MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

6.7900: Machine Learning
Lecture 16

Lecture start: Tues/Thurs 2:35pm

Who's speaking today? Prof. Tamara Broderick

Course website: gradml.mit.edu

Questions? Ask here or on piazza.com/mit/fall2024/67900/

Materials: Slides, video, etc linked from gradml.mit.edu after the lecture (but there is no livestream)

Last Times

I. Various supervised learning methods: linear, polynomial, trees, forests, neural networks

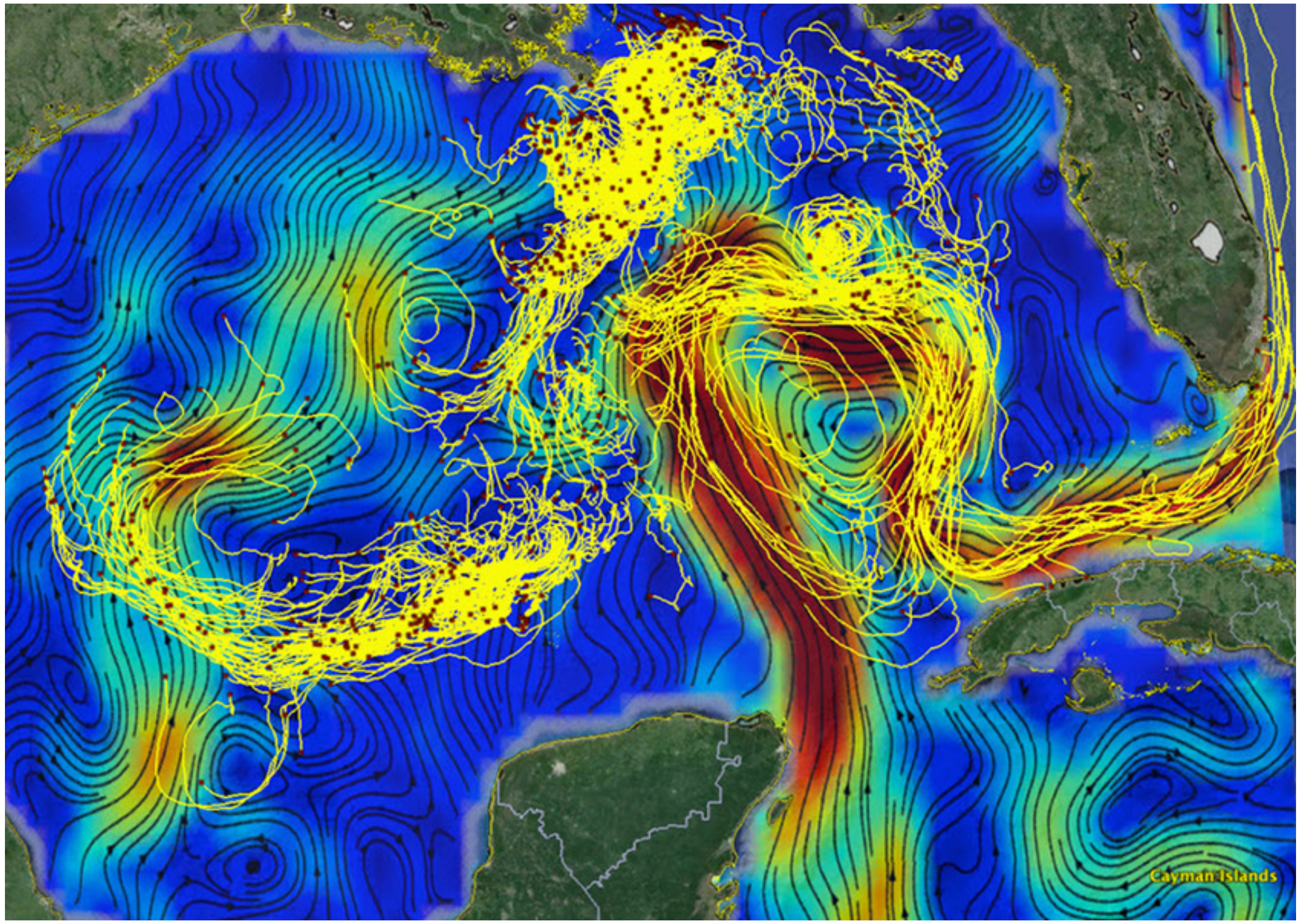
Today

I. A common motif in spatiotemporal (and similar) data

II. Gaussian processes: model and inference

Sub-type of regression

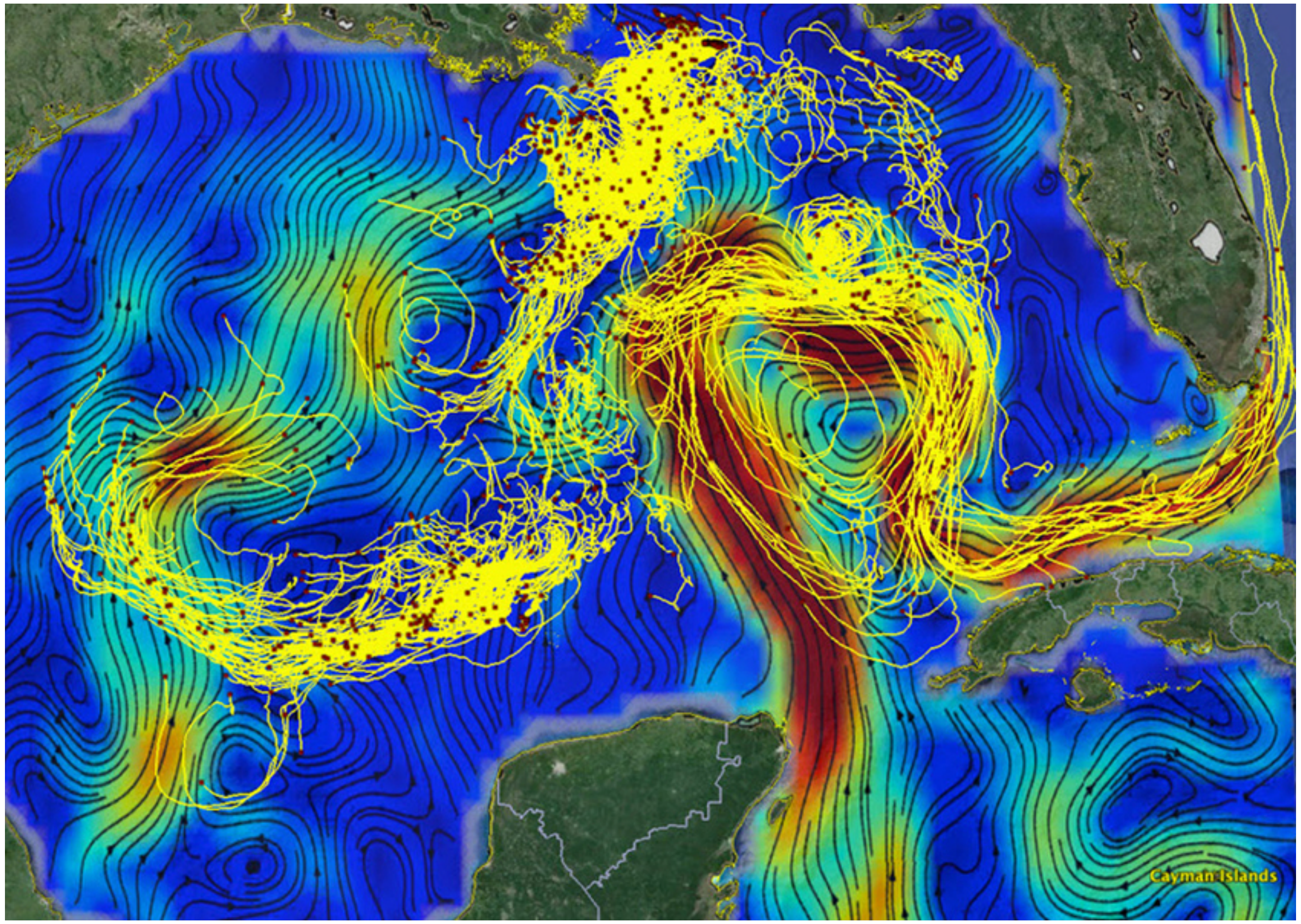
Sub-type of regression



Example:

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Sub-type of regression

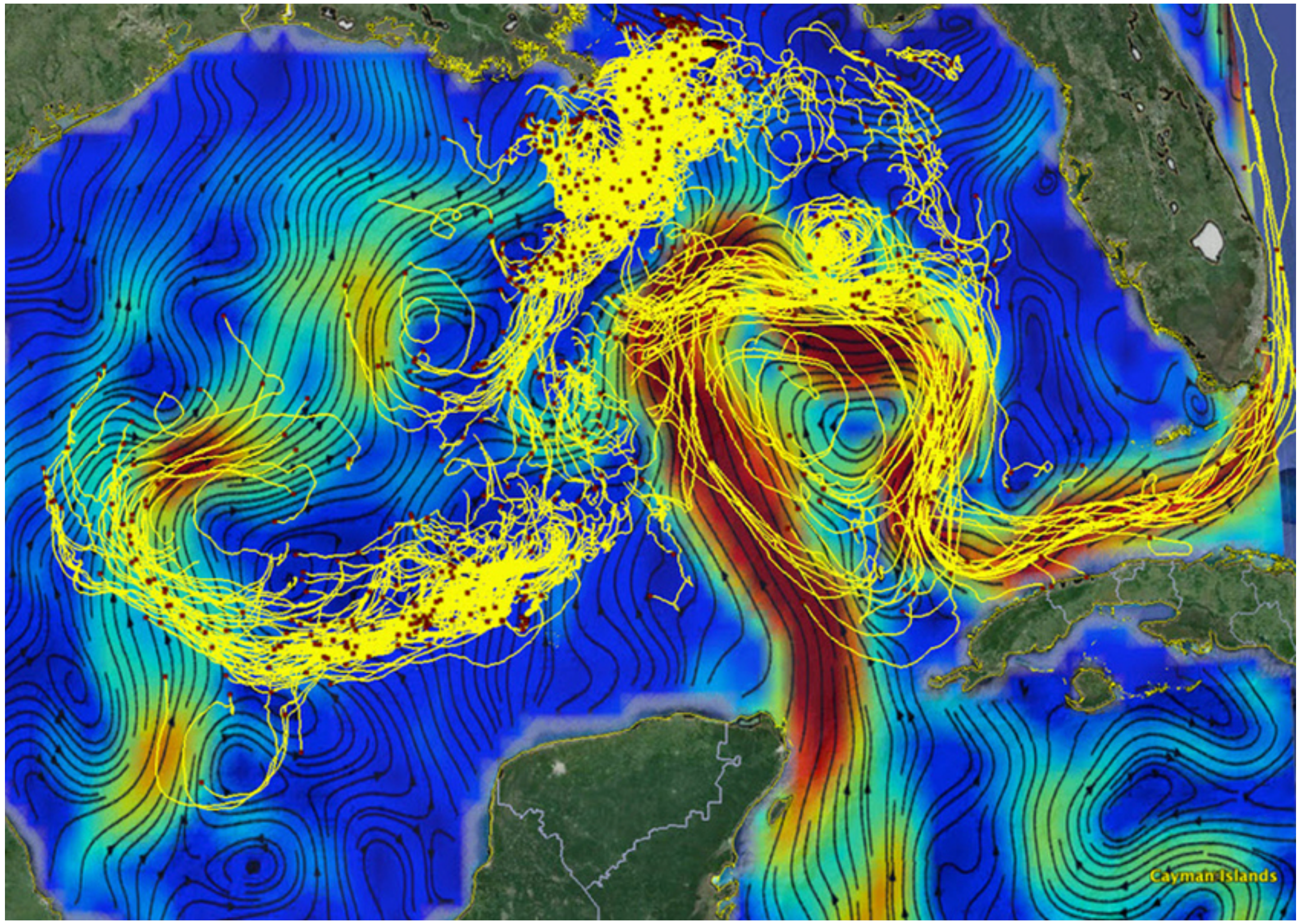


Example:

- The ocean current (velocity vector field) varies by space & time

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Sub-type of regression

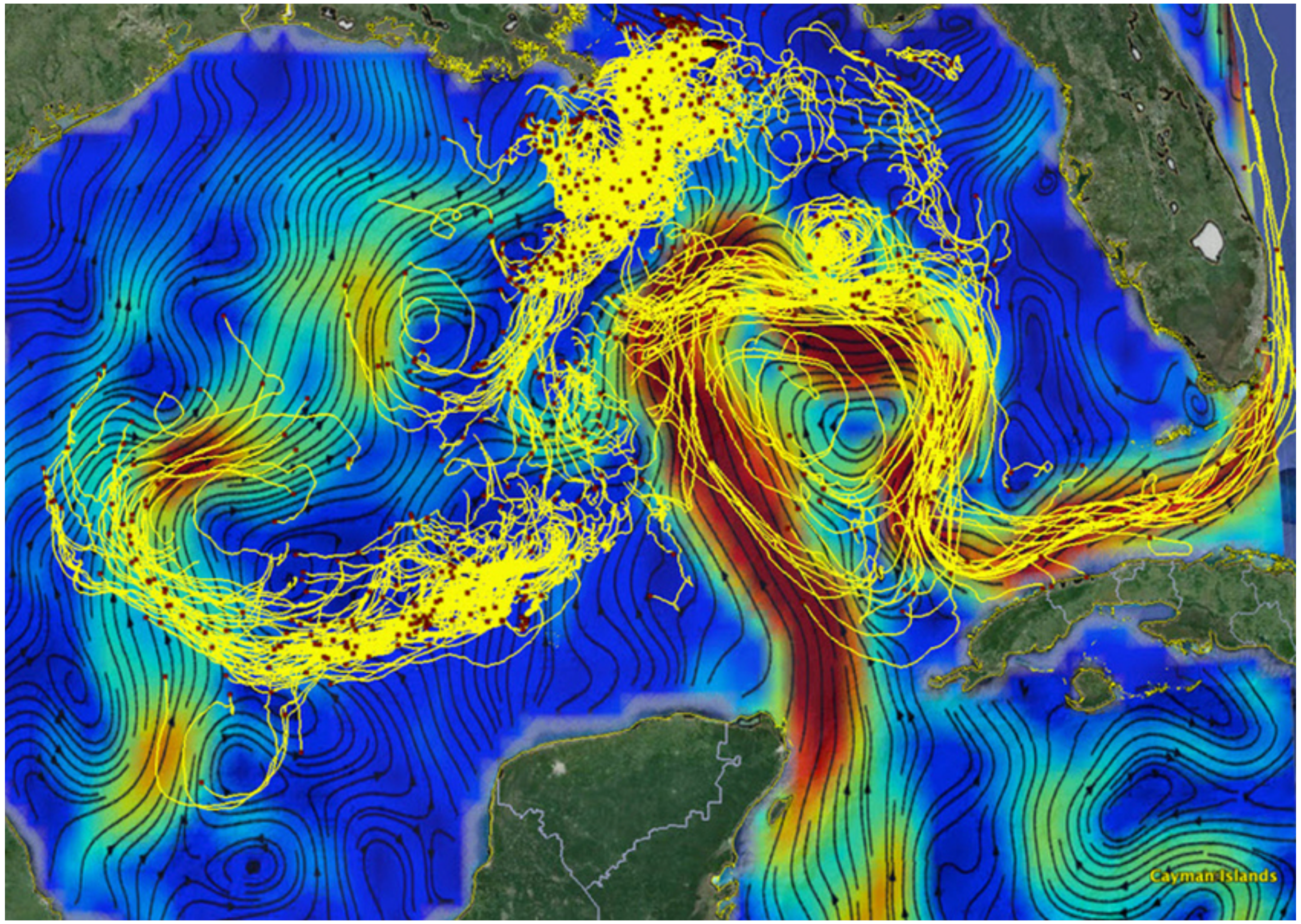


Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Sub-type of regression

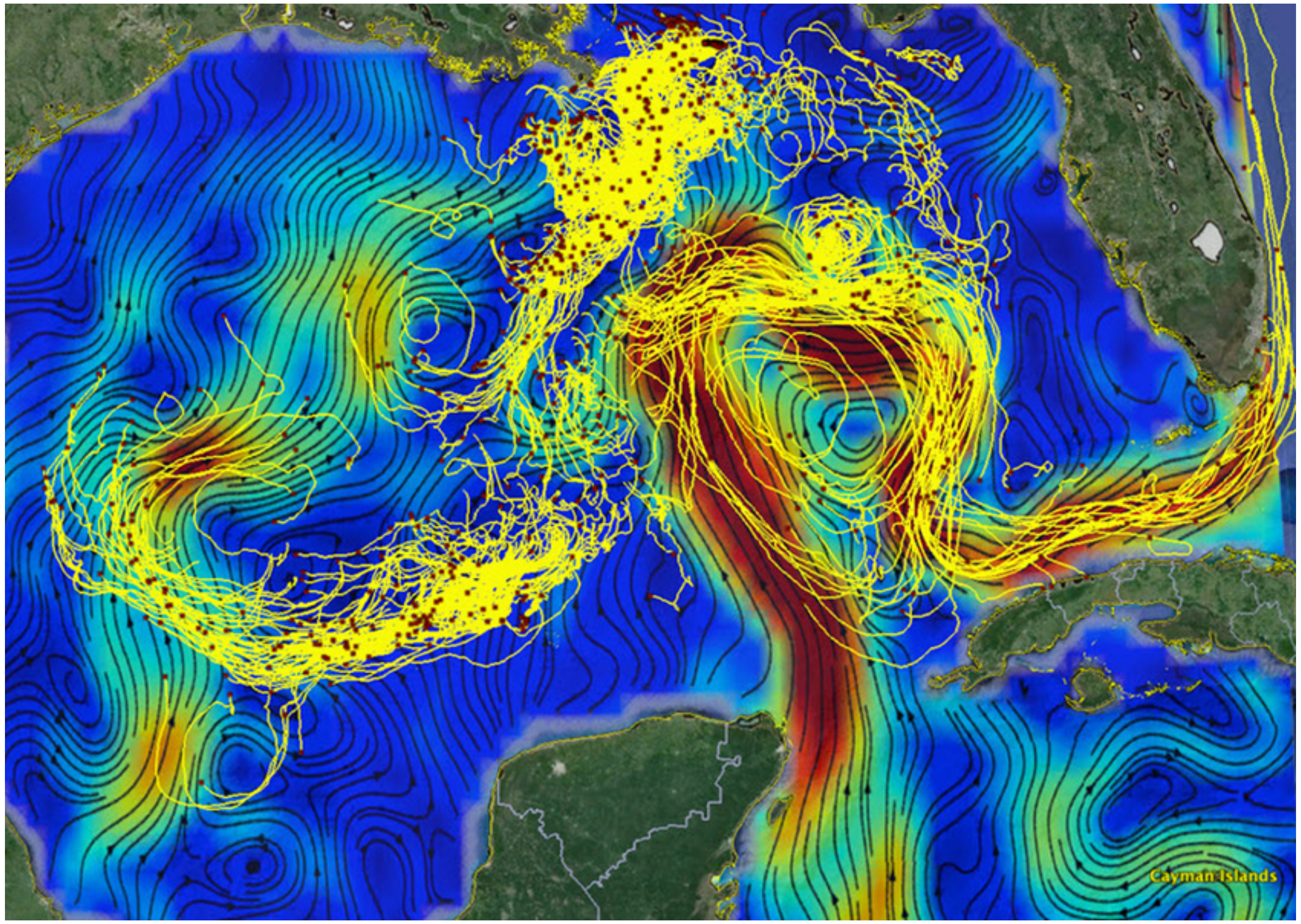


Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Sub-type of regression



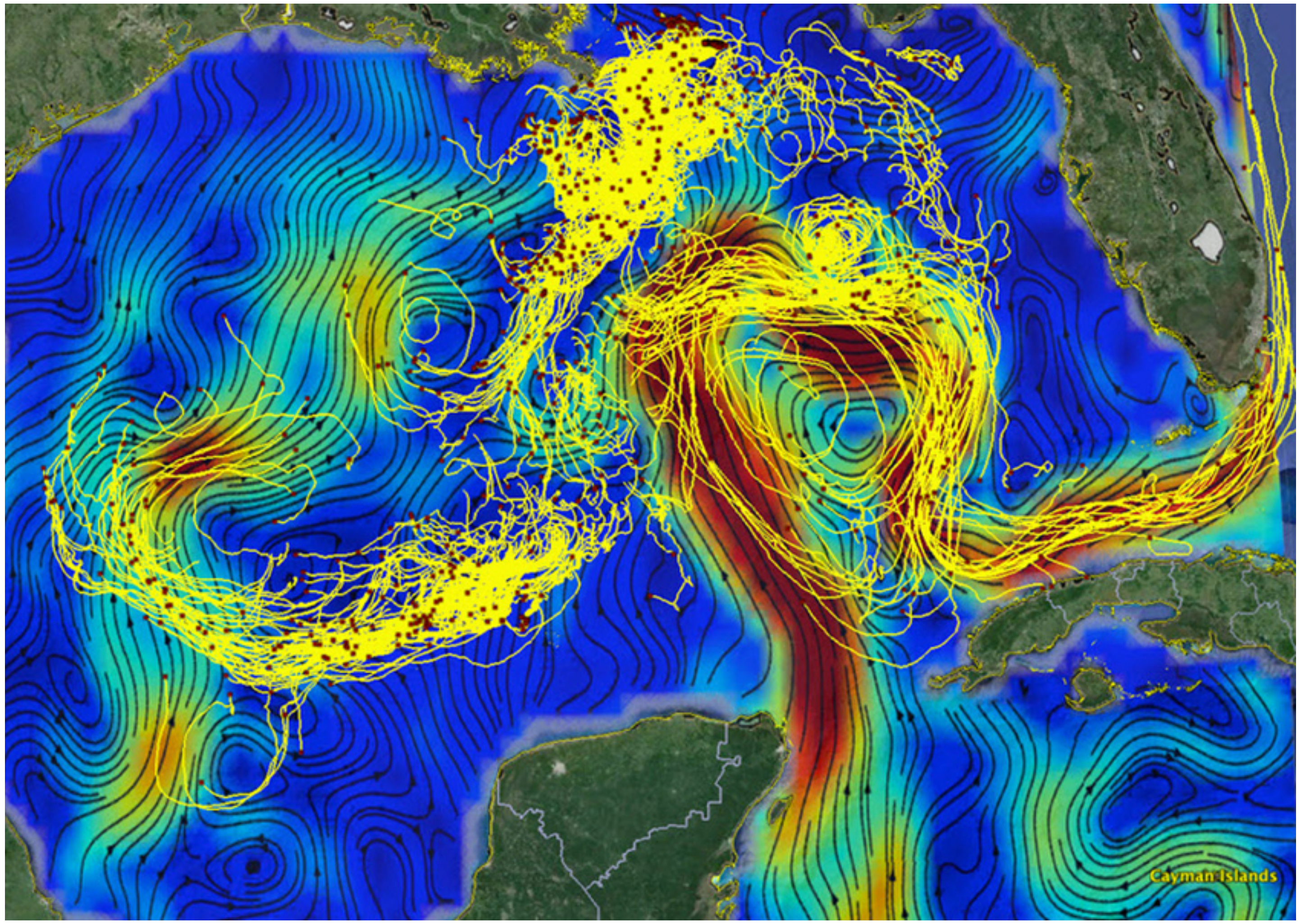
Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Challenges: • Data is difficult to collect and sparse

Sub-type of regression



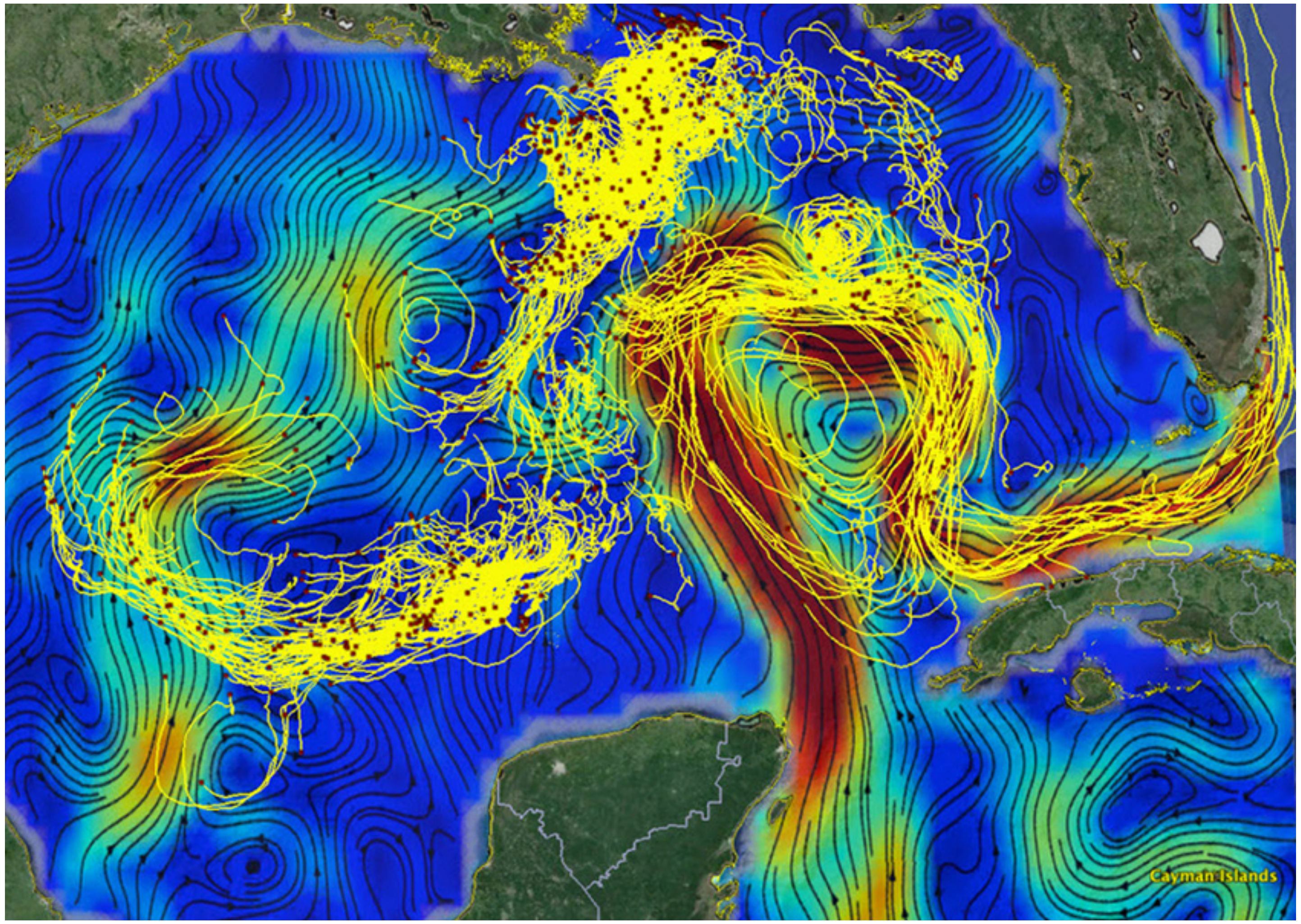
Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Challenges:
- Data is difficult to collect and sparse
- Data is not on a grid or discrete structure

Sub-type of regression



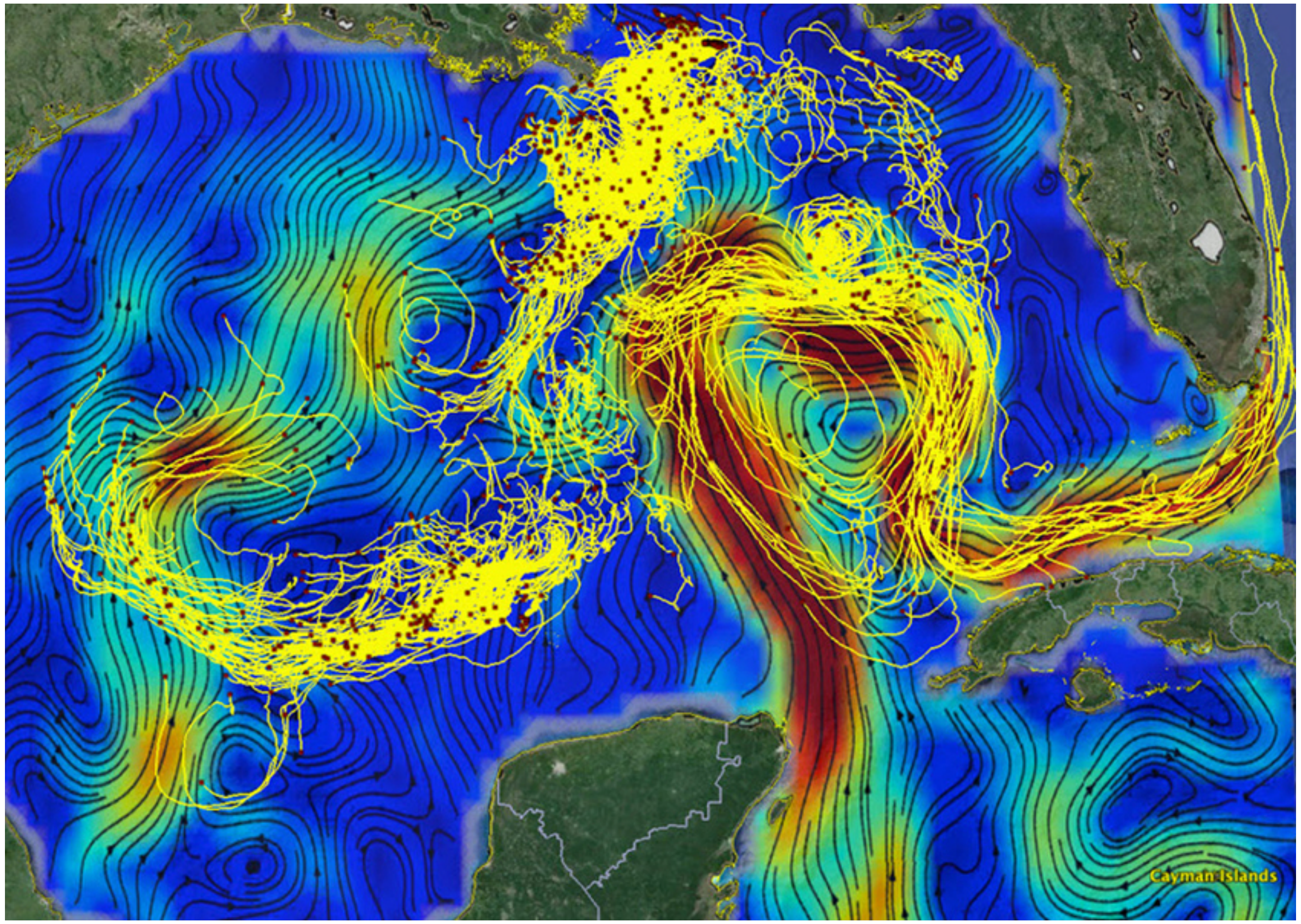
Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Challenges:
- Data is difficult to collect and sparse
- Data is not on a grid or discrete structure
- Current is highly nonlinear but smooth in space-time

Sub-type of regression



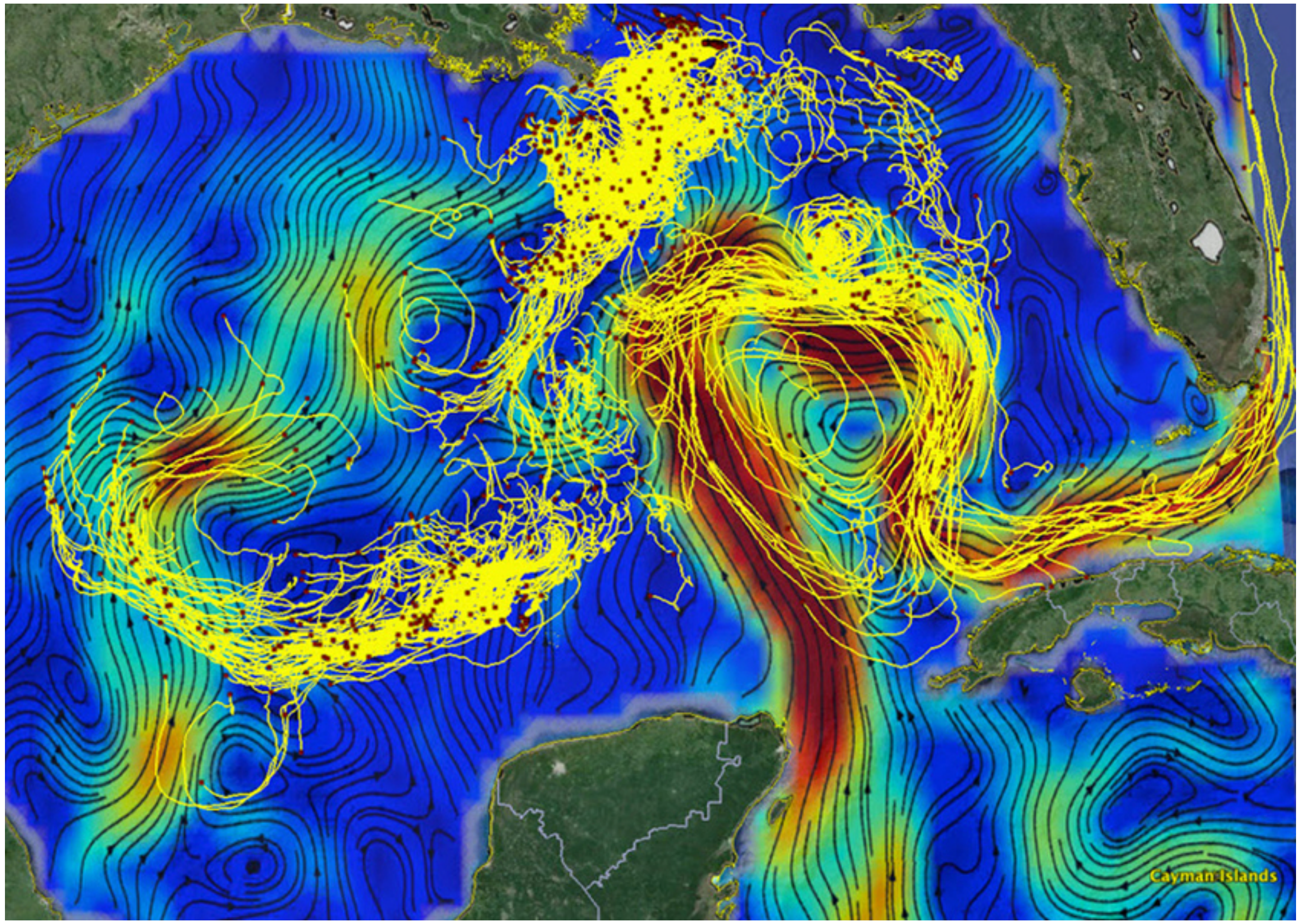
Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Challenges:
- Data is difficult to collect and sparse
- Data is not on a grid or discrete structure
- Current is highly nonlinear but smooth in space-time
- Want uncertainty quantification

Sub-type of regression



Example:

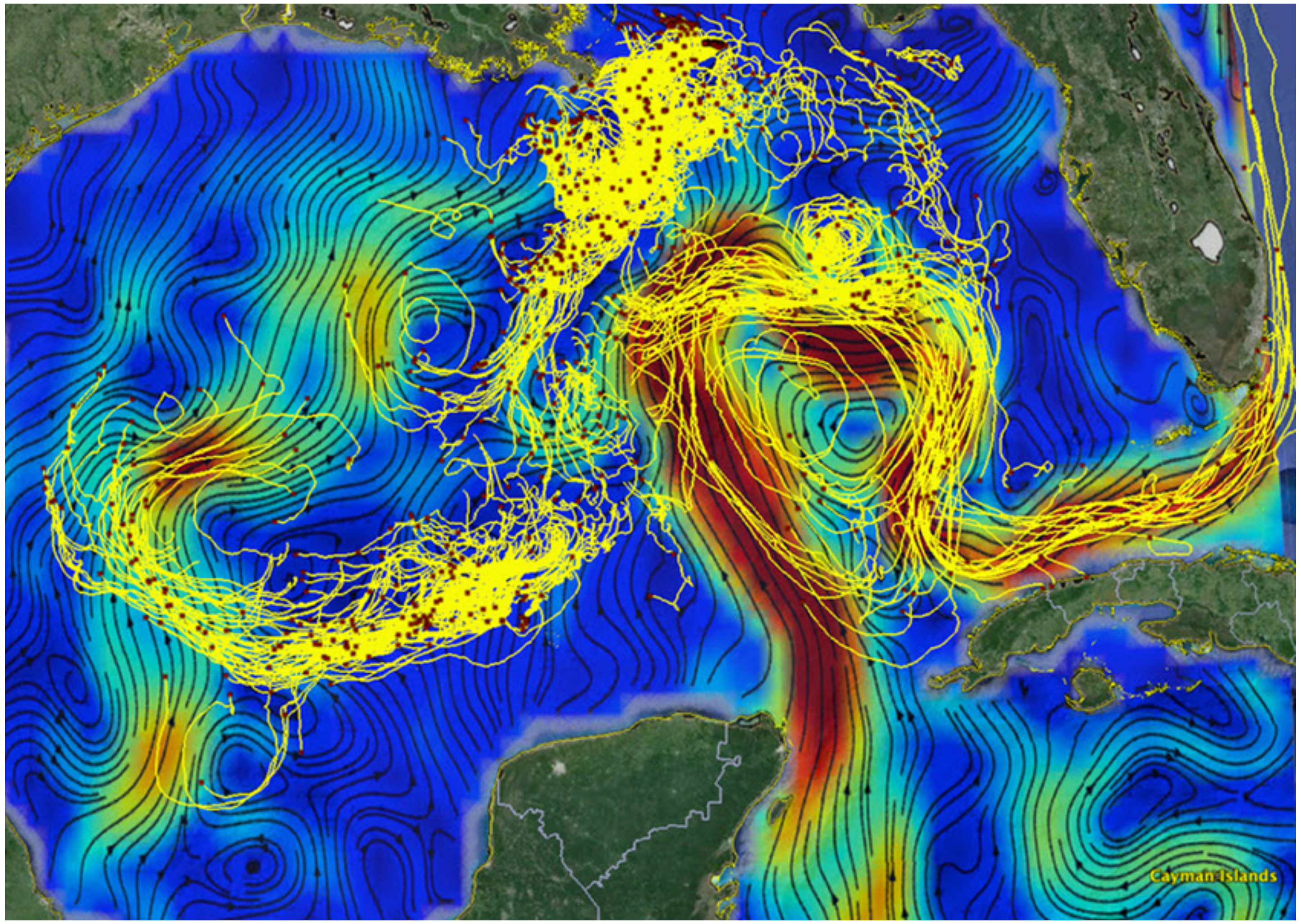
- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Challenges:
- Data is difficult to collect and sparse
- Data is not on a grid or discrete structure
- Current is highly nonlinear but smooth in space-time
- Want uncertainty quantification

Many similar problems: e.g. predicting groundwater quality from well measurements, etc. [Belkhiri et al 2020]

Sub-type of regression



Example:

- The ocean current (velocity vector field) varies by space & time
- Scientists get sparse observations of the current from buoys
- Goal: predict the current at other locations/times

[Ryan, Özgökmen 2023; Zewe 2023; Gonçalves et al 2019; Lodise et al 2020; Berlinghieri et al 2023]

Challenges:
- Data is difficult to collect and sparse
- Data is not on a grid or discrete structure
- Current is highly nonlinear but smooth in space-time
- Want uncertainty quantification

Many similar (low dimensional) problems: e.g. predicting groundwater quality from well measurements, etc. [Belkhiri et al 2020]

Sub-type of regression

Sub-type of regression

Example: “Surrogate model”

Sub-type of regression

Example: “Surrogate model”

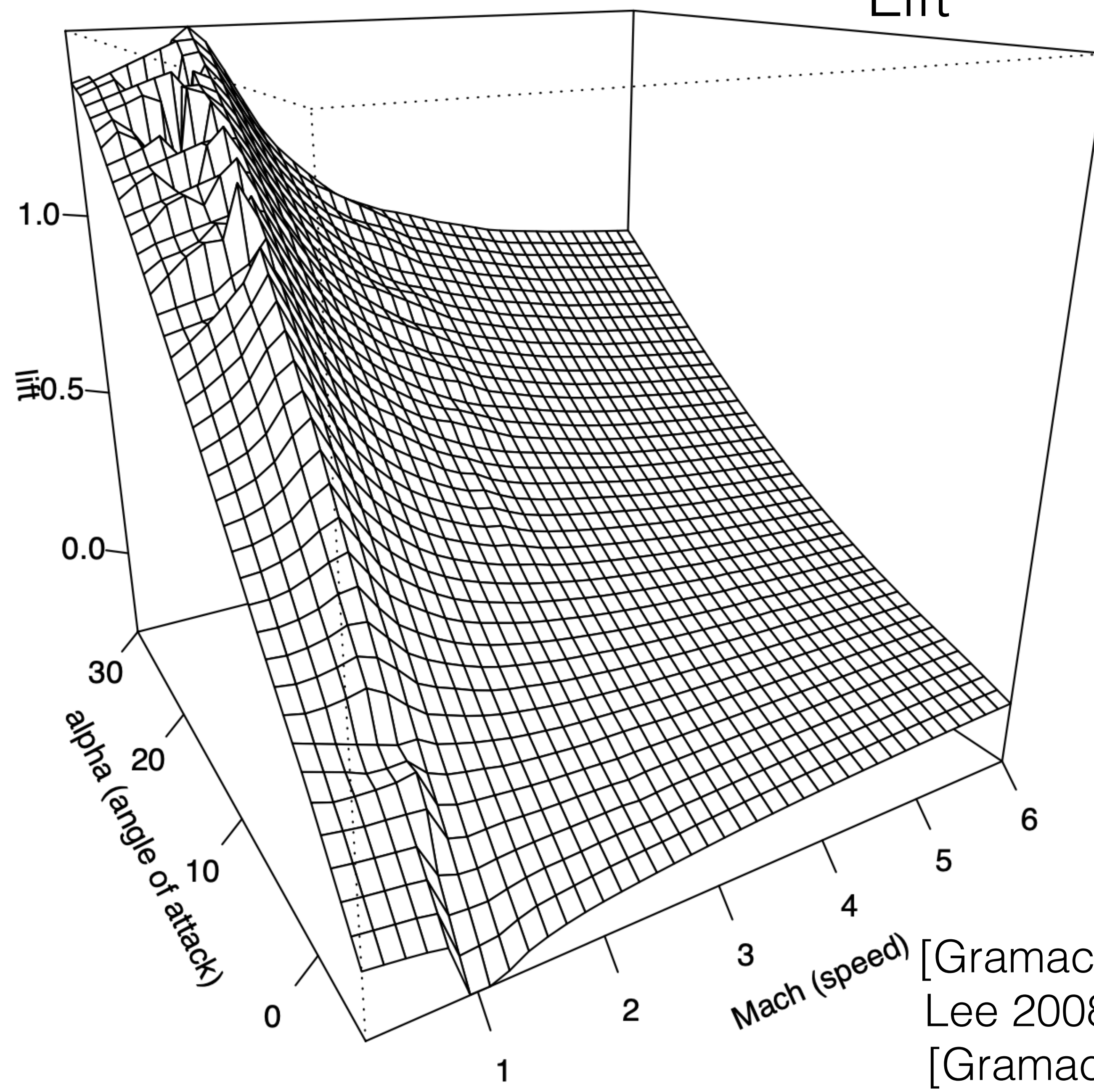
- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle

Sub-type of regression

Example: “Surrogate model”

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle

Lift



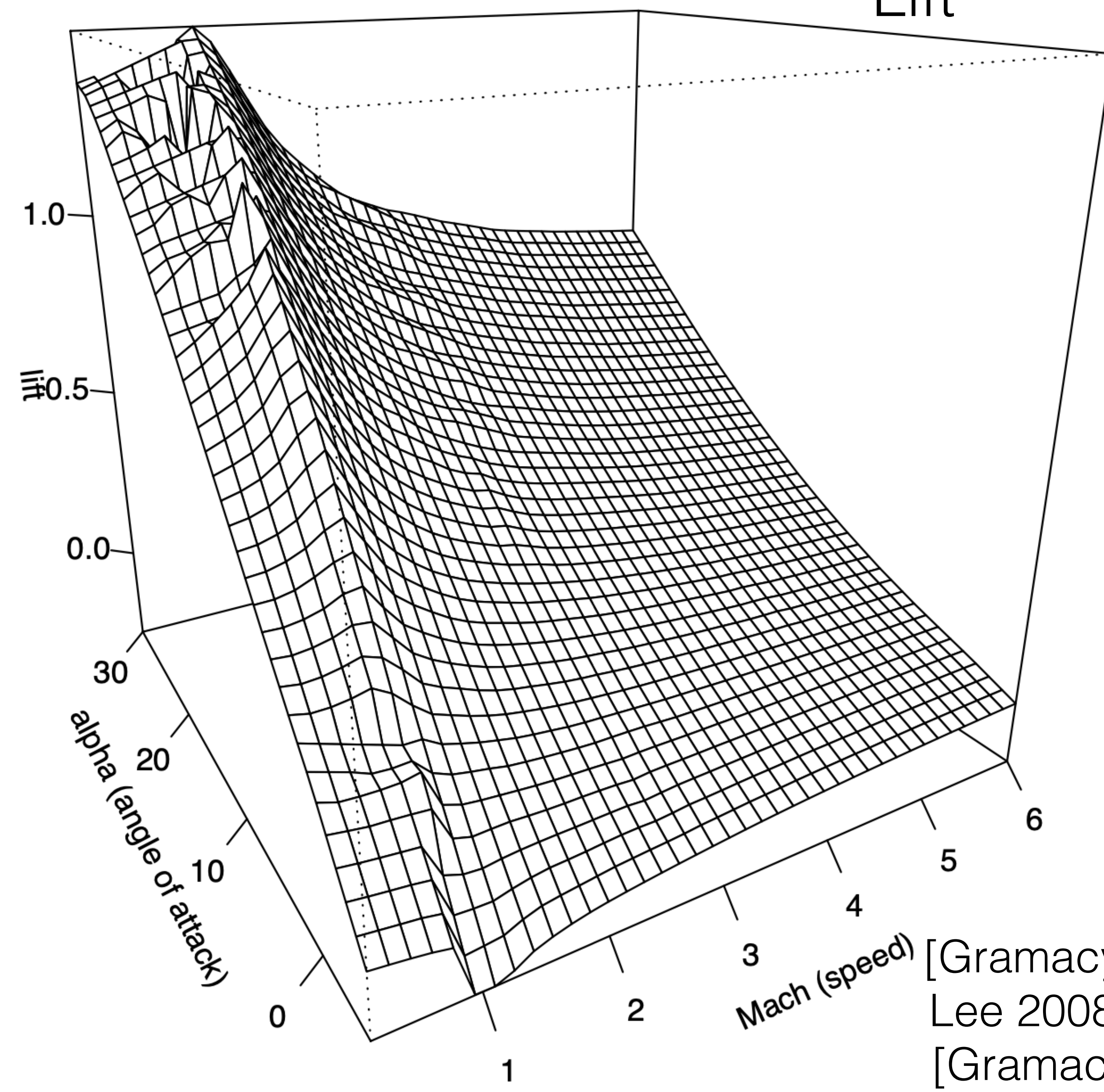
[Gramacy, Lee 2008] [Gramacy 2020]

Sub-type of regression

Example: “Surrogate model”

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle
- Scientists can run expensive simulations at chosen input settings

Lift



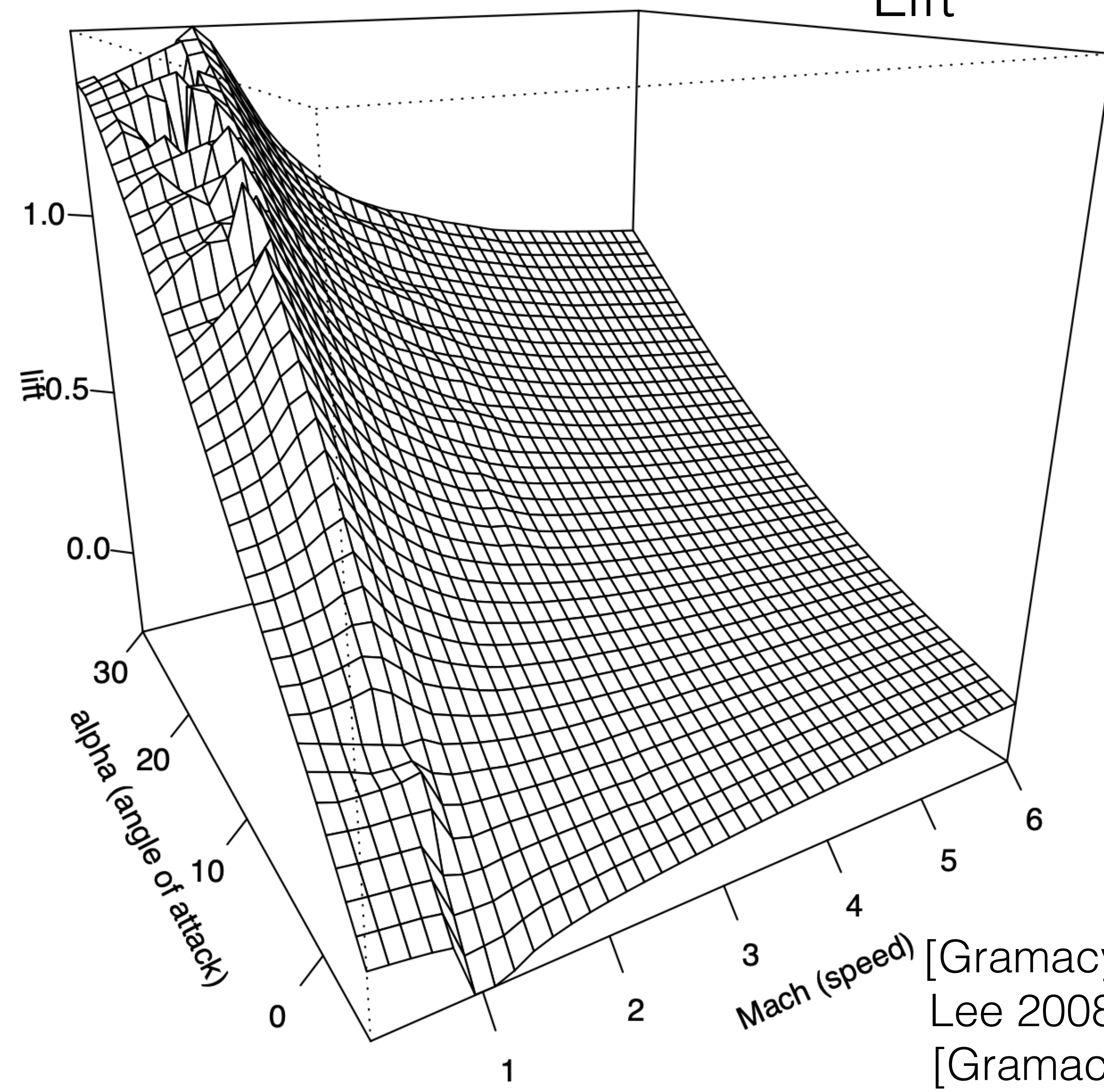
[Gramacy, Lee 2008] [Gramacy 2020]

Sub-type of regression

Example: "Surrogate model"

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle
- Scientists can run expensive simulations at chosen input settings
- Goal: predict how lift varies as a function of these inputs

Lift



[Gramacy, Lee 2008] [Gramacy 2020]

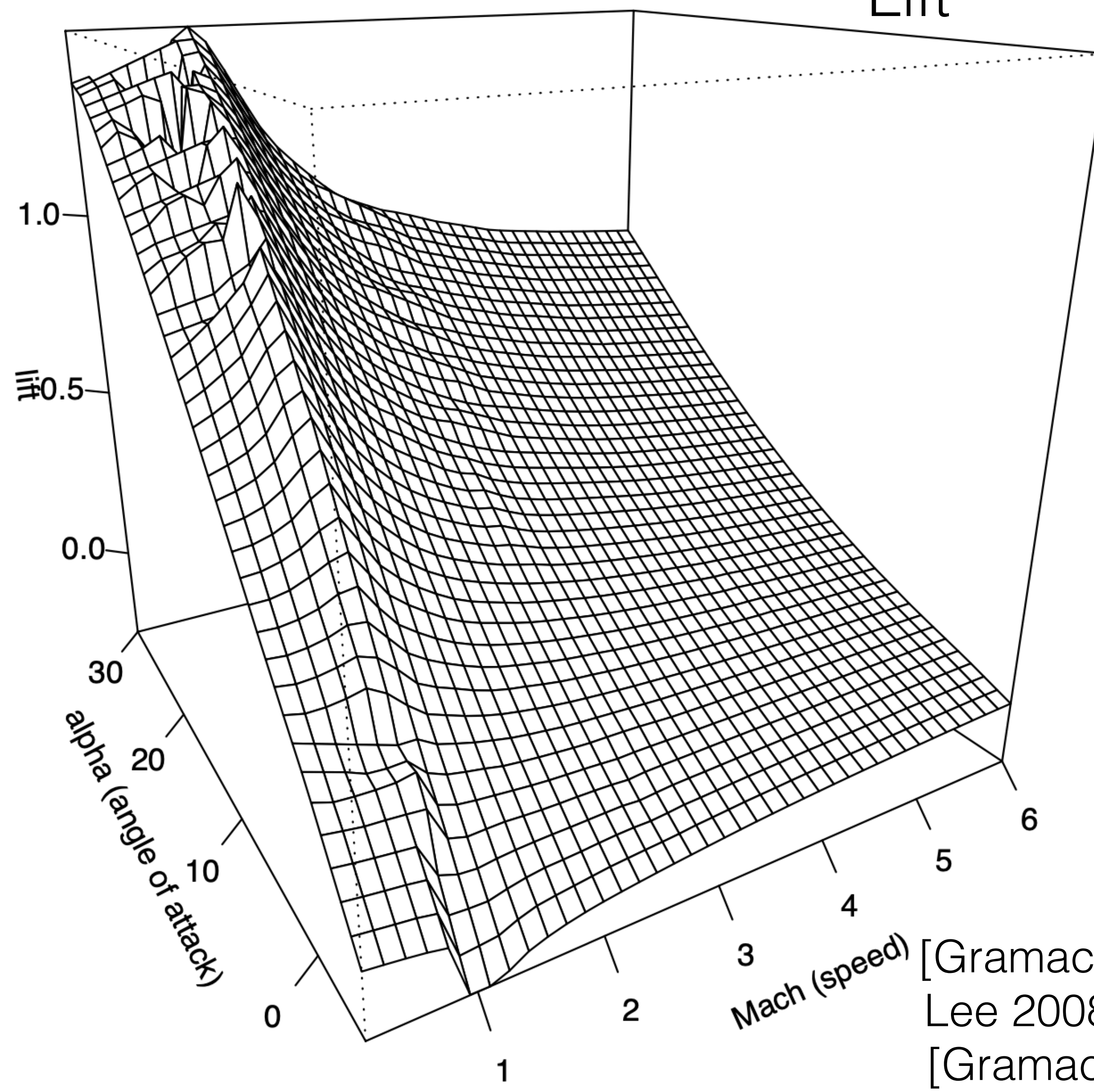
Sub-type of regression

Example: “Surrogate model”

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle
- Scientists can run expensive simulations at chosen input settings
- Goal: predict how lift varies as a function of these inputs

Challenges: • Sparse data

Lift



[Gramacy, Lee 2008] [Gramacy 2020]

Sub-type of regression

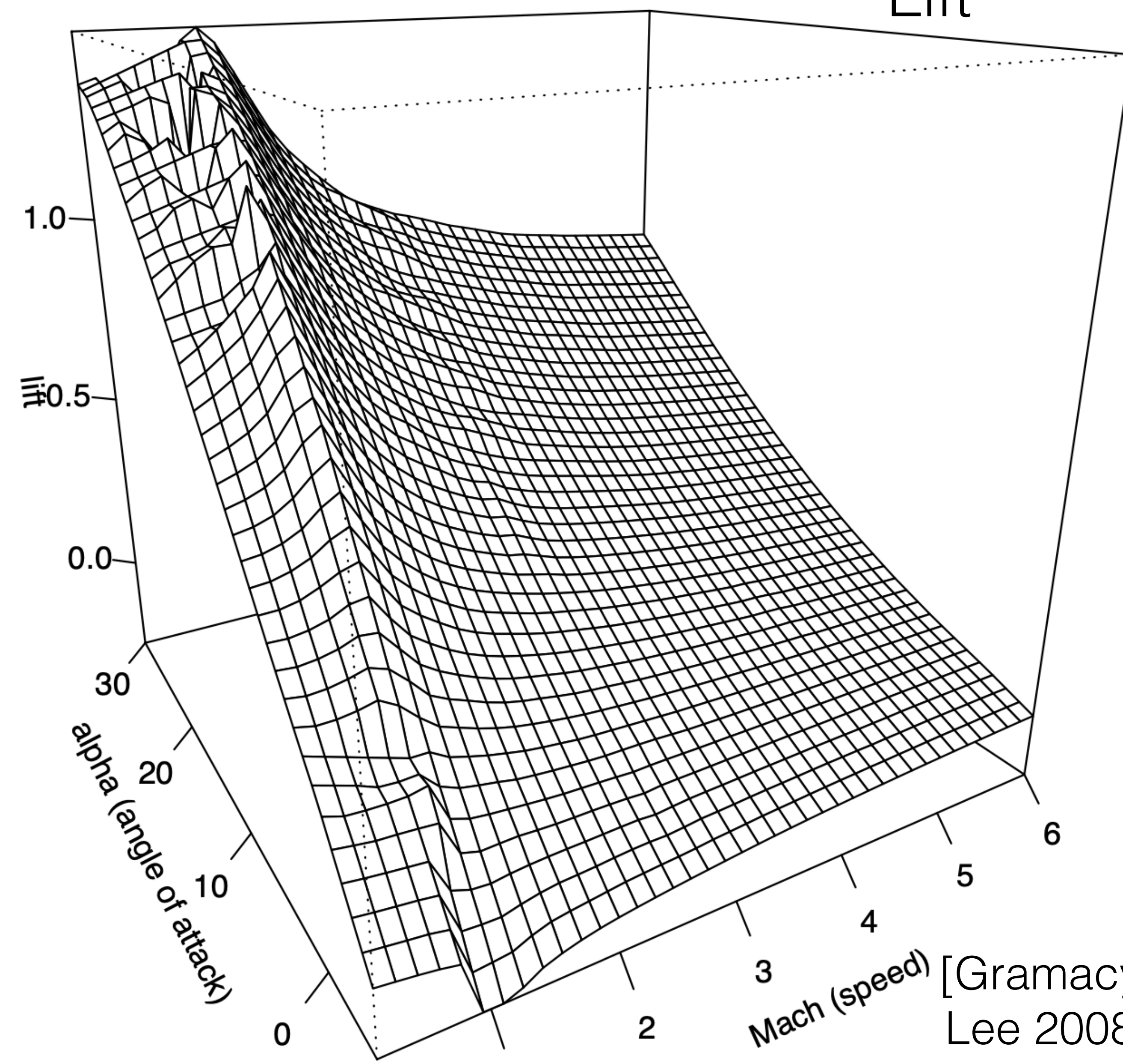
Example: “Surrogate model”

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle
- Scientists can run expensive simulations at chosen input settings
- Goal: predict how lift varies as a function of these inputs

Challenges: • Sparse data

- Lift may have a nonlinear relationship to the inputs

Lift

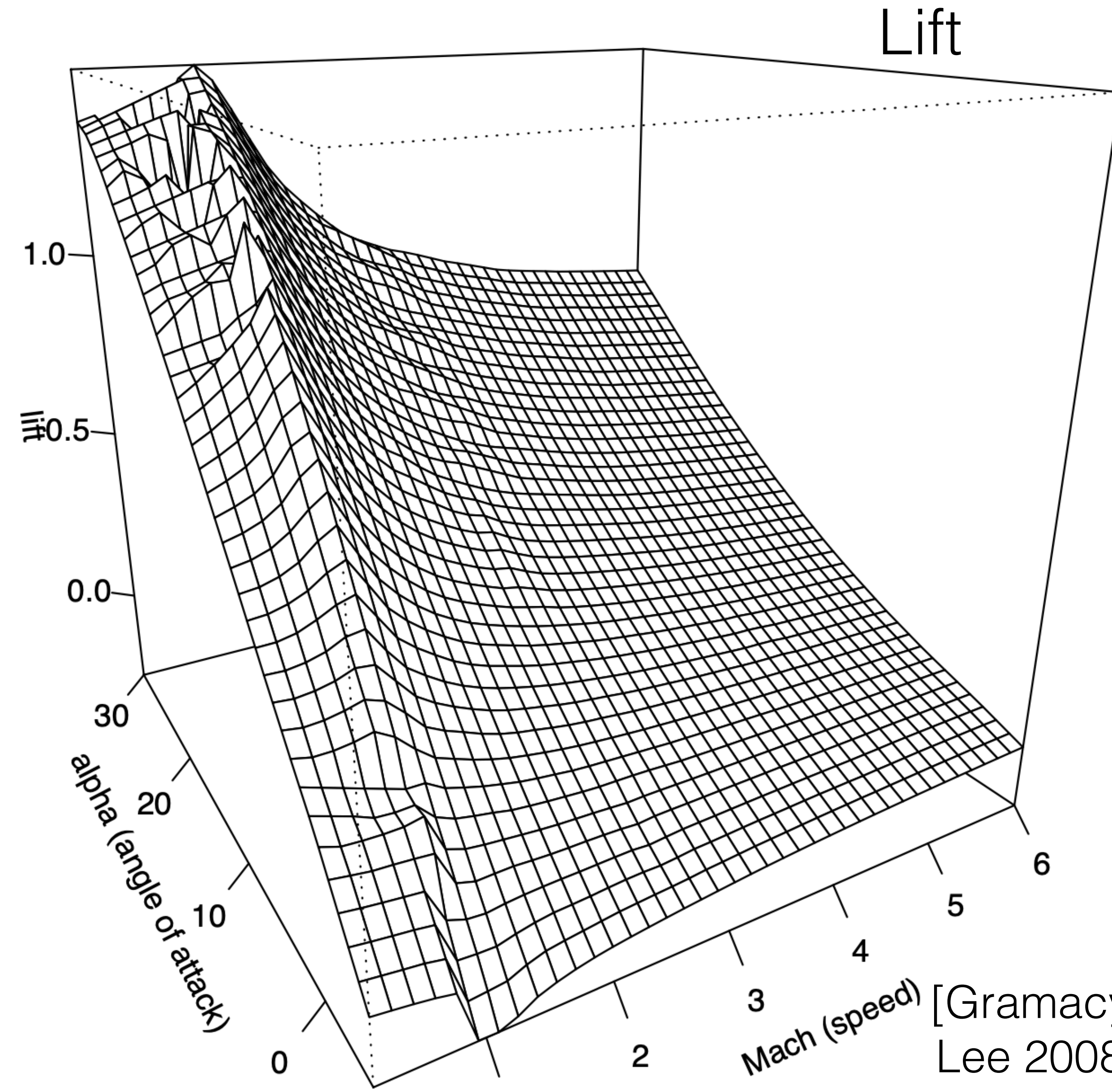


[Gramacy, Lee 2008] [Gramacy 2020]

Sub-type of regression

Example: “Surrogate model”

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle
- Scientists can run expensive simulations at chosen input settings
- Goal: predict how lift varies as a function of these inputs



[Gramacy, Lee 2008] [Gramacy 2020]

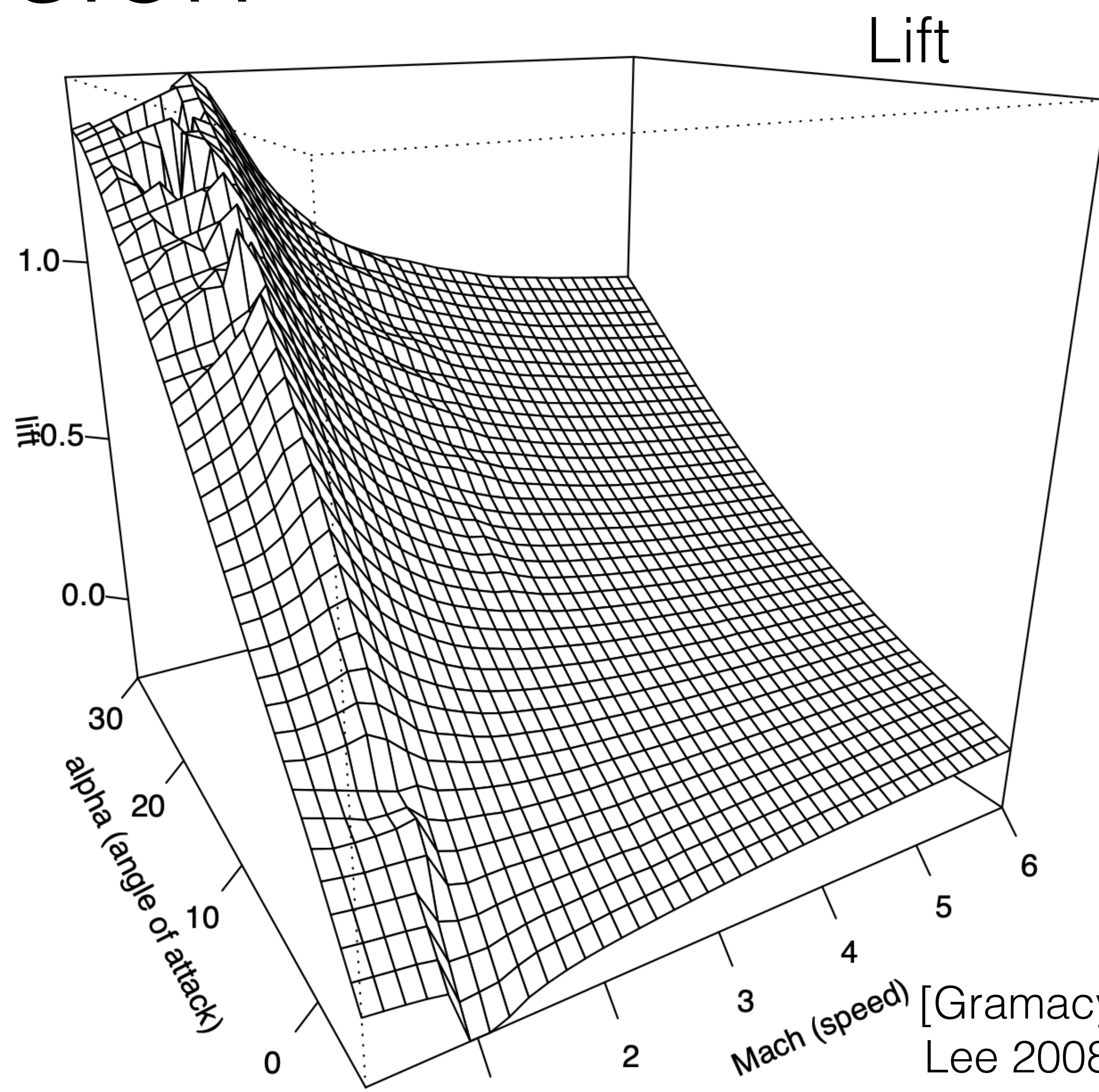
Challenges: • Sparse data

- Lift may have a nonlinear relationship to the inputs
- Want uncertainty quantification

Sub-type of regression

Example: “Surrogate model”

- The lift force of a rocket booster varies as a function of speed at re-entry, angle of attack, and sideslip angle
- Scientists can run expensive simulations at chosen input settings
- Goal: predict how lift varies as a function of these inputs



[Gramacy, Lee 2008] [Gramacy 2020]

Challenges: • Sparse data

- Lift may have a nonlinear relationship to the inputs
- Want uncertainty quantification

Common similar problem: learn (& optimize) performance in ML as a function of scalar hyperparameters [Snoek et al 2012, 2015; Garnett 2023]

- E.g. tune momentum, learning rate, dropout rates, etc.

A Bayesian approach

A Bayesian approach

- p(unknowns | data)

A Bayesian approach

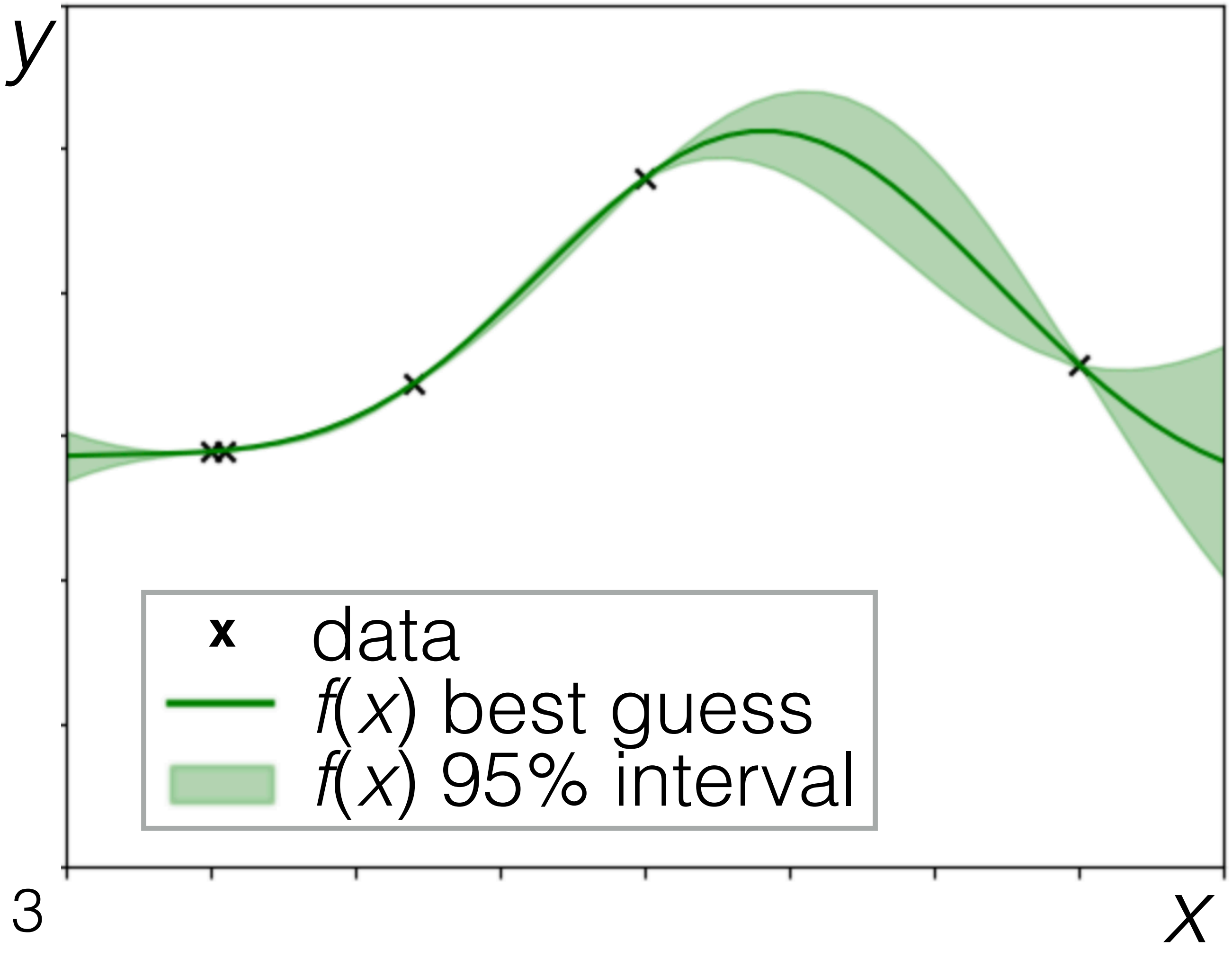
- p(unknowns | data)

Given the data we've seen, what do we know about the underlying function?

A Bayesian approach

- p(unknowns | data)

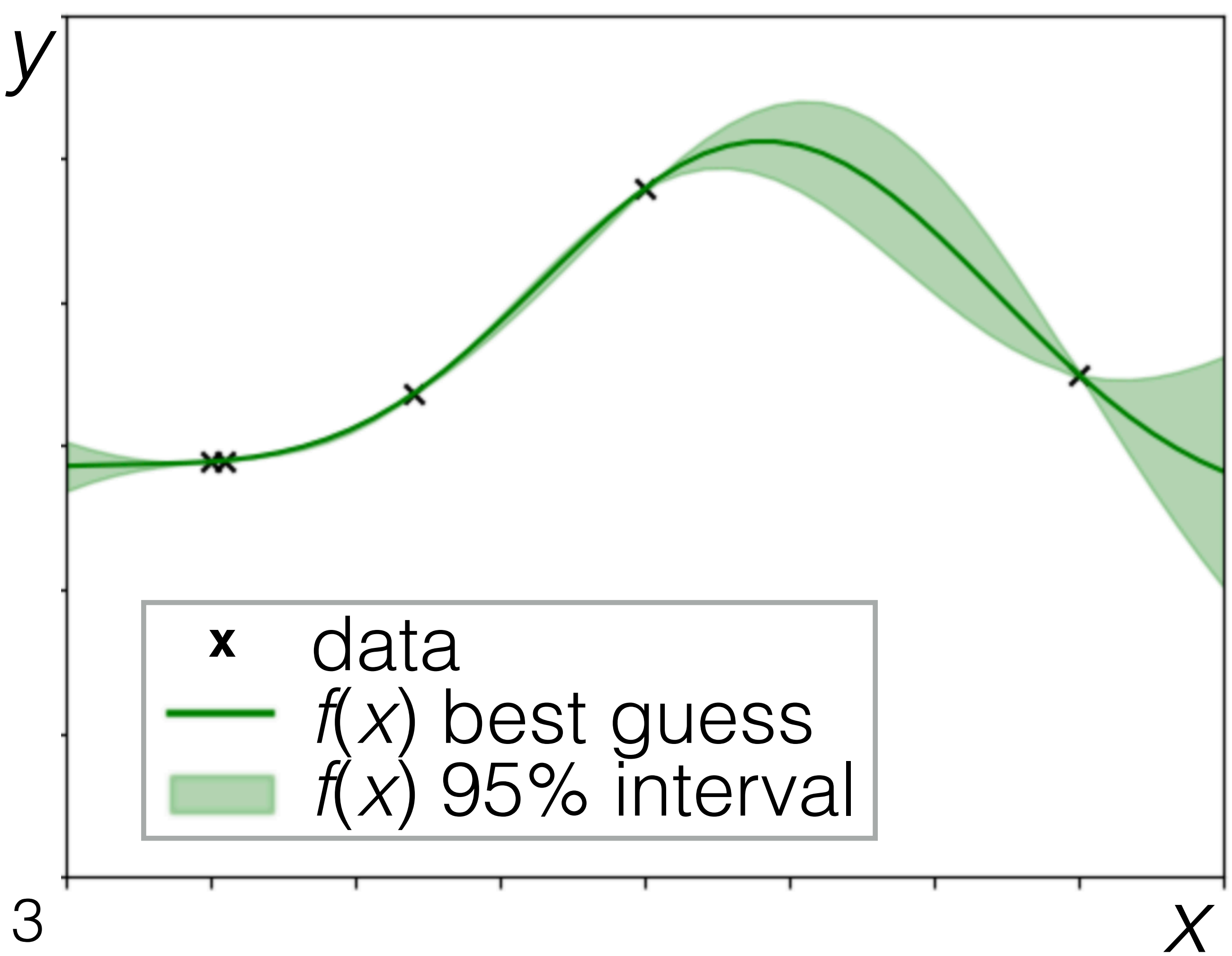
Given the data we've seen, what do we know about the underlying function?



A Bayesian approach

- p(unknowns | data) $\propto$ p(data | unknowns) p(unknowns)

Given the data we've seen, what do we know about the underlying function?

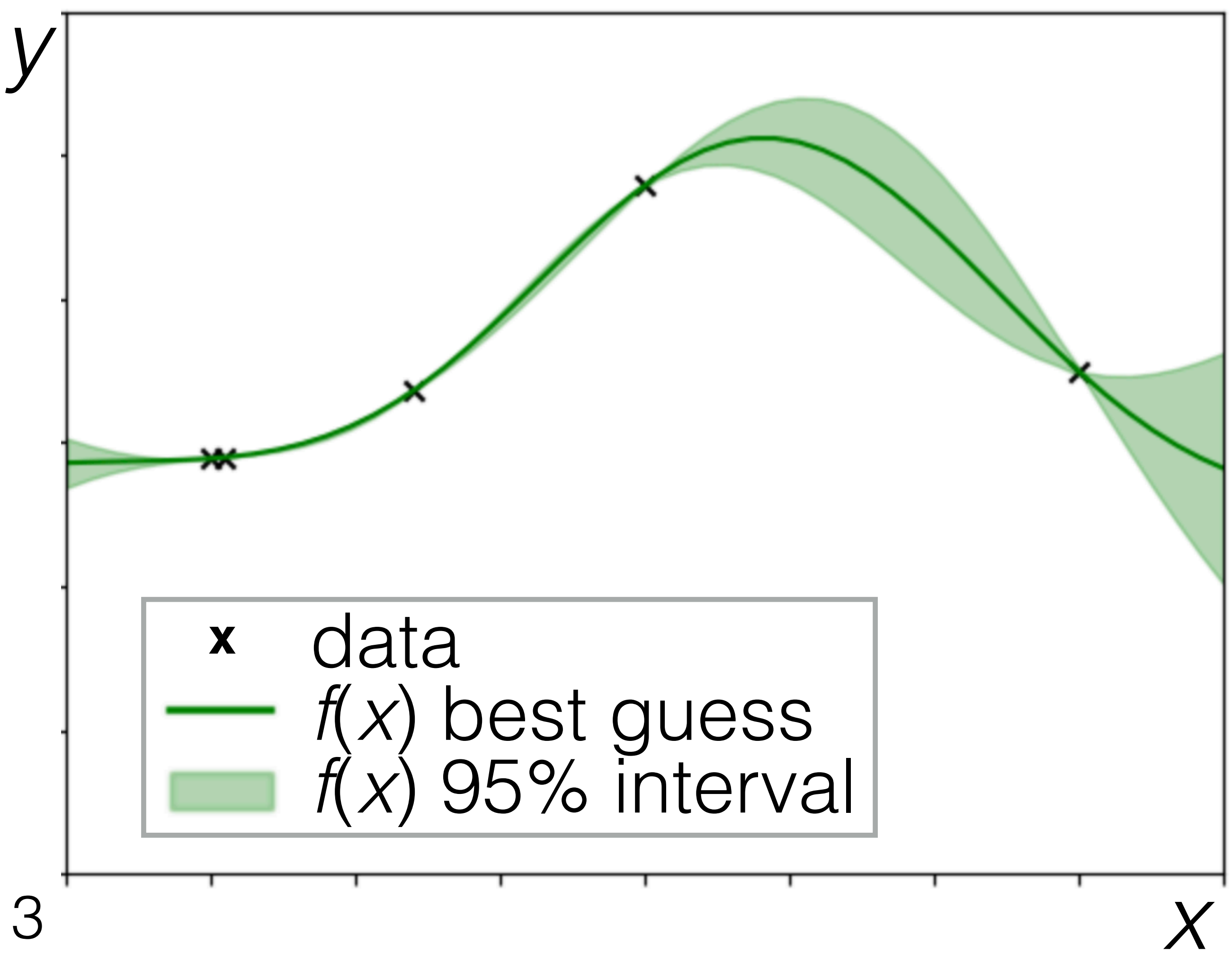


A Bayesian approach

- $\boxed{\text{p(unknowns | data)}} \propto \boxed{\text{p(data | unknowns) p(unknowns)}}$

Given the data we've seen, what do we know about the underlying function?

A model (likelihood & prior) that can generate functions and data of interest

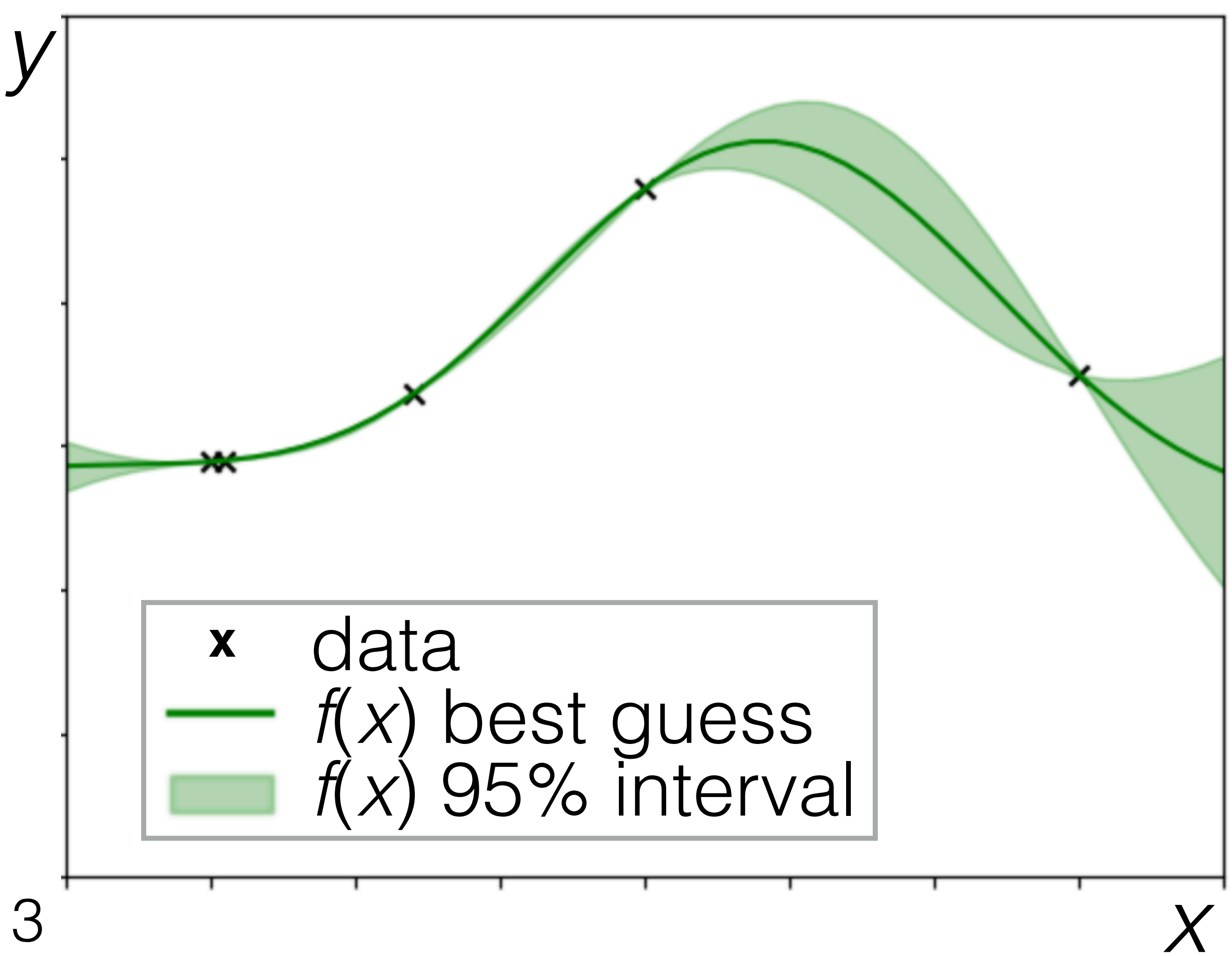


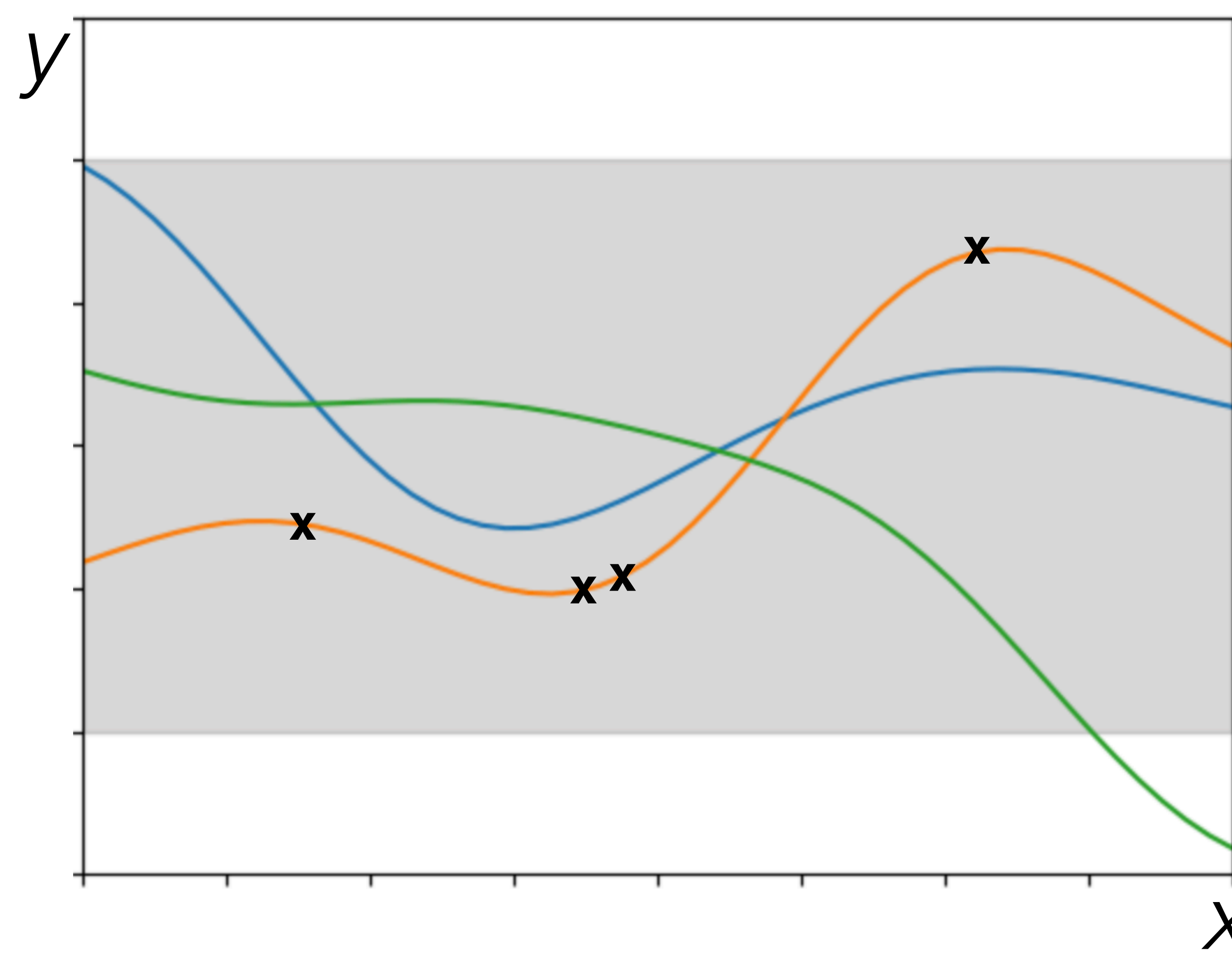
A Bayesian approach

- p(unknowns | data) ∝ p(data | unknowns) p(unknowns)

Given the data we've seen, what do we know about the underlying function?

A model (likelihood & prior) that can generate functions and data of interest





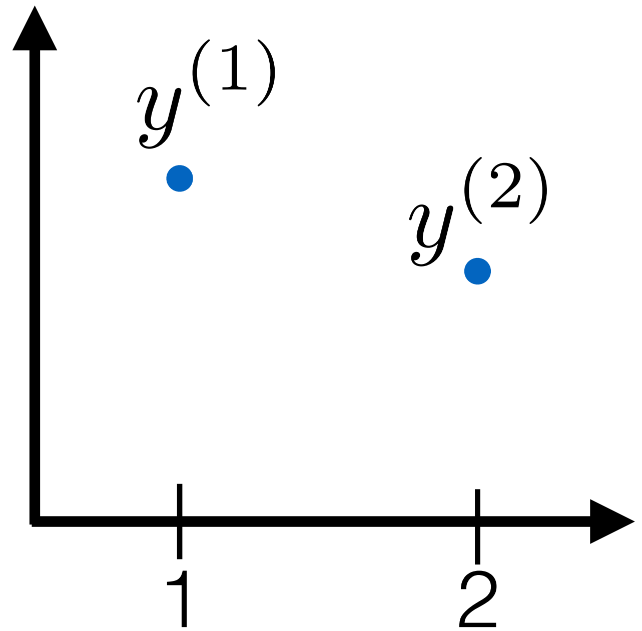
Multivariate Gaussian using locations

Multivariate Gaussian using locations

- *M*=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

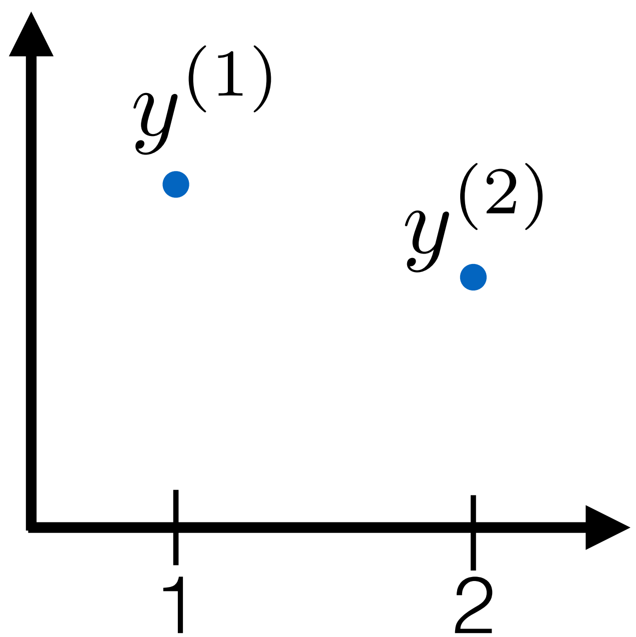
Multivariate Gaussian using locations

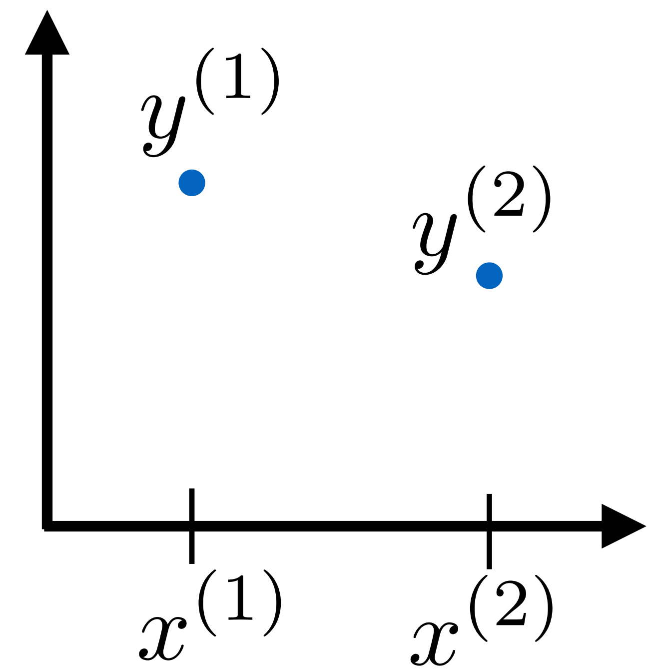
- *M*=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation



Multivariate Gaussian using locations

- M=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

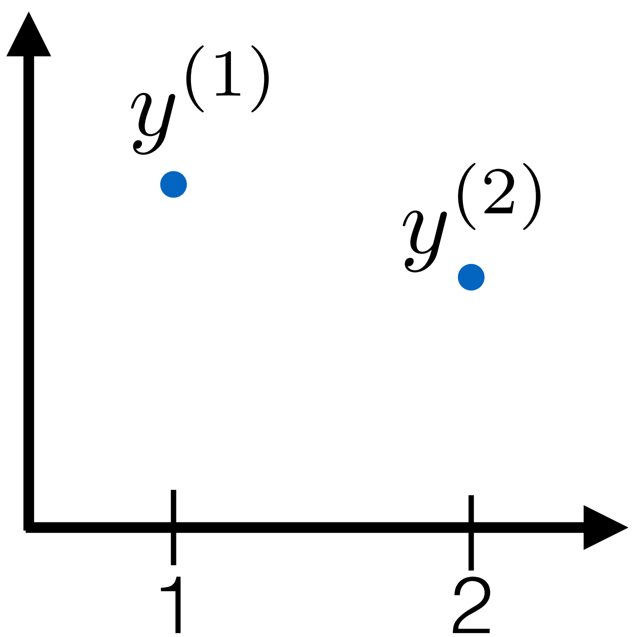


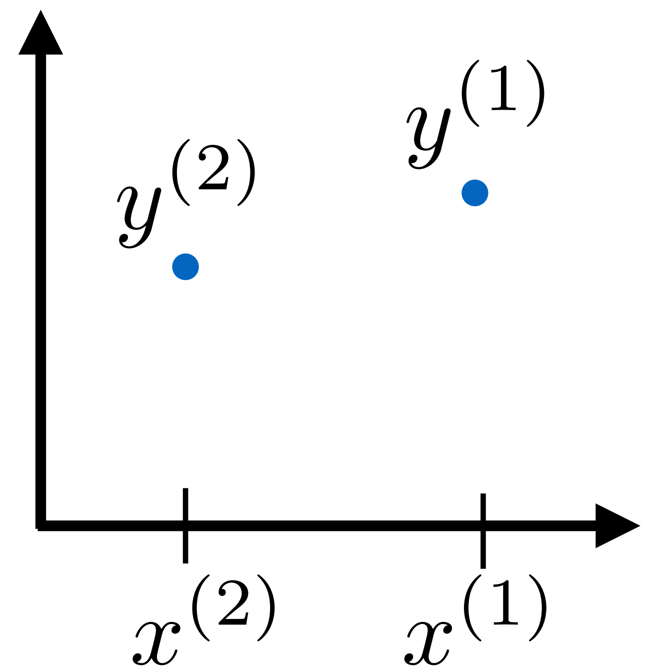


Multivariate Gaussian using locations

- *M*=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$

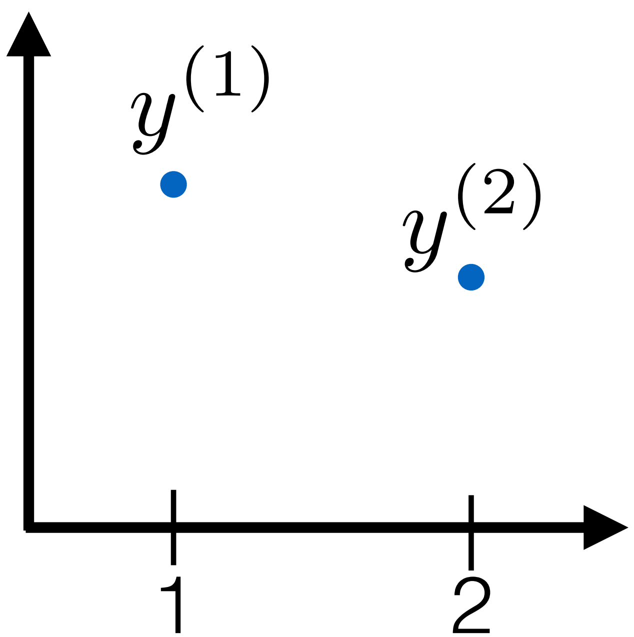
correlation

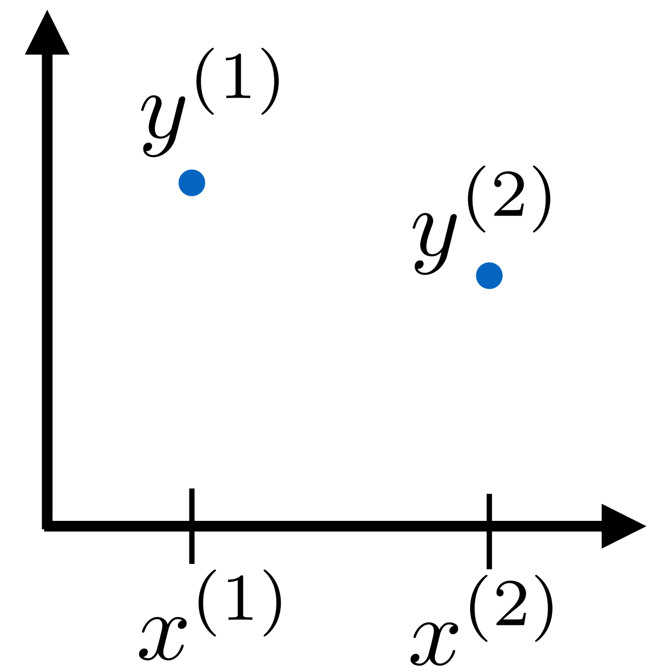




Multivariate Gaussian using locations

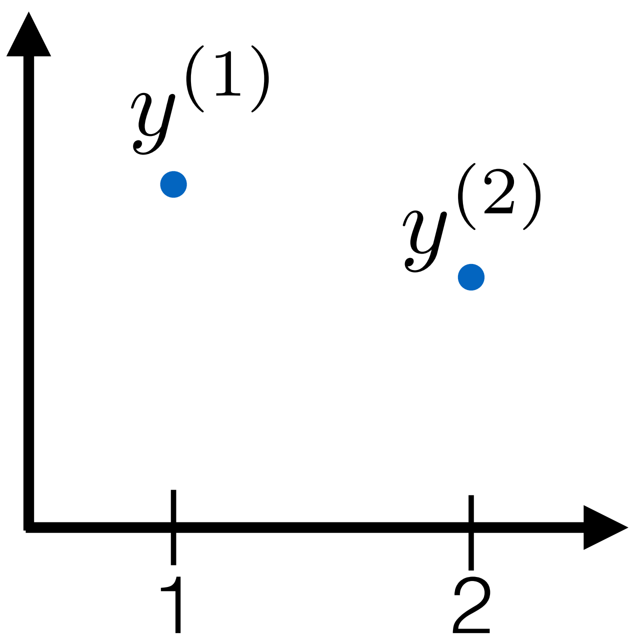
- *M*=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

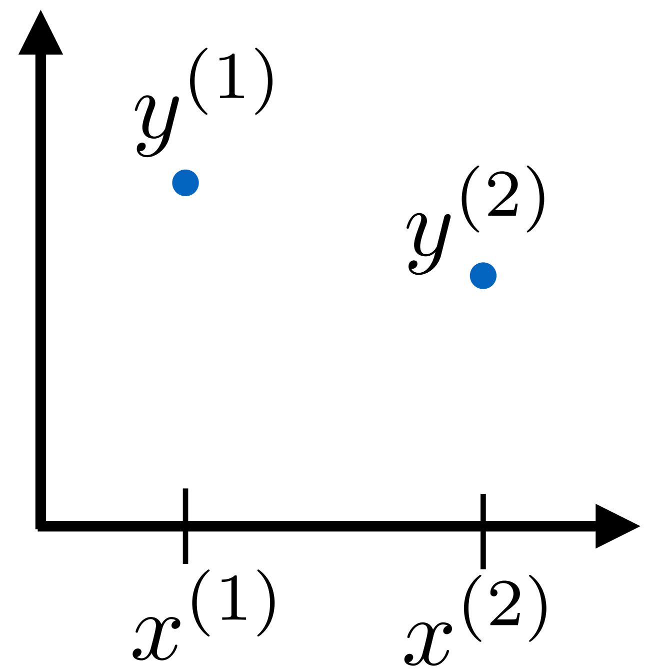




Multivariate Gaussian using locations

- M=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

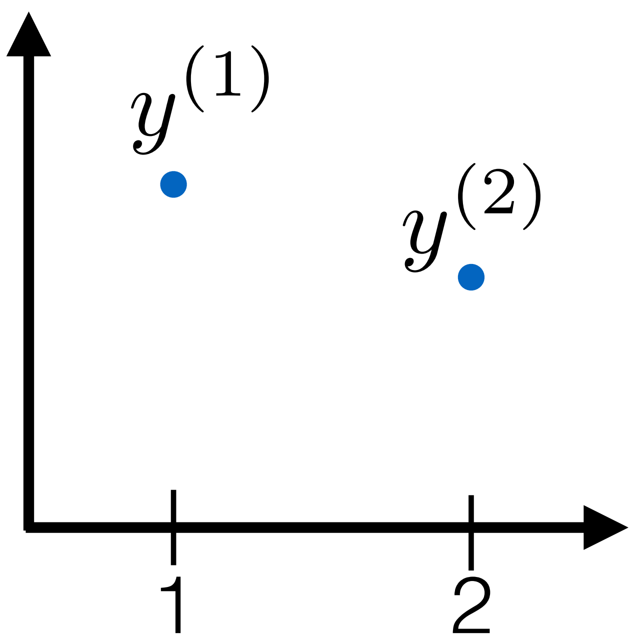


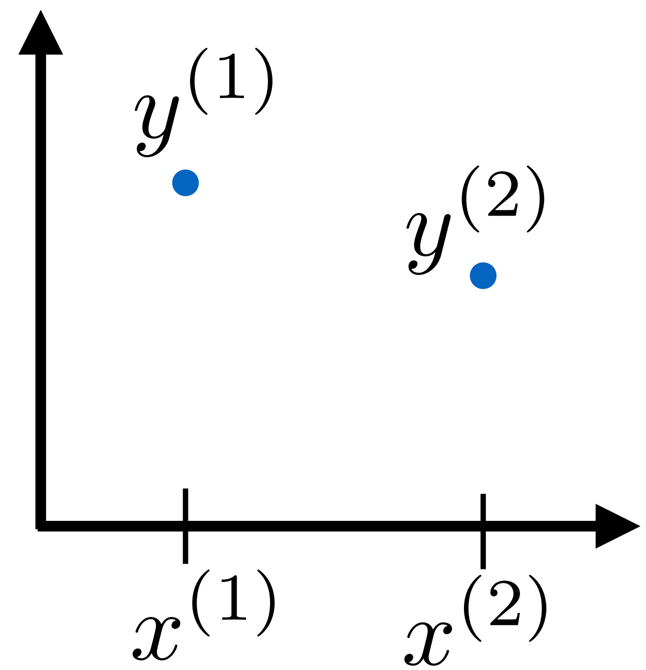


- What if we let the correlation depend on the *x*'s?

Multivariate Gaussian using locations

- *M*=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

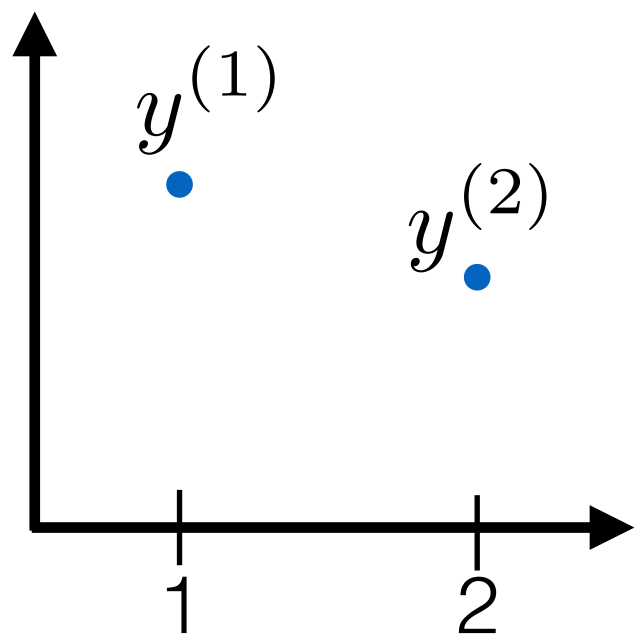


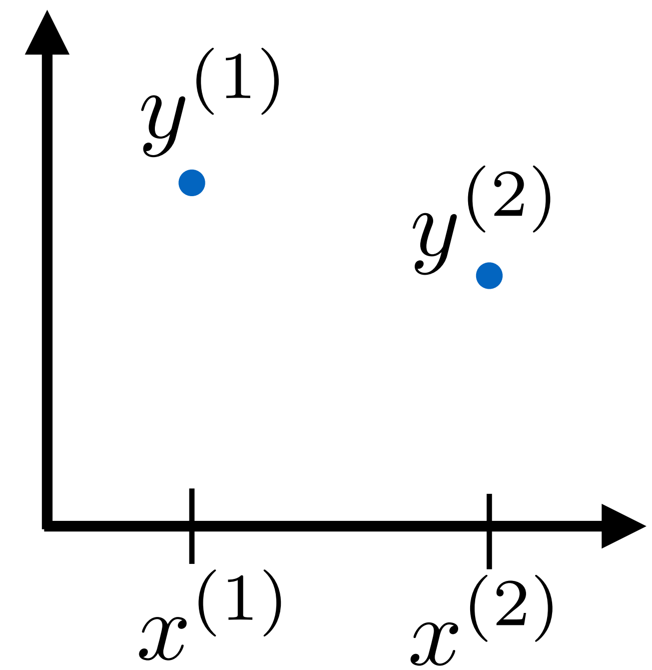


- What if we let the correlation depend on the *x*'s?
  - Let $\rho = \rho(|x^{(1)} - x^{(2)}|)$

Multivariate Gaussian using locations

- M=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

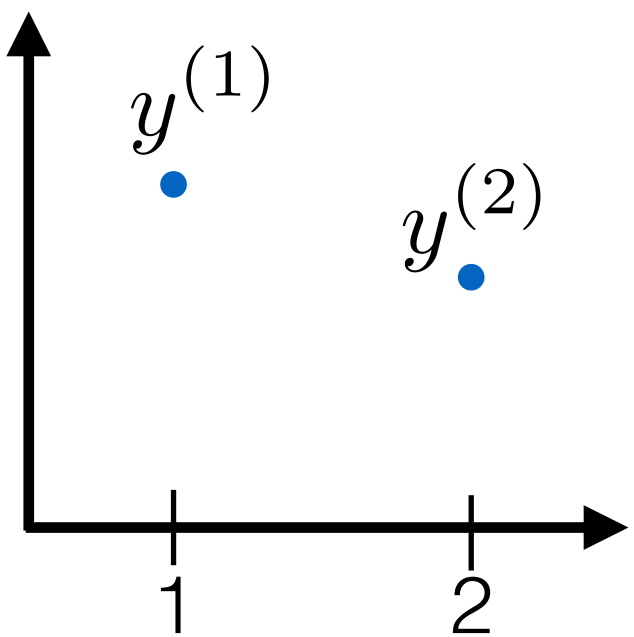


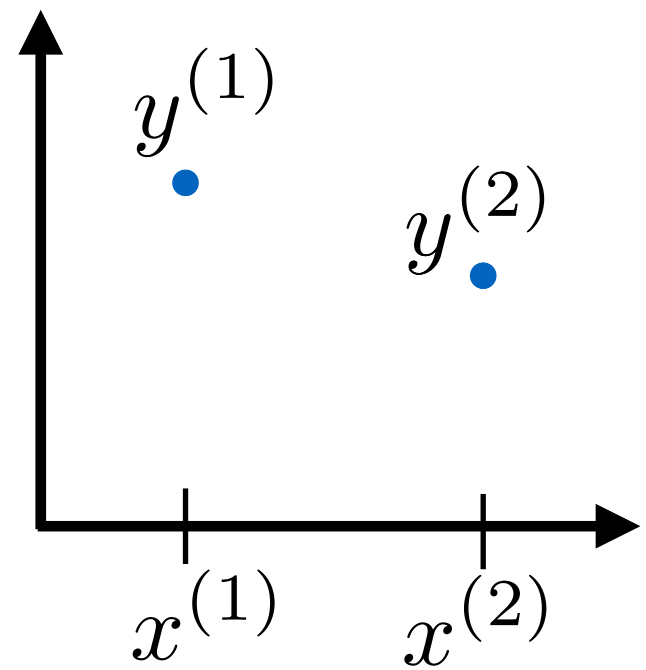


- What if we let the correlation depend on the x's?
  - Let $\rho = \rho(|x^{(1)} - x^{(2)}|)$
    - Where the correlation goes to 1 as the x's get close

Multivariate Gaussian using locations

- *M*=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0, 0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation

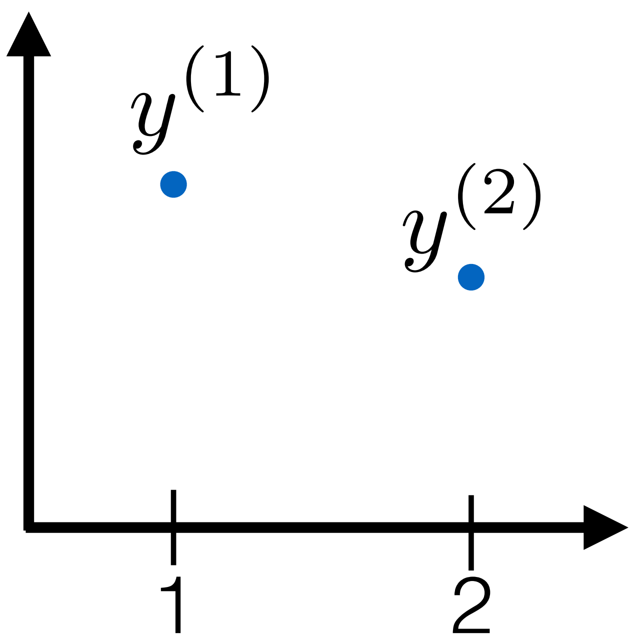


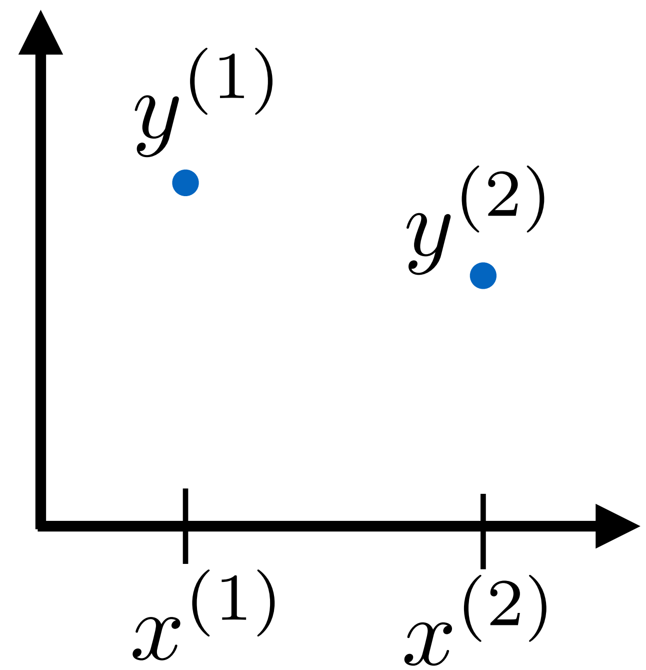


- What if we let the correlation depend on the *x*'s?
  - Let $\rho = \rho(|x^{(1)} - x^{(2)}|)$
    - Where the correlation goes to 1 as the *x*'s get close
    - And goes to 0 as the *x*'s get far

Multivariate Gaussian using locations

- M=2 dimensions: $[y^{(1)}, y^{(2)}]^\top \sim \mathcal{N}(\mu, K)$
  - With $\mu = [0,0]^\top$ and $K = \sigma^2 \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$ correlation





- What if we let the correlation depend on the *x*'s?
  - Let $\rho = \rho(|x^{(1)} - x^{(2)}|)$
    - Where the correlation goes to 1 as the *x*'s get close
    - And goes to 0 as the *x*'s get far

[demo]

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):

$$x^{(m)} \in (-\infty, \infty)$$

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
    $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
  $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
    $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
    $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
      $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
  $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
    $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check:

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
    $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
      $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check: $\rho(0) = 1$

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
  $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
    $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check: $\rho(0) = 1$, $\rho(\Delta)$ is decreasing as Δ increases

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
$$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
$$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check: $\rho(0) = 1$, $\rho(\Delta)$ is decreasing as Δ increases
      - And $\rho(\Delta) \rightarrow 0$ as $\Delta \rightarrow \infty$

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
    $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
      $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check: $\rho(0) = 1$, $\rho(\Delta)$ is decreasing as Δ increases
      - And $\rho(\Delta) \to 0$ as $\Delta \to \infty$

Note: our previous example was the special case where *M*=2

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
    $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
      $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check: $\rho(0) = 1$, $\rho(\Delta)$ is decreasing as Δ increases
      - And $\rho(\Delta) \to 0$ as $\Delta \to \infty$

[demo1, demo2]

Note: our previous example was the special case where *M*=2

Multivariate Gaussian using locations

- Next: Similar setup but an *M*-dimensional Gaussian instead of just 2 dimensions
  - We have *M* locations (need not be in order):
    $$x^{(m)} \in (-\infty, \infty)$$
  - We're going to generate $[y^{(1)}, \ldots, y^{(M)}]^\top \sim \mathcal{N}(\mu, K)$
    - Take $\mu = \mathbf{0}_M$ and *K* such that
      $$K_{m,m'} = \sigma^2 \rho(|x^{(m)} - x^{(m')}|)$$
    - Let's try $\rho(\Delta) = \exp(-\frac{1}{2}\Delta^2)$
      - Check: $\rho(0) = 1$, $\rho(\Delta)$ is decreasing as Δ increases
      - And $\rho(\Delta) \to 0$ as $\Delta \to \infty$

[demo1, demo2]

Note: our previous example was the special case where *M*=2

We just drew random functions from a type of "Gaussian process"!

Gaussian processes

Gaussian processes

GPs for regression are aka
kriging, optimal interpolation, etc.

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: “A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution.” [Rasmussen and Williams 2006; a much much older idea!]

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
    - E.g. the function $f(x)$ is a collection indexed by input x

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: “A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution.” [Rasmussen and Williams 2006; a much much older idea!]
    - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

  - Mean function $m(x) = \mathbb{E}[f(x)]$

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

x could be just about anything, but here we'll assume it's a real vector

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

x could be just about anything, but here we'll assume it's a real vector

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- A common default (e.g. in software) is $m(x) = 0$

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

x could be just about anything, but here we'll assume it's a real vector

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- A common default (e.g. in software) is $m(x) = 0$
- One very commonly used covariance function is the *squared exponential* or *radial basis function (RBF)*

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

x could be just about anything, but here we'll assume it's a real vector

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- A common default (e.g. in software) is $m(x) = 0$
- One very commonly used covariance function is the *squared exponential* or *radial basis function (RBF)*
  - We'll see a more general form later, but for now we're using: $k(x, x') = \sigma^2 \exp(-\frac{1}{2}\|x - x'\|^2)$

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

x could be just about anything, but here we'll assume it's a real vector

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- A common default (e.g. in software) is $m(x) = 0$
- One very commonly used covariance function is the *squared exponential* or *radial basis function (RBF)*
  - We'll see a more general form later, but for now we're using: $k(x, x') = \sigma^2 \exp(-\frac{1}{2}\|x - x'\|^2)$
- For now, assume data is observed without noise

Gaussian processes

GPs for regression are aka kriging, optimal interpolation, etc.

- Definition: "A *Gaussian process* is a collection of random variables, any finite number of which have a joint Gaussian distribution." [Rasmussen and Williams 2006; a much much older idea!]
  - E.g. the function $f(x)$ is a collection indexed by input x
- It is specified by its mean function and covariance function:

$$f \sim \mathcal{GP}(m, k)$$

x could be just about anything, but here we'll assume it's a real vector

  - Mean function $m(x) = \mathbb{E}[f(x)]$
  - Covariance function (a.k.a. *kernel*)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

- A common default (e.g. in software) is $m(x) = 0$
- One very commonly used covariance function is the *squared exponential* or *radial basis function (RBF)*
  - We'll see a more general form later, but for now we're using: $k(x, x') = \sigma^2 \exp(-\frac{1}{2}\|x - x'\|^2)$
- For now, assume data is observed without noise [demo1,2]

Note on “dimension” of the input

Note on “dimension” of the input

- Let’s be careful to separate two types of “dimension”

Note on “dimension” of the input

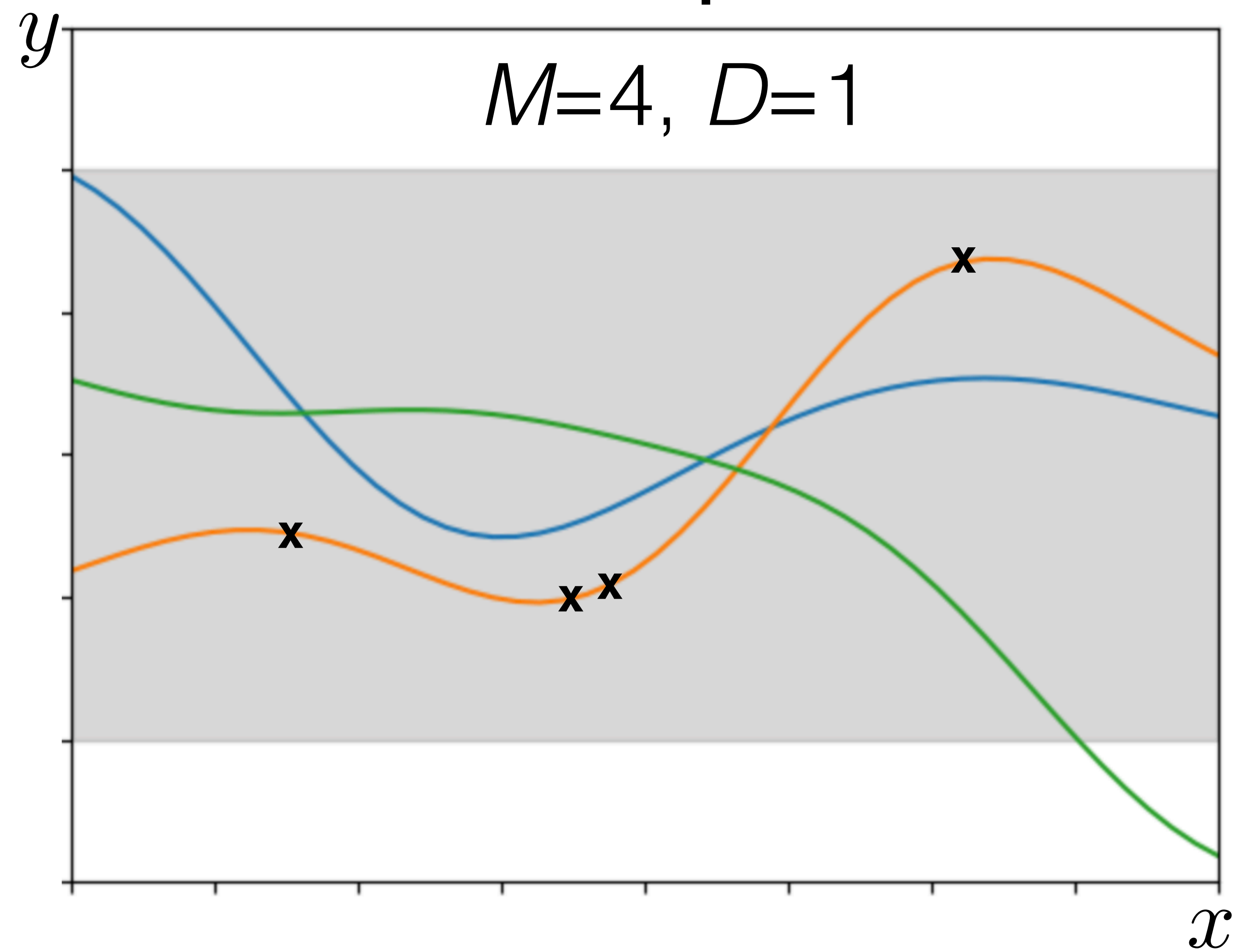
- Let’s be careful to separate two types of “dimension”
  - We’re using a superscript to denote (*M* or *N*) number of points in the space

Note on “dimension” of the input

- Let’s be careful to separate two types of “dimension”
  - We’re using a superscript to denote (M or N) number of points in the space
  - We’ll use a subscript for the (D) different elements of a point’s input vector

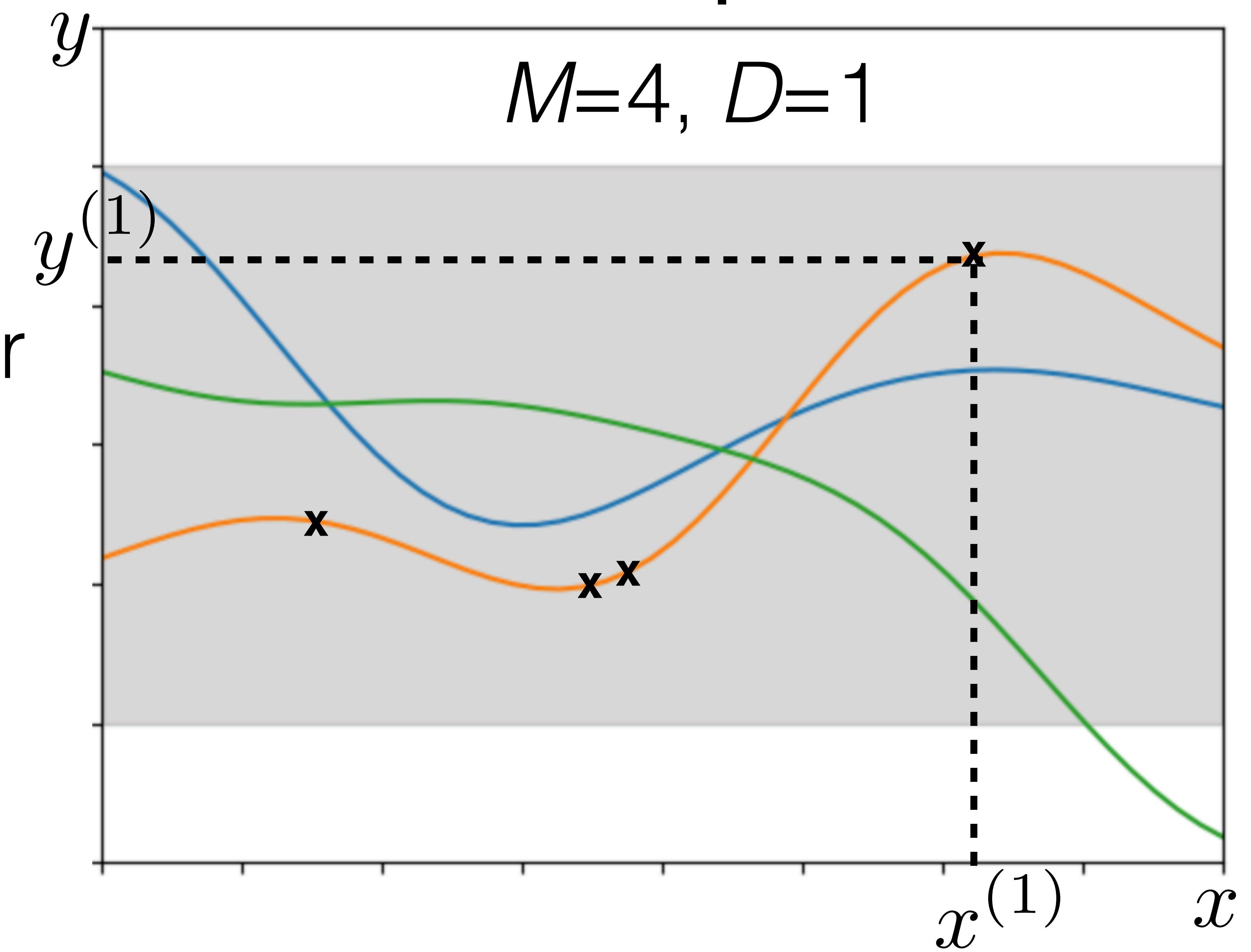
Note on "dimension" of the input

- Let's be careful to separate two types of "dimension"
  - We're using a superscript to denote (M or N) number of points in the space
  - We'll use a subscript for the (D) different elements of a point's input vector



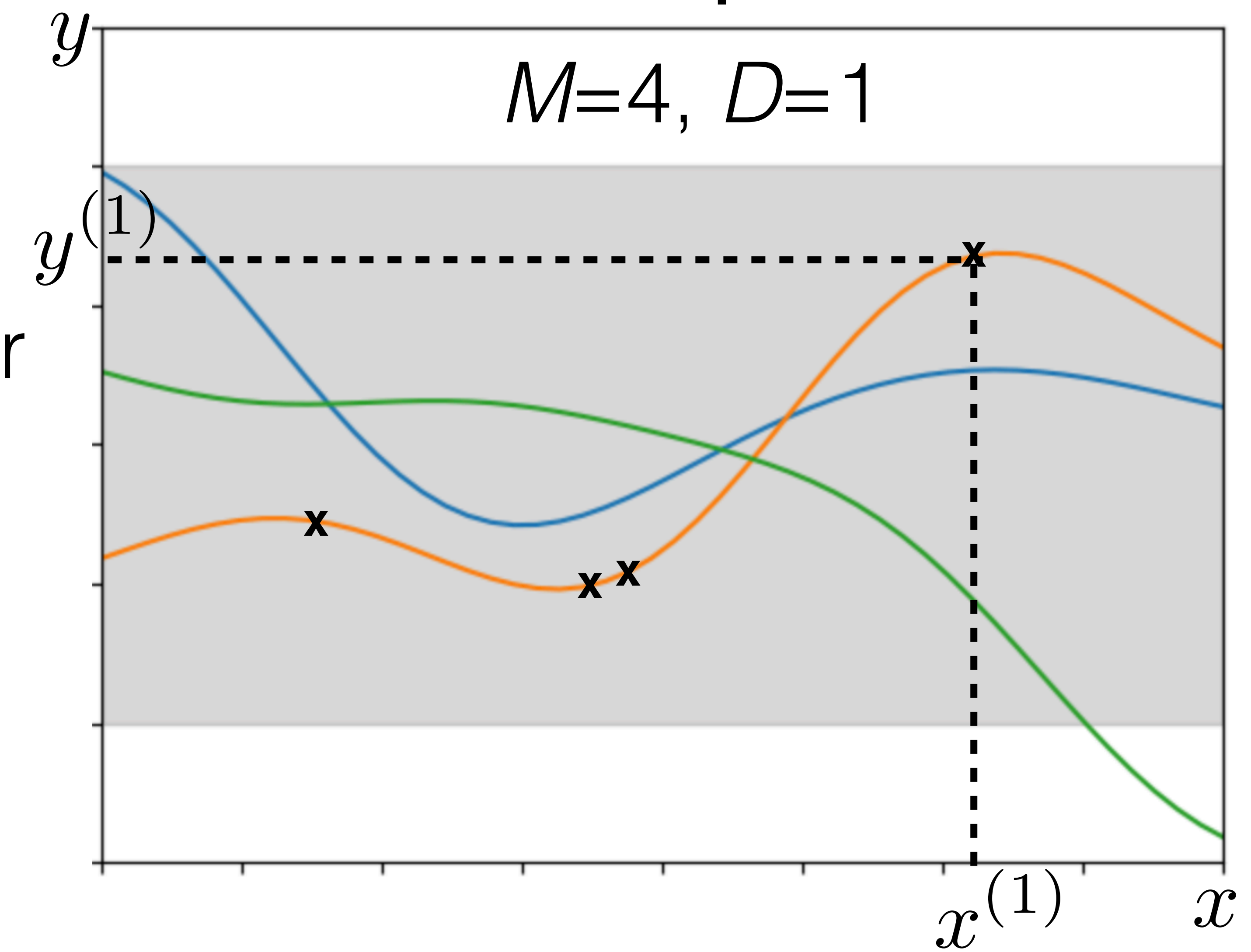
Note on "dimension" of the input

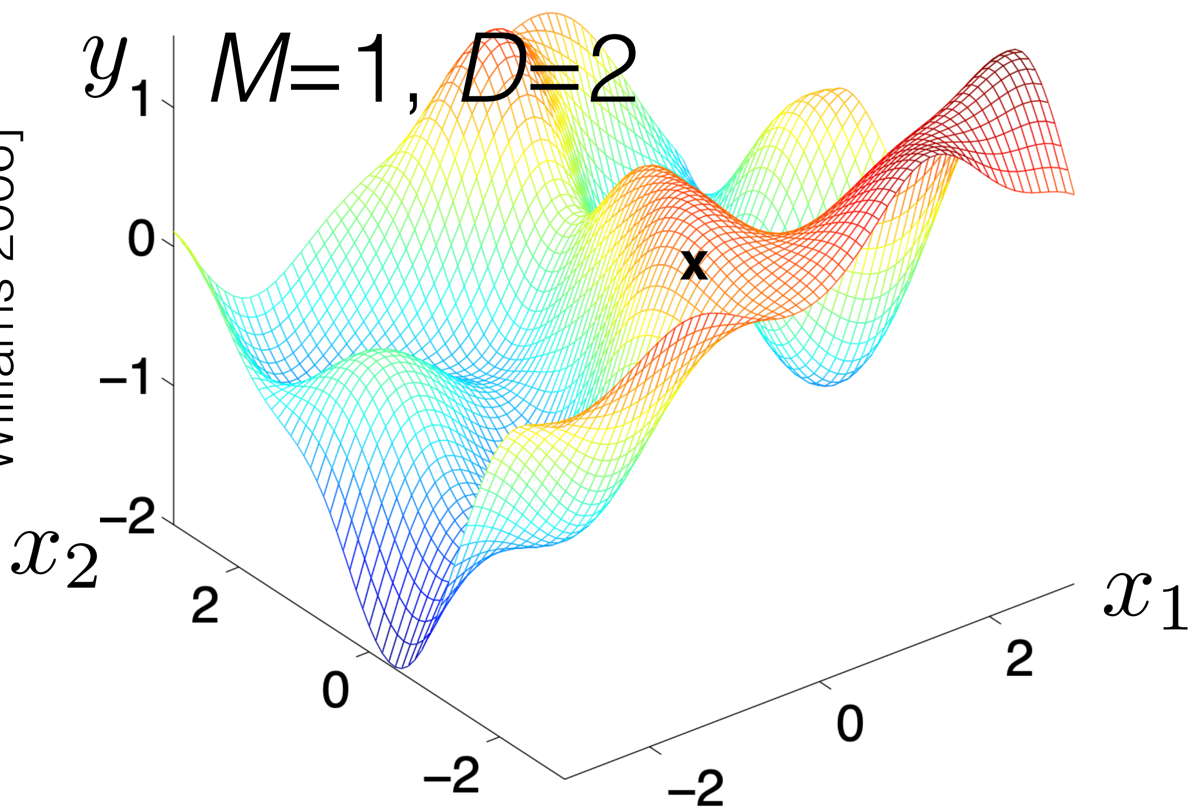
- Let's be careful to separate two types of "dimension"
  - We're using a superscript to denote (*M* or *N*) number of points in the space
  - We'll use a subscript for the (*D*) different elements of a point's input vector



Note on “dimension” of the input

- Let’s be careful to separate two types of “dimension”
  - We’re using a superscript to denote (*M* or *N*) number of points in the space
  - We’ll use a subscript for the (*D*) different elements of a point’s input vector

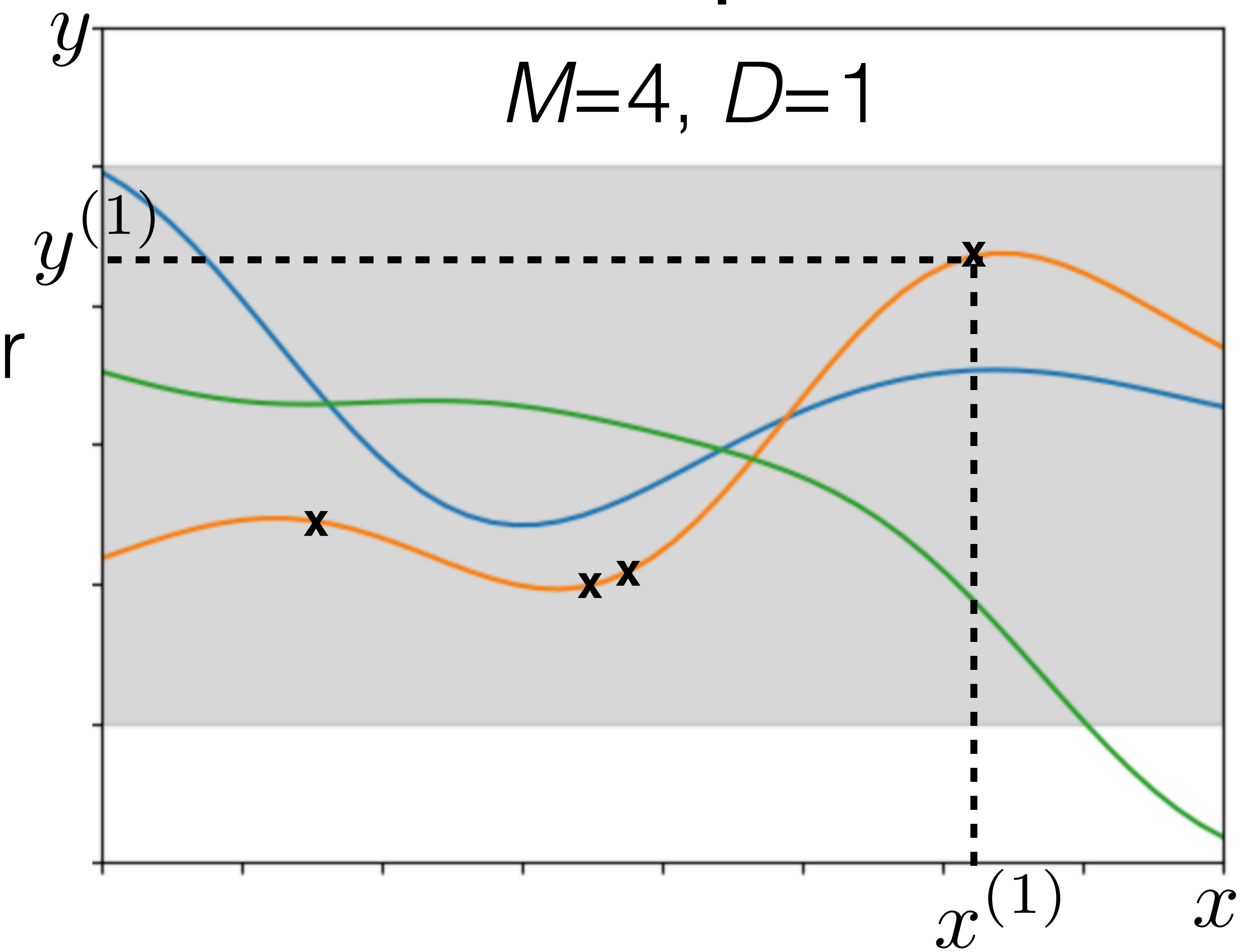


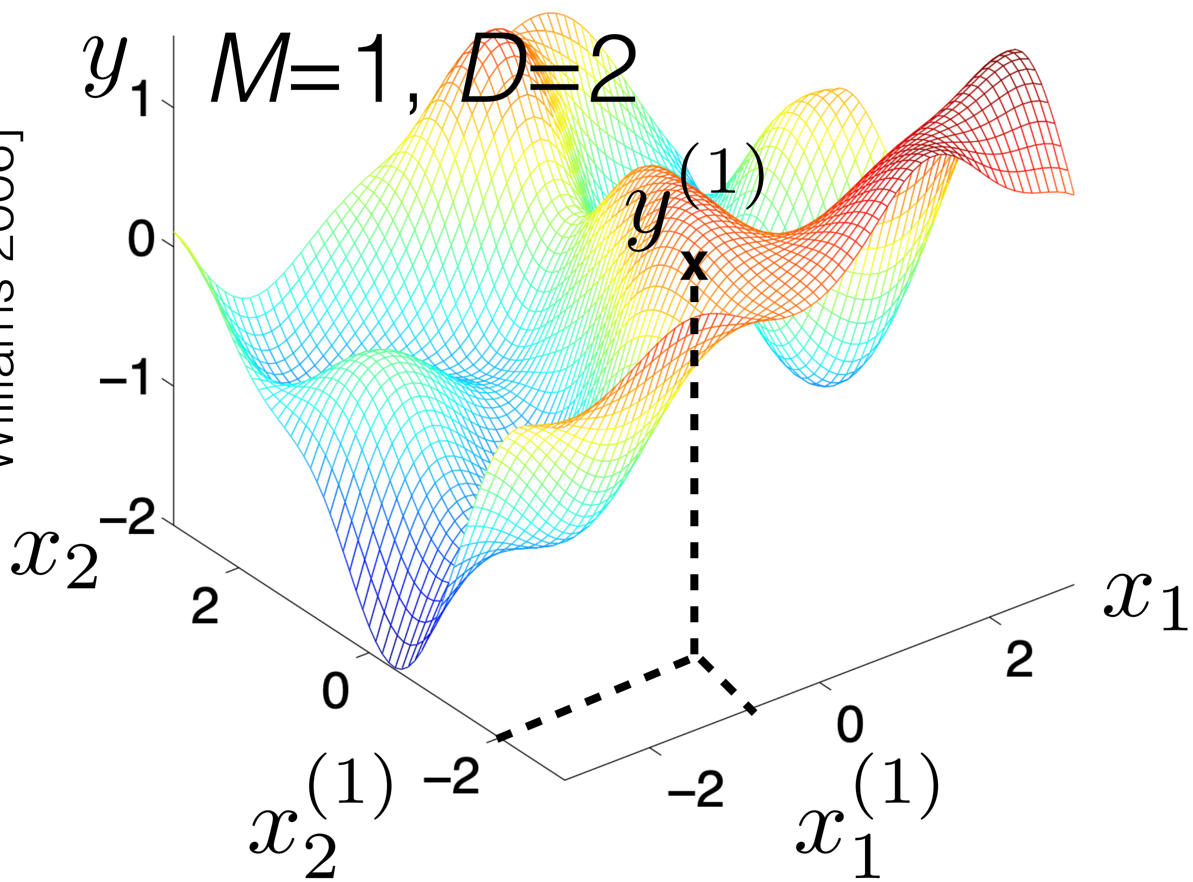


[Rasmussen & Williams 2006]

Note on “dimension” of the input

- Let’s be careful to separate two types of “dimension”
  - We’re using a superscript to denote (*M* or *N*) number of points in the space
  - We’ll use a subscript for the (*D*) different elements of a point’s input vector

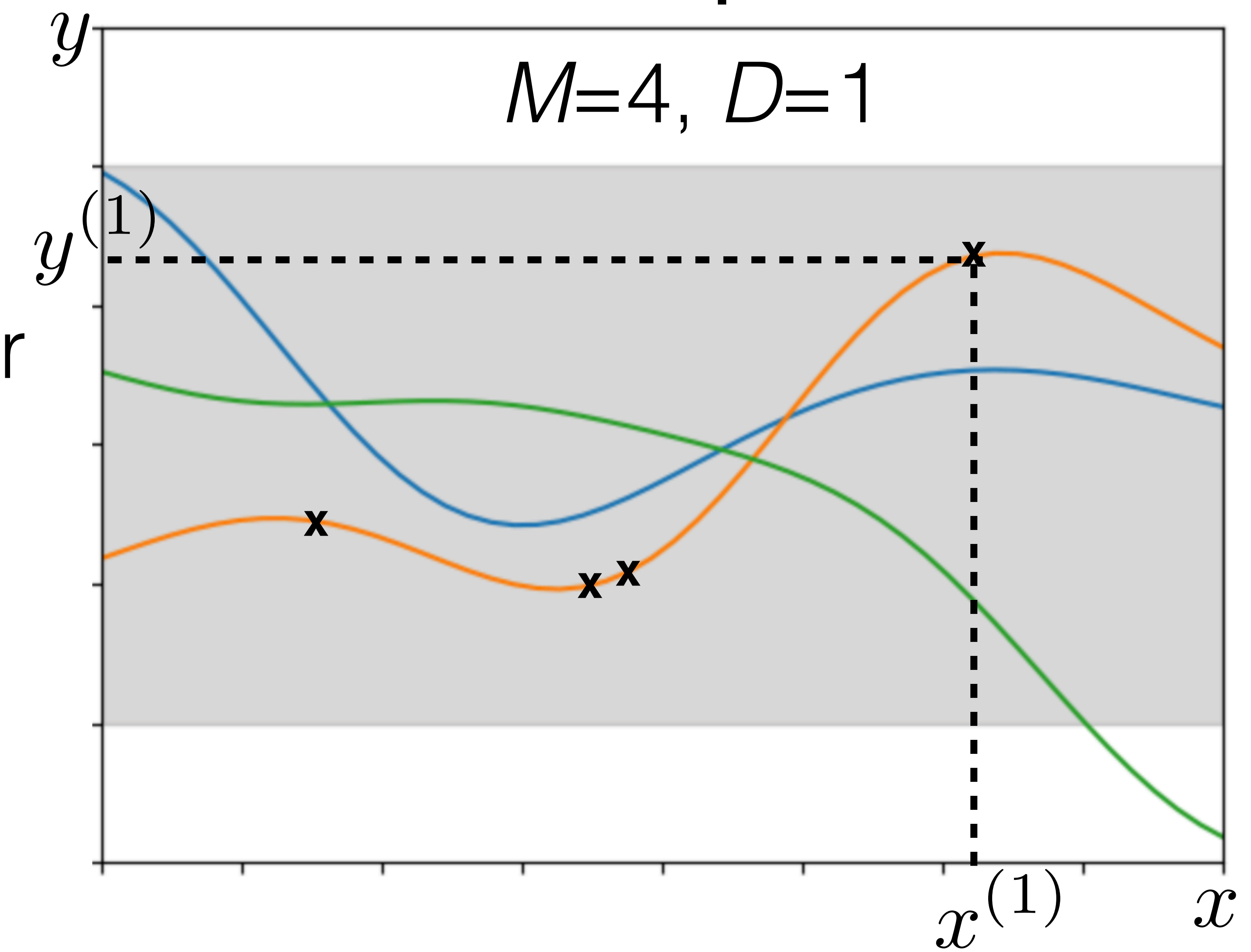


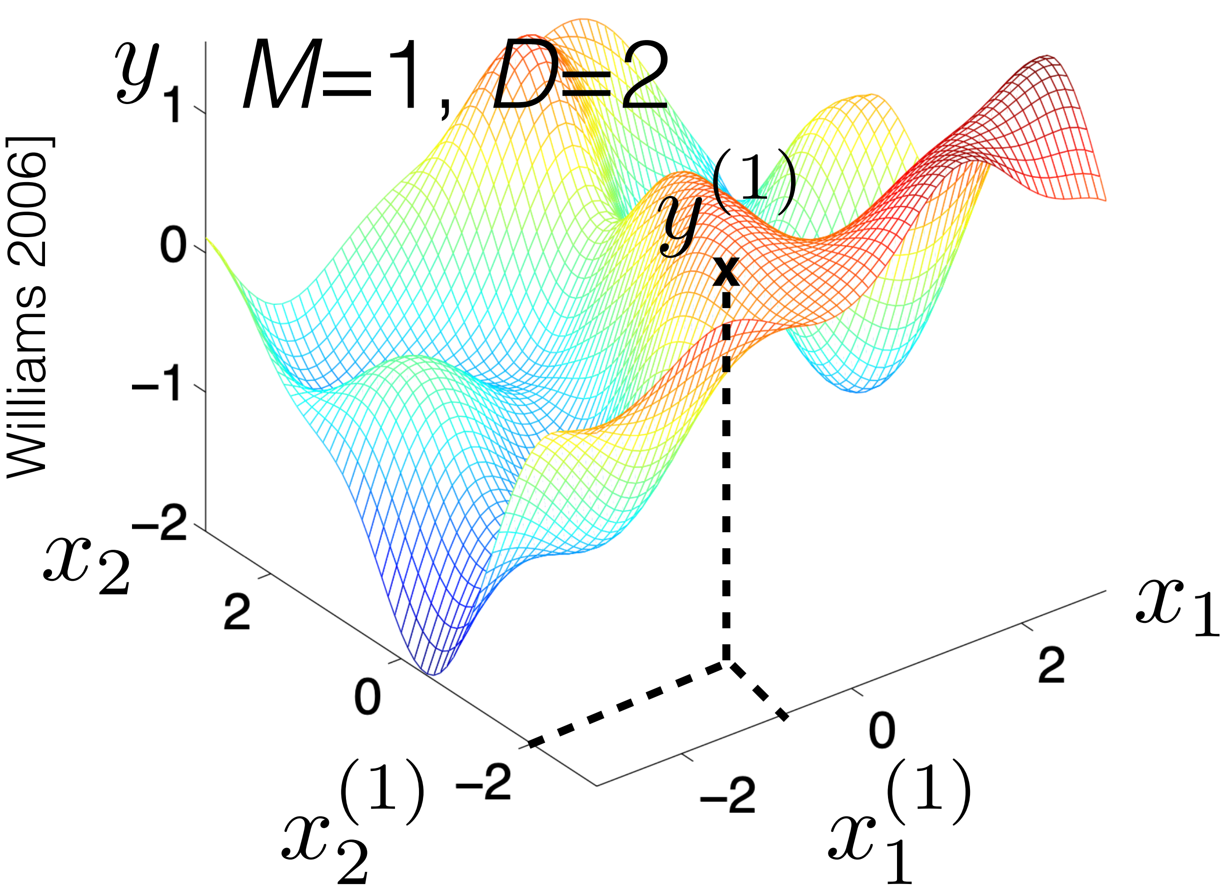


[Rasmussen & Williams 2006]

Note on "dimension" of the input

- Let's be careful to separate two types of "dimension"
  - We're using a superscript to denote (M or N) number of points in the space
  - We'll use a subscript for the (D) different elements of a point's input vector



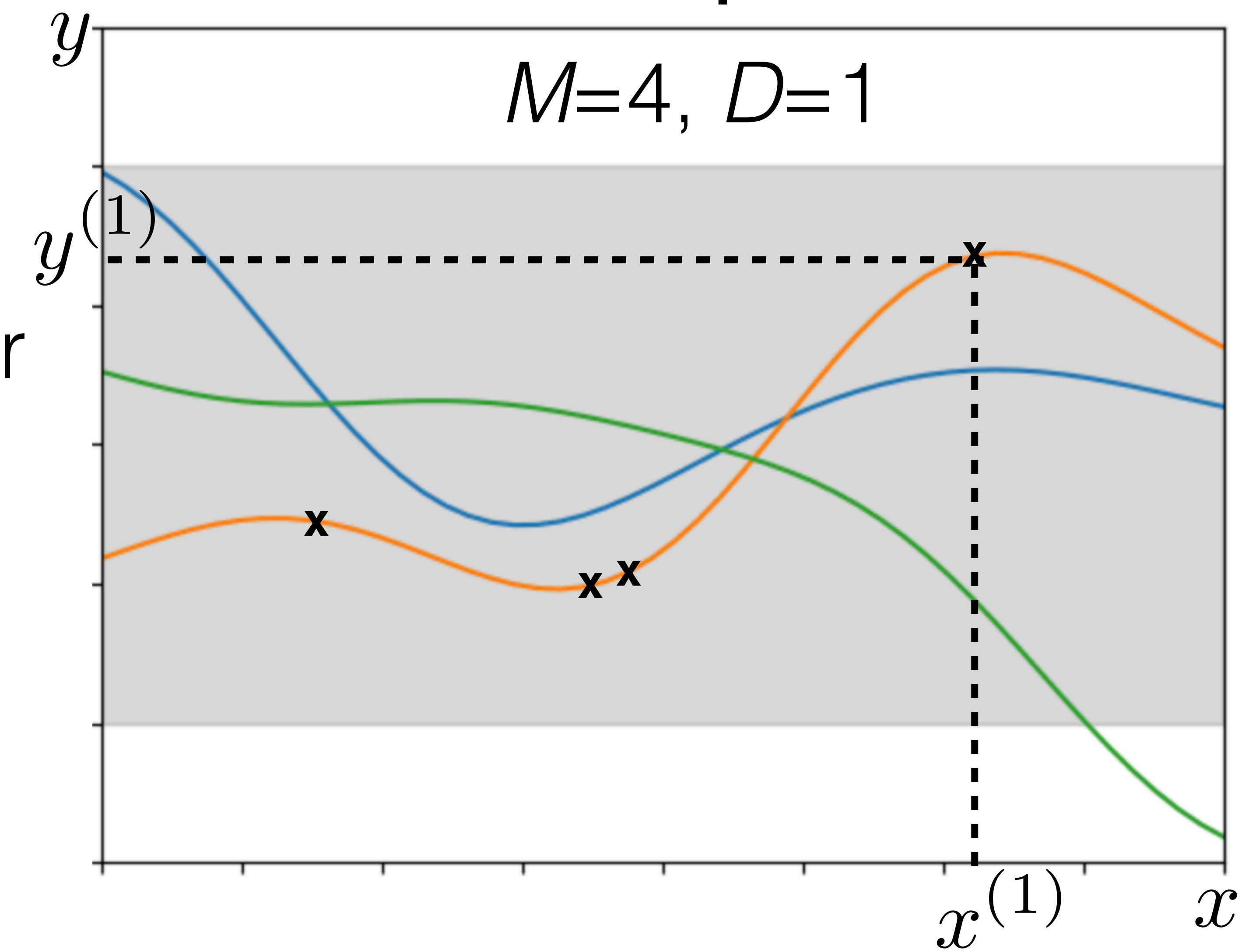


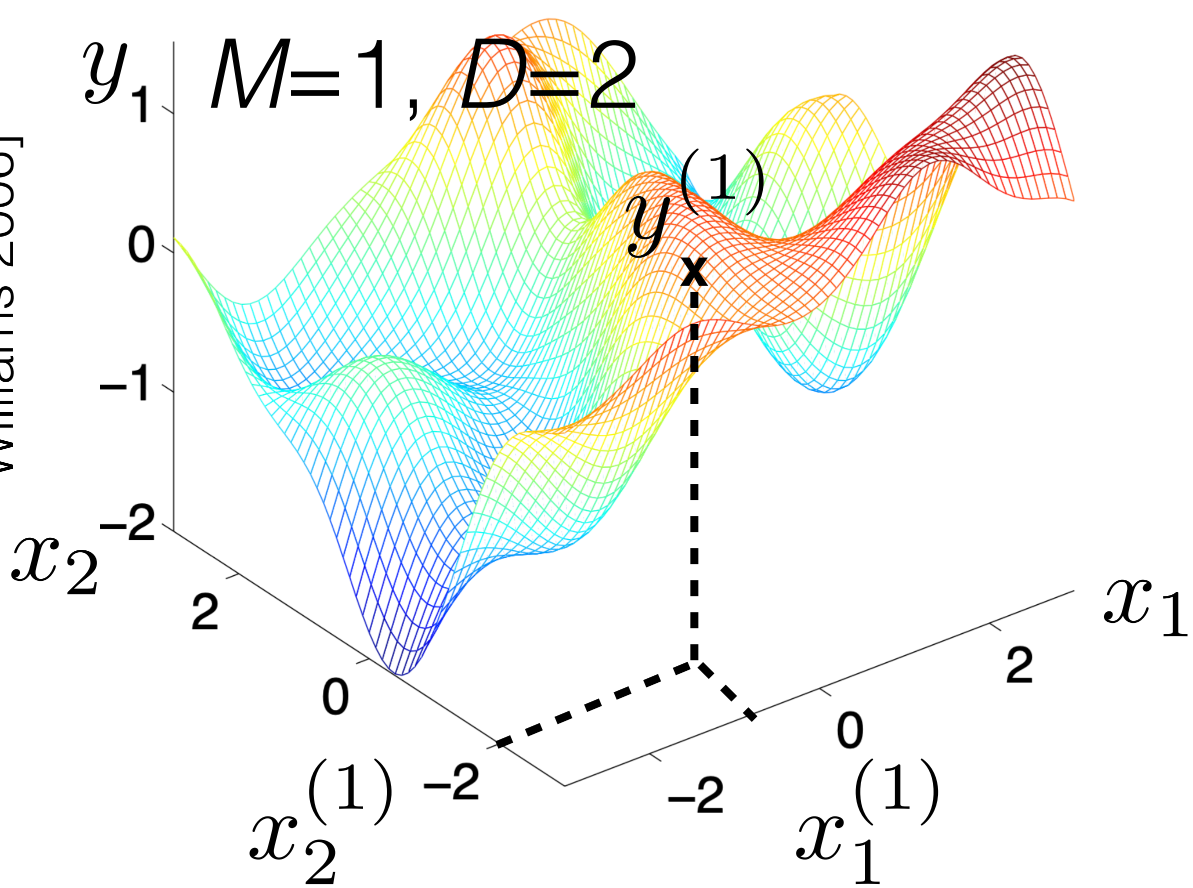
[Rasmussen & Williams 2006]

- Note: all of our real-life examples from the start had number of inputs $D > 1$

Note on “dimension” of the input

- Let’s be careful to separate two types of “dimension”
  - We’re using a superscript to denote (*M* or *N*) number of points in the space
  - We’ll use a subscript for the (*D*) different elements of a point’s input vector





[Rasmussen & Williams 2006]

- Note: all of our real-life examples from the start had number of inputs $D > 1$
- $D = 1$ is much easier to visualize, but might not be representative

Inference about unknowns given data

Inference about unknowns given data

- Let X collect the N “training” data points (indexed 1 to N)

X: NxD

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

X: NxD

Inference about unknowns given data

- Let X collect the N “training” data points (indexed 1 to N)
- Let X' collect the M “test” data points
  - Where we want to evaluate the function

X: NxD

Inference about unknowns given data

- Let X collect the N “training” data points (indexed 1 to N)
- Let X' collect the M “test” data points
  - Where we want to evaluate the function
  - Indexed N+1 to N+M

X: NxD

Inference about unknowns given data

- Let *X* collect the *N* “training” data points (indexed 1 to *N*)
- Let *X’* collect the *M* “test” data points
  - Where we want to evaluate the function
  - Indexed *N*+1 to *N*+*M*

X: *N*x*D*

$$\begin{array}{r} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix}$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points
  - Where we want to evaluate the function
  - Indexed N+1 to N+M

X: NxD

- Then by our model

$$\begin{array}{l} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \boxed{?}$$

Inference about unknowns given data

- Let *X* collect the *N* "training" data points (indexed 1 to *N*)
- Let *X'* collect the *M* "test" data points

  X: *N*x*D*
  - Where we want to evaluate the function
  - Indexed *N*+1 to *N*+*M*

- Then by our model

$$\begin{array}{c} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

X: NxD

  - Where we want to evaluate the function
  - Indexed N+1 to N+M

- Then by our model

$$\begin{array}{l} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\quad ? \quad \right.$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

X: NxD

  - Where we want to evaluate the function
  - Indexed N+1 to N+M

- Then by our model

$$\begin{array}{r} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1},\right.$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

X: NxD

  - Where we want to evaluate the function
  - Indexed N+1 to N+M
- $K(X,X')$ is the NxM matrix with (n,m) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{c} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\Big(\underbrace{\mathbf{0}}_{(N+M)\text{x}1},$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

X: NxD

  - Where we want to evaluate the function
  - Indexed N+1 to N+M
- $K(X,X')$ is the NxM matrix with (n,m) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{c} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix} \right)$$

Inference about unknowns given data

- Let *X* collect the *N* “training” data points (indexed 1 to *N*)
- Let *X’* collect the *M* “test” data points
  - Where we want to evaluate the function
  - Indexed *N*+1 to *N*+*M*
- *K*(*X*,*X’*) is the *N*x*M* matrix with (*n*,*m*) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

X: NxD

$$\begin{array}{c} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \underbrace{\begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}}_{(N+M)\text{x}(N+M)}\right)$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

X: NxD

  - Where we want to evaluate the function
  - Indexed N+1 to N+M
- $K(X,X')$ is the NxM matrix with (n,m) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{c} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \underbrace{\begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}}_{(N+M)\text{x}(N+M)}\right)$$

- The conditional satisfies $f(X') | f(X), X, X' \sim$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)

X: NxD

- Let X' collect the M "test" data points
  - Where we want to evaluate the function
  - Indexed N+1 to N+M
- $K(X,X')$ is the NxM matrix with (n,m) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{r} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \underbrace{\begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}}_{(N+M)\text{x}(N+M)}\right)$$

- The conditional satisfies $f(X') | f(X), X, X' \sim \mathcal{N}$

Inference about unknowns given data

- Let *X* collect the *N* “training” data points (indexed 1 to *N*)
- Let *X’* collect the *M* “test” data points

  X: *N*x*D*
  - Where we want to evaluate the function
  - Indexed *N*+1 to *N*+*M*
- *K*(*X*,*X’*) is the *N*x*M* matrix with (*n*,*m*) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{c} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \underbrace{\begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}}_{(N+M)\text{x}(N+M)}\right)$$

- The conditional satisfies $f(X') | f(X), X, X' \sim \mathcal{N}$
  - Can compute mean & covariance in closed form with Gaussian facts *M*x1 *M*x*M*

Inference about unknowns given data

- Let X collect the N “training” data points (indexed 1 to N)
- Let X' collect the M “test” data points

X: NxD

  - Where we want to evaluate the function
  - Indexed N+1 to N+M
- $K(X,X')$ is the NxM matrix with (n,m) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{l} N\text{x}1 \\ M\text{x}1 \end{array} \begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \underbrace{\begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}}_{(N+M)\text{x}(N+M)}\right)$$

- The conditional satisfies $f(X') | f(X), X, X' \sim \mathcal{N}$
  - Can compute mean & covariance in closed form with Gaussian facts

Mx1 $\qquad$ MxM

- We’ll infer $f(X')$ given our simulated data; recall we’re using
$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

Inference about unknowns given data

- Let X collect the N "training" data points (indexed 1 to N)
- Let X' collect the M "test" data points

  X: NxD
  - Where we want to evaluate the function
  - Indexed N+1 to N+M
- $K(X,X')$ is the NxM matrix with (n,m) entry $k(x^{(n)}, x^{(N+m)})$
- Then by our model

$$\begin{array}{r} N\text{x}1 \\ M\text{x}1 \end{array}\begin{bmatrix} f(X) \\ f(X') \end{bmatrix} \sim \mathcal{N}\left(\underbrace{\mathbf{0}}_{(N+M)\text{x}1}, \underbrace{\begin{bmatrix} K(X,X) & K(X,X') \\ K(X',X) & K(X',X') \end{bmatrix}}_{(N+M)\text{x}(N+M)}\right)$$

- The conditional satisfies $f(X') | f(X), X, X' \sim \mathcal{N}$
  - Can compute mean & covariance in closed form with Gaussian facts

    Mx1 MxM

- We'll infer $f(X')$ given our simulated data; recall we're using

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

[demo1,2]

Squared exponential kernel revisited

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale?

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale?
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of *f*(*x*) a priori?

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo]
- We've been using this particular kernel:
$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$
  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in$?

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of $f(x)$ a priori?
    - At one x, with ~95% probability a priori, $f(x) \in (-2, 2)$

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of $f(x)$ a priori?
    - At one x, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:
$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$
  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data
  - What counts as "close" in *x*?

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data
  - What counts as "close" in *x*?

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data
  - What counts as "close" in *x*?

$$\exp(-\tfrac{1}{2}2^2) \approx 0.14 \quad \exp(-\tfrac{1}{2}3^2) \approx 0.011 \quad \exp(-\tfrac{1}{2}4^2) \approx 0.00034$$

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:

$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data
  - What counts as "close" in *x*?

$$\exp(-\tfrac{1}{2}2^2) \approx 0.14 \quad \exp(-\tfrac{1}{2}3^2) \approx 0.011 \quad \exp(-\tfrac{1}{2}4^2) \approx 0.00034$$

- What can we do to handle different *x* and *f*(*x*) scales?

Squared exponential kernel revisited

- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:
  $$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$
  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data
  - What counts as "close" in *x*?

$$\exp(-\tfrac{1}{2}2^2) \approx 0.14 \quad \exp(-\tfrac{1}{2}3^2) \approx 0.011 \quad \exp(-\tfrac{1}{2}4^2) \approx 0.00034$$

- What can we do to handle different *x* and *f*(*x*) scales?
  - Standardization in *y* can help; in *x*, can still be hiccups

Squared exponential kernel revisited

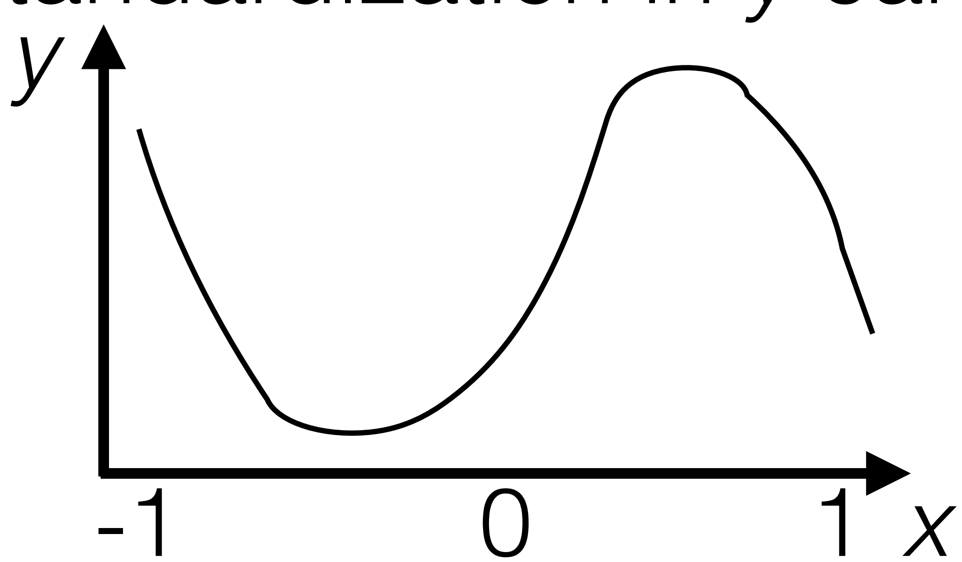
- What if we happened to measure our data on a different scale? [demo1, demo2]
- We've been using this particular kernel:

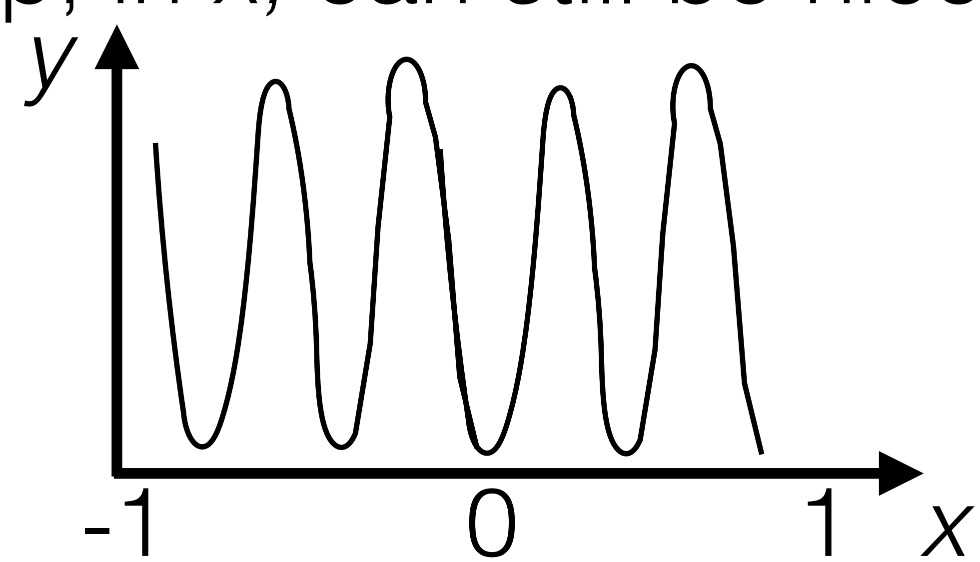
$$k(x, x') = \sigma^2 \exp(-\tfrac{1}{2}(x - x')^2), \sigma = 1$$

  - What do we expect from the scale of *f*(*x*) a priori?
    - At one *x*, with ~95% probability a priori, $f(x) \in (-2, 2)$
    - Marginal variance cannot increase with data
  - What counts as "close" in *x*?

$$\exp(-\tfrac{1}{2}2^2) \approx 0.14 \quad \exp(-\tfrac{1}{2}3^2) \approx 0.011 \quad \exp(-\tfrac{1}{2}4^2) \approx 0.00034$$

- What can we do to handle different *x* and *f*(*x*) scales?
  - Standardization in *y* can help; in *x*, can still be hiccups





Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2})$$

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \underbrace{\sigma^2}_{\text{signal variance}} \exp\Big(-\frac{1}{2}\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2}\Big)$$

signal variance lengthscales

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \underbrace{\sigma^2}_{\text{signal variance}} \exp(-\frac{1}{2}\sum_{d=1}^{D}\frac{(x_d - x'_d)^2}{\underbrace{\ell_d^2}_{\text{lengthscales}}})$$

  - *Parameters* (here, *f*) parametrize the distribution of the data. If we knew them, we could generate the data.

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \underbrace{\sigma^2}_{\text{signal variance}} \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\underbrace{\ell_d^2}_{\text{lengthscales}}})$$

  - *Parameters* (here, *f*) parametrize the distribution of the data. If we knew them, we could generate the data.
    - GPs: *nonparametric* model: infinite # of latent params

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \underbrace{\sigma^2}_{\text{signal variance}} \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\underbrace{\ell_d^2}_{\text{lengthscales}}})$$

  - *Parameters* (here, *f*) parametrize the distribution of the data. If we knew them, we could generate the data.
    - GPs: *nonparametric* model: infinite # of latent params
  - *Hyperparameters* parametrize the distribution of the parameters. If known, we could generate the parameters.

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2})$$

signal variance lengthscales

  - *Parameters* (here, *f*) parametrize the distribution of the data. If we knew them, we could generate the data.
    - GPs: *nonparametric* model: infinite # of latent params
  - *Hyperparameters* parametrize the distribution of the parameters. If known, we could generate the parameters.
- Algorithm:

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{1}{2}\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2})$$

signal variance　　　lengthscales

  - *Parameters* (here, *f*) parametrize the distribution of the data. If we knew them, we could generate the data.
    - GPs: *nonparametric* model: infinite # of latent params
  - *Hyperparameters* parametrize the distribution of the parameters. If known, we could generate the parameters.
- Algorithm:
  - Fit a value for the hyperparameters using the data.

Squared exponential kernel revisited

- A common option in practice and in software is to fit the *hyperparameters* of a more general *squared exponential kernel* from data
  - More general form of the squared exponential:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp(-\frac{1}{2} \sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{\ell_d^2})$$

signal variance lengthscales

  - *Parameters* (here, *f*) parametrize the distribution of the data. If we knew them, we could generate the data.
    - GPs: *nonparametric* model: infinite # of latent params
  - *Hyperparameters* parametrize the distribution of the parameters. If known, we could generate the parameters.
- Algorithm:
  - Fit a value for the hyperparameters using the data.
  - Given those values, now compute and report the mean and uncertainty intervals.

References (1/1)

Belkhiri, L., Tiri, A., & Mouni, L. (2020). Spatial distribution of the groundwater quality using kriging and Co-kriging interpolations. *Groundwater for Sustainable Development*, 11, 100473.

Berlinghieri, R., et al. (2023). Gaussian processes at the Helm(holtz): A more fluid model for ocean currents. *ICML*.

Binois, M., & Wycoff, N. (2022). A survey on high-dimensional Gaussian process modeling with application to Bayesian optimization. ACM Transactions on Evolutionary Learning and Optimization, 2(2), 1-26.

Garnett, R. (2023). *Bayesian Optimization*. Cambridge University Press.

Gramacy, R. B., & Lee, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. Journal of the American Statistical Association, 103(483), 1119-1130.

Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC.

Ryan, E., & Özgökmen, T. Image credit.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *NeurIPS.*

Snoek, J., et al. (2015). Scalable Bayesian optimization using deep neural networks. *ICML* (pp. 2171-2180). PMLR.

Williams, C. K., & Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Zewe, A. “A better way to study ocean currents.” MIT News. May 17, 2023.