Regression with Linear Factored Functions

Wendelin Böhmer <wendelin@ni.tu-berlin.de>
Klaus Obermayer <oby@ni.tu-berlin.de>
Neural Information Processing Group, Technische Universität Berlin

DFG Schwerpunktprogramm 1527 Autonomous Learning

Abstract

Many applications that use empirically estimated functions face a curse of dimensionality, because integrals over most function classes must be approximated by sampling. This paper introduces a novel regression-algorithm that learns linear factored functions (LFF). This class of functions has structural properties that allow to analytically solve certain integrals and to calculate point-wise products. Applications like belief propagation and reinforcement learning can exploit these properties to break the curse and speed up computation. We derive a regularized greedy optimization scheme, that learns factored basis functions during training. The novel regression algorithm performs competitively to Gaussian processes on benchmark tasks, and the learned LFF functions are with 4-9 factored basis functions on average very compact.

Linear Factored Functions (LFF)

$$f(\vec{x}) := \vec{a}^\top \vec{\psi}(\vec{x}) := \vec{a}^\top \Big[\prod_{k=1}^{d} \vec{\psi}^k(x_k) \Big] := \sum_{i=1}^{m} a_i \prod_{k=1}^{d} \sum_{j=0}^{m_k} B^k_{ji} \phi^k_j(x_k)$$

factorizing inner products

$$\vartheta(d\vec{x}) = \prod_{k=1}^{d} \vartheta(dx_k)$$

$$\langle \psi_i, \psi_j \rangle_\vartheta = \prod_{k=1}^{d} \langle \psi^k_i, \psi^k_j \rangle_{\vartheta^k}$$

Fourier cosine base

$$\phi^k_j(x_k) := \sqrt{2} \cos(j\,\pi\, x_k)$$

$$\phi^k_0(x_k) := 1\,, \quad \langle \phi^k_i, \phi^k_j \rangle_{\vartheta^k} = \delta_{ij}$$

- In the limit equivalent to the space of *square-integrable functions.*
- Allows analytical **marginalization** and **point-wise multiplication**.

Regularization

- Regression with **virtual** or **noisy samples**

$$\inf_f \mathcal{C}[f|\mu,\chi] \quad := \quad \inf_f \iint \xi(d\vec{x})\, \chi(d\vec{z}|\vec{x}) \Big(f(\vec{z}) - \mu(\vec{x}) \Big)^2$$

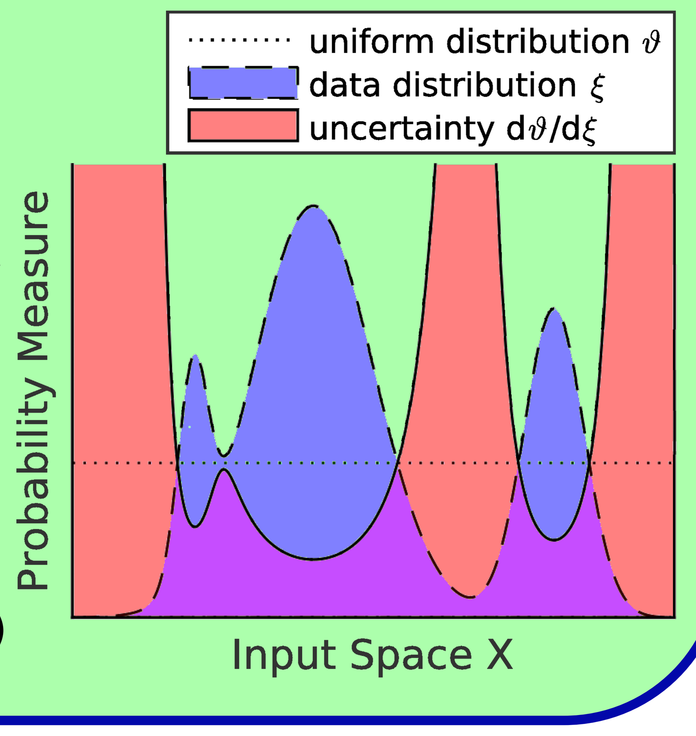
- Gaussian **noise assumption** with **uncertainty** scaled variance

$$\int \chi(d\vec{z}|\vec{x})\, (\vec{z}-\vec{x}) = \vec{0}\,, \quad \int \chi(d\vec{z}|\vec{x})\, (\vec{z}-\vec{x})(\vec{z}-\vec{x})^\top = \tfrac{d\vartheta}{d\xi}(\vec{x}) \cdot \mathbf{\Sigma}\,, \quad \forall \vec{x} \in \mathcal{X}$$

- First order **Taylor approximation**

$$\tilde{\mathcal{C}}[f] \quad := \quad \underbrace{\|f-\mu\|^2_\xi}_{\text{least-squares}} + \sum_{k=1}^{d} \sigma^2_k \underbrace{\big\|\tfrac{\partial}{\partial x_k} f\big\|^2_\vartheta}_{\text{regularization}}$$

- Regularization enforces **smoothness**



- Over-fitting in rarely sampled regions
- Under-fitting in often sampled regions
- **Uncertainty measure** scales variance
- Inverse PDF (Radon-Nikodym derivative)

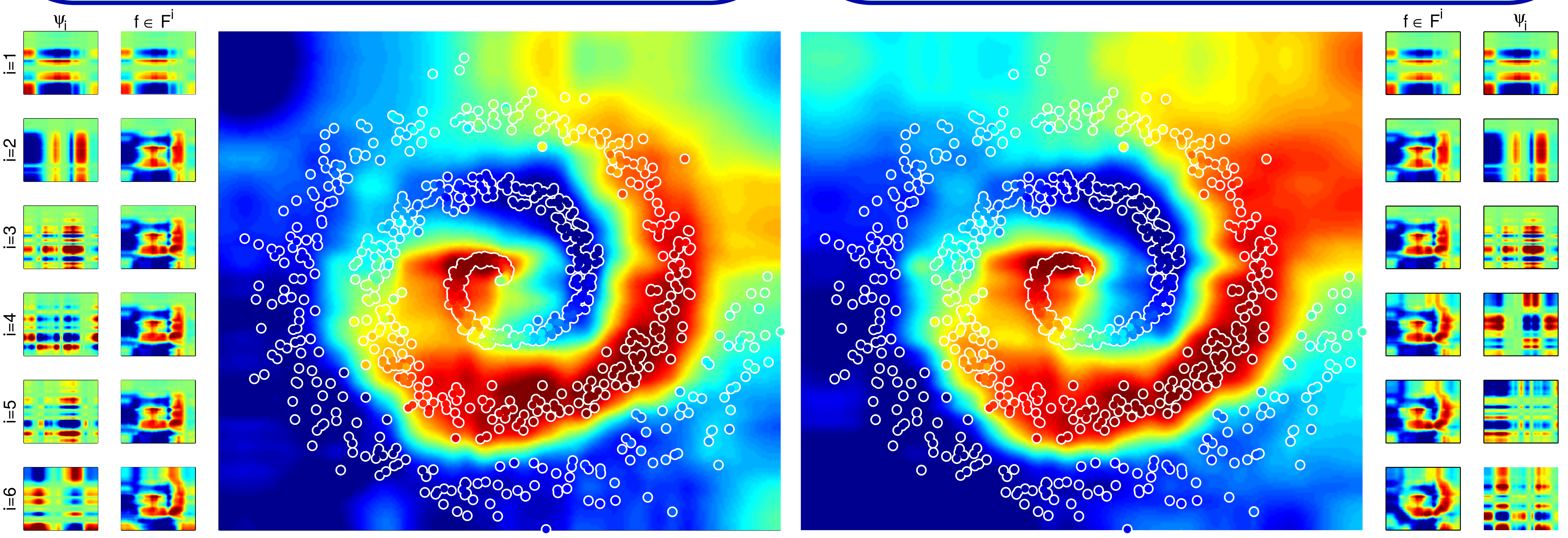
Optimization

- Cost function $\tilde{\mathcal{C}}[f]$ for LFF $f \in \mathcal{F}^m$ **non-convex**
- Learn **one** factored basis function $g \in \mathcal{F}$ **at a time**

$$\inf_{g \in \mathcal{F}} \tilde{\mathcal{C}}[f+g] \qquad \text{s.t.} \quad \|g^k\|_{\vartheta^k} = 1\,, \quad \forall k \in \{1,\dots,d\}$$

- Analytical solution for one factor function $g^k : \mathcal{X}_k \to \mathbb{R}$
- Repeat with random dimensions *k* until convergence
- Greedy LFF regression algorithm:
  - "inner loop" optimizes one basis function *g* at a time
  - "outer loop" learns coefficients a_i with *ordinary least squares* (OLS)

while new factored basis function can improve solution **do**
  initialize new basis function *g* as constant function
  while optimization improves cost in Equation 6 **do**
    for random input dimension k **do**
      calculate optimal solution for g^k without changing other g^l
    end for
  end while // new basis function *g* has converged
  add g to set of factored basis functions and solve OLS
end while // regression has converged



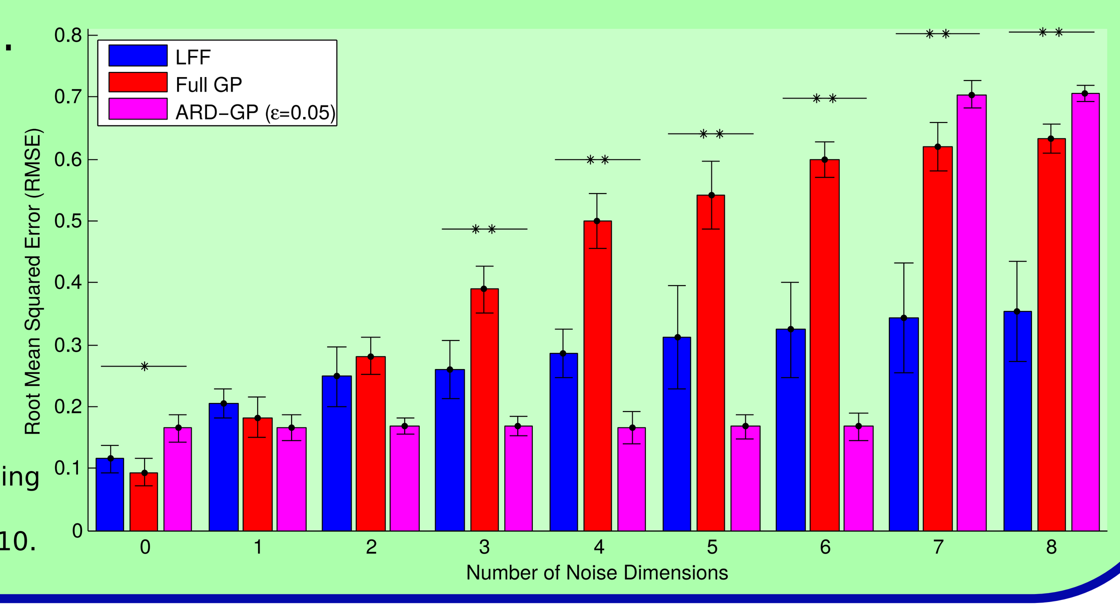
Evaluation: Spiral Toy Example

- Labels from sinus
- n = 1000 2D-spiral samples
- Not easily factorizable
- 10-fold crossvalidation

$$\vec{x}_t = 6\tfrac{t}{n} \begin{bmatrix} \cos(6\frac{t}{n}\pi) \\ \sin(6\frac{t}{n}\pi) \end{bmatrix} + \mathcal{N}(\vec{0}, \tfrac{t^2}{4n^2}\mathrm{I})$$

$$y_t = \sin\big(4\tfrac{t}{n}\pi\big)\,, \quad \forall t \in \{1,\dots,1000\}$$

- Additional noise dim. (normal distributed)
- No information
- Full RBF-GP does not generalize well
- ARD-GP adapts RBF kernel parameters

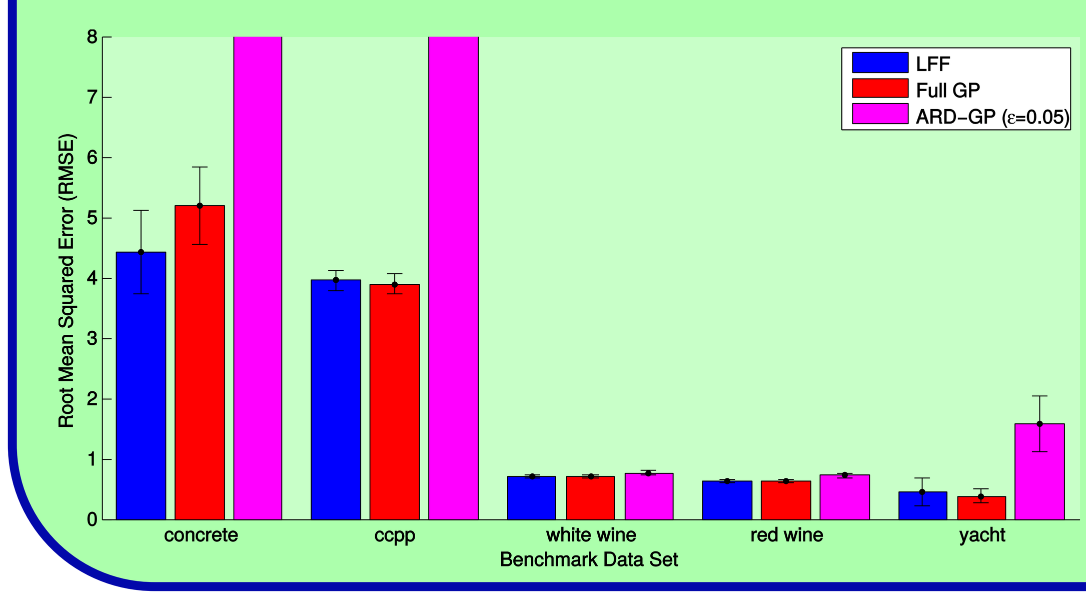


C.E. Rasmussen and H. Nickisch: Gaussian Processes for Machine Learning (GPML) Toolbox. *Journal of Machine Learning Research*, 11:3011-3015, 2010.

Evaluation: UCI Repository

DATA SET	*d*	*n*	# SV	RMSE LFF	RMSE GP	*m* LFF	*h* LFF	*h* GP
Concrete	8	1030	927	4.429 ± 0.69	5.196 ± 0.64	4.2 ± 0.8	3.00	0.05
CCPP	4	9568	2000	3.957 ± 0.17	3.888 ± 0.17	8.8 ± 2.0	1.96	1.14
White Wine	11	4898	2000	0.707 ± 0.02	0.708 ± 0.03	4.2 ± 0.4	4.21	0.69
Red Wine	11	1599	1440	0.632 ± 0.03	0.625 ± 0.03	4.7 ± 0.7	3.25	0.13
Yacht	6	308	278	0.446 ± 0.23	0.383 ± 0.11	4.2 ± 0.6	0.43	0.005

n samples with ***d*** dimensions, learned by ***m*** LFF bases in ***h*** hrs.



Concrete compression strength
I-C. Yeh: Modeling the stength of high performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797-1808, 1998.

Combined cycle power plant
P. Tüfekci: Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electical Power & Energy Systems*, 60:126-140, 2014.

Wine quality (white and red)
P. Cortez et al.: Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47 (4):547-553, 2009.

Yacht hydrodynamics
J. Gettirsma et al.: Geometry, resistance and stability of the delft systematic yacht hull series. *International Shipbuilding Progress*, 28:276-297, 1981.