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."

Outline



"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regressions + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."
Regularization
"Avoid large parameter vectors."

Outline


"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regression + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."
Regularization
"Avoid large parameter vectors."

Regularization

- **Idea**: penalize large parameter vectors to
  - avoid overfitting / achieve sparse parameter vectors

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} (\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}}) \tag{10}$$

Regularization

- **Idea**: penalize large parameter vectors to
  - avoid overfitting / achieve sparse parameter vectors

$$\mathbf{a}^* = \arg\min_{\mathbf{a}}(\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}}) \qquad (10)$$

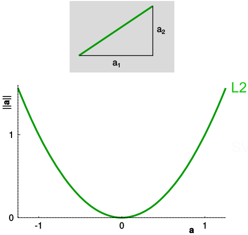
L^2-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_2 = \left(\sum_{d=1}^{D} |a_d|^2 \right)^{\frac{1}{2}}$$

Euclidean distance

$$\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}.$$

"Thikonov Regularization"
"Ridge Regression"



Regularization

- **Idea**: penalize large parameter vectors to
  - avoid overfitting / achieve sparse parameter vectors

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \Big(\underbrace{\frac{1}{2} \|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2} \|\mathbf{a}\|^2}_{\text{small parameters}} \Big) \tag{10}$$

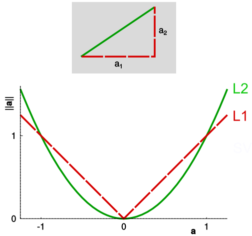
L^2-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_2 = \left(\sum_{d=1}^{D} |a_d|^2 \right)^{\frac{1}{2}}$$

Euclidean distance

$$\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}.$$

"Thikonov Regularization"
"Ridge Regression"



L^1-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_1 = \left(\sum_{d=1}^{D} |a_d|^1 \right)^{\frac{1}{1}} = \sum_{d=1}^{D} |a_d|$$

Manhattan distance

no closed-form solution . . .

"LASSO Regularization"

Regularization

- **Idea**: penalize large parameter vectors to
  - avoid overfitting / achieve sparse parameter vectors

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \Big(\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}} \Big) \qquad (10)$$

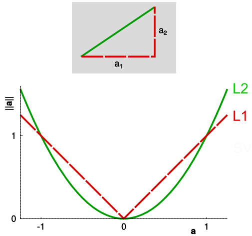
L^2-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_2 = \left(\sum_{d=1}^{D} |a_d|^2 \right)^{\frac{1}{2}}$$

Euclidean distance

$$\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}.$$

"Thikonov Regularization"
"Ridge Regression"



L^1-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_1 = \left(\sum_{d=1}^{D} |a_d|^1 \right)^{\frac{1}{1}} = \sum_{d=1}^{D} |a_d|$$

Manhattan distance

no closed-form solution . . .

"LASSO Regularization"

Use combination of L^1 and L^2: "Elastic Nets"

Regularization

- **Idea**: penalize large parameter vectors to
  - avoid overfitting / achieve sparse parameter vectors

$$\mathbf{a}^* = \arg\min_{\mathbf{a}}(\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}}) \quad (10)$$

L^2-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_2 = \left(\sum_{d=1}^{D} |a_d|^2\right)^{\frac{1}{2}}$$

Euclidean distance

$$\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}.$$

"Thikonov Regularization"
"Ridge Regression"

L^1-norm for $\|\mathbf{a}\|$

$$\|\mathbf{a}\|_1 = \left(\sum_{d=1}^{D} |a_d|^1\right)^{\frac{1}{1}} = \sum_{d=1}^{D} |a_d|$$

Manhattan distance

no closed-form solution ...

"LASSO Regularization"

Use combination of L^1 and L^2: "Elastic Nets"

Regularization

- **Idea**: penalize large parameter vectors to
  - avoid overfitting / achieve sparse parameter vectors

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} (\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}}) \tag{10}$$

Michael Littman and Charles Isbell feat Infinite Harmony

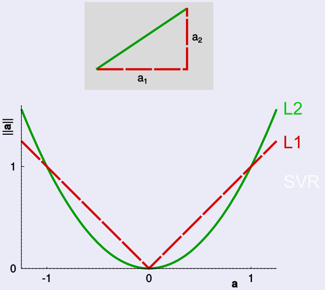
"Overfitting A Cappella"

Beyond squares

$$\mathbf{a}^* = \arg\min_{\mathbf{a}}\Big(\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}} \Big) \tag{11}$$

Penalty on parameters **a**

(regularization)



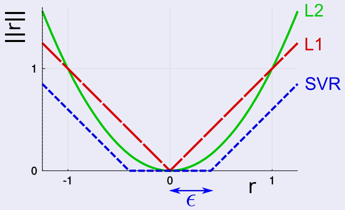
Beyond squares

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} (\underbrace{\frac{1}{2}\|\mathbf{y} - \mathbf{X}^\intercal \mathbf{a}\|^2}_{\text{fit data}} + \underbrace{\frac{\lambda}{2}\|\mathbf{a}\|^2}_{\text{small parameters}}) \qquad (11)$$

Penalty on residuals r_n

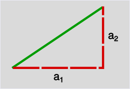
(fit data)

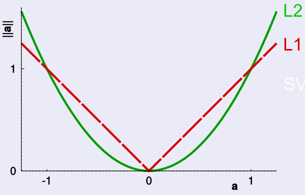
L_2: least squares
L_1: least deviations
L_ε: support vector regression



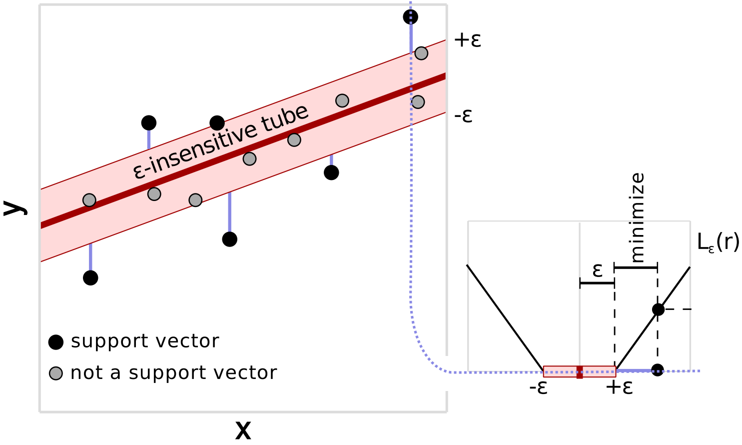
Penalty on parameters **a**

(regularization)



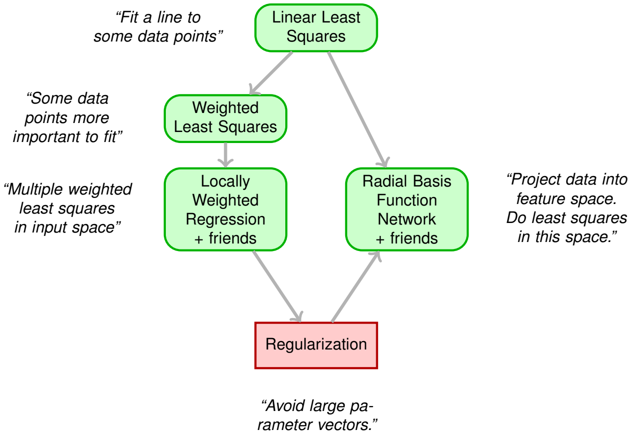


Linear Support Vector Regression



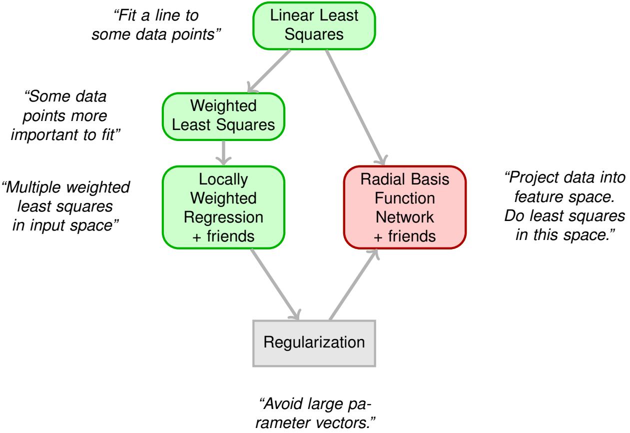
- No closed-form solution, but efficient optimizers exist

Outline



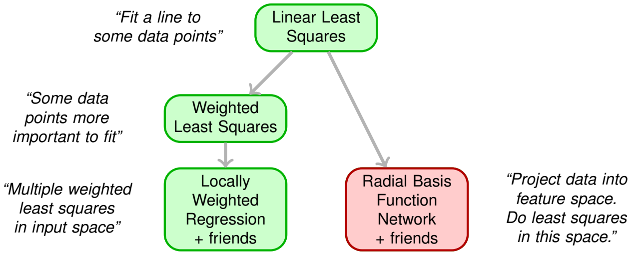
“Fit a line to some data points”
Linear Least Squares
“Some data points more important to fit”
Weighted Least Squares
“Multiple weighted least squares in input space”
Locally Weighted Regression + friends
Radial Basis Function Network + friends
“Project data into feature space. Do least squares in this space.”
Regularization
“Avoid large parameter vectors.”

Outline



"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regression + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."
Regularization
"Avoid large parameter vectors."

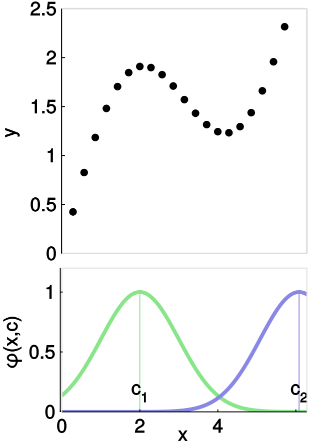
Outline



"Fit a line to
some data points"
Linear Least
Squares
"Some data
points more
important to fit"
Weighted
Least Squares
"Multiple weighted
least squares
in input space"
Locally
Weighted
Regression
+ friends
Radial Basis
Function
Network
+ friends
"Project data into
feature space.
Do least squares
in this space."

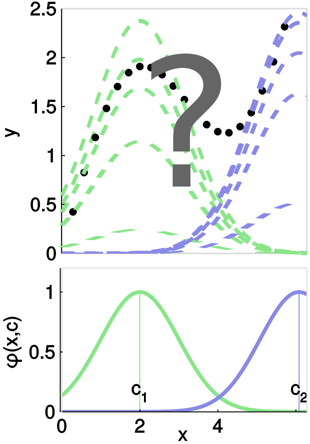
Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \mathbf{c}_e). \tag{12}$$



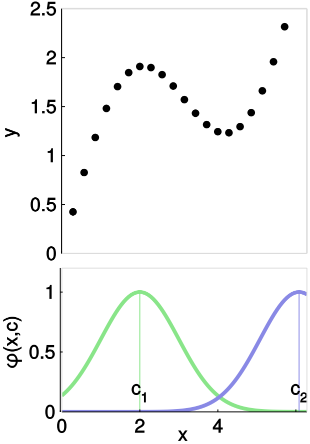
Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \mathbf{c}_e). \tag{12}$$



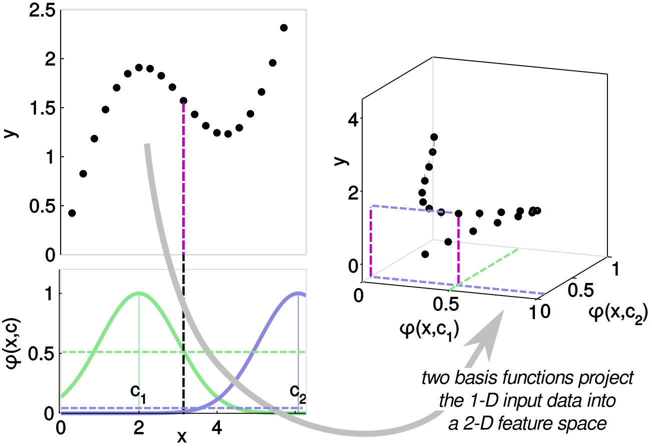
Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \mathbf{c}_e). \quad (12)$$



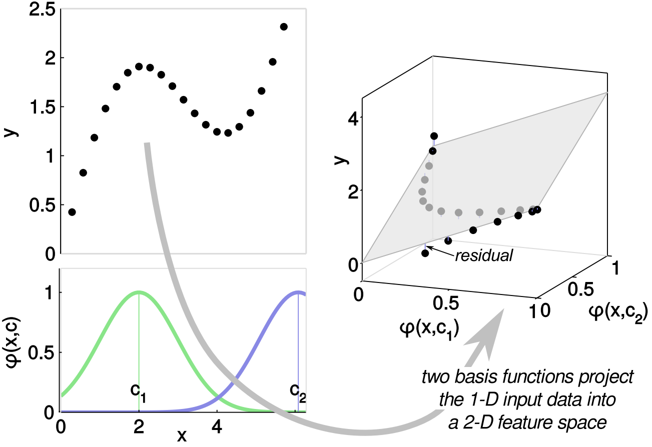
Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \mathbf{c}_e). \qquad (12)$$



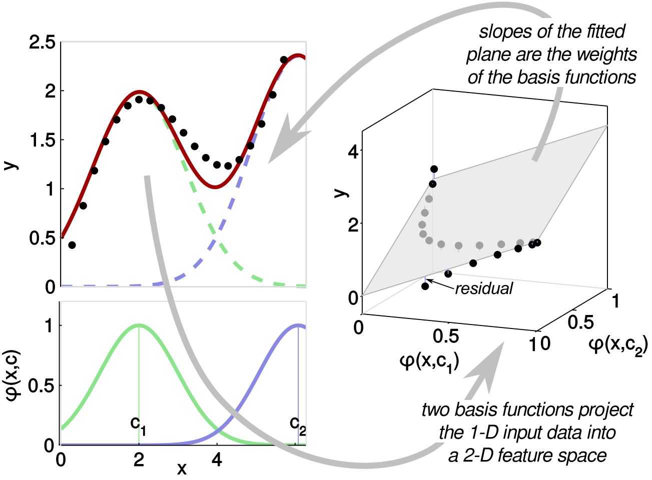
Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \mathbf{c}_e). \qquad (12)$$



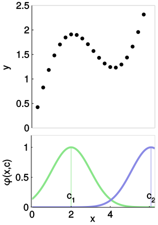
Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \mathbf{c}_e). \tag{12}$$



Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \boldsymbol{\theta}_e). \qquad (13)$$

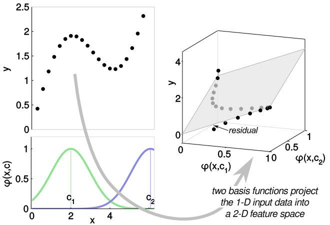


Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \boldsymbol{\theta}_e). \qquad (13)$$

Feature matrix (analogous to design matrix $\mathbf{X}$)

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{c}_1) & \phi(\mathbf{x}_1, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_1, \mathbf{c}_E) \\ \phi(\mathbf{x}_2, \mathbf{c}_1) & \phi(\mathbf{x}_2, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_2, \mathbf{c}_E) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N, \mathbf{c}_1) & \phi(\mathbf{x}_N, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_N, \mathbf{c}_E) \end{bmatrix} \qquad (14)$$



Radial Basis Function Network

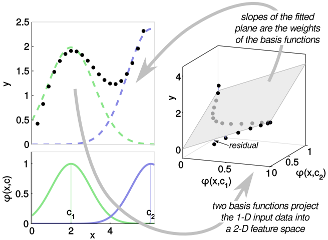
$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \boldsymbol{\theta}_e). \qquad (13)$$

Feature matrix (analogous to design matrix $\mathbf{X}$)

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{c}_1) & \phi(\mathbf{x}_1, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_1, \mathbf{c}_E) \\ \phi(\mathbf{x}_2, \mathbf{c}_1) & \phi(\mathbf{x}_2, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_2, \mathbf{c}_E) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N, \mathbf{c}_1) & \phi(\mathbf{x}_N, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_N, \mathbf{c}_E) \end{bmatrix} \qquad (14)$$

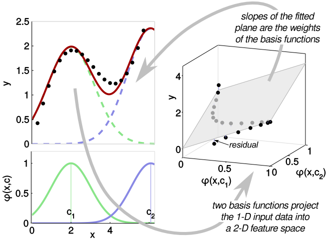
Least squares solution

$$\mathbf{w}^* = (\boldsymbol{\Theta}^\intercal \boldsymbol{\Theta})^{-1} \boldsymbol{\Theta}^\intercal \mathbf{y}. \qquad (15)$$



Radial Basis Function Network

$$f(\mathbf{x}) = \sum_{e=1}^{E} w_e \cdot \phi(\mathbf{x}, \boldsymbol{\theta}_e). \tag{13}$$



Feature matrix (analogous to design matrix $\mathbf{X}$)

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{c}_1) & \phi(\mathbf{x}_1, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_1, \mathbf{c}_E) \\ \phi(\mathbf{x}_2, \mathbf{c}_1) & \phi(\mathbf{x}_2, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_2, \mathbf{c}_E) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\mathbf{x}_N, \mathbf{c}_1) & \phi(\mathbf{x}_N, \mathbf{c}_2) & \cdots & \phi(\mathbf{x}_N, \mathbf{c}_E) \end{bmatrix} \tag{14}$$

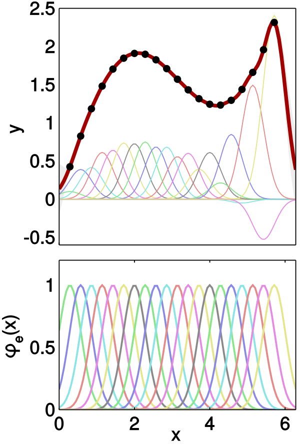
Least squares solution

$$\mathbf{w}^* = (\boldsymbol{\Theta}^\intercal \boldsymbol{\Theta})^{-1} \boldsymbol{\Theta}^\intercal \mathbf{y}. \tag{15}$$

Kernel Ridge Regression

- Like a RBFN, but every data point is the center of a basis function

$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n \cdot k(\mathbf{x}, \mathbf{x}_n). \tag{16}$$



"Gram matrix"(analogous to design matrix **X**)

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{17}$$

$$\mathbf{w}^* = (\mathbf{K}^\intercal \mathbf{K})^{-1} \mathbf{K}^\intercal \mathbf{y} \tag{18}$$

$$= \mathbf{K}^{-1} \mathbf{y}, \tag{19}$$

(20)

Kernel Ridge Regression

- Like a RBFN, but every data point is the center of a basis function
- Uses L^2 regularization

$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n \cdot k(\mathbf{x}, \mathbf{x}_n). \tag{16}$$

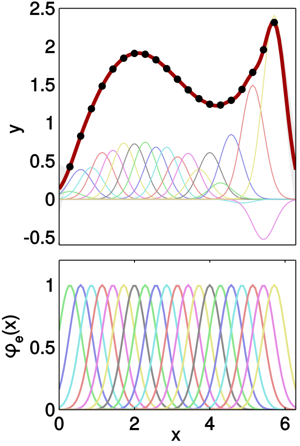
“Gram matrix”(analogous to design matrix $\mathbf{X}$)

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{17}$$

$$\mathbf{w}^* = (\mathbf{K}^\top \mathbf{K})^{-1} \mathbf{K}^\top \mathbf{y} \tag{18}$$

$$= \mathbf{K}^{-1} \mathbf{y}, \tag{19}$$

$$\mathbf{w}^* = (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{y} \quad \text{with } L^2 \text{ regularization} \tag{20}$$



(Radial) Basis Function Networks

Beyond radial basis functions

- Cosines: Ridge Regression with Random Fourier Features
- Sigmoids: Extreme Learning Machines (MLFF with 1 hidden)
- Boxcars: model trees (as decision trees, but for regression)
- Kernels: every data point is the center of a radial basis function

(Radial) Basis Function Networks

Beyond radial basis functions

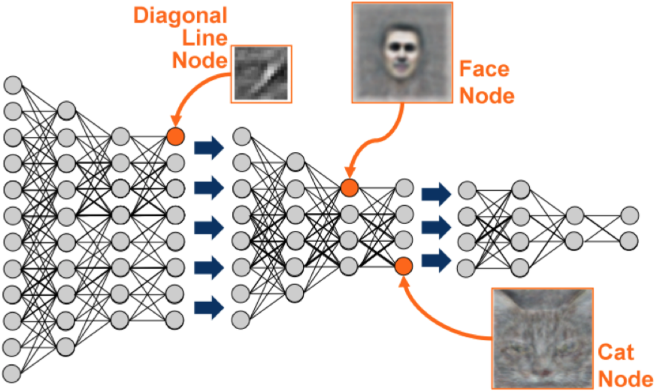
- Cosines: Ridge Regression with Random Fourier Features
- Sigmoids: Extreme Learning Machines (MLFF with 1 hidden)
- Boxcars: model trees (as decision trees, but for regression)
- Kernels: every data point is the center of a radial basis function

- Since least squares is at the heart of all of these
  - incremental versions $\leftarrow$ recursive least squares
  - apply L^2 regularization (still closed form)

least squares inside

Freek, aren't you being a bit shallow?

- Deep learning great when you
  - do not know the features
  - know the features to be hierarchically organized



John Smart

Freek, aren't you being a bit shallow?

- Deep learning great when you
  - do not know the features
  - know the features to be hierarchically organized

Rajeswaran A, Lowrey K, Todorov E and Kakade S. (2017).
Towards generalization and simplicity in continuous control
Neural Information Processing Systems (NIPS).

Table 1: Final performances of the policies

Task	Linear		RBF		NN
	stoc	mean	stoc	mean	TRPO
Swimmer	362	**366**	361	365	131
Hopper	3466	3651	3590	**3810**	3668
Cheetah	3810	4149	6477	**6620**	4800
Walker	4881	5234	5631	**5867**	5594
Ant	3980	4607	4297	4816	**5007**
Humanoid	5873	6440	6237	**6849**	6482

Table 2: Number of episodes to achieve threshold

Task	Th.	Linear	RBF	TRPO+NN
Swimmer	325	**1450**	1550	N-A
Hopper	3120	13920	**8640**	10000
Cheetah	3430	11250	6000	**4250**
Walker	4390	36840	25680	**14250**
Ant	3580	39240	**30000**	73500
Humanoid	5280	**79800**	96720	87000

A neural network perspective

All these models can be considered (degenerate) neural networks!

A neural network perspective

All these models can be considered (degenerate) neural networks!

Backpropagation can be used in all these models!

Linear model

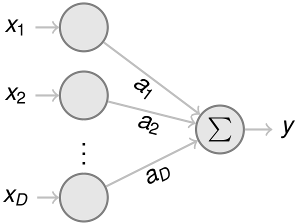


Figure: Network representation of a linear model. Activation is... linear!

RBFN

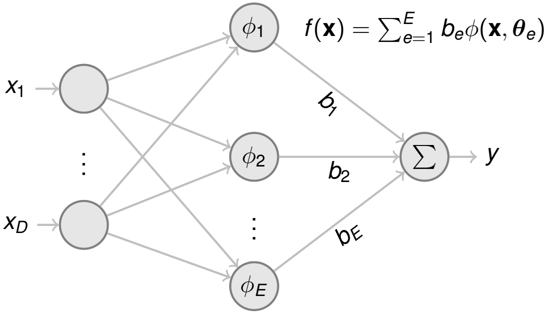


Figure: The RBFN model. ϕ_e is an abbreviation of $\phi(\mathbf{x}, \boldsymbol{\theta}_e)$

RRRFF

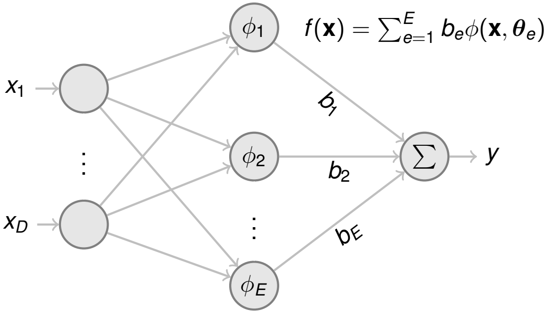


Figure: The RRRFF model. ϕ_e is an abbreviation of $\phi(\mathbf{x}, \boldsymbol{\theta}_e)$

SVR

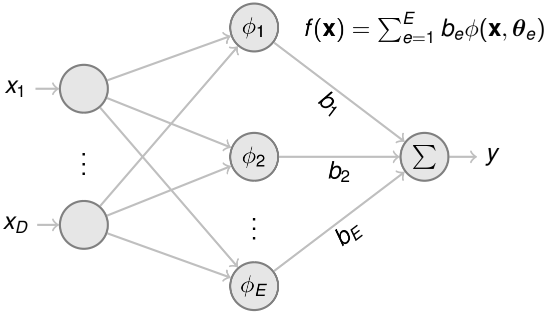


Figure: The SVR model. ϕ_e is an abbreviation of $\phi(\mathbf{x}, \boldsymbol{\theta}_e)$

Regression trees

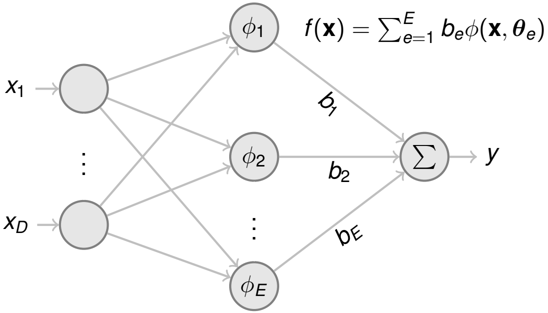


Figure: The regression trees model. ϕ_e is an abbreviation of $\phi(\mathbf{x}, \boldsymbol{\theta}_e)$

Extreme learning machine

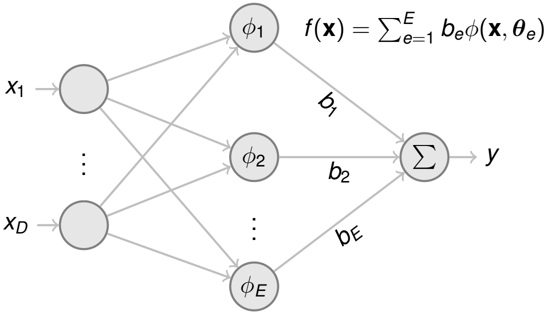


Figure: The extreme learning machine model. ϕ_e is an abbreviation of $\phi(\mathbf{x}, \boldsymbol{\theta}_e)$

Extreme Learning Machine vs. (Deep) Neural Networks

- ELM: sigmoid act. function, no hidden layer, random features
- ANN: sigmoid act. function, hidden layers, learned features

KRR and GPR

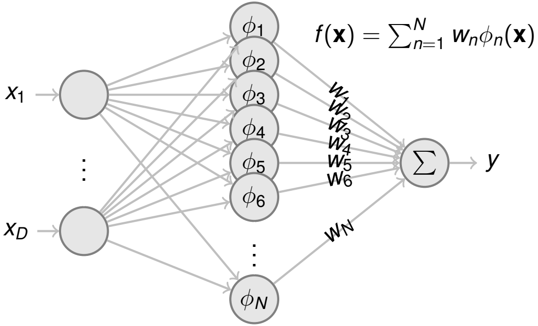


Figure: The function model used in KRR and GPR, as a network.

Locally weighted regression

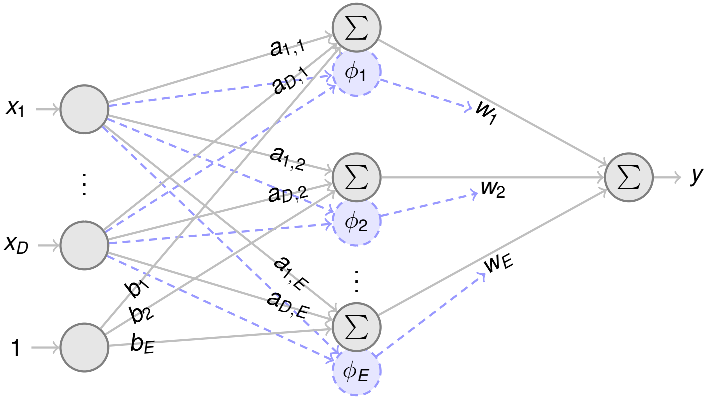
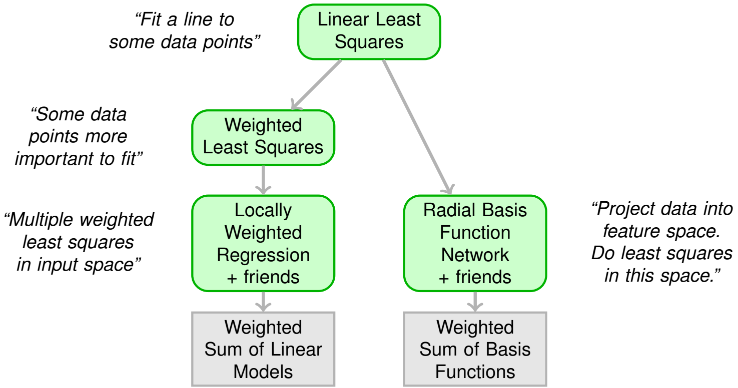


Figure: Function model in Locally Weighted Regressions, represented as a feedforward neural network. The functions $\phi_e(\mathbf{x})$ generate the weights w_e from the hidden nodes – which contain linear sub-models $(\mathbf{a}_e^\intercal \mathbf{x} + b_e)$ – to the output node. Here, ϕ_e is an abbreviation of $\phi(\mathbf{x}, \boldsymbol{\theta}_e)$

Conclusion

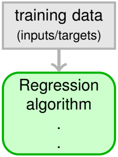


"Fit a line to some data points"
Linear Least Squares
"Some data points more important to fit"
Weighted Least Squares
"Multiple weighted least squares in input space"
Locally Weighted Regression + friends
Radial Basis Function Network + friends
"Project data into feature space. Do least squares in this space."
Weighted Sum of Linear Models
Weighted Sum of Basis Functions

Conclusion: Generic batch regression flow-chart

Algorithm

least squares: $\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}$



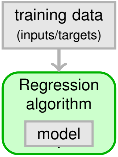
Conclusion: Generic batch regression flow-chart

Algorithm

least squares: $\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$

Model

linear model: $f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$



Conclusion: Generic batch regression flow-chart

Algorithm

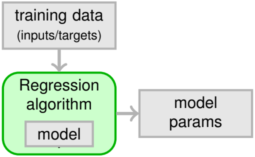
least squares: $\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}$

Model

linear model: $f(\mathbf{x}) = \mathbf{a}^\intercal \mathbf{x}$

Model parameters

slopes: $\mathbf{a}$



Conclusion: Generic batch regression flow-chart

Algorithm

least squares: $\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}$

Meta parameters

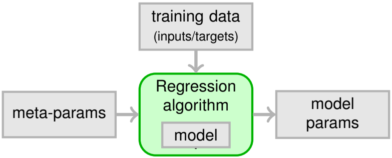
regularization: λ

Model

linear model: $f(\mathbf{x}) = \mathbf{a}^\intercal \mathbf{x}$

Model parameters

slopes: $\mathbf{a}$



Conclusion: Generic batch regression flow-chart

Algorithm

least squares: $\mathbf{a}^* = (\lambda \mathbf{I} + \mathbf{X}^\intercal \mathbf{X})^{-1} \mathbf{X}^\intercal \mathbf{y}$

Meta parameters

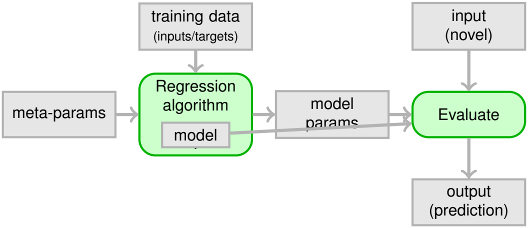
regularization: λ

Model

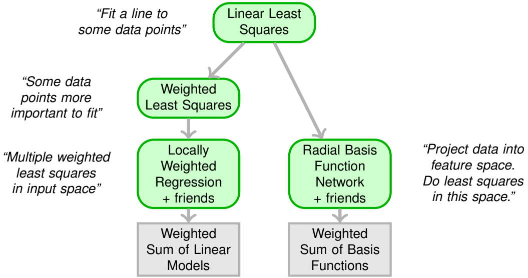
linear model: $f(\mathbf{x}) = \mathbf{a}^\intercal \mathbf{x}$

Model parameters

slopes: $\mathbf{a}$



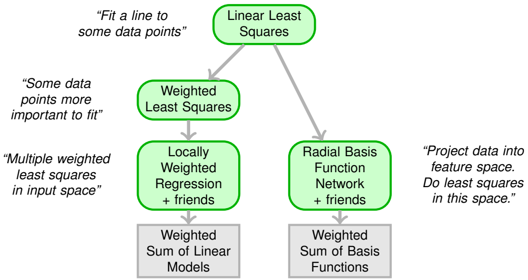
Conclusion



$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot (b_e + \mathbf{a}_e^{\intercal}\mathbf{x}) \qquad \text{Weighted sum of linear models} \tag{21}$$

$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot \; w_e \qquad \text{Weighted sum of basis functions} \tag{22}$$

Conclusion

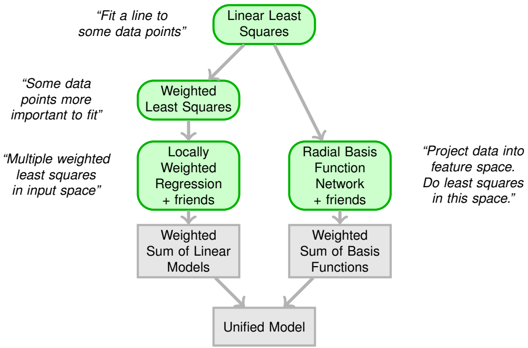


$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot (b_e + \mathbf{a}_e^\intercal \mathbf{x}) \qquad \text{Weighted sum of linear models} \qquad (21)$$

$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot w_e \qquad \text{Weighted sum of basis functions} \qquad (22)$$

(22) is a special case of (21) with $\mathbf{a}_e = \mathbf{0}$ and $b_e \equiv w_e$

Conclusion

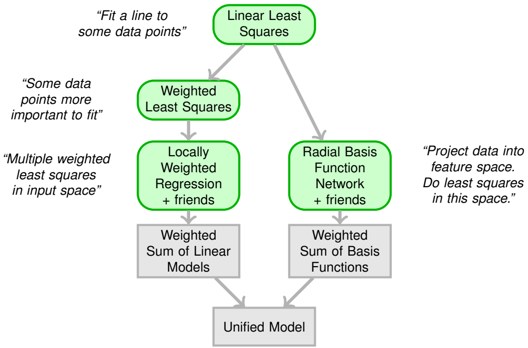


$$f(\mathbf{x}) = \sum\nolimits_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot (b_e + \mathbf{a}_e^{\mathsf{T}}\mathbf{x}) \qquad \text{Weighted sum of linear models} \qquad (21)$$

$$f(\mathbf{x}) = \sum\nolimits_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot w_e \qquad \text{Weighted sum of basis functions} \qquad (22)$$

(22) is a special case of (21) with $\mathbf{a}_e = \mathbf{0}$ and $b_e \equiv w_e$

Conclusion



Freek Stulp and Olivier Sigaud (2015).
Many regression algorithms, one unified model - A review.
Neural Networks.

$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot (b_e + \mathbf{a}_e^\intercal \mathbf{x}) \qquad \text{Weighted sum of linear models} \qquad (21)$$

$$f(\mathbf{x}) = \sum_{e=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_e) \cdot w_e \qquad \text{Weighted sum of basis functions} \qquad (22)$$

(22) is a special case of (21) with $\mathbf{a}_e = \mathbf{0}$ and $b_e \equiv w_e$

Conclusion

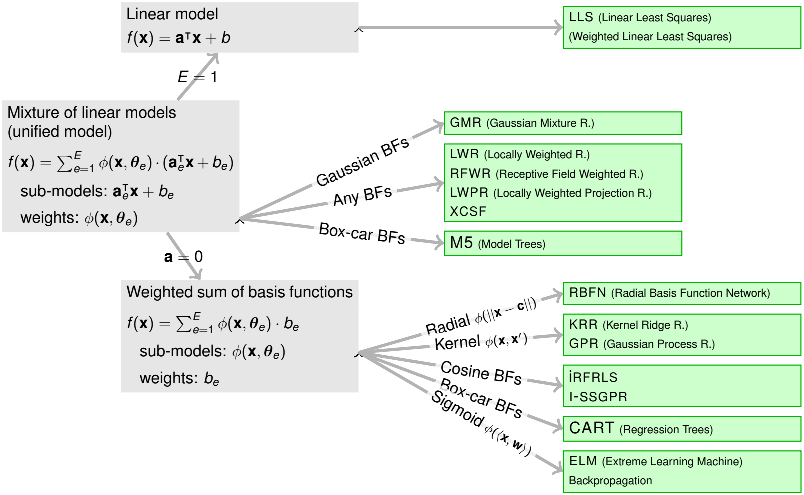


Figure: Classification of regression algorithms, based only on the model used to represent the underlying function.

Many toolkits available

- Python
  - scikit-learn: `http://scikit-learn.org`
  - StatsModels: `http://www.statsmodels.org/`
  - PbDlib: `http://calinon.ch/codes.htm`
  - dmpbbo: `https://github.com/stulp/dmpbbo`
- Matlab
  - curvefit: `https://www.mathworks.com/help/curvefit/linear-and-nonlinear-regression.html`
  - PbDlib: `http://calinon.ch/codes.htm`
- C++
  - PbDlib: `http://calinon.ch/codes.htm`
  - dmpbbo: `https://github.com/stulp/dmpbbo`

Personal Favourites

Gaussian process regression

+ Very few assumptions
+ Meta-parameters estimated from data itself
+ Estimates variance also
+ Works in high dimensions
- Training/query times increase with amount of data
- Not easy to make incremental

Gaussian mixture regression

+ Estimates variance also
+ Algorithm is inherently incremental
+ Some meta-parameters, but easy to tune
+ Fast training times
- Training only stable for low input dimensions

Locally Weighted Regressions

+ Fast query times, fast training
+ Few meta-parameters, and easy to set
+ Stable learning results (batch)
- Not incremental
- No variance estimate

Deep Learning

+ Automatic extraction of (hierarchical) features

Conclusion

- Don’t think about these regression algorithms as being unique
    - Similar algorithms that use different subsets of algorithmic features

- All these models are essentially shallow neural networks with different basis functions

Conclusion

- Don’t think about these regression algorithms as being unique
  - Similar algorithms that use different subsets of algorithmic features
- All these models are essentially shallow neural networks with different basis functions

Thank you for your attention!

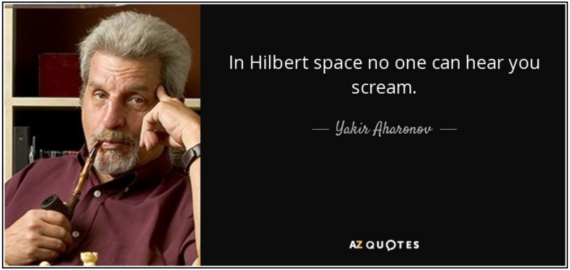
Appendix

Gaussian Process Regression

"Given a Gaussian process on some topological space T, with a continuous covariance kernel $C(\cdot, \cdot) : T \times T \rightarrow R$, we can associate a Hilbert space, which is the reproducing kernel Hilbert space of real-valued functions on T, with C as kernel function."

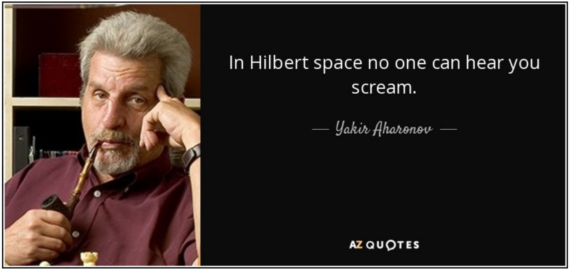
Gaussian Process Regression

"Given a Gaussian process on some topological space T, with a continuous covariance kernel $C(\cdot, \cdot) : T \times T \to R$, we can associate a Hilbert space, which is the reproducing kernel Hilbert space of real-valued functions on T, with C as kernel function."



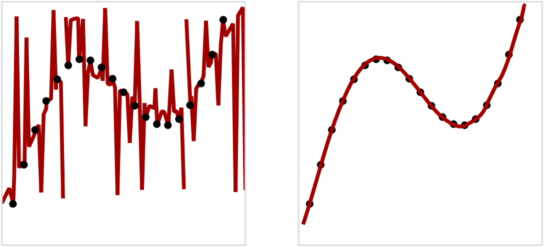
Gaussian Process Regression

"Given a Gaussian process on some topological space T, with a continuous covariance kernel $C(\cdot,\cdot) : T \times T \to R$, we can associate a Hilbert space, which is the reproducing kernel Hilbert space of real-valued functions on T, with C as kernel function."



Instead of screaming, let's talk about what it means to be *smooth*.

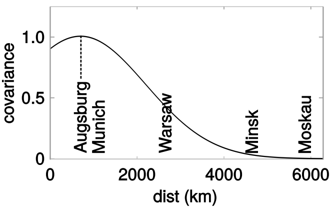
Gaussian Process Regression



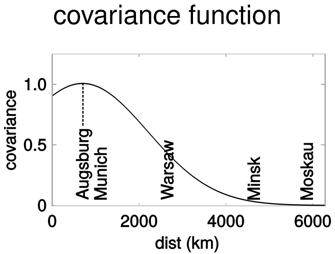
- Points that are close in the input space should be close in the output space.
  - Cities that are close geographically have similar temperatures (on average)
  - Taller people have larger shoe sizes (on average)
- Shoe size **covaries** with height

Gaussian Process Regression – Covariance Function

covariance function



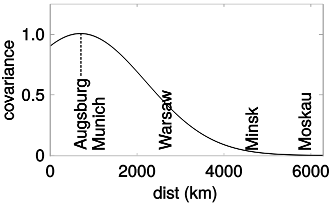
Gaussian Process Regression – Covariance Function



	x_{Aug}	x_{Muc}	x_{War}	x_{Min}	x_{Mos}
x_{Aug}	1.00	0.96	0.42	0.02	0.00

Gaussian Process Regression – Covariance Function

covariance function



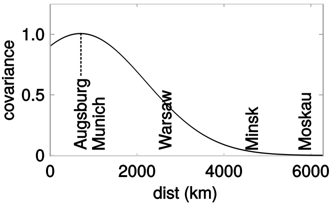
covariance matrix (Gram matrix)

$\mathbf{K}(\mathbf{X},\mathbf{X}) =$

	x_{Aug}	x_{Muc}	x_{War}	x_{Min}	x_{Mos}
x_{Aug}	1.00	0.96	0.42	0.02	0.00
x_{Muc}	0.96	1.00	0.59	0.04	0.00
x_{War}	0.42	0.59	1.00	0.32	0.10
x_{Min}	0.02	0.04	0.32	1.00	0.80
x_{Mos}	0.00	0.00	0.10	0.80	1.00

Gaussian Process Regression – Covariance Function

covariance function



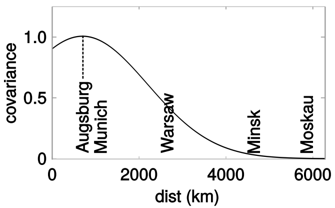
covariance matrix (Gram matrix)

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \begin{array}{c} x_{\text{Aug}} \\ x_{\text{Muc}} \\ x_{\text{War}} \\ x_{\text{Min}} \\ x_{\text{Mos}} \end{array} \overset{\begin{array}{ccccc} x_{\text{Aug}} & x_{\text{Muc}} & x_{\text{War}} & x_{\text{Min}} & x_{\text{Mos}} \end{array}}{\begin{bmatrix} 1.00 & 0.96 & 0.42 & 0.02 & 0.00 \\ 0.96 & 1.00 & 0.59 & 0.04 & 0.00 \\ 0.42 & 0.59 & 1.00 & 0.32 & 0.10 \\ 0.02 & 0.04 & 0.32 & 1.00 & 0.80 \\ 0.00 & 0.00 & 0.10 & 0.80 & 1.00 \end{bmatrix}}$$

- Remarks
  - Basis function has very specific interpretation: covariance
  - No temperature measurements **y** have been made yet
  - Prior: assume temperature is 0°C

Gaussian Process Regression – Covariance Function

covariance function



covariance matrix (Gram matrix)

$$\mathbf{K}(\mathbf{X},\mathbf{X}) = \begin{array}{c} \\ x_{\text{Aug}} \\ x_{\text{Muc}} \\ x_{\text{War}} \\ x_{\text{Min}} \\ x_{\text{Mos}} \end{array} \begin{array}{c} \begin{array}{ccccc} x_{\text{Aug}} & x_{\text{Muc}} & x_{\text{War}} & x_{\text{Min}} & x_{\text{Mos}} \end{array} \\ \begin{bmatrix} 1.00 & 0.96 & 0.42 & 0.02 & 0.00 \\ 0.96 & 1.00 & 0.59 & 0.04 & 0.00 \\ 0.42 & 0.59 & 1.00 & 0.32 & 0.10 \\ 0.02 & 0.04 & 0.32 & 1.00 & 0.80 \\ 0.00 & 0.00 & 0.10 & 0.80 & 1.00 \end{bmatrix} \end{array}$$

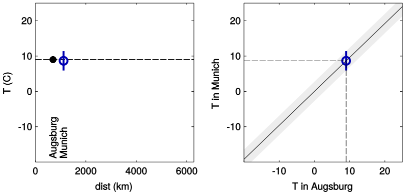
- Remarks
  - Basis function has very specific interpretation: covariance
  - No temperature measurements $\mathbf{y}$ have been made yet
  - Prior: assume temperature is 0°C

Question

Expected temperature in Munich, given $9^\circ C$ in Augsburg?

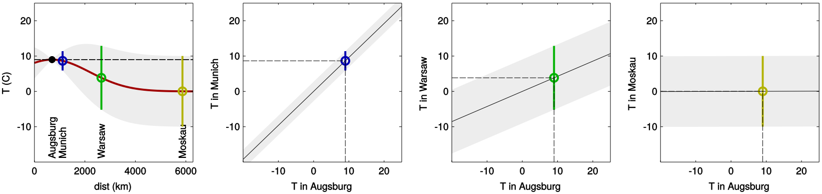
(condition on $T_{\text{Aug}} = 9$, i.e. $E\left[T_{\text{Muc}} \mid T_{\text{Aug}} = 9\right]$)

Gaussian Process Regression – Example



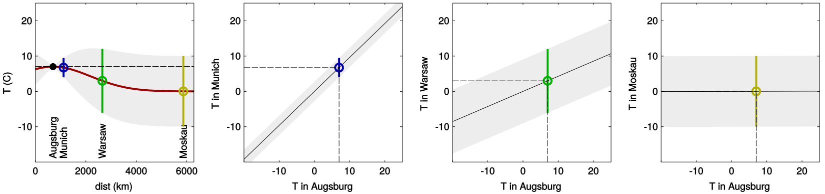
$$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$$

Gaussian Process Regression – Example



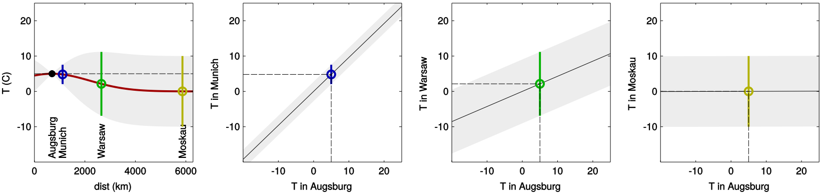
$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$ $k(x_{\text{War}}, x_{\text{Aug}}) = 0.42$ $k(x_{\text{Mos}}, x_{\text{Aug}}) = 0.00$

Gaussian Process Regression – Example



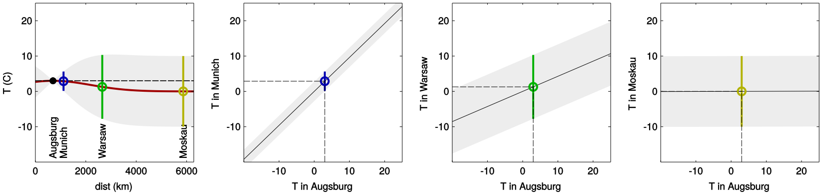
$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$ $k(x_{\text{War}}, x_{\text{Aug}}) = 0.42$ $k(x_{\text{Mos}}, x_{\text{Aug}}) = 0.00$

Gaussian Process Regression – Example



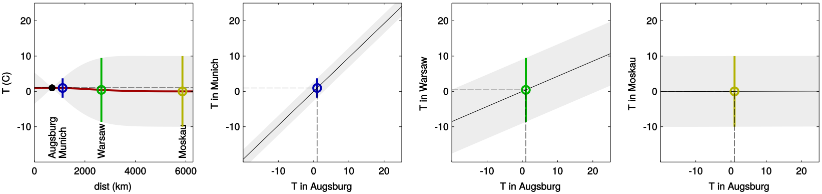
$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$ $\quad k(x_{\text{War}}, x_{\text{Aug}}) = 0.42$ $\quad k(x_{\text{Mos}}, x_{\text{Aug}}) = 0.00$

Gaussian Process Regression – Example



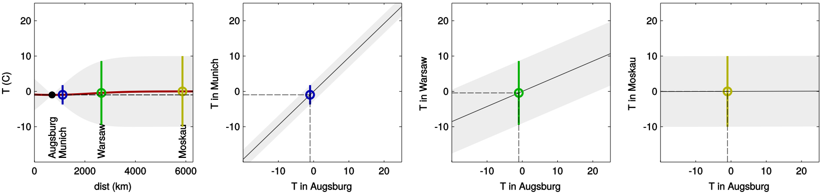
$k(x_{\mathbf{Muc}}, x_{\mathbf{Aug}}) = 0.96$ $k(x_{\mathbf{War}}, x_{\mathbf{Aug}}) = 0.42$ $k(x_{\mathbf{Mos}}, x_{\mathbf{Aug}}) = 0.00$

Gaussian Process Regression – Example



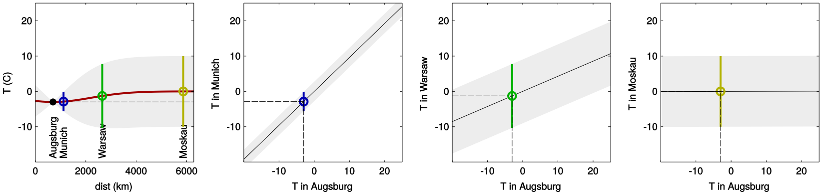
$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$ $k(x_{\text{War}}, x_{\text{Aug}}) = 0.42$ $k(x_{\text{Mos}}, x_{\text{Aug}}) = 0.00$

Gaussian Process Regression – Example



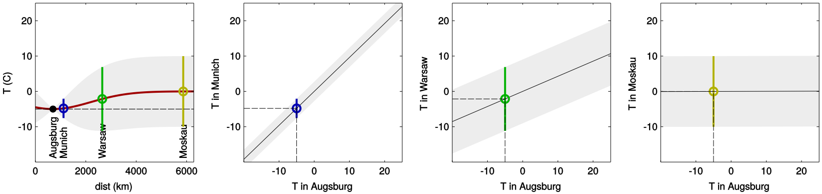
$k(x_{\mathbf{Muc}}, x_{\mathbf{Aug}}) = 0.96$ $k(x_{\mathbf{War}}, x_{\mathbf{Aug}}) = 0.42$ $k(x_{\mathbf{Mos}}, x_{\mathbf{Aug}}) = 0.00$

Gaussian Process Regression – Example



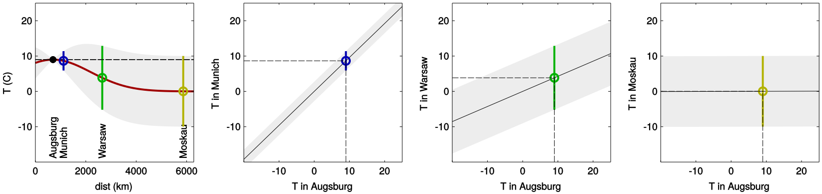
$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$ $k(x_{\text{War}}, x_{\text{Aug}}) = 0.42$ $k(x_{\text{Mos}}, x_{\text{Aug}}) = 0.00$

Gaussian Process Regression – Example



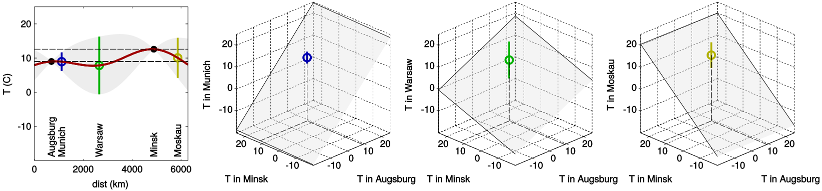
$k(x_{\mathbf{Muc}}, x_{\mathbf{Aug}}) = 0.96$ $\quad$ $k(x_{\mathbf{War}}, x_{\mathbf{Aug}}) = 0.42$ $\quad$ $k(x_{\mathbf{Mos}}, x_{\mathbf{Aug}}) = 0.00$

Gaussian Process Regression – Example



$k(x_{\text{Muc}}, x_{\text{Aug}}) = 0.96$ $k(x_{\text{War}}, x_{\text{Aug}}) = 0.42$ $k(x_{\text{Mos}}, x_{\text{Aug}}) = 0.00$

Gaussian Process Regression – Example

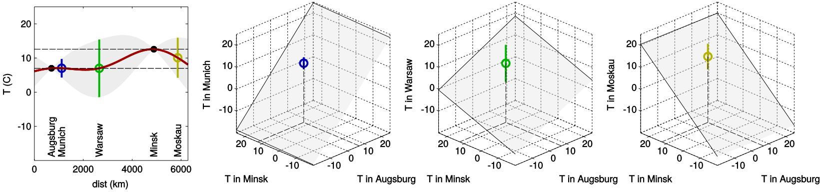


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example

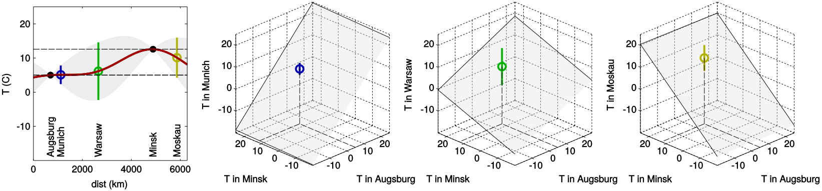


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example

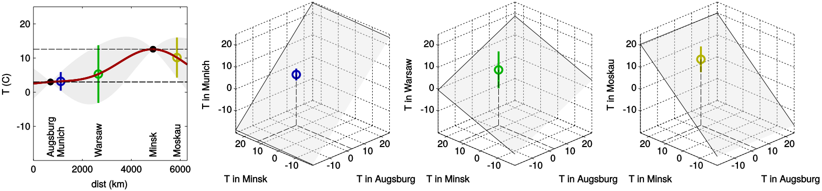


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example

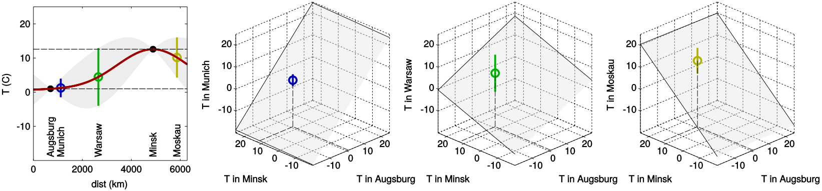


$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) =$ [0.96 0.04]

$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) =$ [0.42 0.32]

$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) =$ [0.00 0.8]

Gaussian Process Regression – Example

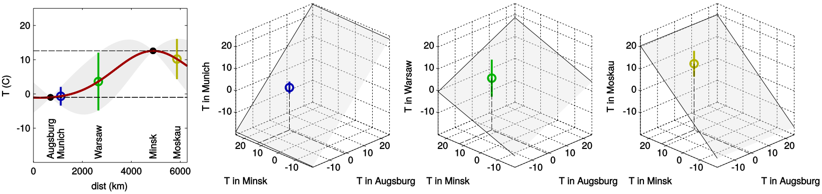


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example

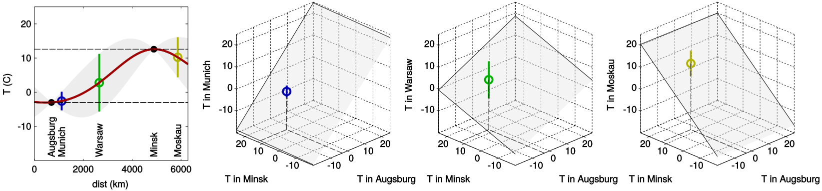


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example

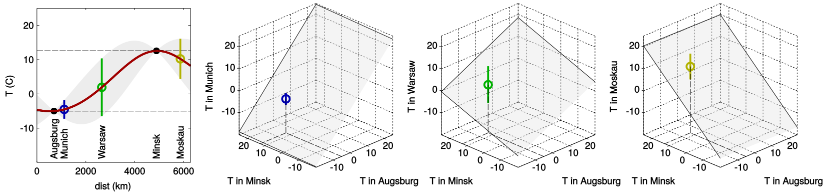


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example

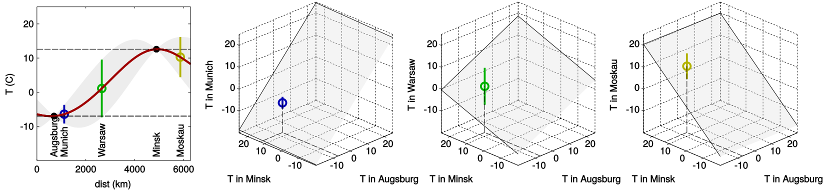


$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) =$ [0.96 0.04]

$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) =$ [0.42 0.32]

$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) =$ [0.00 0.8]

Gaussian Process Regression – Example

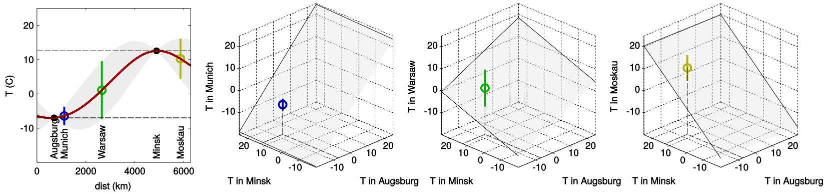


$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

Gaussian Process Regression – Example



$\mathbf{k}(x_{\mathbf{Muc}}, [x_{\mathbf{Aug}}\ x_{\mathbf{Min}}]) =$ [0.96 0.04]

$\mathbf{k}(x_{\mathbf{War}}, [x_{\mathbf{Aug}}\ x_{\mathbf{Min}}]) =$ [0.42 0.32]

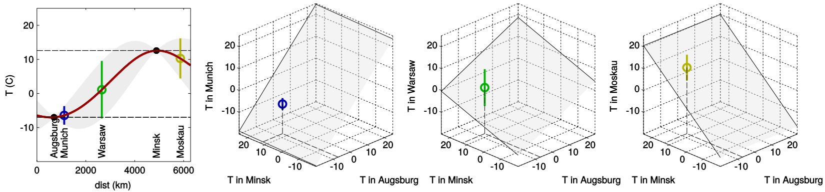
$\mathbf{k}(x_{\mathbf{Mos}}, [x_{\mathbf{Aug}}\ x_{\mathbf{Min}}]) =$ [0.00 0.8]

What are the plane slopes?

$$\bar{y}_q = \overbrace{\mathbf{k}(\mathbf{x}_q, \mathbf{X})}^{\text{see above}} \quad \underbrace{\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}} \tag{23}$$

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{matrix} & x_{\mathbf{Aug}} & x_{\mathbf{Min}} \\ x_{\mathbf{Aug}} & 1.00 & 0.02 \\ x_{\mathbf{Min}} & 0.02 & 1.00 \end{matrix}$$

Gaussian Process Regression – Example



$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$

$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$

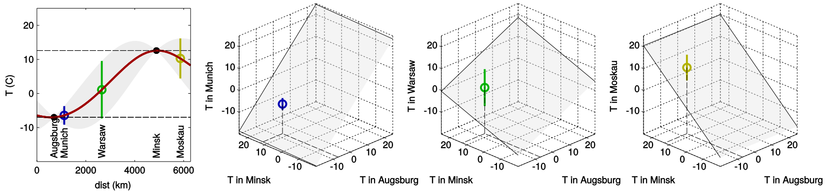
$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$

What are the plane slopes?

$$\bar{y}_q = \overbrace{\mathbf{k}(\mathbf{x}_q, \mathbf{X})}^{\text{see above}} \quad \underbrace{\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}}_{\text{Least squares!}} \qquad (23)$$

$$\mathbf{K}(\mathbf{X}, \mathbf{X}) = \begin{array}{c} \\ x_{\text{Aug}} \\ x_{\text{Min}} \end{array} \begin{array}{cc} x_{\text{Aug}} & x_{\text{Min}} \\ \end{array} \left[\begin{array}{cc} 1.00 & 0.02 \\ 0.02 & 1.00 \end{array}\right]$$

Gaussian Process Regression – Example



$$\mathbf{k}(x_{\text{Muc}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.96\ 0.04]$$

$$\mathbf{k}(x_{\text{War}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.42\ 0.32]$$

$$\mathbf{k}(x_{\text{Mos}}, [x_{\text{Aug}}\ x_{\text{Min}}]) = [0.00\ 0.8]$$

What are the plane slopes?

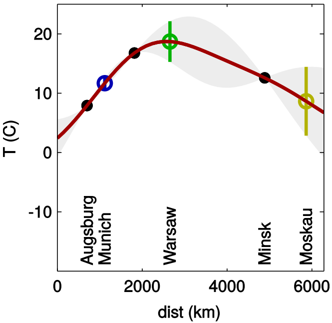
$$\bar{y}_q = \overbrace{\mathbf{k}(\mathbf{x}_q, \mathbf{X})}^{\text{see above}} \quad \underbrace{\mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}}_{\text{Least squares!}} \qquad (23)$$

Kernel Regression

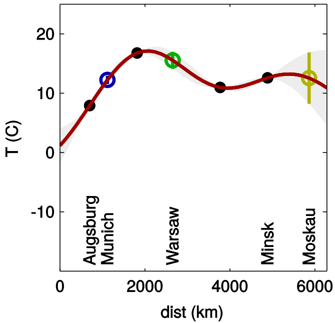
$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n \cdot k(\mathbf{x}, \mathbf{x}_n)$$

$$\mathbf{w}^* = \mathbf{K}(\mathbf{X}, \mathbf{X})^{-1}\mathbf{y}$$

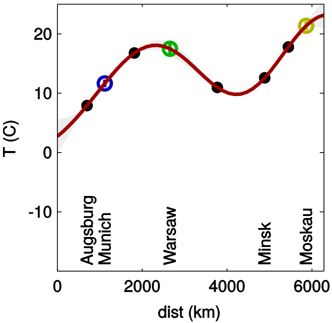
Gaussian Process Regression – Example



Gaussian Process Regression – Example

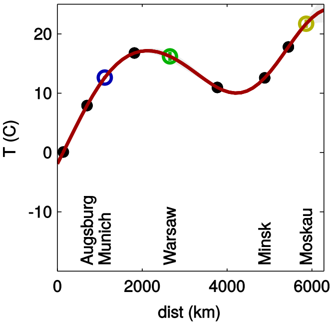


Gaussian Process Regression – Example



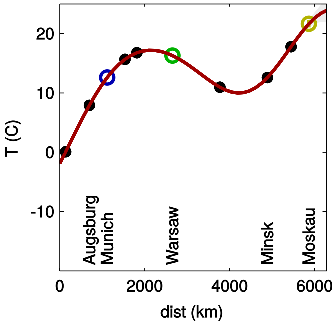
The more measurements become available,
the more certain we become

Gaussian Process Regression – Example



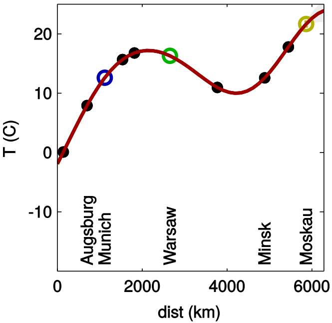
The more measurements become available,
the more certain we become

Gaussian Process Regression – Example



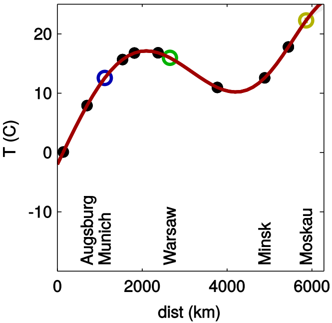
The more measurements become available,
the more certain we become

Gaussian Process Regression – Example



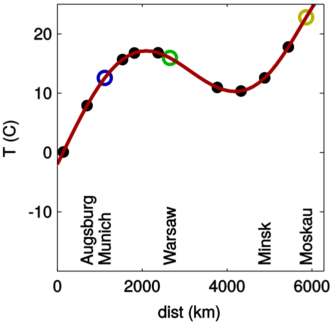
The more measurements become available,
the more certain we become

Gaussian Process Regression – Example



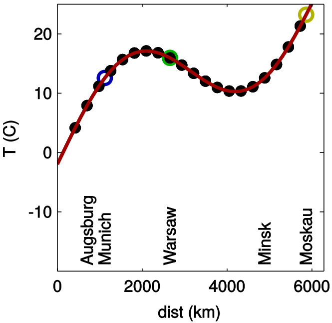
The more measurements become available,
the more certain we become

Gaussian Process Regression – Example



The more measurements become available,
the more certain we become

Gaussian Process Regression – Example



The more measurements become available,
the more certain we become