Gaussian Processes - Part II
Kernel Algebra

Philipp Hennig

MLSS 2013
30 August 2013

Max Planck Institute for Intelligent Systems
Department of Empirical Inference
Tübingen, Germany

Reminder: Yesterday

Gaussians are the linear algebra of inference

- Gaussians are closed under
    - linear projection / marginalization / sum rule
    - linear restriction / conditioning / product rule

⇒ they provide the linear algebra of inference

- combine with nonlinear features ϕ, get nonlinear regression
- in fact, number of features can be infinite

⇒ (nonparametric) Gaussian process regression

A few open questions

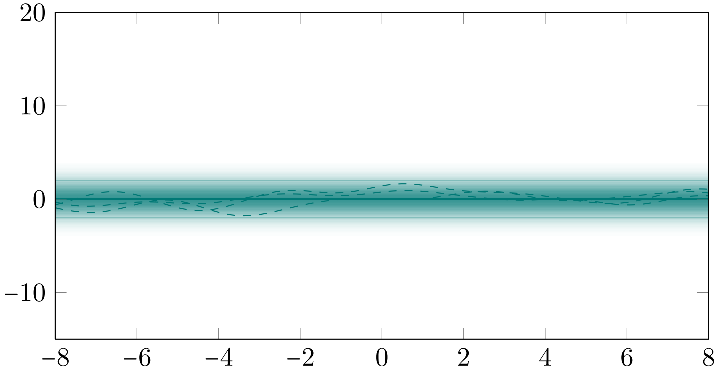
- so what are kernels? What is the set of kernels?
- how should we design GP models?s
- how powerful are those models?s

Scaling Outputs

```
k = @(a,b)(1.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
```

$$v^\top k v \geq 0 \ \forall v \qquad \Rightarrow \qquad v^\top \theta^2 k v = \theta^2 v^\top k v \geq 0 \ \forall v$$

$$p(f) = \mathcal{GP}(f; \mu, k) \qquad \Rightarrow \qquad \mathrm{var}[f(x)] = \theta^2 k(x,x) \overset{w.l.o.g.}{=} \theta^2$$

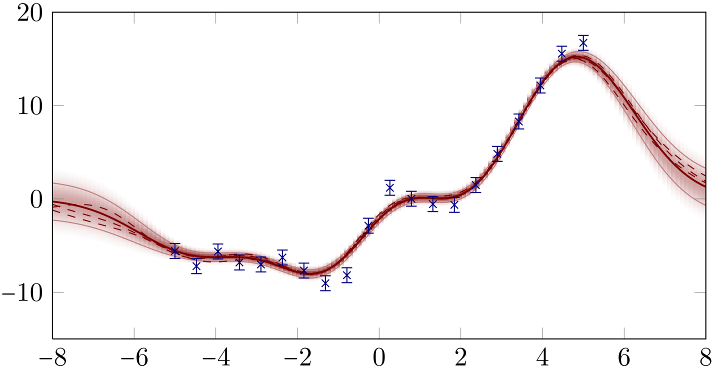


Scaling Outputs

```
k = @(a,b)(1.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
```

$$v^\top k v \geq 0 \; \forall v \qquad \Rightarrow \qquad v^\top \theta^2 k v = \theta^2 v^\top k v \geq 0 \; \forall v$$

$$p(f) = \mathcal{GP}(f; \mu, k) \qquad \Rightarrow \qquad \operatorname{var}[f(x)] = \theta^2 k(x, x) \overset{w.l.o.g.}{=} \theta^2$$

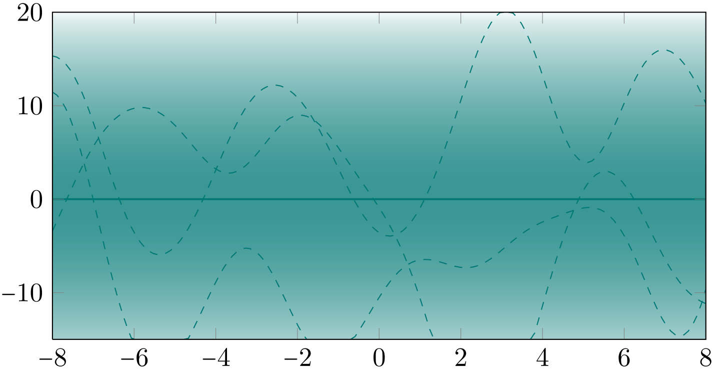


Scaling Outputs

```
k = @(a,b)(10.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
```

$$v^\top k v \geq 0 \;\forall v \quad\Rightarrow\quad v^\top \theta^2 k v = \theta^2 v^\top k v \geq 0 \;\forall v$$

$$p(f) = \mathcal{GP}(f; \mu, k) \quad\Rightarrow\quad \operatorname{var}[f(x)] = \theta^2 k(x,x) \overset{w.l.o.g.}{=} \theta^2$$

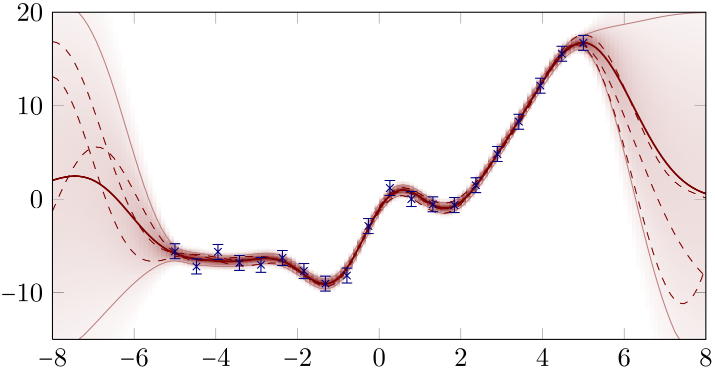


Scaling Outputs

```
k = @(a,b)(10.^2 * exp(-(bsxfun(@minus,a./2,b’./2)).^2));
```

$$v^\top k v \geq 0\ \forall v \qquad \Rightarrow \qquad v^\top \theta^2 k v = \theta^2 v^\top k v \geq 0\ \forall v$$

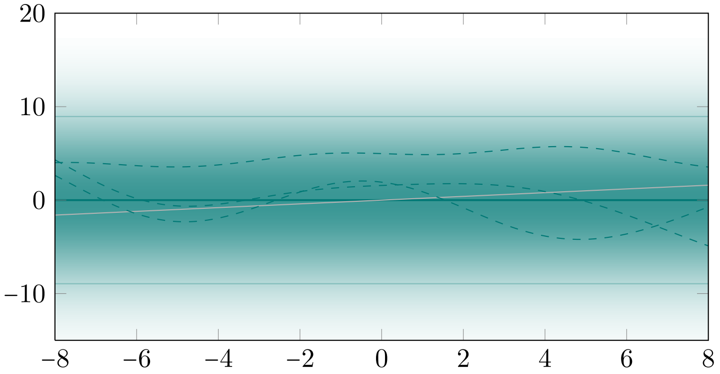
$$p(f) = \mathcal{GP}(f;\mu,k) \qquad \Rightarrow \qquad \mathrm{var}[f(x)] = \theta^2 k(x,x) \overset{w.l.o.g.}{=} \theta^2$$



Scaling Inputs

```
kSE = @(a,b)(exp(-(bsxfun(@minus,a,b')).^2)); phi = @(a)(a/5);
k   = @(a,b)(20 * kSE(phi(a),phi(b)));
```

$$k(a,b) = \sum\!\!\!\!\!\!\int_{\ell} \eta_\ell(a)\eta_\ell(b)^\intercal \qquad \Rightarrow \qquad k(\phi(a),\phi(b)) = \sum\!\!\!\!\!\!\int_{\ell} \eta_\ell(\phi(a))\eta_\ell(\phi(b))^\intercal$$

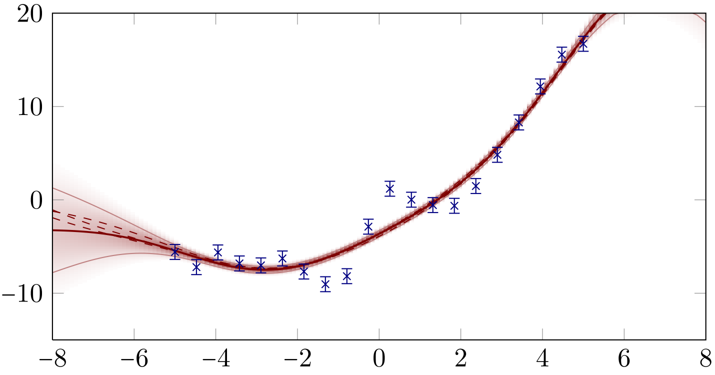


- $k(a,b)$ is pos. semidef. ⇒ $k(\phi(a),\phi(b))$ is pos. semidef.

Scaling Inputs

```
kSE = @(a,b)(exp(-(bsxfun(@minus,a,b')).^2)); phi = @(a)(a/5);
k   = @(a,b)(20 * kSE(phi(a),phi(b)));
```

$$k(a,b) = \sum\!\!\!\!\int_\ell \eta_\ell(a)\eta_\ell(b)^\intercal \qquad \Rightarrow \qquad k(\phi(a),\phi(b)) = \sum\!\!\!\!\int_\ell \eta_\ell(\phi(a))\eta_\ell(\phi(b))^\intercal$$

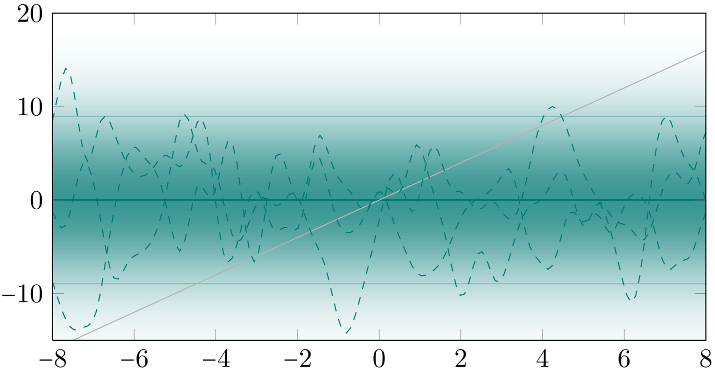


- $k(a,b)$ is pos. semidef. ⇒ $k(\phi(a),\phi(b))$ is pos. semidef.

Scaling Inputs

```
kSE = @(a,b)(exp(-(bsxfun(@minus,a,b’)).^2)); phi = @(a)(a*2);
k   = @(a,b)(20 * kSE(phi(a),phi(b)));
```

$$k(a,b) = \sum\!\!\!\!\!\!\int_\ell \eta_\ell(a)\eta_\ell(b)^\intercal \qquad \Rightarrow \qquad k(\phi(a),\phi(b)) = \sum\!\!\!\!\!\!\int_\ell \eta_\ell(\phi(a))\eta_\ell(\phi(b))^\intercal$$

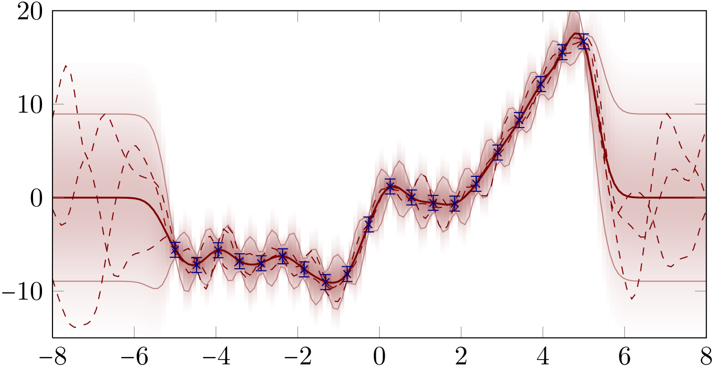


- $k(a,b)$ is pos. semidef. ⇒ $k(\phi(a),\phi(b))$ is pos. semidef.

Scaling Inputs

```
kSE = @(a,b)(exp(-(bsxfun(@minus,a,b')).^2)); phi = @(a)(a*2);
k   = @(a,b)(20 * kSE(phi(a),phi(b)));
```

$$k(a,b) = \sum_\ell \eta_\ell(a)\eta_\ell(b)^\intercal \qquad \Rightarrow \qquad k(\phi(a),\phi(b)) = \sum_\ell \eta_\ell(\phi(a))\eta_\ell(\phi(b))^\intercal$$

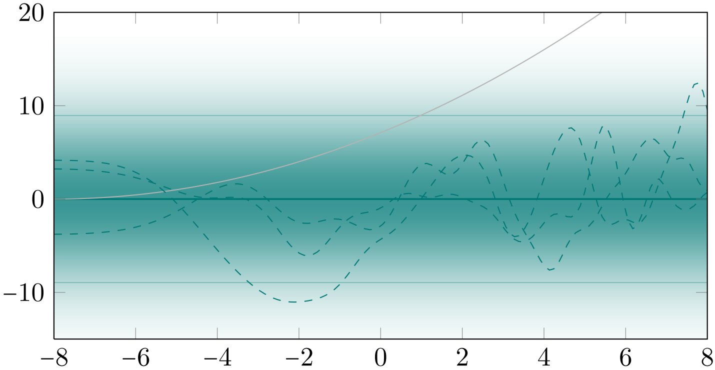


- $k(a,b)$ is pos. semidef. ⇒ $k(\phi(a),\phi(b))$ is pos. semidef.

Scaling Inputs

```
kSE = @(a,b)(exp(-(bsxfun(@minus,a,b')).^2)); phi = @(a)(((a+9)./5).^2);
k   = @(a,b)(20 * kSE(phi(a),phi(b)));
```

$$k(a,b) = \sum\!\!\!\!\int_\ell \eta_\ell(a)\eta_\ell(b)^\intercal \qquad \Rightarrow \qquad k(\phi(a),\phi(b)) = \sum\!\!\!\!\int_\ell \eta_\ell(\phi(a))\eta_\ell(\phi(b))^\intercal$$

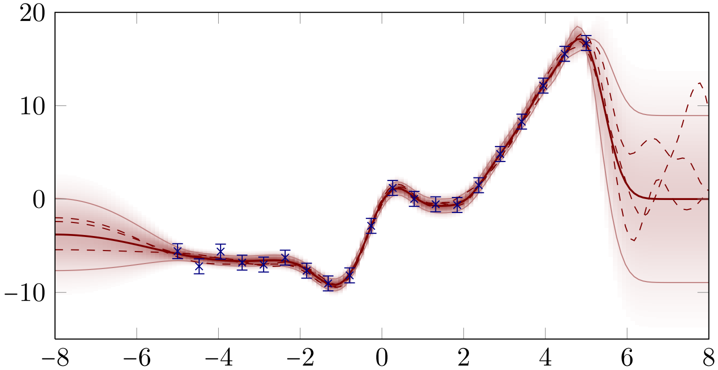


Caution: This can have unintended consequences is ϕ is not monotonic (long range interactions!)

Scaling Inputs

```
kSE = @(a,b)(exp(-(bsxfun(@minus,a,b’)).^2)); phi = @(a)(((a+9)./5).^2);
k   = @(a,b)(20 * kSE(phi(a),phi(b)));
```

$$k(a,b) = \sum\!\!\!\!\!\!\int_\ell \eta_\ell(a)\eta_\ell(b)^\intercal \qquad \Rightarrow \qquad k(\phi(a),\phi(b)) = \sum\!\!\!\!\!\!\int_\ell \eta_\ell(\phi(a))\eta_\ell(\phi(b))^\intercal$$

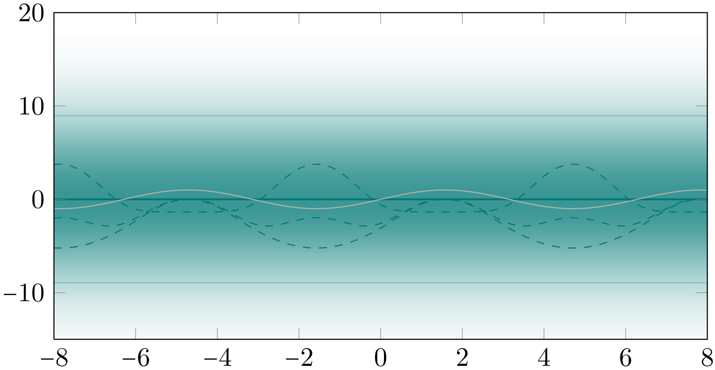


Caution: This can have unintended consequences is ϕ is not monotonic (long range interactions!)

Scaling Inputs – Example: periodic functions

D.J.C. MacKay, 1998

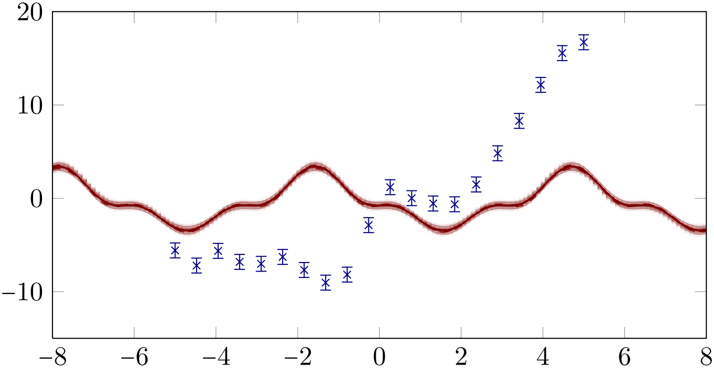
```
phi = @(a)(sin(a)); kSE = @(a,b)(20 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
k   = @(a,b)(kSE(phi(a),phi(b)));
```



Scaling Inputs – Example: periodic functions

D.J.C. MacKay, 1998

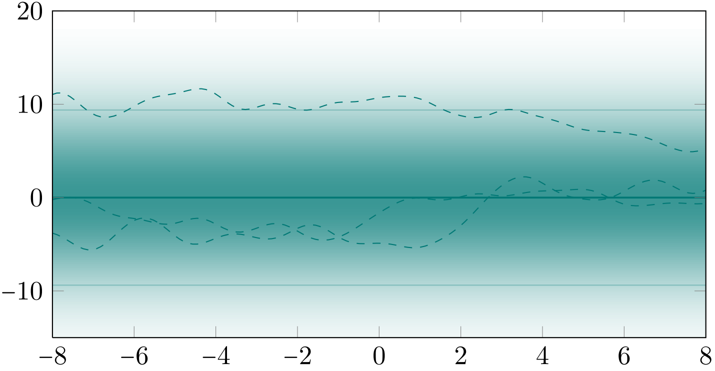
```
phi = @(a)(sin(a)); kSE = @(a,b)(20 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
k   = @(a,b)(kSE(phi(a),phi(b)));
```



Sums of Kernels are Kernels

```
k1 = @(a,b)(4.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2 ./ 10.^2));
k2 = @(a,b)(1.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2 ./ 0.5^2));
k  = @(a,b)(k1(a,b) + k2(a,b));
```

$$v^\top (k^1_{XX} + k^2_{XX})v = v^\top k^1_{XX} v + v^\top k^2_{XX} v \geq 0$$

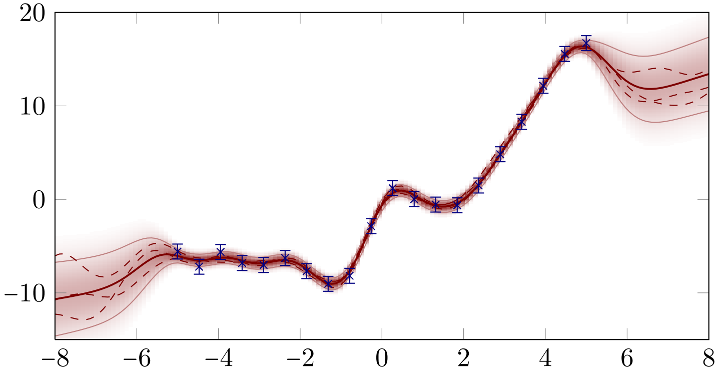


Intuition: similarity under k^1 OR k^2.

Sums of Kernels are Kernels

```
k1 = @(a,b)(4.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2 ./ 10.^2));
k2 = @(a,b)(1.^2 * exp(-(bsxfun(@minus,a./2,b'./2)).^2 ./ 0.5^2));
k  = @(a,b)(k1(a,b) + k2(a,b));
```

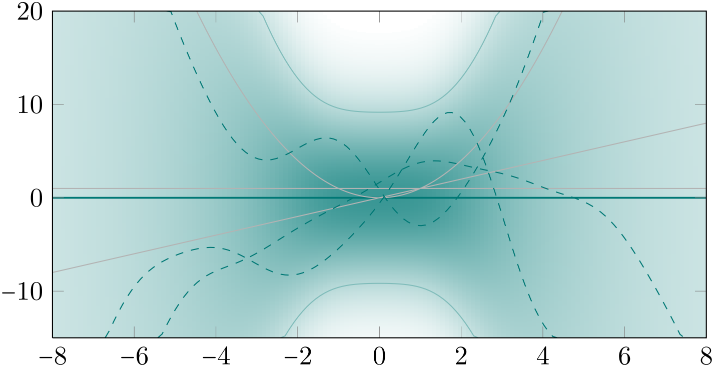
$$v^{\top}(k^1_{XX} + k^2_{XX})v = v^{\top}k^1_{XX}v + v^{\top}k^2_{XX}v \geq 0$$



Intuition: similarity under k^1 OR k^2.

Sums of Kernel and Parametric Features

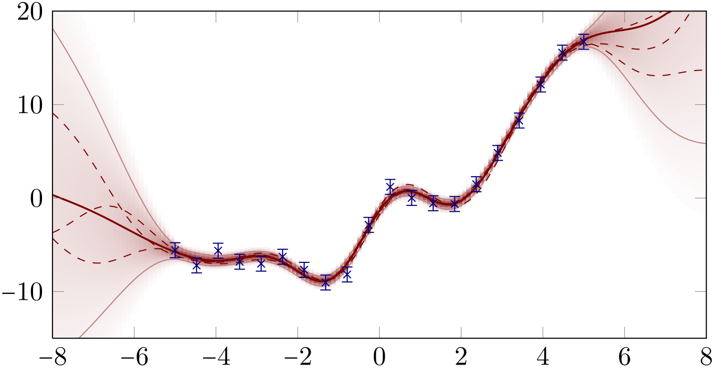
```
phi = @(a)(bsxfun(@power,a,[0:2]));
k   = @(a,b)(20 * exp(-(bsxfun(@minus,a./2,b'./2)).^2) + phi(a)*phi(b)');
```



see Rasmussen & Williams, §2.7 for an efficient implementation

Sums of Kernel and Parametric Features

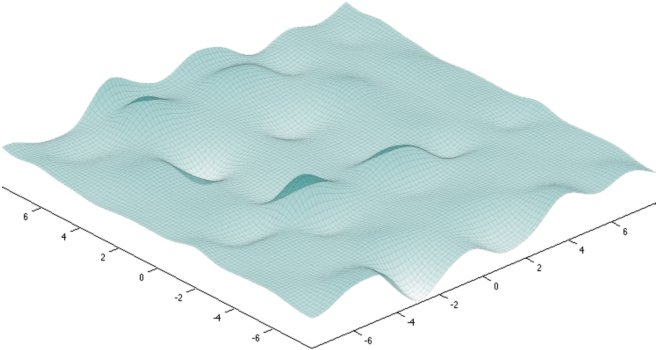
```
phi = @(a)(bsxfun(@power,a,[0:2]));
k   = @(a,b)(20 * exp(-(bsxfun(@minus,a./2,b'./2)).^2) + phi(a)*phi(b)');
```



see Rasmussen & Williams, §2.7 for an efficient implementation

Multiple Inputs

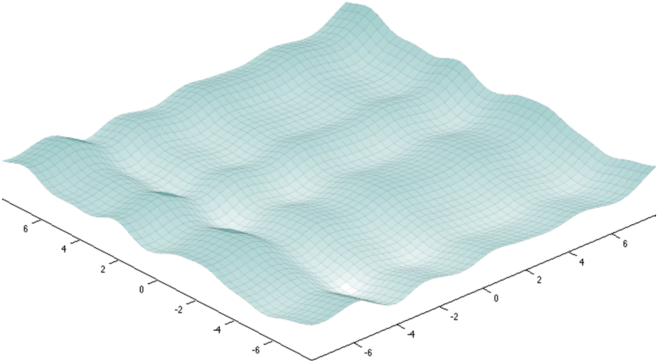
just a quick reminder



Additive Models

```
k = @(a,b)(kSE(a(:,1),b(:,1)) + kSE(a(:,2),b(:,2)));
```

Hastie & Tibshirani, 1990

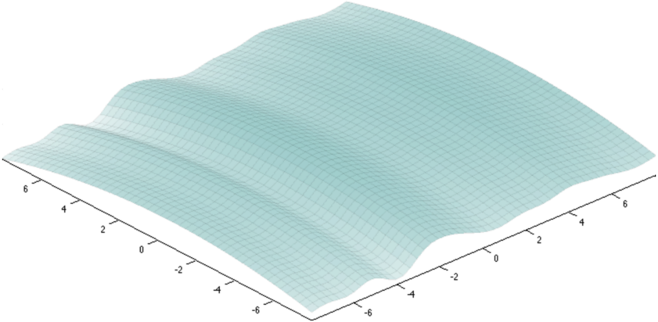


$$k(a,b) = \sum_{d}^{D} k_d(a_d, b_d)$$

Additive Models

```
phi = @(a)(bsxfun(@power,a,[0:2]));
k   = @(a,b)(kSE(a(:,1),b(:,1)) + phi(a(:,2))*phi(b(:,2))');
```

Wahba, 1990, Rasmussen & Williams, 2006



$$k(a,b) = \sum_d^D k_d(a_d, b_d)$$

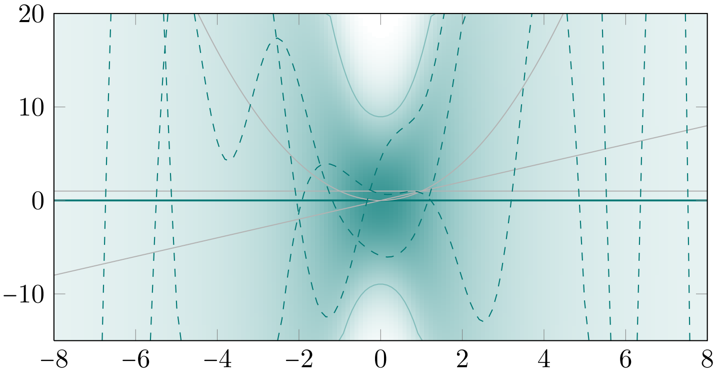
- use structure of k_{XX} to drastically lower inference cost
- generalize to $k(a,b) = \sum_d^D k_d(a_d,b_d) + \sum_i^D \sum_j^{i-1} k_{ij}(a_i,a_j,b_i,b_j)$ to get functional ANOVA

Products of Kernels are Kernels

```
phi = @(a)(bsxfun(@power,a,[0:2]));
k1  = @(a,b)(20 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
k   = @(a,b)(k1(a,b) .* (phi(a) * phi(b)'));
```

Theorem (I. Schur (proof in Bapat, 1997, Million 2007))

If A and B are positive semidefinite, then $A \odot B$ (=`A.*B`) is semidefinite.



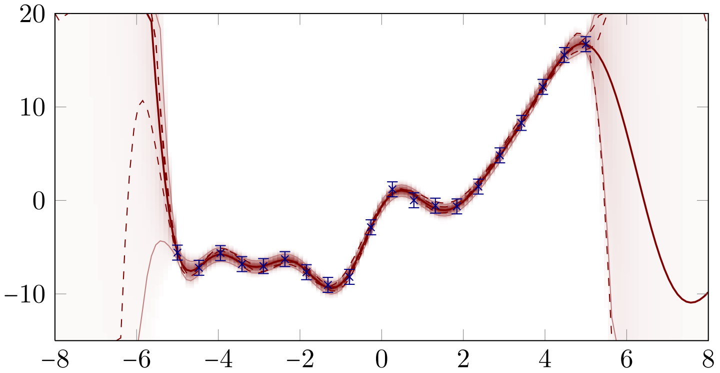
Intuition: similarity under k^1 AND k^2.

Products of Kernels are Kernels

```
phi = @(a)(bsxfun(@power,a,[0:2]));
k1  = @(a,b)(20 * exp(-(bsxfun(@minus,a./2,b'./2)).^2));
k   = @(a,b)(k1(a,b) .* (phi(a) * phi(b)'));
```

Theorem (I. Schur (proof in Bapat, 1997, Million 2007))

If A and B are positive semidefinite, then $A \odot B$ (=`A.*B`) is semidefinite.



Intuition: similarity under k^1 AND k^2.

Summary: Kernel design

Mercer kernels form a semiring

- k is positive semidefinite ⇒ αk for $\alpha \in \mathbb{R}_+$ is positive semidefinite
  e.g. to change signal variance
- $k(a,b)$ is pos. semidef. ⇒ $k(\phi(a),\phi(b))$ is pos. semidef.
  e.g. to change length scale
- k_1, k_2 is positive semidefinite ⇒ $k_1 + k_2$ is positive semidefinite
  e.g. to encode OR similarity
- k_1, k_2 is positive semidefinite ⇒ $k_1 \odot k_2$ is positive semidefinite
  e.g. to encode AND similarity

These rules can encode prior knowledge in Gaussian models.

If your model has no parameters, you haven’t found them yet.

Of all those hyperparameters, which ones should I use?
And how should I set them?

Can I get away with using few, or no hyperparameters?

How should I choose all those parameters?

they are everywhere

- $f(x) = \sum_{i=1}^{?} x^i w_i$
- $k(a,b) = \theta^2 \exp\left(-\frac{(a-b)^2}{2\lambda}\right)$
- $p(y \mid f, \sigma) = \mathcal{N}(y; f_x, \sigma^2 I)$
- $k(a,b) = \theta^2 k_1(a,b) + k_2(a,b) \cdot k_3(a,b) + k_4(\phi(a), \phi(b))$

Hierarchical Bayesian Inference

announce your hypotheses, and let mathematics do the magic

- sum rule

$$p(y \mid \mathcal{M}) = \int p(y \mid f, \mathcal{M}) p(f \mid \mathcal{M}) \, \mathrm{d}f$$

- for Gaussians:

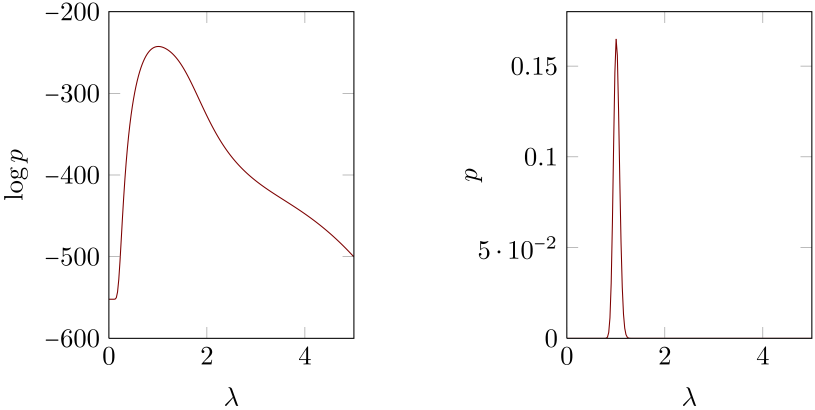
$$\begin{aligned} p(y \mid \mathcal{M}) &= \int \mathcal{N}(y; f_X, \sigma^2 I) \mathcal{GP}(f; \mu, k) \, \mathrm{d}f \\ &= \mathcal{N}(y; \mu_X, k_{XX} + \sigma^2 I) \end{aligned}$$

- Bayes' Theorem

$$p(\mathcal{M} \mid y) = \frac{p(y \mid \mathcal{M}) p(\mathcal{M})}{p(y)}$$

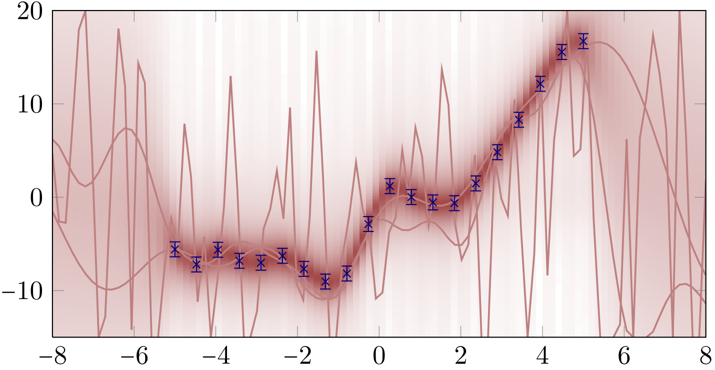
Model Selection

for the SE kernel



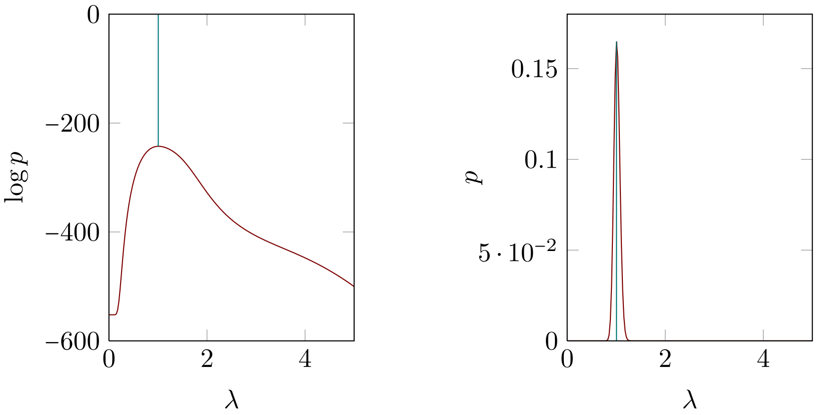
So how do you actually do this?

Markov Chain Monte Carlo



A Shortcut

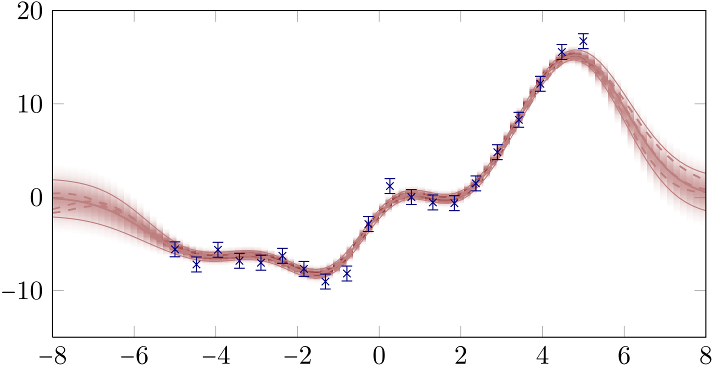
Type-II Maximum Likelihood



- frequentist estimator with good properties
- gives a Gaussian process: analytically desirable
- but ignores all other hypotheses
- maximum need not be good representer of whole distribution!

A Shortcut

Type-II Maximum Likelihood



So do we get away without making assumptions?

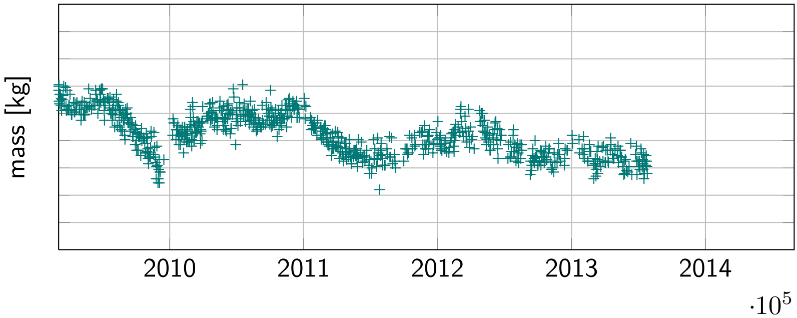
There is no unique natural way of choosing models

- parametrization of hypothesis classes is not unique
- there is always a “just so” hypothesis
- Minimizing training error gives overfitting
- Class of models is unbounded
- Choosing the “least complex” model is not well-defined.
- If your “model has no tuning parameters”, you haven’t found them yet.
- If your “model makes no assumptions”, you haven’t found them yet.

Inference requires assumptions.

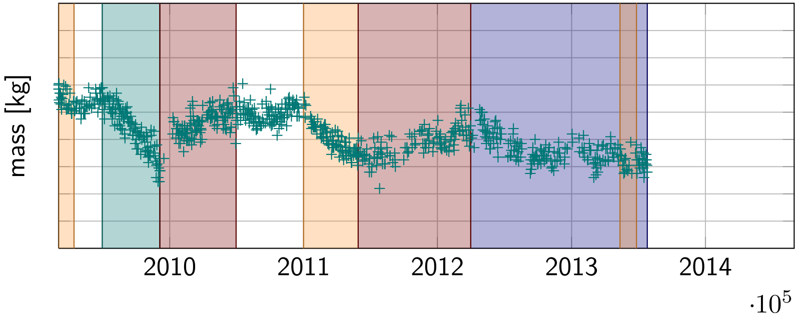
Building Explicit Models for Physical Processes

How to loose weight



Building Explicit Models for Physical Processes

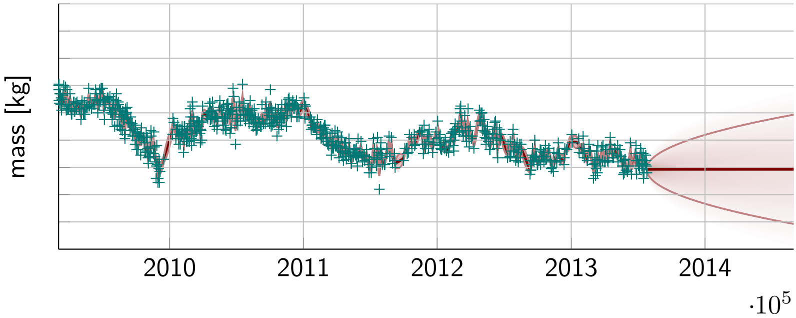
How to loose weight



- running ($4\times$ / week, ≥ 7k): 1 Jul 2008 – 5 Dec 2009
- slacking: 1 Jan 2011 – 30 Aug 2011
- dieting: 1 Jan 2011 – 30 Aug 2011
- gym ($2\times$ / week): 1 Apr 2012 –
- vegetarian diet: 13 May 2013 – 27 Jun 2013

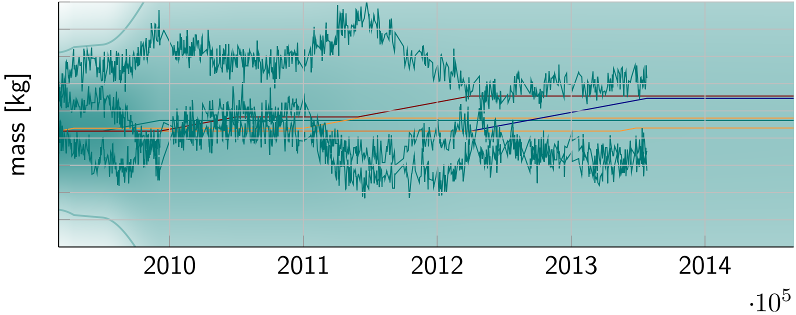
You can’t learn without making assumptions

Wiener process prior: `k = @(a,b)(bsxfun(@min,a-d0,b’-d0) ./ ell);`



Designing the Model

Use prior knowledge



- constant effects for each action: $\phi_{\text{run}}, \phi_{\text{diet}}, \phi_{\text{gym}}, \phi_{\text{veg}}, \phi_{\text{slack}}$
- random steps $k_{\text{Wiener}}(a, b)$
- random fluctuations $k_{\text{SE}}(a, b; \lambda = 1s\text{day})$
- measurement noise $\sigma = 0.1\text{g}$ (known)

$$k = \theta_{\text{SE}}^2 k_{\text{SE}} + \theta_{\text{W}}^2 k_{\text{Wiener}} + \theta_{\text{eff}}^2 (\phi_{\text{run}}\phi_{\text{run}}^\top + \phi_{\text{diet}}\phi_{\text{diet}}^\top + \phi_{\text{slack}}\phi_{\text{slack}}^\top + \phi_{\text{gym}}\phi_{\text{gym}}^\top + \phi_{\text{veg}}\phi_{\text{veg}}^\top)$$

Infer superimposed functions

Gaussians are closed under linear maps . . .

$$f_t = f_t^{\text{SE}} + f_t^{\text{Wiener}} + +\phi_t^{\text{gym}} w_{\text{gym}} + \phi_t^{\text{veg}} w_{\text{veg}} + \phi_t^{\text{diet}} w_{\text{diet}} + \phi_t^{\text{slack}} w_{\text{slack}} + \phi_t^{\text{run}} w_{\text{run}}$$

$$p(f) = \mathcal{N}\left[\underbrace{\begin{pmatrix} \delta(t-T) \\ \delta(t-T) \\ \phi_t^{\text{lin}} \\ \vdots \\ \phi_t^{\text{run}} \end{pmatrix}^{\top}}_{=:A^\top} \begin{pmatrix} f_{\text{SE}} \\ f_{\text{Wiener}} \\ w_{\text{diet}} \\ \vdots \\ w_{\text{run}} \end{pmatrix} ; A^\top \mu, A \underbrace{\begin{pmatrix} k_{\text{SE}} & & & & \\ & k_{\text{W}} & & & \\ & & \sigma^2_{\text{diet}} & & \\ & & & \ddots & \\ & & & & \sigma^2_{\text{run}} \end{pmatrix}}_{\Sigma} A \right]$$

$$p(f \mid y) = \mathcal{N}(f; \mu - \Sigma A \underbrace{(A^\top \Sigma A + \sigma^2 I)^{-1}}_{=:K^{-1}} (Y - A^\top \mu), \Sigma - \Sigma A K^{-1} A^\top \Sigma)$$

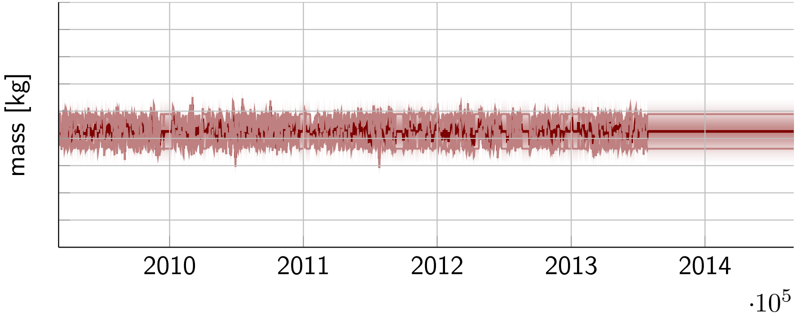
$$p(w_{\text{run}}) = \mathcal{N}(w_{\text{run}}; \mu_{\text{run}} - \sigma^2_{\text{run}} \phi_T^{\text{run}\top} K^{-1} (Y - A^\top \mu), \sigma^2_{\text{run}} - \sigma^2_{\text{run}} \phi_T^{\text{run}\top} K^{-1} \phi^{\text{run}} \sigma^2_{\text{run}})$$

$$p(f^{\text{SE}}) = \mathcal{N}(f^{\text{SE}}; \mu_{\text{SE}} - k_{tT}^{\text{SE}} K^{-1} (Y - A^\top \mu), k_{tt}^{\text{run}} - k_{tT}^{\text{run}} K^{-1} k_{Tt}^{\text{run}})$$

Individual effects

systematic effects

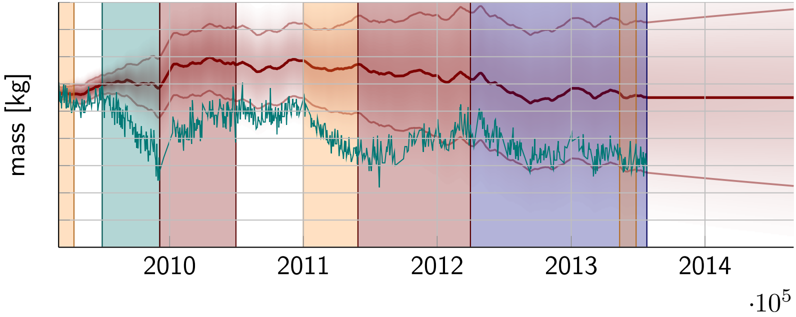
	μ	σ
running	-29 g/day	± 7 g/day
slacking	$+10$ g/day	± 4 g/day
dieting	-21 g/day	± 6 g/day
gym	-2 g/day	± 3 g/day
vegetarian diet	0 g/day	± 0 g/day



Individual effects

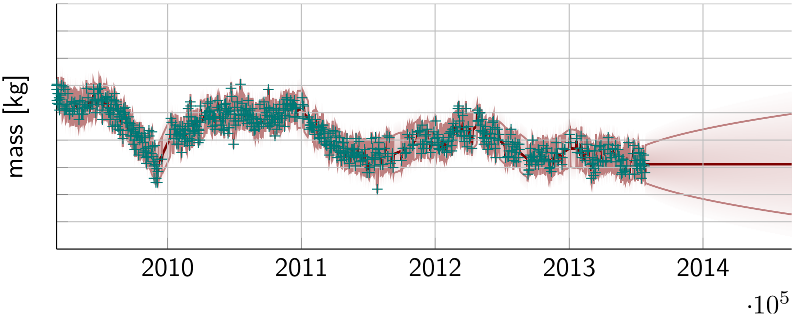
Wiener drift

	μ	σ
running	-29 g/day	± 7 g/day
slacking	$+10$ g/day	± 4 g/day
dieting	-21 g/day	± 6 g/day
gym	-2 g/day	± 3 g/day
vegetarian diet	0 g/day	± 0 g/day



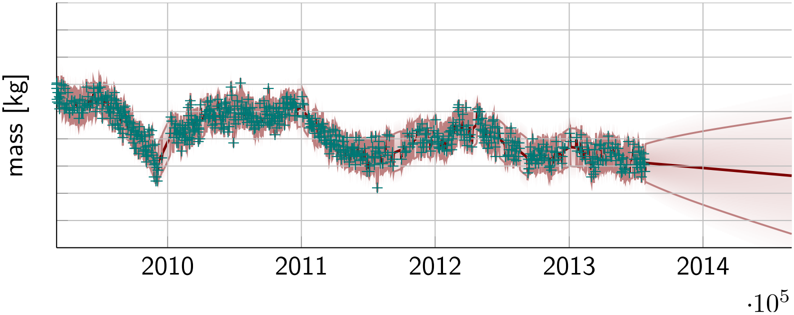
So What?

prediction doing nothing



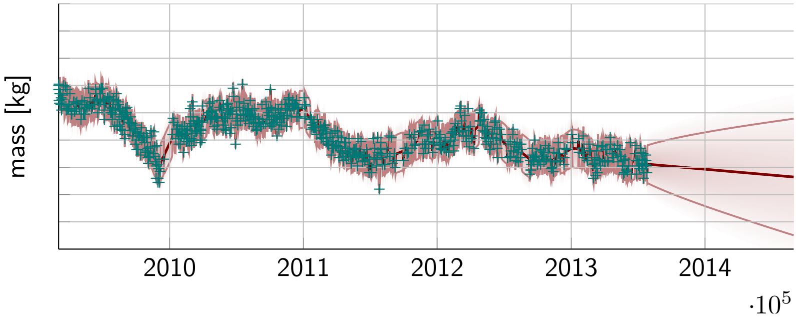
So What?

prediction going to the gym

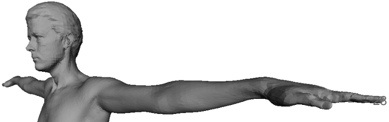


So What?

prediction going to the gym



combine (correlate) with other measurements to predict changes in body shape, exercise performance, . . .
see Karsten Roth’s ’weightulator’ app



Structured Linear Regression Models

powerful, but not magic

Gaussian nonlinear regression models

- can model nontrivial nonlinear effects
- can incorporate (Gaussian) measurement noise
- can separate nonlinear effects from each other

but they are not magic!

- all predictions subject to nontrivial prior
- hyperparameter choices depend on other effects modeled

This is true for literally all of science to various degrees!

ℓ_2 regularised least-squares

is Gaussian regression

$$p(f_X \mid y) = \frac{p(y \mid f_X)p(f_X)}{p(y)} = \frac{\mathcal{N}(y; f_X, \sigma^2 I)\mathcal{N}(f_X; m_X, k_{XX})}{\mathcal{N}(y; m_X, k_{XX} + \sigma^2 I)}$$

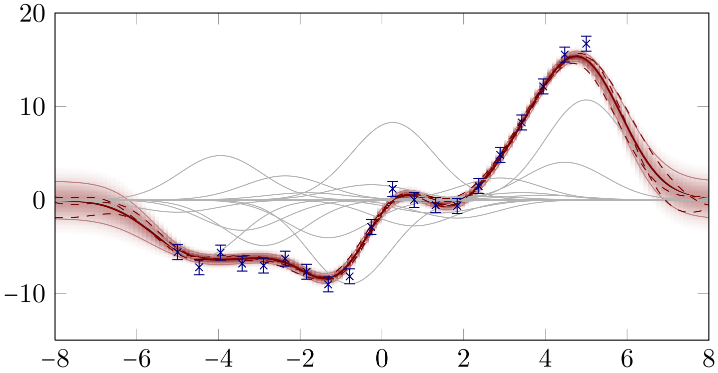
$$-2 \log p(f \mid y) = (y - f_X)^\top \sigma^{-2} I (y - f_X) + (f_X - m_X)^\top k_{XX}^{-1} (f_X - m_X) + \text{const.}$$

$$= \sigma^{-2} \|y - f_X\|_I^2 + \|f_X - m_X\|_{k_{XX}}^2 + \text{const.}$$

- the GP posterior mean is the regularised least-squares estimate
- aka kernel ridge regression
- regularizers are priors
- this also means a lot of theoretical concepts translate.
  But not all of them…

Reproducing Kernel Hilbert Spaces

the very rough story



- posterior mean $k_{xX}(k_{XX} + \sigma^2 I)^{-1} y = k_{xX}\alpha$
- (leaving out lots of details) the reproducing kernel Hilbert space (RKHS) of k is the space of $f(x) = \sum_{i=1}^{N} k(x, x_i)\alpha_i$
- note: for nondegenerate kernels, GP samples are almost surely not in the RKHS! The RKHS idea principally applies to the mean

Can GPs learn anything?

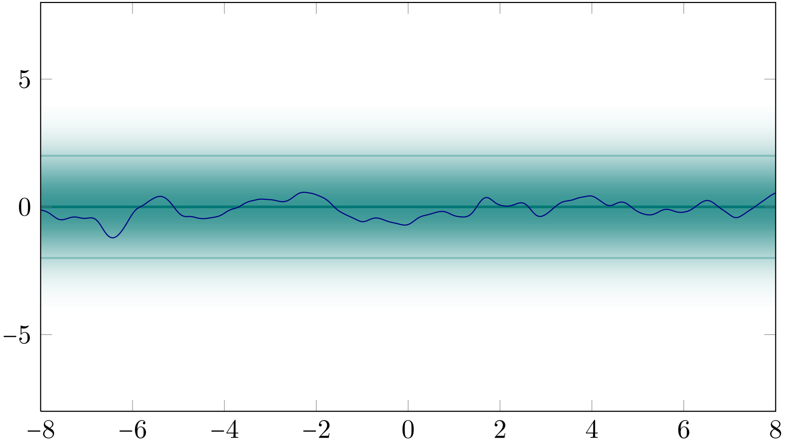
universal kernels

Micchelli, Xu, Zhang, 2006

- For some kernels over $\mathbb{R}^M$, for example the square exponential, the RKHS lies dense in the space of continuous functions
- so does this mean such GPs can learn any function?

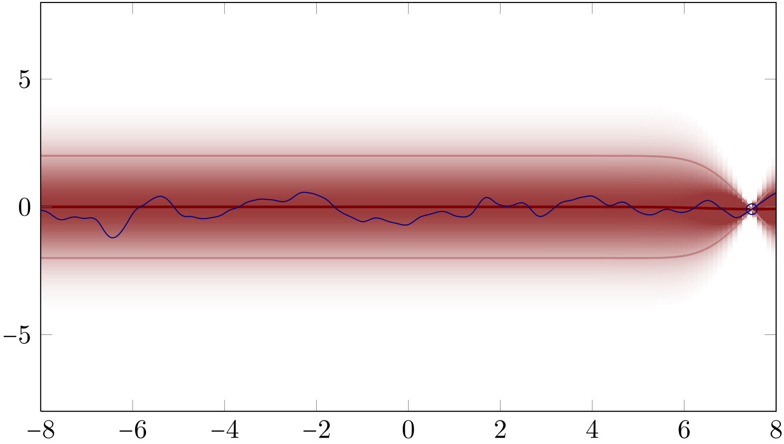
Universal RKHSs

an experiment – prior



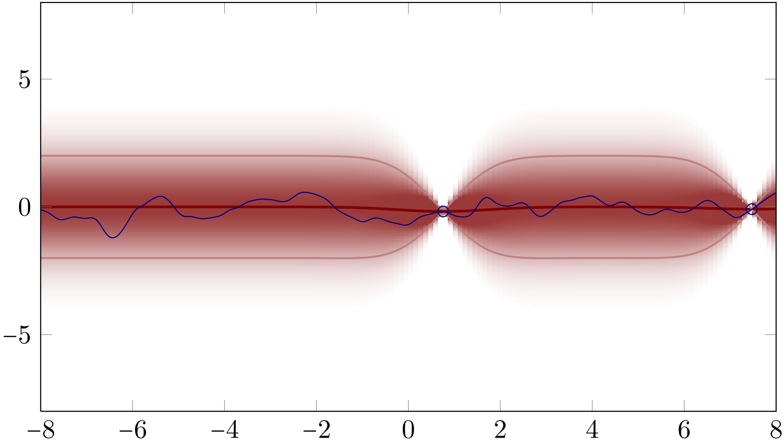
Universal RKHSs

an experiment – 1 evaluation



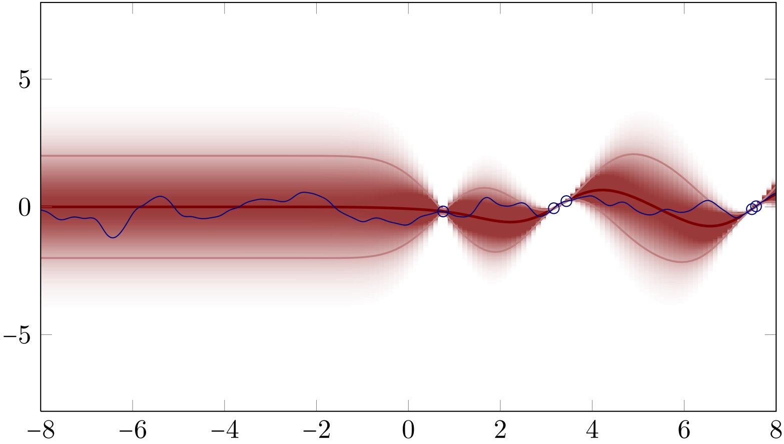
Universal RKHSs

an experiment – 2 evaluations



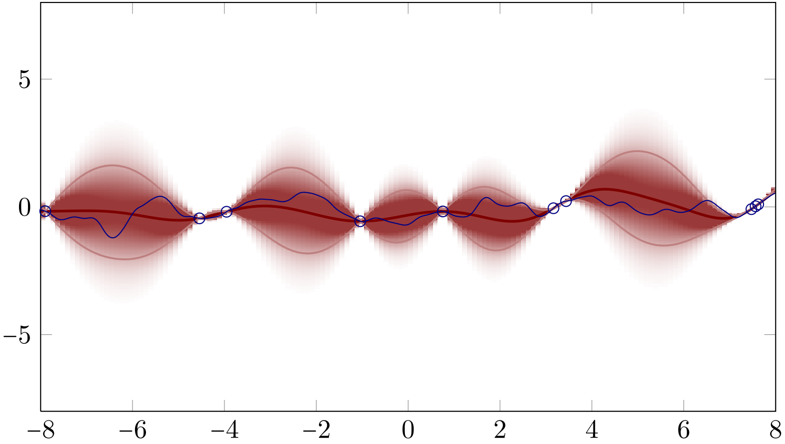
Universal RKHSs

an experiment – 5 evaluations



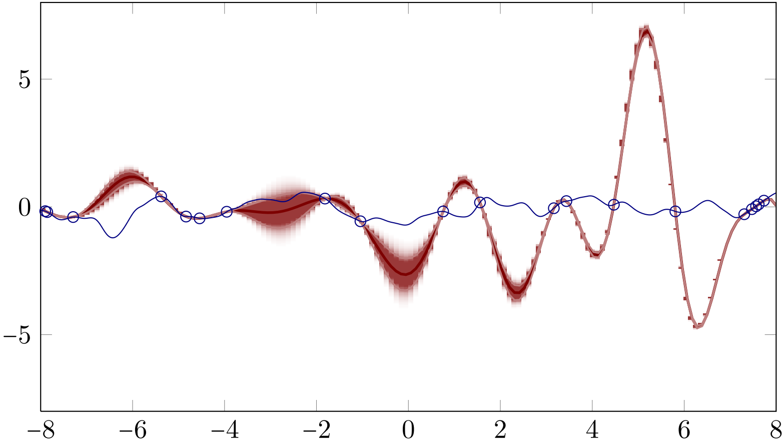
Universal RKHSs

an experiment – 10 evaluations



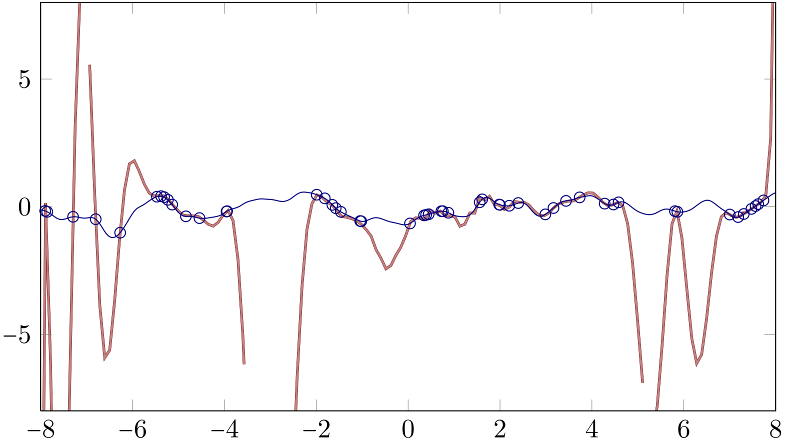
Universal RKHSs

an experiment – 20 evaluations



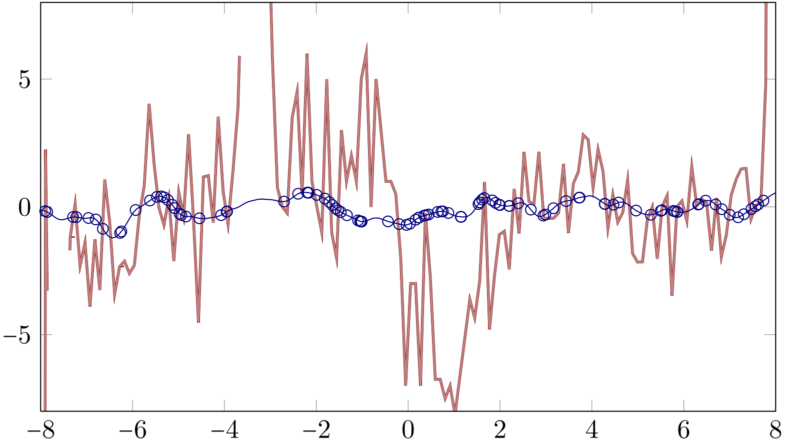
Universal RKHSs

an experiment – 50 evaluations



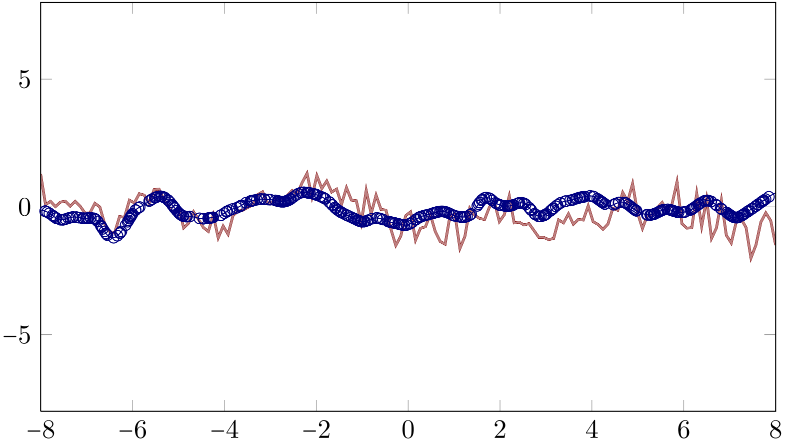
Universal RKHSs

an experiment – 100 evaluations



Universal RKHSs

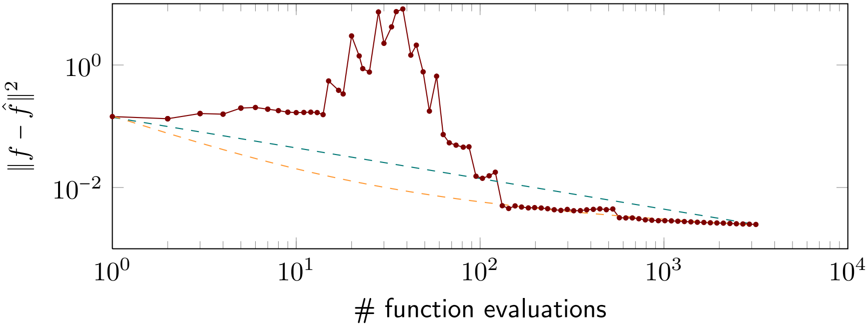
an experiment – 500 evaluations



Convergence Rates are Important

non-obvious aspects of f can ruin convergence

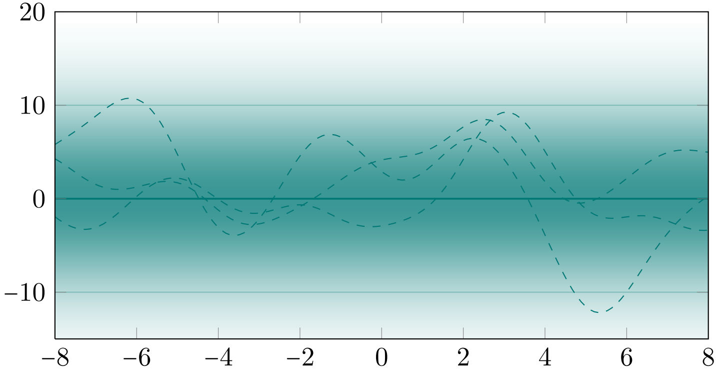
v.d.Vaart & v.Zanten, 2011



If f is “not well represented” by the kernel (has low prior density), the number of datapoints required to achieve ϵ error can be exponential in ϵ. Outside of the observation range, there are no guarantees at all.

A Tale of Frequentists and Bayesians

Gaussian / ℓ_2 regression is an interesting case, because the exact same method is studied on both sides.

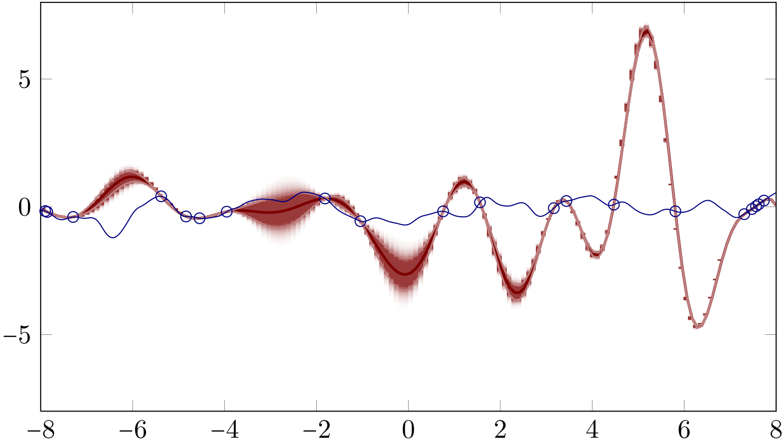


Bayesian: If this generative model is correct, this inference is optimal!

Frequentist: This estimator can learn everything given enough data!

A Tale of Frequentists and Bayesians

Gaussian / ℓ_2 regression is an interesting case, because the exact same method is studied on both sides.



Bayesian: Well, you haven't used the right prior!

Frequentist: Well, you haven't collected ∞ samples yet!

A Tale of Frequentists and Bayesians

a few things people actually say

The insufferable Frequentist, on kernel ridge regression

- “My method makes no assumptions because there are no priors.”
- “Just use the RBF [squared-exp] kernel. It is universal.”
- “I can show consistency and universality. The Bayesian can’t. Therefore my method is more mathematically pure (i.e. better).”

The insufferable Bayesian, on Gaussian process posterior means

- “The prior is subjective. If you don’t like it, you can always change it. I don’t need to worry about what happens out of model. If the model were wrong, I’d just use a different one.”
- “The posterior faithfully represents all available information. We can therefore use it to guide exploration (even though we never really check model validity).”

A Tale of Frequentists and Bayesians

the linear version of an old battle

The “Bayesian” (probabilistic) view

- is particularly helpful for small datasets and extrapolation
- gives an intuition for model properties, assumptions
- allows hierarchical extension, “complete toolbox”
- can help build good models

The “frequentist” (asymptotic) view

- is particularly helpful for the large dataset limit, interpolation
- gives an intuition for model limitations
- can offer efficient computational “shortcuts”
- can help build general models

Frequentist: “If the assumptions are correct,
this is the worst that could happen.”

Bayesian: “If the prior is strictly correct,
the posterior is the exact, optimal answer.”

Summary

> Il ne faudrait pas avoir uns sort de superstition pour la méthode des moindres carrés. [. . .] Elle suppose, en effect, qu'il n'y a pas d'erreur systématique, et *il y en a toujours*.
>
> Henri Poincaré
> *Calcul des probabilités*, 1896

- Kernels can be combined algebraically to build expressive models
  . . . but there is no universal model
- Hyperparameters can be inferred by hierarchical inference
  . . . but the result always depends on hyperpriors
- Gaussian regression allows inference on superposed effects
  . . . but priors need to be analysed carefully
- Both Bayesian and frequentist interpretations are helpful
  blindly trust neither your prior nor asymptotic statements
- Gaussians are a fundamental concept, used widely

Bibliography

- D.J.C. MacKay
  Introduction to Gaussian Processes
  in Bishop, C.M. (ed.), Neural Networks and Machine Learning, Springer, 1998
- T.J. Hastie & R.J. Tibshirani
  Generalized Additive Models
  Chapman & Hall, 1990
- E. Million
  The Hadamard Product
  Tech. Report
- R. Bapat
  Nonnegative Matrices and Applications
  Cambridge UP, 1997
- C.E. Rasmussen & C.K.I. Williams
  Gaussian Processes for Machine Learning
  MIT Press, 2006
- G. Wahba
  Spline Models for Observational Data
  SIAM CBMS-NSF reg. conf. series in applied mathematics, 1990
- C.A. Micchelli, Y. Xu, H. Zhang
  Universal Kernels
  JMLR **7** (2006), pp. 2651–2667
- A. van der Vaart & H. van Zanten
  Information rates of nonparametric Gaussian process methods
  JMLR **12** (2011), pp. 2095–2119

Eigenfunctions

Kernels really are infinitely large positive semidefinite matrices

An eigenfunction $\phi : \mathbb{X} \rightharpoonup \mathbb{C}$ obeys

$$\int k(a,b)\phi(a)\,\mathrm{d}\nu(x) = \lambda\phi(b)$$

Theorem (Mercer)

For a positive definite kernel. Then, ν^2-almost everywhere,

$$k(a,b) = \sum_{i=1}^{\infty} \lambda_i \phi_i(a)\phi_i^*(b)$$

The eigenfunctions can be chosen orthonormal, i.e. $\int \phi_i(x)\phi_j(x)\,\mathrm{d}x = \delta_{ij}$

So a GP puts mass on the space spanned by the eigenfunctions. What is this space?

the RKHS

definition

- posterior mean $k_{xX}(k_{XX} + \sigma^2 I)^{-1} y = k_{xX}\alpha$
- the reproducing kernel Hilbert space (RKHS) of k is the space of

$$f(x) = \sum_{i=1}^{N} f_i \phi_i(x) \quad \text{s.t.} \quad \sum_{i=1}^{N} f_i^2 / \lambda_i < \infty$$

- this space, with the inner product

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i}^{N} \frac{f_i g_i}{\lambda_i}$$

  is a Hilbert space. It is uniquely defined[1] by k. (It is also *reproduced* by k, i.e. $\langle f, k(\cdot, x) \rangle_{\mathcal{H}} = f(x)$).
- so is the GP a distribution over that Hilbert space?

[1] Moor-Aronszajn theorem. Aronszajn, 1950

Are GPs distributions on the RKHS?

no!

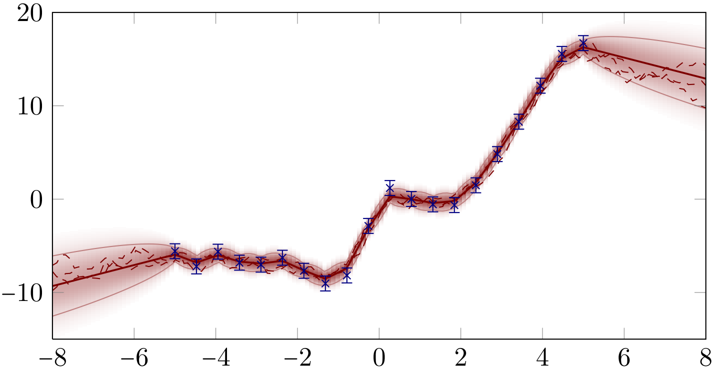
- to sample $f \sim \mathcal{GP}(0, k)$, draw $f_i \sim \mathcal{N}(0, \lambda_i), \forall i = 1, \dots, N$, then

$$f(x) = \sum_i^N f_i \phi_i(x) \quad \Rightarrow \quad \mathsf{E}[\|f\|_{\mathcal{H}}^2] = \mathsf{E}[\langle f, f\rangle_{\mathcal{H}}] = \sum_{i=1}^N \frac{\mathsf{E}[f_i^2]}{\lambda_i} = \sum_{i=1}^N 1$$

- for nondegenerate kernels ($N = \infty$), GP samples are almost surely not in the RKHS! The posterior mean is more regular (usually: smoother) than almost all samples.
- samples from a GP are “just outside” of the RKHS in that they are almost surely not of finite norm, but of the right algebraic form.

So what?

This is not just a technical point. Example: linear splines



- RKHS: piecewise linear, i.e. smooth almost everywhere
- GP samples: non-differentiable almost everywhere
- when you think about the mean, think of the RKHS. But remember that samples from a GP can be very different from the mean.