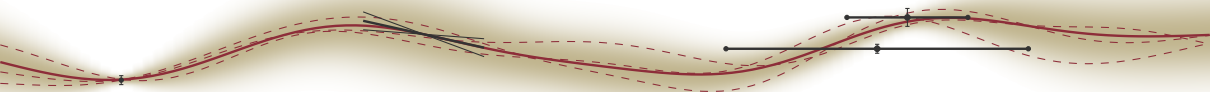
Probabilistic Inference and Learning
Lecture 04
Parametric Gaussian Regression

Philipp Hennig

29 October 2018

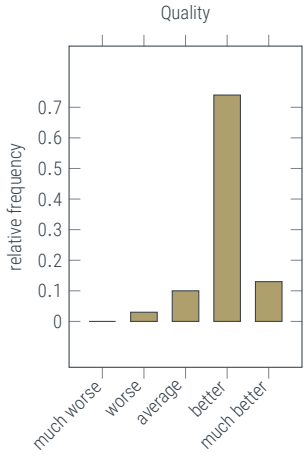
Eberhard Karls
Universität Tübingen

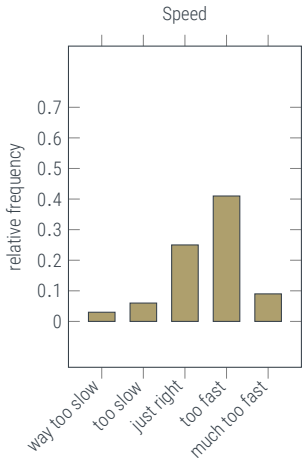
Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning

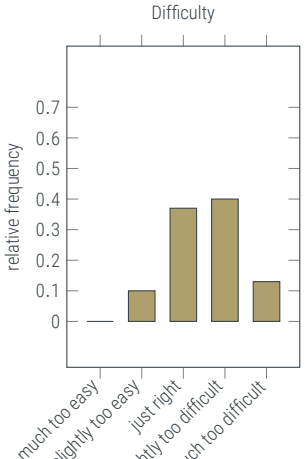


Last Lecture: Debrief

Feedback dashboard







Last Lecture: Debrief

Detailed Feedback

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Things you did not like:

- **use of blackboard** !
- examples
- plots
- too fast, more proofs required
- too easy, we know this already
- too many equations
- the pace is too high to take notes on a computer (equations are hard)

Things you did not understand:

- map from ν to x in Gaussian examples
- why is $p(x_1 \mid x_2)$ unnormalizable in x_2, yet still a Gaussian?
- proofs
- what does "explaining away" mean?
- why is axis aligned $\Leftrightarrow$ independence?

Things you enjoyed:

- **plots / visualizations**
- use of blackboard
- properties of Gaussians
- historical notes
- break
- examples

Overview of Lectures so far:

0. Introduction to Reasoning under Uncertainty
   - Probabilities are the mathematical formalization of uncertainty
1. Probabilistic Reasoning
   - Probabilities extend deductive to plausible reasoning. Conditional independence affects complexity
2. Probabilities over Continuous Variables
   - Probability **densities** distribute probability over continuous domains
3. Gaussian Probability Distributions
   - Gaussians map probabilistic inference to **linear algebra**

Today:

- using Gaussian distributions to **learn/infer functions** — the base case of **supervised learning**

Recap: Gaussian Distributions

Gaussian distributions provide the linear algebra of inference

- products of Gaussians are Gaussians $\quad C := (A^{-1} + B^{-1})^{-1} \quad c := C(A^{-1}a + B^{-1}b)$

$$\mathcal{N}(x; a, A)\mathcal{N}(x; b, B) = \mathcal{N}(x; c, C)\mathcal{N}(a; b, A + B)$$

- marginals of Gaussians are Gaussians

$$\int \mathcal{N}\left[\begin{bmatrix} x \\ y \end{bmatrix}; \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right] dy = \mathcal{N}(x; \mu_x, \Sigma_{xx})$$

- (linear) conditionals of Gaussians are Gaussians

$$p(x \mid y) = \frac{p(x, y)}{p(y)} = \mathcal{N}\left(x; \mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(y - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}\right)$$

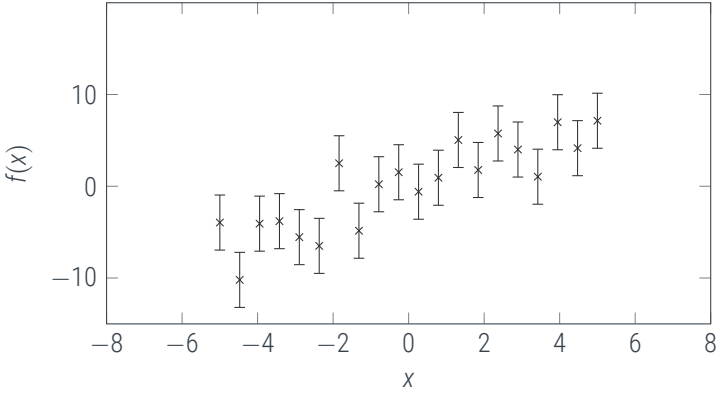
- linear projections of Gaussians are Gaussians

$$p(z) = \mathcal{N}(z; \mu, \Sigma) \quad \Rightarrow \quad p(Az) = \mathcal{N}(Az, A\mu, A\Sigma A^\intercal)$$

Bayesian inference becomes linear algebra $\qquad p(x) = \mathcal{N}(x; \mu, \Sigma) \qquad p(y \mid x) = \mathcal{N}(y; A^\intercal x + b, \Lambda)$

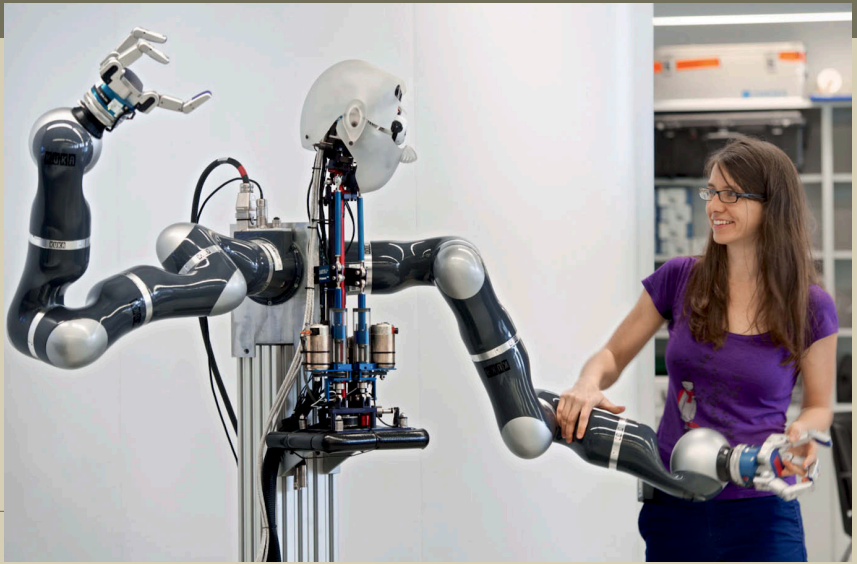
$$p(B^\intercal x + c \mid y) = \mathcal{N}[B^\intercal x + c; B^\intercal \mu + c + B^\intercal \Sigma A(A^\intercal \Sigma A + \Lambda)^{-1}(y - A^\intercal \mu - b), B^\intercal \Sigma B - B^\intercal \Sigma A(A^\intercal \Sigma A + \Lambda)^{-1} A^\intercal \Sigma B]$$

given: $\boldsymbol{y} \in \mathbb{R}^N, \quad p(\boldsymbol{y} \mid f) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{f}(\boldsymbol{x}), \sigma^2 I_N)$. What is f?

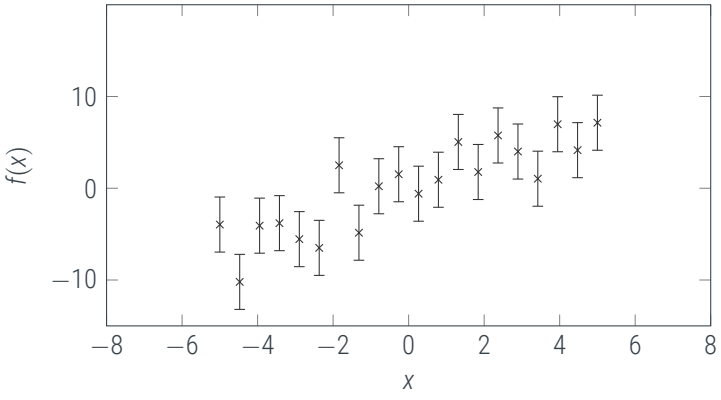


Supervised Regression

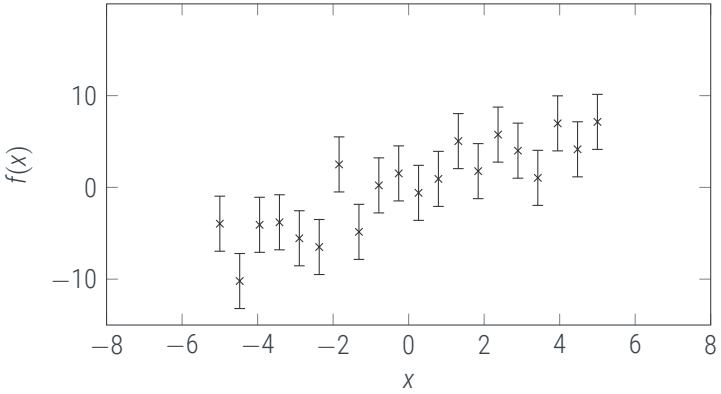
A data set

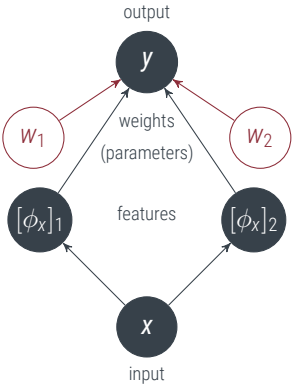


given: $\boldsymbol{y} \in \mathbb{R}^N, \quad p(\boldsymbol{y} \mid f) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{f}(\boldsymbol{x}), \sigma^2 I_N)$. What is f?



Assume **linear** function $f(x) = w_1 + w_2 x = \phi_x^\intercal \boldsymbol{w}$ with **features** $\phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix} =: \phi_x$

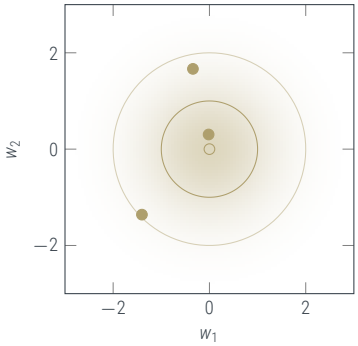




A linear **generative** model

if every variable is Gaussian and every relationship is linear, all marginals and conditionals are also Gaussian

$$f(x) = w_1 + w_2 x = \phi_x^\intercal w$$
$$p(w) = \mathcal{N}(w; \mu, \Sigma)$$

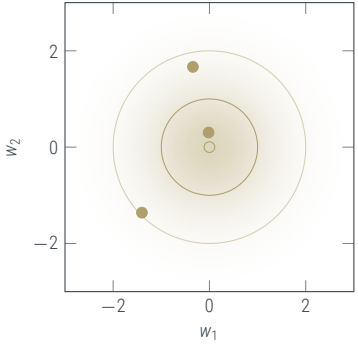


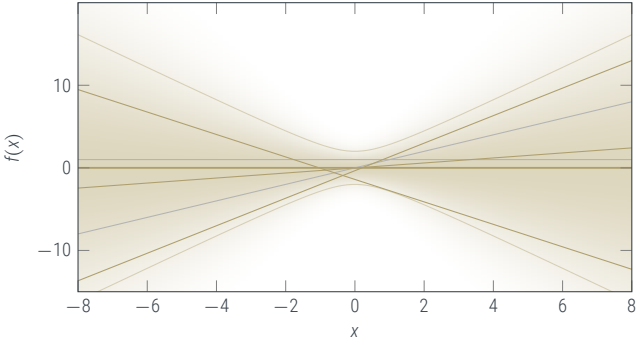
A linear **generative** model

if every variable is Gaussian and every relationship is linear, all marginals and conditionals are also Gaussian

$$f(x) = w_1 + w_2 x = \phi_x^\intercal w$$
$$p(w) = \mathcal{N}(w; \mu, \Sigma)$$

$$p(f) = \mathcal{N}(f; \phi_x^\intercal \mu, \phi_x^\intercal \Sigma \phi_x)$$

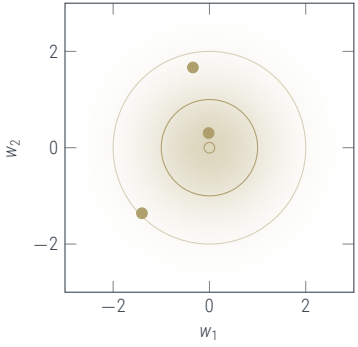




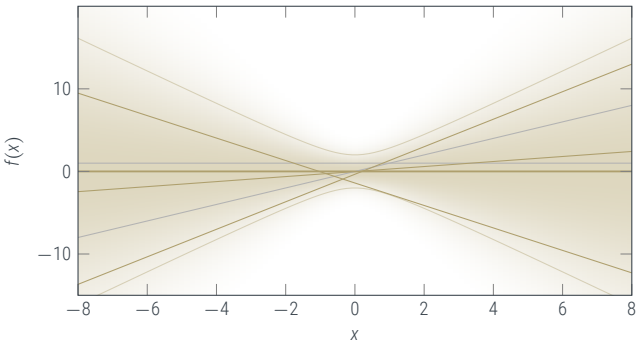
A linear **generative** model

if every variable is Gaussian and every relationship is linear, all marginals and conditionals are also Gaussian

$$f(x) = w_1 + w_2 x = \phi_x^\intercal w$$
$$p(w) = \mathcal{N}(w; \mu, \Sigma)$$



$$p(f) = \mathcal{N}(f; \phi_x^\intercal \mu, \phi_x^\intercal \Sigma \phi_x)$$



Notation

this will become exceedingly helpful later on

Dataset: $X := \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{X}^N, \boldsymbol{y} := \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \in \mathbb{R}^N$. We will use the following very sloppy notation, sloppily

$$\phi_x := \phi(x) = \begin{bmatrix} 1 \\ x \end{bmatrix} \in \mathbb{R}^F \qquad \phi_X := \begin{bmatrix} \phi(x_1) & \phi(x_2) & \cdots & \phi(x_N) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_N \end{bmatrix} \in \mathbb{R}^{F \times N}$$

$$f_x := f(x) \in \mathbb{R} \qquad f_X := \phi_X^\intercal \boldsymbol{w} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) \\ \phi_1(x_2) & \phi_2(x_2) \\ \vdots & \vdots \\ \phi_1(x_N) & \phi_2(x_N) \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \phi_{x_1}^\intercal \boldsymbol{w} \\ \phi_{x_2}^\intercal \boldsymbol{w} \\ \vdots \\ \phi_{x_N}^\intercal \boldsymbol{w} \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_N) \end{bmatrix} \in \mathbb{R}^N$$

Think of f as an infinitely long vector, indexed by x:

$$\boldsymbol{v} \in \mathbb{R}^N, I \in \mathbb{N}^d \Rightarrow v_I := [v_{I_1}, \ldots, v_{I_d}] \in \mathbb{R} \quad \Longleftrightarrow \quad f \in \mathbb{R}^\infty, X \in \mathbb{R}^N \Rightarrow f_X := [f_{x_1}, \ldots, f_{x_N}] \in \mathbb{R}^N.$$

Gaussian Inference on a linear function

weight space / function space

$$\text{prior} \quad p(w) = \mathcal{N}(w; \mu, \Sigma) \quad \Rightarrow \quad p(f) = \mathcal{N}(f_X; \phi_X^\intercal \mu, \phi_X \Sigma \phi_X)$$

$$\text{likelihood} \quad p(y \mid w, \phi_X) = \mathcal{N}(y; \phi_X^\intercal w, \sigma^2 I) = \mathcal{N}(y; f_X, \sigma^2 I)$$

Gaussian Inference on a linear function

weight space / function space

$$
\begin{aligned}
\text{prior} \quad p(w) &= \mathcal{N}(w; \mu, \Sigma) \quad \Rightarrow \quad p(f) = \mathcal{N}(f_X; \phi_x^\intercal \mu, \phi_x \Sigma \phi_x) \\
\text{likelihood} \quad p(y \mid w, \phi_X) &= \mathcal{N}(y; \phi_X^\intercal w, \sigma^2 I) = \mathcal{N}(y; f_X, \sigma^2 I) \\
\text{posterior on } \boldsymbol{w} \quad p(w \mid \boldsymbol{y}, \phi_X) &= \mathcal{N}(w; \mu + \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal \mu), \\
&\qquad \Sigma - \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}\phi_X^\intercal \Sigma) \\
&= \mathcal{N}\Big(w; (\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1} \left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right), \\
&\qquad (\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\Big)
\end{aligned}
$$

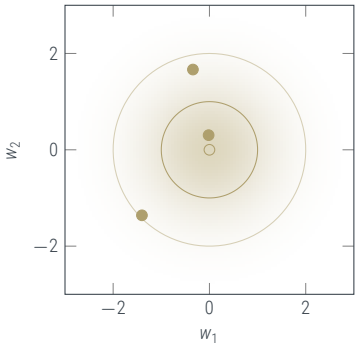
Gaussian Inference on a linear function

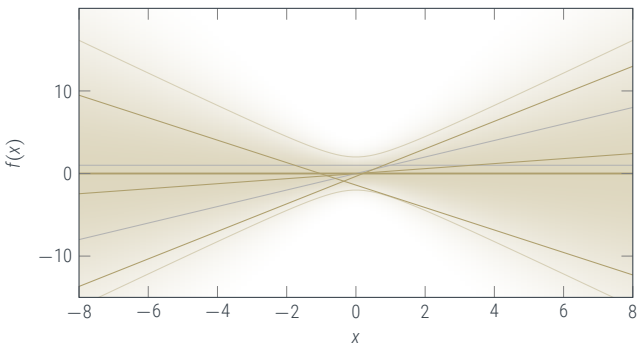
weight space / function space

$$
\begin{aligned}
\text{prior} \quad & p(w) = \mathcal{N}(w; \mu, \Sigma) \quad \Rightarrow \quad p(f) = \mathcal{N}(f_X; \phi_X^\intercal \mu, \phi_X \Sigma \phi_X) \\
\text{likelihood} \quad & p(y \mid w, \phi_X) = \mathcal{N}(y; \phi_X^\intercal w, \sigma^2 I) = \mathcal{N}(y; f_X, \sigma^2 I) \\
\text{posterior on } \boldsymbol{w} \quad & p(w \mid \boldsymbol{y}, \phi_X) = \mathcal{N}(w; \mu + \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal \mu), \\
& \qquad \Sigma - \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}\phi_X^\intercal \Sigma) \\
& \qquad = \mathcal{N}\Big(w; (\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right), \\
& \qquad (\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\Big) \\
\text{posterior on } f \quad & p(f_x \mid y, \phi_X) = \mathcal{N}(f_x; \phi_x^\intercal \mu + \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal \mu), \\
& \qquad \phi_x^\intercal \Sigma \phi_x - \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1} \phi_X^\intercal \Sigma \phi_x) \\
& \qquad \mathcal{N}\Big(f_x; \phi_x(\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right), \\
& \qquad \phi_x(\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\phi_x^\intercal\Big)
\end{aligned}
$$

$$p(w \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(w; \mu + \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal\mu), \Sigma - \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)$$

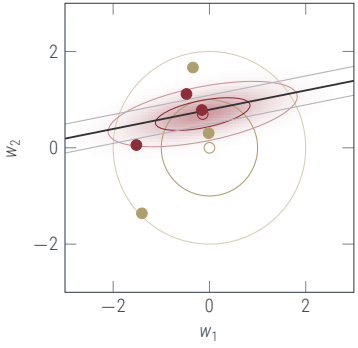
$$p(f_x \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(f_x; \phi_x^\intercal\mu + \phi_x^\intercal\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal\mu), \phi_x^\intercal\left(\Sigma - ^\intercal\ \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)\phi_x\right)$$

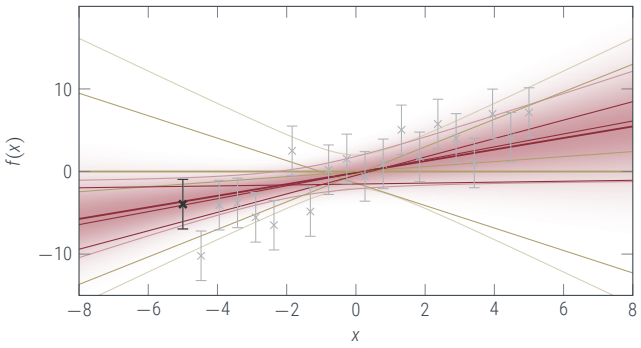




$$p(w \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(w; \mu + \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal\mu), \Sigma - \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)$$

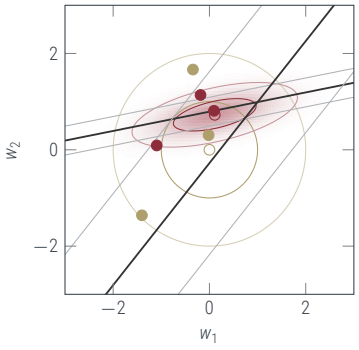
$$p(f_x \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(f_x; \phi_x^\intercal\mu + \phi_x^\intercal\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal\mu), \phi_x^\intercal\left(\Sigma - ^\intercal\ \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)\phi_x\right)$$

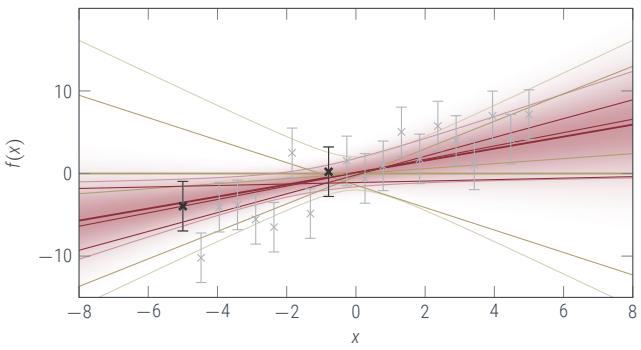




$$p(w \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(w; \mu + \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal\mu), \Sigma - \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)$$

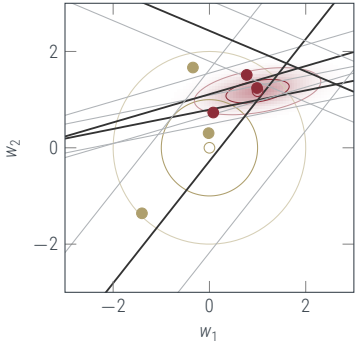
$$p(f_x \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(f_x; \phi_x^\intercal\mu + \phi_x^\intercal\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal\mu), \phi_x^\intercal\left(\Sigma - {}^\intercal\,\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)\phi_x\right)$$

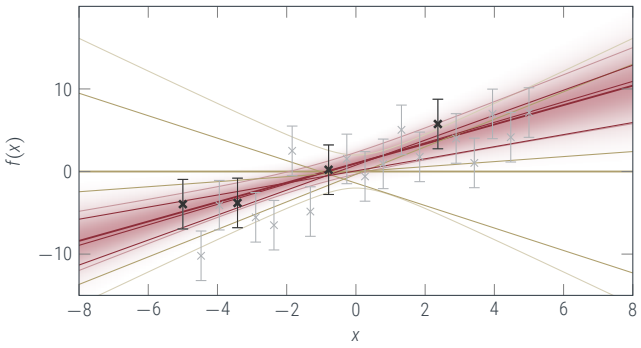




$$p(w \mid \mathbf{y}, \phi_X) = \mathcal{N}\left(w; \mu + \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(\mathbf{y} - \phi_X^\intercal\mu), \Sigma - \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)$$

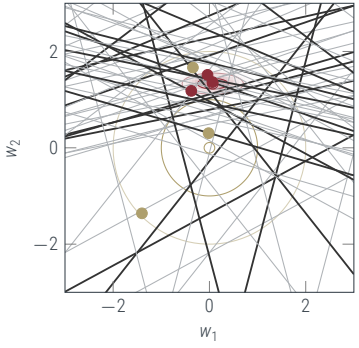
$$p(f_x \mid \mathbf{y}, \phi_X) = \mathcal{N}\left(f_x; \phi_x^\intercal\mu + \phi_x^\intercal\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal\mu), \phi_x^\intercal\left(\Sigma - {}^\intercal\,\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)\phi_x\right)$$

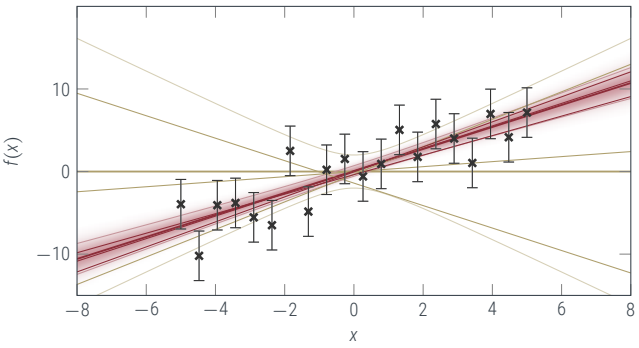




$$p(w \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(w; \mu + \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal\mu), \Sigma - \Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)$$

$$p(f_x \mid \boldsymbol{y}, \phi_X) = \mathcal{N}\left(f_x; \phi_x^\intercal\mu + \phi_x^\intercal\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal\mu), \phi_x^\intercal\left(\Sigma - {}^\intercal\,\Sigma\phi_X(\phi_X^\intercal\Sigma\phi_X + \sigma^2 I)^{-1}\phi_X^\intercal\Sigma\right)\phi_x\right)$$





Aside: Sampling from a Gaussian

Part I: Sampling from a Standard Gaussian — The Box-Muller Transform

[Box & Muller, 1958; also Paley & Wiener, 1934]

Illustration: (u/Cmglee, CC BY-SA 3.0)

https://upload.wikimedia.org/wikipedia/commons/1/1f/Box-Muller_transform_visualisation.svg

procedure RANDN(n)
- $u_0 \leftarrow$ RAND // draw first unit random
- **for** i=1,…,n **do**
  - $u_i \leftarrow$ RAND // draw unit random
  - $r \leftarrow \sqrt{-2 \ln u_{i-1}}$ // compute radius
  - $\theta \leftarrow 2\pi u_i$ // compute angle
  - **if** $i\%2 == 0$ **then**
    - $z_i \leftarrow r \cos\theta$ // transform to Euclidean
  - **else**
    - $z_i \leftarrow r \sin\theta$ // transform to Euclidean
  - **end if**
- **end for**
- **return** $z \sim \mathcal{N}(0, I_n)$

end procedure

Improvements

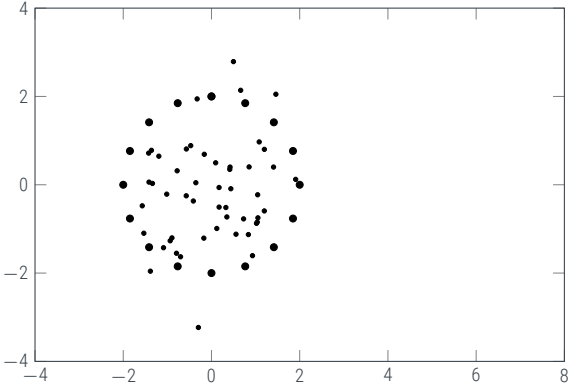
- Marsaglia's polar form: draw from unit circle instead of square (using rejection sampling). This avoids sin and cos, but requires rejection. Typically faster
- Marsaglia's Ziggurat algorithm: iteratively refined rejection sampling. Even faster, but tough to understand intuitively.

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance


```
U,D = eig(Sigma)
```

```
x = dot(dot(U,diag(sqrt(D))),randn(n,3)) + mu
```



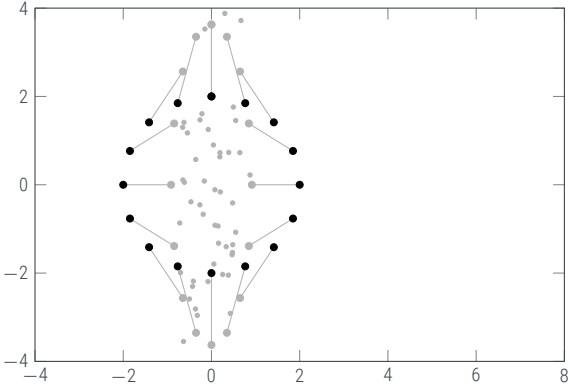
$$\Sigma = UDU^\intercal \qquad x = u$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance


```
U,D = eig(Sigma)
```

```
x = dot(dot(U,diag(sqrt(D))),randn(n,3)) + mu
```



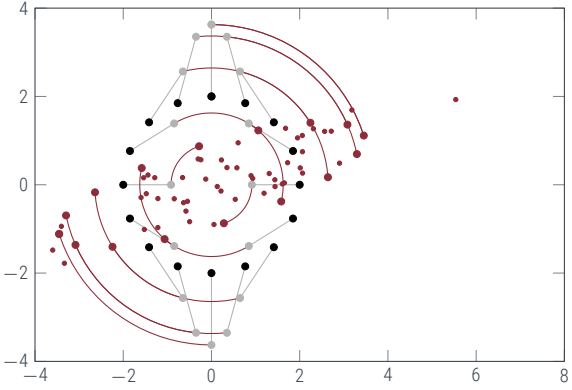
$$\Sigma = UDU^\intercal \qquad x = D^{1/2}u$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance


```
U,D = eig(Sigma)
```

```
x = dot(dot(U,diag(sqrt(D))),randn(n,3)) + mu
```



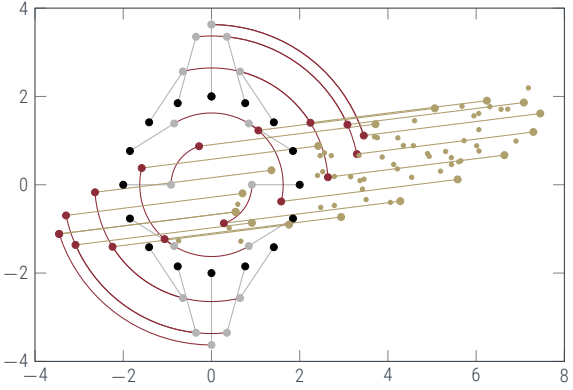
$$\Sigma = UDU^\intercal \qquad x = UD^{1/2}u$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
U,D = eig(Sigma)
```

```
x = dot(dot(U,diag(sqrt(D))),randn(n,3)) + mu
```

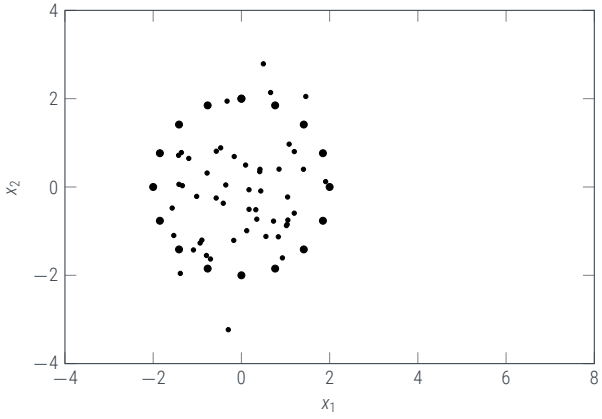


$$\Sigma = UDU^\intercal \qquad x = UD^{1/2}u + \mu$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
x = dot(cholesky(Sigma),randn(n,3)) + mu
```

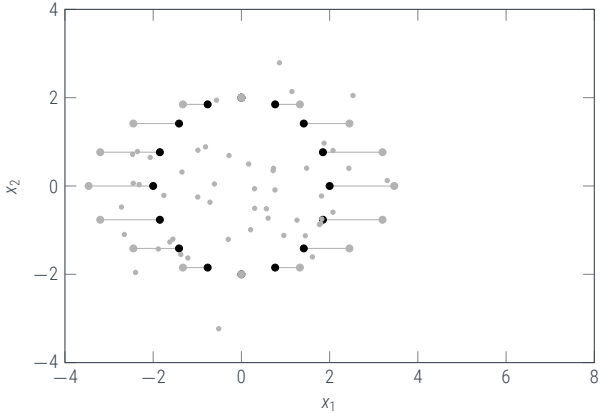


$$x = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
x = dot(cholesky(Sigma),randn(n,3)) + mu
```

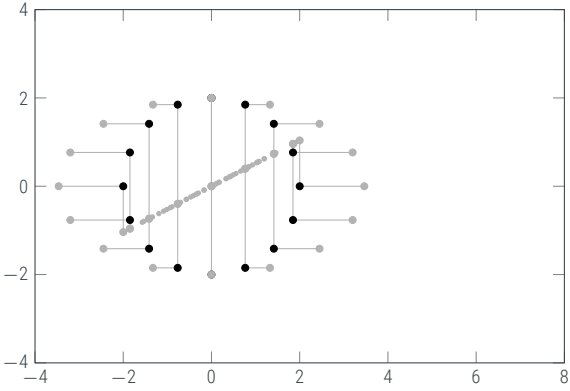


$$x = \begin{pmatrix} \sqrt{\Sigma_{11}} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
x = dot(cholesky(Sigma),randn(n,3)) + mu
```

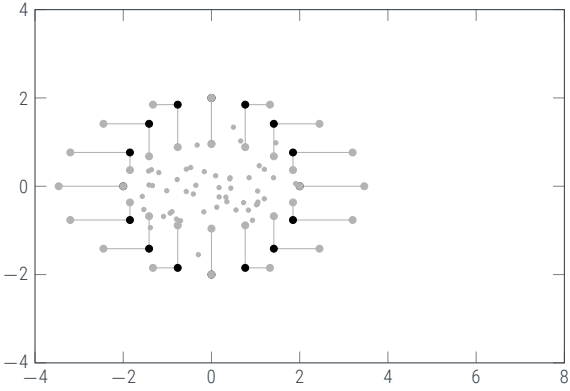


$$x = \begin{pmatrix} \sqrt{\Sigma_{11}} & 0 \\ \Sigma_{11}/\sqrt{\Sigma_{12}} & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
x = dot(cholesky(Sigma),randn(n,3)) + mu
```

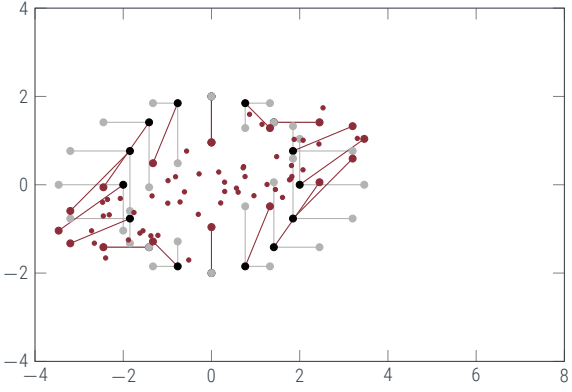


$$x = \begin{pmatrix} \sqrt{\Sigma_{11}} & 0 \\ 0 & \sqrt{\Sigma_{22} - \Sigma_{12}/\Sigma_{22}} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
x = dot(cholesky(Sigma),randn(n,3)) + mu
```

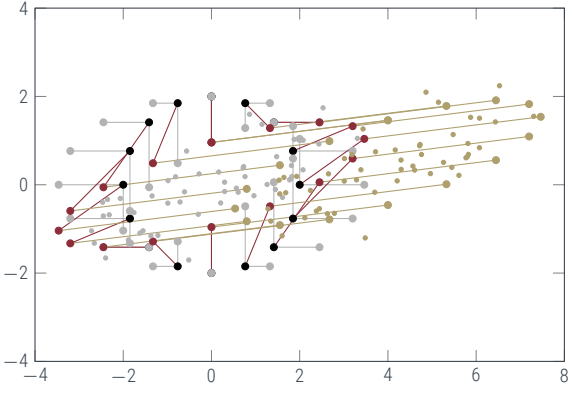


$$x = \begin{pmatrix} \sqrt{\Sigma_{11}} & 0 \\ \Sigma_{11}/\sqrt{\Sigma_{12}} & \sqrt{\Sigma_{22} - \Sigma_{12}/\Sigma_{22}} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Aside: Sampling from a Gaussian

Part II: Mapping to non-standard mean and covariance

```
x = dot(cholesky(Sigma),randn(n,3)) + mu
```



$$x = \begin{pmatrix} \sqrt{\Sigma_{11}} & 0 \\ \Sigma_{11}/\sqrt{\Sigma_{12}} & \sqrt{\Sigma_{22} - \Sigma_{12}/\Sigma_{22}} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \mu$$

- cost: $n^3/3$ FLOPS, about $1/4$ of Eigenvalue decomposition
- error bounded by $||\Sigma||_2$ (sqrt of largest eigenvalue) Golub & van Loan, 2013, 4.2.6

That's it for the Algebra

Gaussian Inference on a linear function

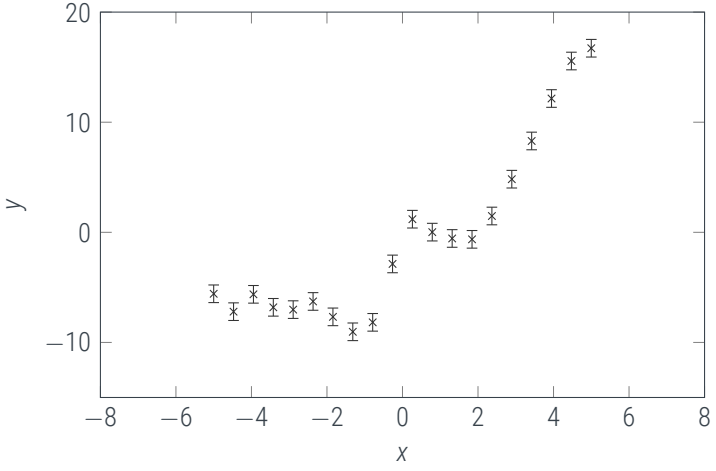
$$
\begin{aligned}
\text{prior} \quad p(w) &= \mathcal{N}(w; \mu, \Sigma) \quad \Rightarrow \quad p(f) = \mathcal{N}(f_x; \phi_x^\intercal \mu, \phi_x \Sigma \phi_x) \\
\text{likelihood} \quad p(y \mid w, \phi_x) &= \mathcal{N}(y; \phi_X^\intercal w, \sigma^2 I) = \mathcal{N}(y; f_X, \sigma^2 I) \\
\text{posterior on } \boldsymbol{w} \quad p(w \mid \boldsymbol{y}, \phi_X) &= \mathcal{N}(w; \mu + \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(\boldsymbol{y} - \phi_X^\intercal \mu), \\
&\qquad \Sigma - \Sigma\phi_X(\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}\phi_X^\intercal \Sigma) \\
&= \mathcal{N}\Big(w; (\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right), \\
&\qquad (\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\Big) \\
\text{posterior on } f \quad p(f_x \mid y, \phi_X) &= \mathcal{N}(f_x; \phi_x^\intercal \mu + \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}(y - \phi_X^\intercal \mu), \\
&\qquad \phi_x^\intercal \Sigma \phi_x - \phi_x^\intercal \Sigma \phi_X (\phi_X^\intercal \Sigma \phi_X + \sigma^2 I)^{-1}\phi_X^\intercal \Sigma \phi_x) \\
&\qquad \mathcal{N}\Big(f_x; \phi_x(\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\left(\Sigma^{-1}\mu + \sigma^{-2}\phi_X \boldsymbol{y}\right), \\
&\qquad \phi_x(\Sigma^{-1} + \sigma^{-2}\phi_X^\intercal \phi_X)^{-1}\phi_x^\intercal\Big)
\end{aligned}
$$

Code

`Gaussian_Linear_Regression.ipynb`

A more Realistic Dataset

General linear regression

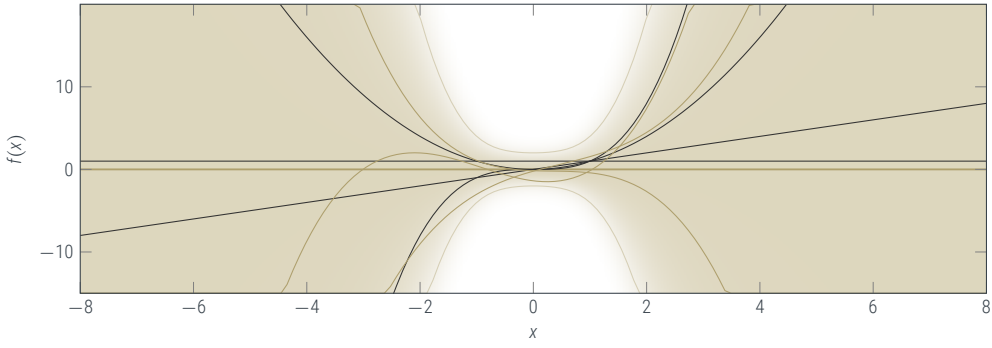


$$f(x) = w_1 + w_2 x = \phi_x^\intercal w$$

$$\phi_x := \begin{bmatrix} 1 \\ x \end{bmatrix}$$

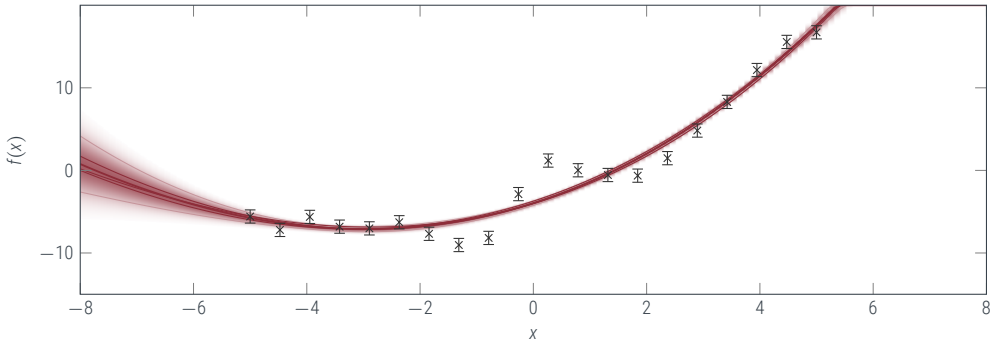
Cubic Regression

$$f(x) = \phi(x)^\intercal w \qquad \phi(x) = \begin{bmatrix} 1 & x & x.^2 & x.^3 \end{bmatrix}^\intercal$$



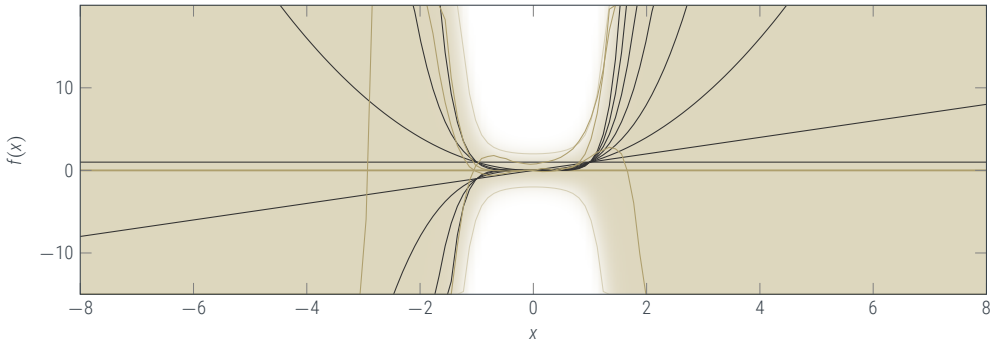
Cubic Regression

$$f(x) = \phi(x)^\intercal w \qquad \phi(x) = \begin{bmatrix} 1 & x & x.^2 & x.^3 \end{bmatrix}^\intercal$$



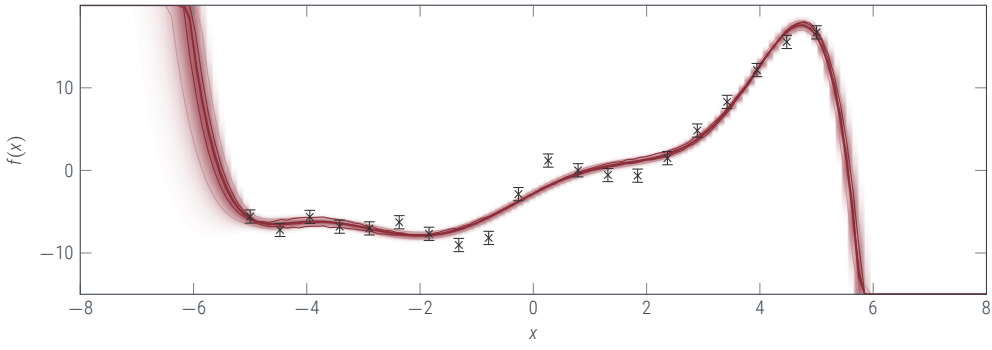
Septic Regression (?)

$$f(x) = \phi(x)^\intercal w \qquad \phi(x) = \begin{bmatrix} 1 & x & x.^2 & \cdots & x.^7 \end{bmatrix}^\intercal$$



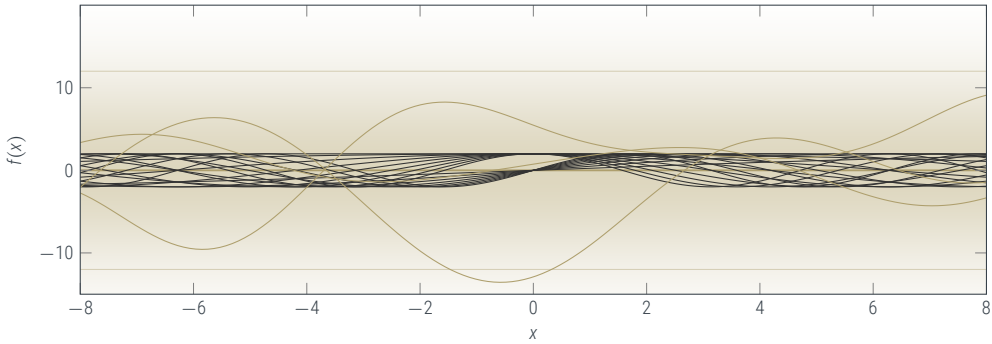
Septic Regression (?)

$$f(x) = \phi(x)^\intercal w \qquad \phi(x) = \begin{bmatrix} 1 & x & x.^2 & \cdots & x.^7 \end{bmatrix}^\intercal$$



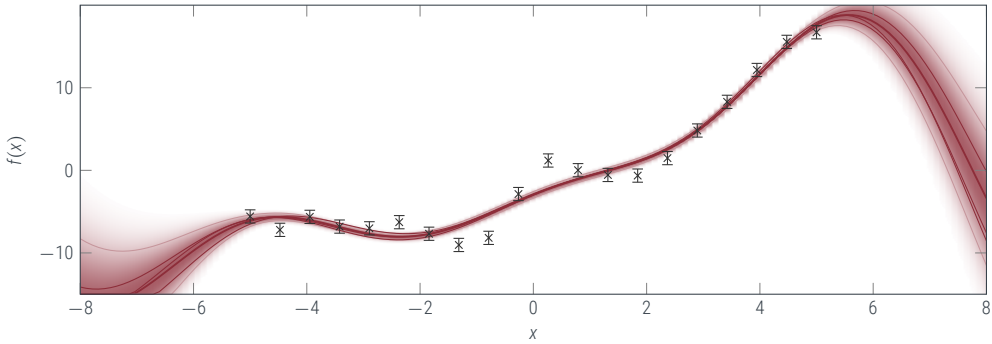
Fourier Regression

$$f(x) = \phi(x)^\intercal w \qquad \phi(x) = \begin{bmatrix} \cos(x) & \cos(2x) & \cos(3x) & \dots & \sin(x) & \sin(2x) & \dots \end{bmatrix}^\intercal$$



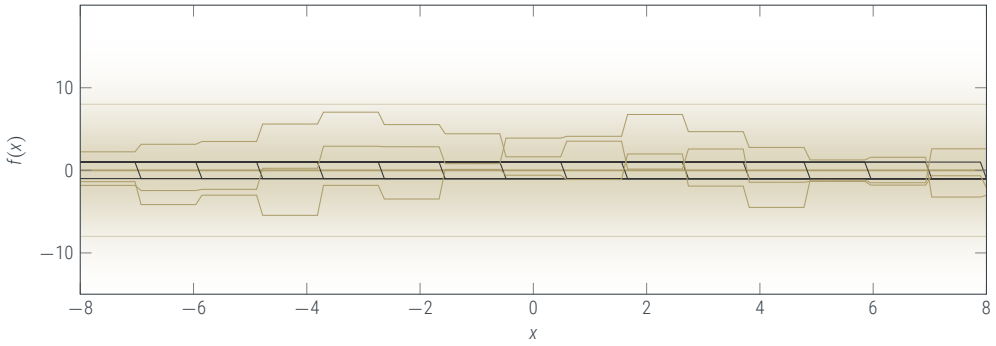
Fourier Regression

$$f(x) = \phi(x)^\intercal w \qquad \phi(x) = \begin{bmatrix} \cos(x) & \cos(2x) & \cos(3x) & \dots & \sin(x) & \sin(2x) & \dots \end{bmatrix}^\intercal$$



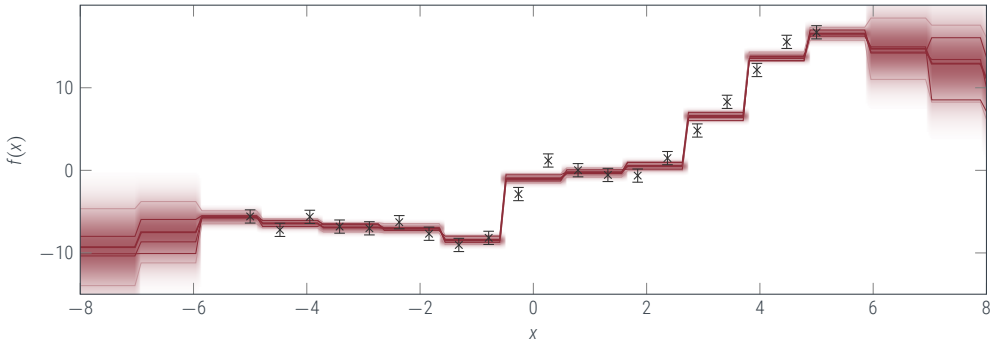
Pixel Regression

$$\phi(x) = -1 + 2\begin{bmatrix}\theta(x-8) & \theta(8-x) & \theta(x-7) & \theta(7-x) & \dots\end{bmatrix}^\intercal$$



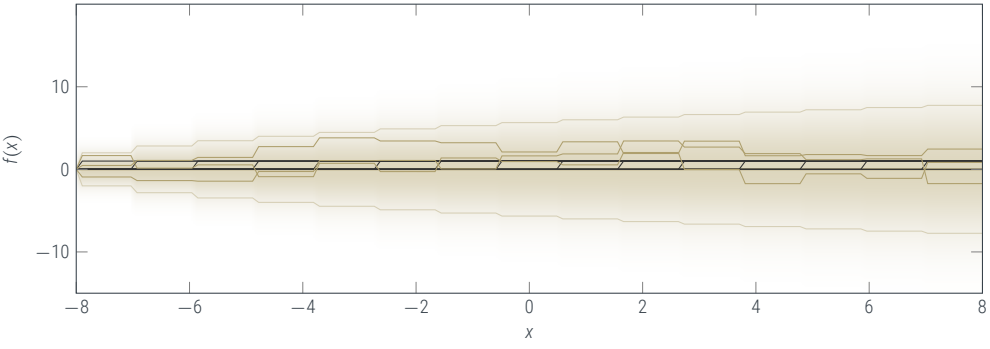
Pixel Regression

$$\phi(x) = -1 + 2\begin{bmatrix}\theta(x-8) & \theta(8-x) & \theta(x-7) & \theta(7-x) & \dots\end{bmatrix}^\intercal$$



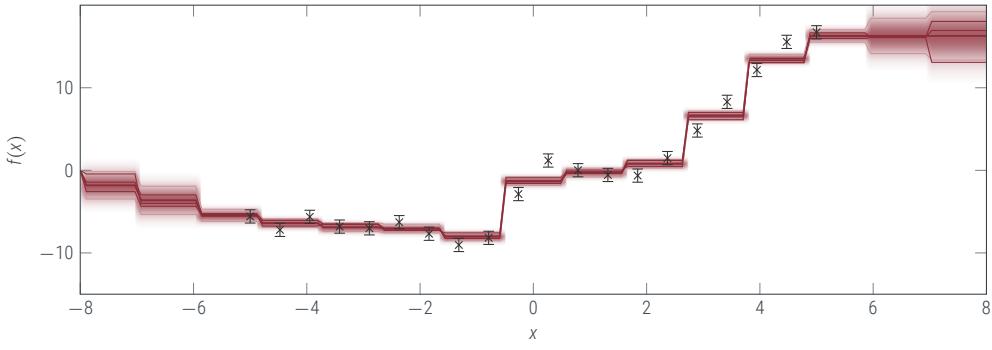
Switch Regression

$$\phi(x) = \begin{bmatrix} \theta(x-8) & \theta(8-x) & \theta(x-7) & \theta(7-x) & \dots \end{bmatrix}^\intercal$$



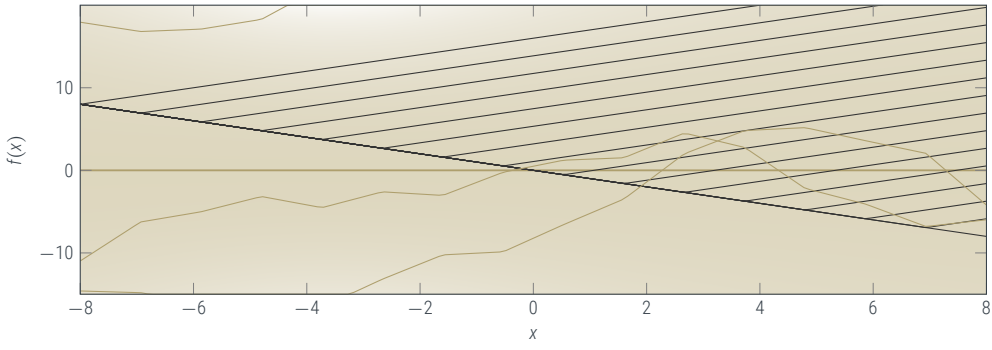
Switch Regression

$$\phi(x) = \begin{bmatrix} \theta(x-8) & \theta(8-x) & \theta(x-7) & \theta(7-x) & \dots \end{bmatrix}^\intercal$$



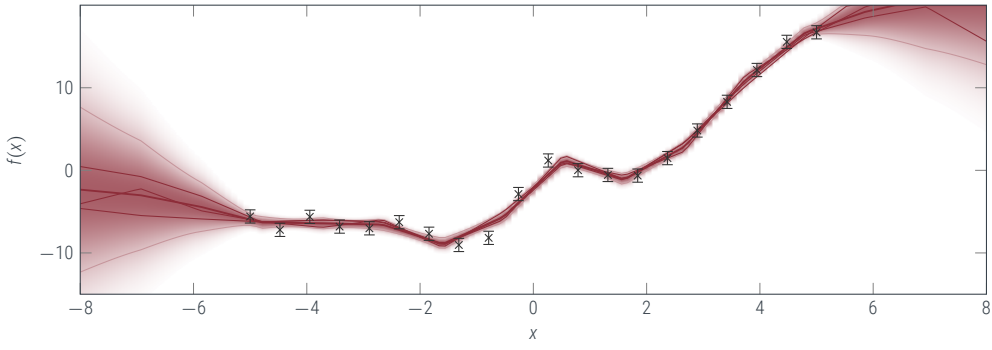
V Regression

$$\phi(x) = \begin{bmatrix} |x-8|-8 & |x-7|-7 & |x-6|-6 & \dots \end{bmatrix}^\intercal$$



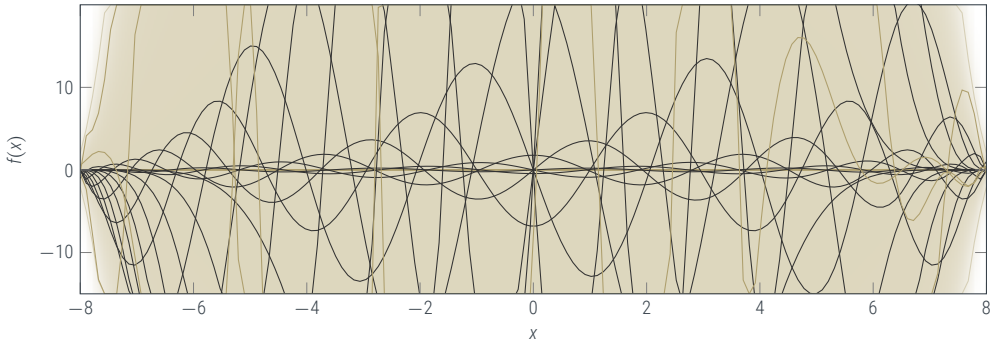
V Regression

$$\phi(x) = \begin{bmatrix} |x-8|-8 & |x-7|-7 & |x-6|-6 & \dots \end{bmatrix}^\intercal$$



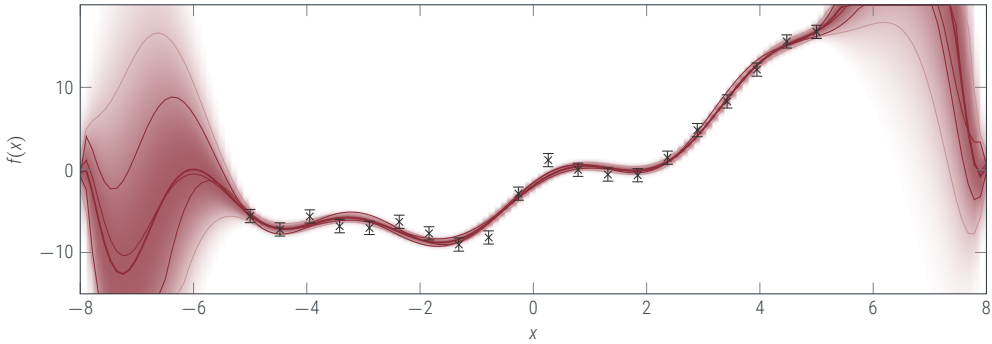
Legendre Regression

$$\phi(x) = \left[b^0 P_0(x), b^1 P_1(x), \ldots, b^{13} P_{13}(x)\right]^\intercal \qquad P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}(x^2 - 1)^n$$



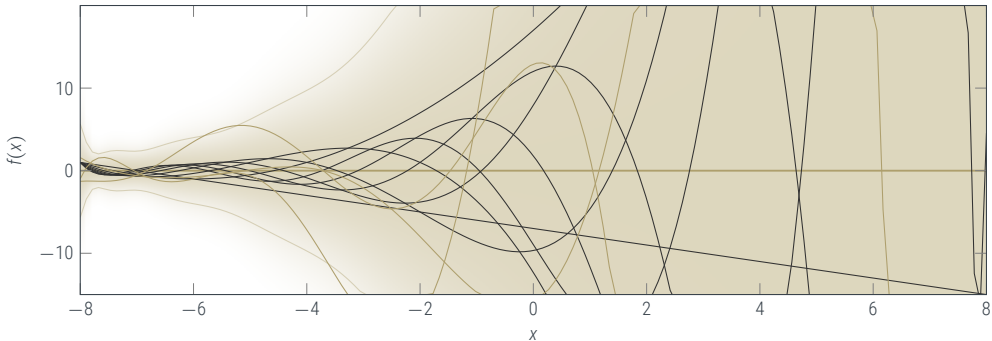
Legendre Regression

$$\phi(x) = \left[b^0 P_0(x), b^1 P_1(x), \ldots, b^{13} P_{13}(x)\right]^\intercal \qquad P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}(x^2 - 1)^n$$



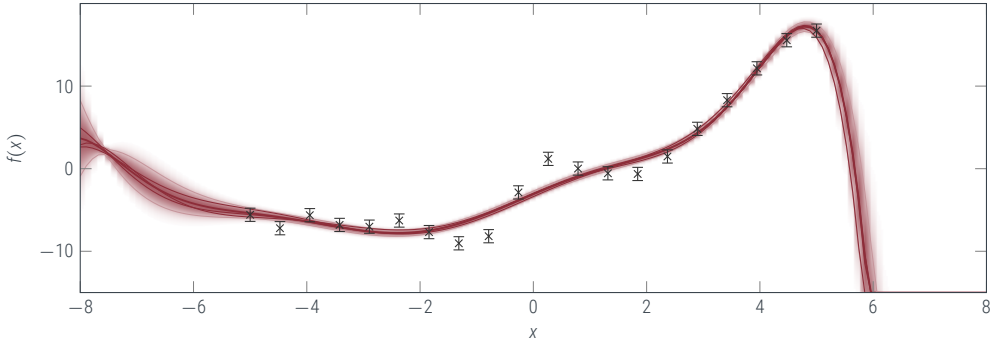
Laguerre Regression

$$\phi(x) = \left[\frac{1}{n!} \left(\frac{d}{dx} - 1 \right)^n x^n \right]^\intercal_{n=1,\dots,8}$$



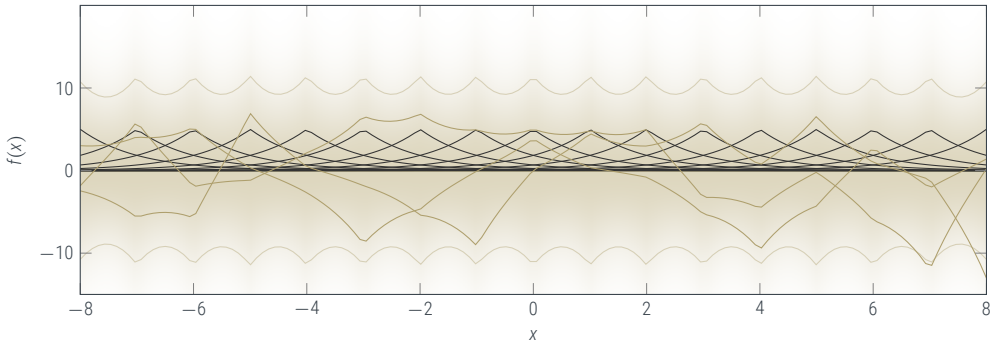
Laguerre Regression

$$\phi(x) = \left[\frac{1}{n!} \left(\frac{d}{dx} - 1 \right)^n x^n \right]^\intercal_{n=1,\ldots,8}$$



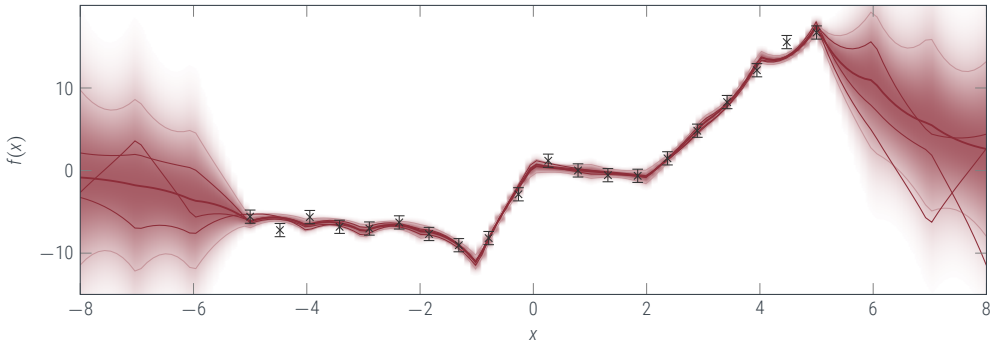
Eiffel Tower Regression

$$\phi(x) = \begin{bmatrix} e^{-|x-8|} & e^{-|x-7|} & e^{-|x-6|} & \dots \end{bmatrix}^\intercal$$



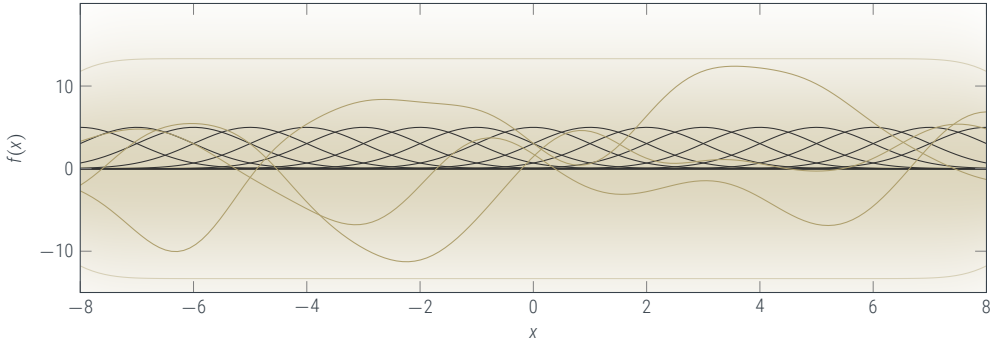
Eiffel Tower Regression

$$\phi(x) = \begin{bmatrix} e^{-|x-8|} & e^{-|x-7|} & e^{-|x-6|} & \dots \end{bmatrix}^\intercal$$



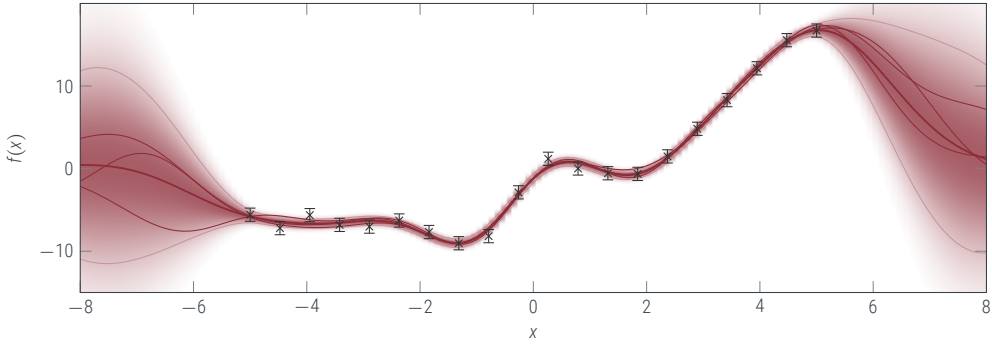
Bell Curve Regression

$$\phi(x) = \begin{bmatrix} e^{-\frac{1}{2}(x-8)^2} & e^{-\frac{1}{2}(x-7)^2} & e^{-\frac{1}{2}(x-6)^2} & \dots \end{bmatrix}^\intercal$$



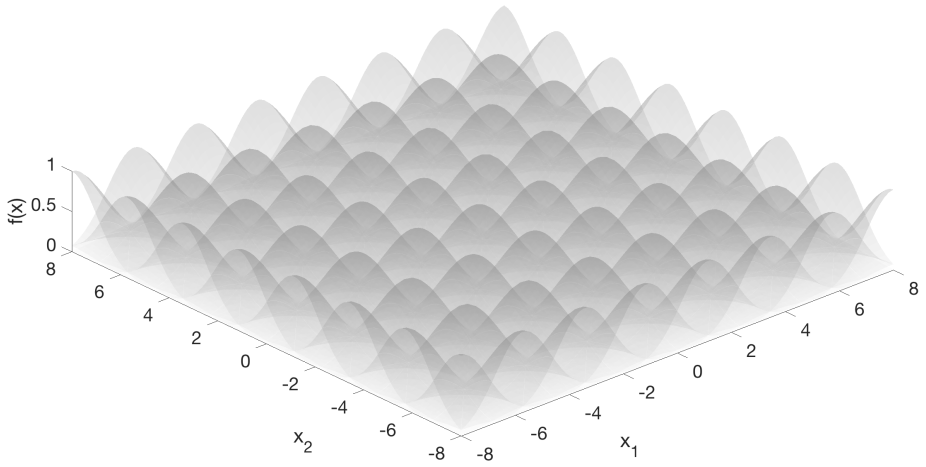
Bell Curve Regression

$$\phi(x) = \begin{bmatrix} e^{-\frac{1}{2}(x-8)^2} & e^{-\frac{1}{2}(x-7)^2} & e^{-\frac{1}{2}(x-6)^2} & \dots \end{bmatrix}^\intercal$$



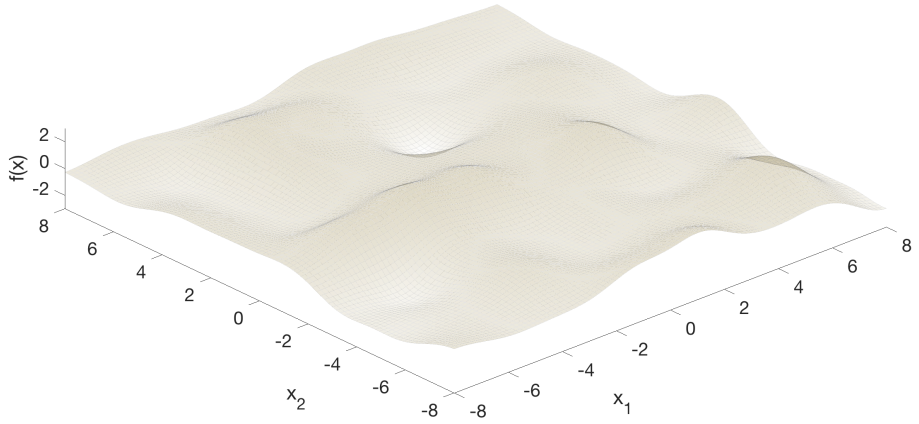
Multiple Inputs

Input domain $\mathbb{X}$ can be anything



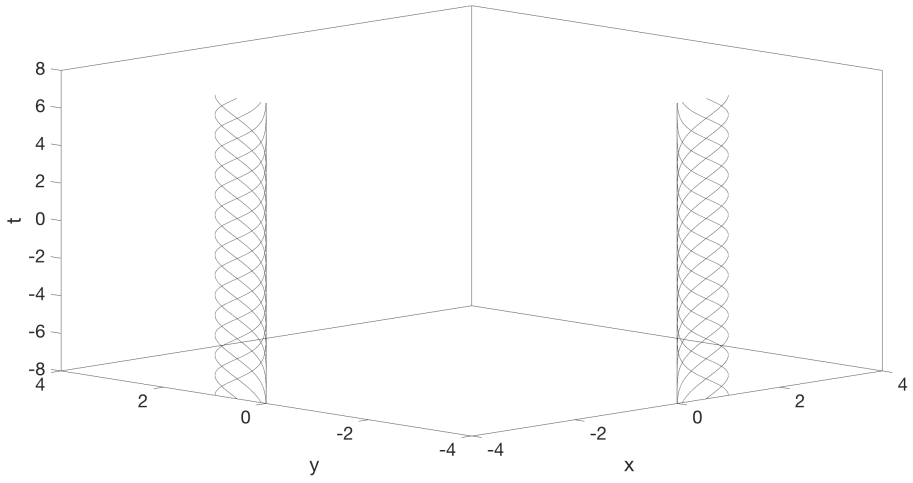
Multiple Inputs

Input domain $\mathbb{X}$ can be anything



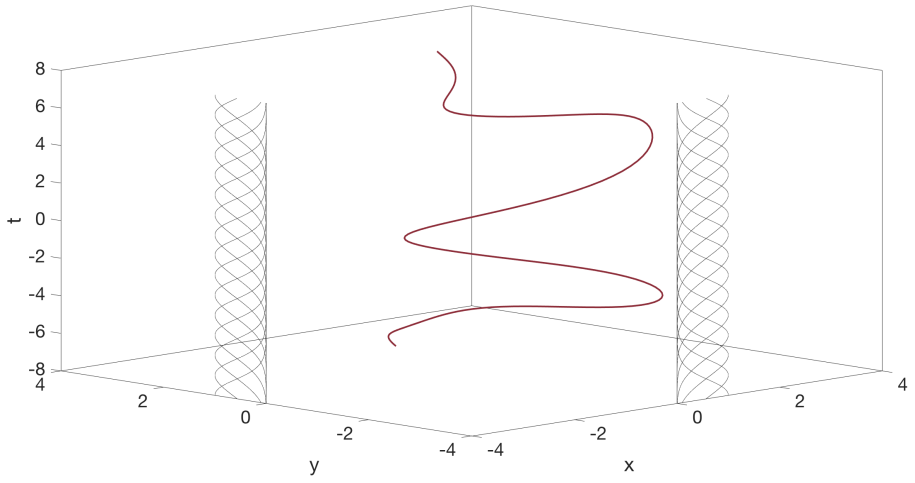
Multiple Outputs

The output domain can be anything isomorphic to $\mathbb{R}^N$



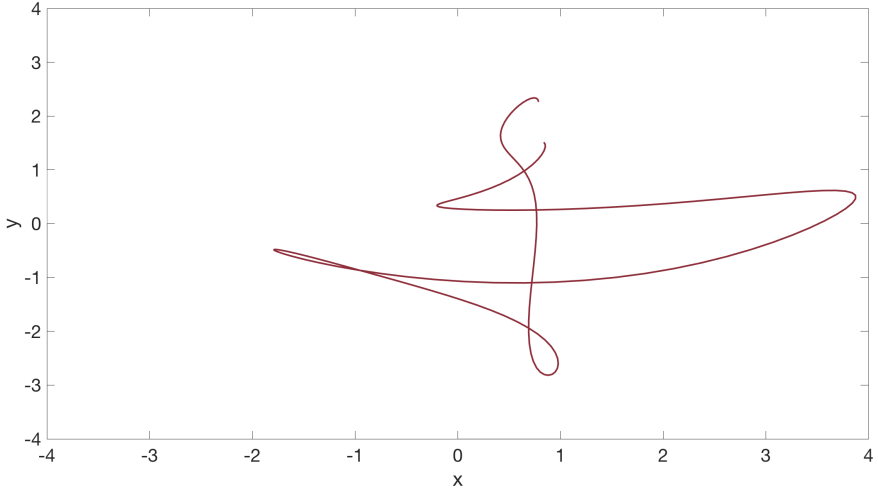
Multiple Outputs

The output domain can be anything isomorphic to $\mathbb{R}^N$



Multiple Outputs

The output domain can be anything isomorphic to $\mathbb{R}^N$



Summary:

- Gaussian distributions can be used to learn functions
- Analytical inference is possible using general linear models

$$f(x) = \phi(x)^\intercal \boldsymbol{w} = \phi_x^\intercal \boldsymbol{w}$$

- Then the posterior on both w and f is Gaussian
- The choice of features $\phi : \mathbb{X} \to \mathbb{R}$ is essentially unconstrained