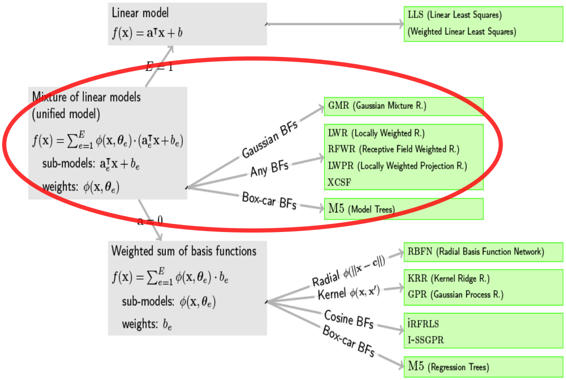
Regression

3. Locally Weighted Regression Methods

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud

Reminder: Outline of methods



- Multiple local and weighted least square regressions (shown with LWR)

Stulp, F. and Sigaud, O. (2015) Many regression algorithms, one unified model: A review. *Neural Networks*, 69:60–79

Locally Weighted Regression

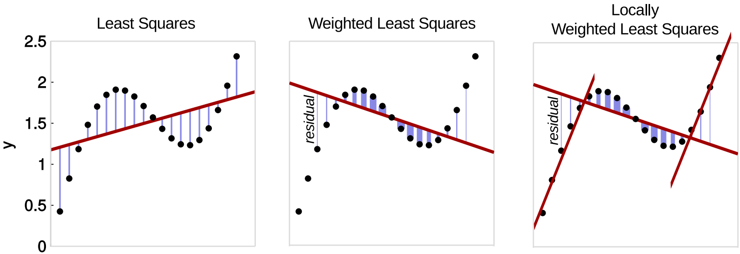
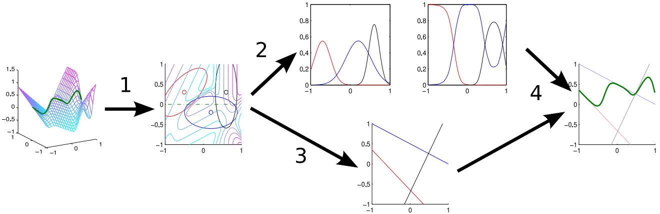


Figure: The thickness of the lines indicates the weights.

- General idea:
  - Split the function domain into linear parts
  - Give more weight to the centers
- Local linear models are tuned with Least Squares
- The importance of datapoints is represented by a Gaussian function

William S Cleveland and Susan J Devlin (1988) Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American statistical association*, 83(403):596–610.

Locally Weighted Regression: Processes



1. Define the split into regions (receptive fields): by hand, data-driven, evolutionary
2. Determine relative importance of domains
3. Find linear models in the regions
4. Combine all linear models

Atkeson, C. (1991) Using locally weighted regression for robot learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 958–963

Combining linear models

- There are E features, or receptive fields (RF)
- Each RF is defined as a Gaussian $\phi(\mathbf{x}, \boldsymbol{\theta}_i) = e^{-\frac{1}{2}(\mathbf{x}-\mu_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\mu_i)}$ with $\boldsymbol{\theta}_i = (\mu_i, \boldsymbol{\Sigma}_i)$
- Each RF tunes a local linear model

$$\Psi_i(\mathbf{x}) = \mathbf{a}_i^\intercal \mathbf{x} + b_i$$

- Gaussians tell you how much each RF contributes to the output

$$y = \frac{\sum_{i=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_i) \Psi_i(\mathbf{x})}{\sum_{i=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_i)}$$

Batch learning (1)

- Consider a batch of N $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{1 \leq i \leq N}$ data.
-
$$y = f(\mathbf{x}) = \frac{\sum_{i=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_i)\Psi_i(\mathbf{x})}{\sum_{i=1}^{E} \phi(\mathbf{x}, \boldsymbol{\theta}_i)}$$
with $\Psi_i(\mathbf{x}) = w(\mathbf{x})^\mathsf{T}\boldsymbol{\theta}_i$ and $w(\mathbf{x}) = (\mathbf{x}_1 \ \mathbf{x}_2 \cdots \mathbf{x}_d \ 1)^\mathsf{T}$.
- Each local model is computed using the following locally weighted error:

$$\begin{aligned}\epsilon_i(\boldsymbol{\theta}_i) &= \frac{1}{2N}\sum_{j=1}^{N} \phi(\mathbf{x}^{(j)}, \boldsymbol{\theta}_i)\left(y^{(j)} - \Psi_i(\mathbf{x}^{(j)})\right)^2 \\ &= \frac{1}{2N}\sum_{j=1}^{N} \phi(\mathbf{x}^{(j)}, \boldsymbol{\theta}_i)\left(y^{(j)} - w(\mathbf{x}^{(j)})^\mathsf{T}\boldsymbol{\theta}_i\right)^2.\end{aligned}$$

Batch learning (2)

- As with the least squares method, we try to cancel out the gradient:

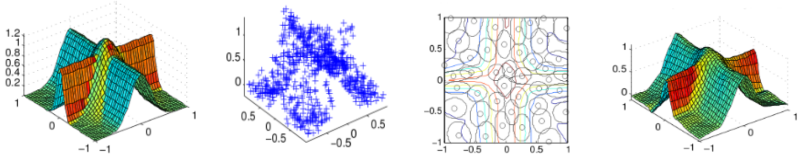
$$-\frac{1}{N}\sum_{j=1}^{N}\phi(\mathbf{x}^{(j)},\boldsymbol{\theta}_i)w(\mathbf{x}^{(j)})\left(y^{(j)}-w(\mathbf{x}^{(j)})^{\mathsf{T}}\boldsymbol{\theta}_i\right)=0.$$

- Therefore, we pose $\boldsymbol{\theta}_i = A_i^{\sharp} b_i$, with:

$$A_i = \sum_{j=1}^{N}\phi(\mathbf{x}^{(j)},\boldsymbol{\theta}_i)w(\mathbf{x}^{(j)})w(\mathbf{x}^{(j)})^{\mathsf{T}}$$

$$b_i = \sum_{j=1}^{N}\phi(\mathbf{x}^{(j)},\boldsymbol{\theta}_i)w(\mathbf{x}^{(j)})y^{(j)}.$$

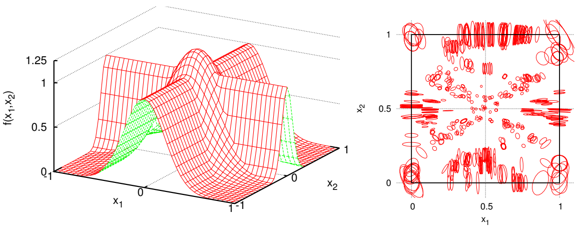
LWPR: general goal



- Non-linear function approximation in very large spaces
- Using PLS to project linear models in a smaller space
- Adds receptive fields around non covered datapoints, moves the shape, cannot remove them
- Good along local trajectories

Schaal, S., Atkeson, C. G., and Vijayakumar, S. (2002). Scalable techniques from nonparametric statistics for real time robot learning. *Applied Intelligence*, 17(1):49–60.

XCSF: overview

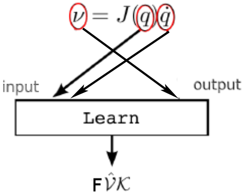


- XCSF is a Learning Classifier System [Holland(1975)]
- Linear models weighted by Gaussian functions (similar to LWPR)
- Linear models are updated using RLS
- Gaussian functions adaptation: Σ_i^{-1} and c_i are updated using a GA
- Condensation: reduce population to generalize better

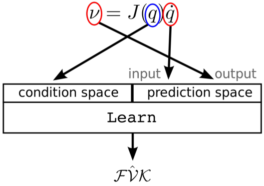
Wilson, S. W. (2001) Function approximation with a classifier system. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 974–981, San Francisco, California, USA. Morgan Kaufmann

XCSF: key feature

- Distinguish the space of linear models (prediction space) and the space of weights (condition space)
- Forward kinematics: $\dot{\boldsymbol{\xi}} = F_\theta(\mathbf{q}, \dot{\mathbf{q}})$ $\dot{\boldsymbol{\xi}} = J(\mathbf{q})\,\dot{\mathbf{q}}$
- Forward dynamics: $\ddot{\mathbf{q}} = G_\theta(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\Gamma})$ $\ddot{\mathbf{q}} = A(\mathbf{q})^{-1}(\boldsymbol{\Gamma} - \boldsymbol{n}(\mathbf{q}, \dot{\mathbf{q}}))$



Forward kinematics with LWPR

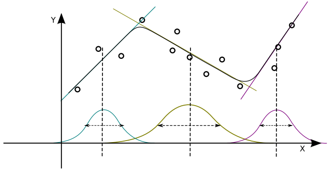


Forward kinematics with XCSF

- Example: learning forward kinematics $(\mathbf{x} = < \mathbf{q}, \dot{\mathbf{q}} >)$
- LWPR: $\hat{f}(\mathbf{x}) = \sum_{i=1}^{E} \phi((\mathbf{q}, \dot{\mathbf{q}}), \boldsymbol{\theta}_i) \cdot (b_i + \mathbf{a}_i^\intercal(\mathbf{q}, \dot{\mathbf{q}}))$
- XCSF: $f(\mathbf{x}) = \sum_{i=1}^{E} \phi(\mathbf{q}, \boldsymbol{\theta}_i) \cdot (b_i + \mathbf{a}_i^\intercal \dot{\mathbf{q}})$

Butz, M., Pedersen, G., and Stalph, P. (2009) Learning sensorimotor control structures with XCSF: redundancy exploitation and dynamic control. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 1171–1178. ACM

GMR



$$y = \sum_{k=1}^{K} h_k(\mathbf{x})(\mu_{k,Y} + \Sigma_{k,YX}\Sigma_{k,Y}^{-1}(\mathbf{x} - \mu_{k,X}))$$

With

$$\mu_k = [\mu_{k,X}^T, \mu_{k,Y}^T]^T \text{ and } \mathbf{\Sigma}_k = \begin{pmatrix} \Sigma_{k,X} & \Sigma_{k,XY} \\ \Sigma_{k,YX} & \Sigma_{k,YX} \end{pmatrix}$$

- From input-output manifold to input-output function
- Same representation as the others, using $\boldsymbol{\theta}^\intercal = \Sigma_{i,YX}\Sigma_{i,Y}^{-1}$ and $b_i = \mu_{i,Y} - \Sigma_{i,YX}\Sigma_{i,X}^{-1}\mu_{i,X}$
- We get

$$\tilde{y} = \sum_{i=1}^{E} \frac{\pi_i \phi(\mathbf{x}, \boldsymbol{\theta}_i)}{\sum_{l=1}^{E} \pi_l \phi(\mathbf{x}, \boldsymbol{\theta}_l)} (\boldsymbol{\theta}^\intercal \mathbf{x} + b_i),$$

- Same as usual + scaling with the priors $\pi_i \rightarrow \pi_i = 1$ in standard model.
- Incorporates Bayesian variance estimation $\rightarrow$ The richest representation

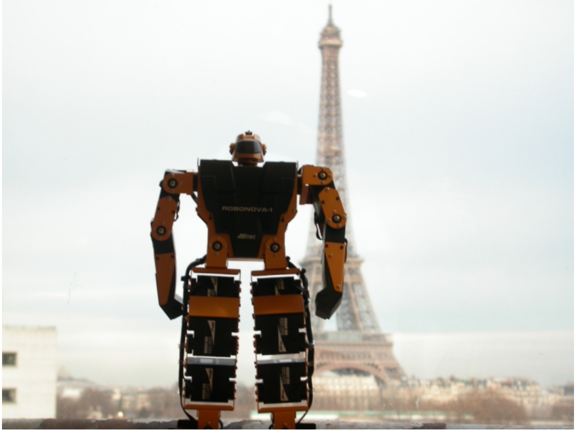
Hersch, M., Guenter, F., Calinon, S., & Billard, A. (2008) "Dynamical system modulation for robot learning via kinesthetic demonstrations." *IEEE Transactions on Robotics*, 24(6), 1463–1467

LWR methods: main features

Algo	LWR	LWPR	GMR	XCSF
Number of RFs	fixed	growing	fixed	adaptive
Position of RFs	fixed	fixed	adaptive	adaptive
Size of RFs	fixed	adaptive	adaptive	adaptive

- The main differences are in meta-parameter tuning
- Fewer hyperparameters is better, but less flexibility

Any question?



Send mail to: `Olivier.Sigaud@upmc.fr`

C.G. Atkeson.
Using locally weighted regression for robot learning.
In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pp. 958–963, Apr 1991.

M.V. Butz, G.K.M. Pedersen, and P.O. Stalph.
Learning sensorimotor control structures with XCSF: redundancy exploitation and dynamic control.
In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 1171–1178. ACM, 2009.

William S Cleveland and Susan J Devlin.
Locally weighted regression: an approach to regression analysis by local fitting.
Journal of the American statistical association, 83(403):596–610, 1988.

Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude Billard.
Dynamical system modulation for robot learning via kinesthetic demonstrations.
IEEE Transactions on Robotics, 24(6):1463–1467, 2008.

J. H. Holland.
Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.
University of Michigan Press, Ann Arbor, MI, 1975.

S. Schaal, C. G. Atkeson, and S. Vijayakumar.
Scalable techniques from nonparametric statistics for real time robot learning.
Applied Intelligence, 17(1):49–60, 2002.

Freek Stulp and Olivier Sigaud.
Many regression algorithms, one unified model: A review.
Neural Networks, 69:60–79, 2015.

S. W. Wilson.
Function approximation with a classifier system.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 974–981, San Francisco, California, USA, 7-11 July 2001. Morgan Kaufmann.

ISBN 1-55860-774-9.