But what is a Gaussian process?

Regression while knowing how certain you are

Quan Nguyen

Who am I?

- **Quan Nguyen**
- Ph.D. student – Bayesian machine learning, decision-making under uncertainty

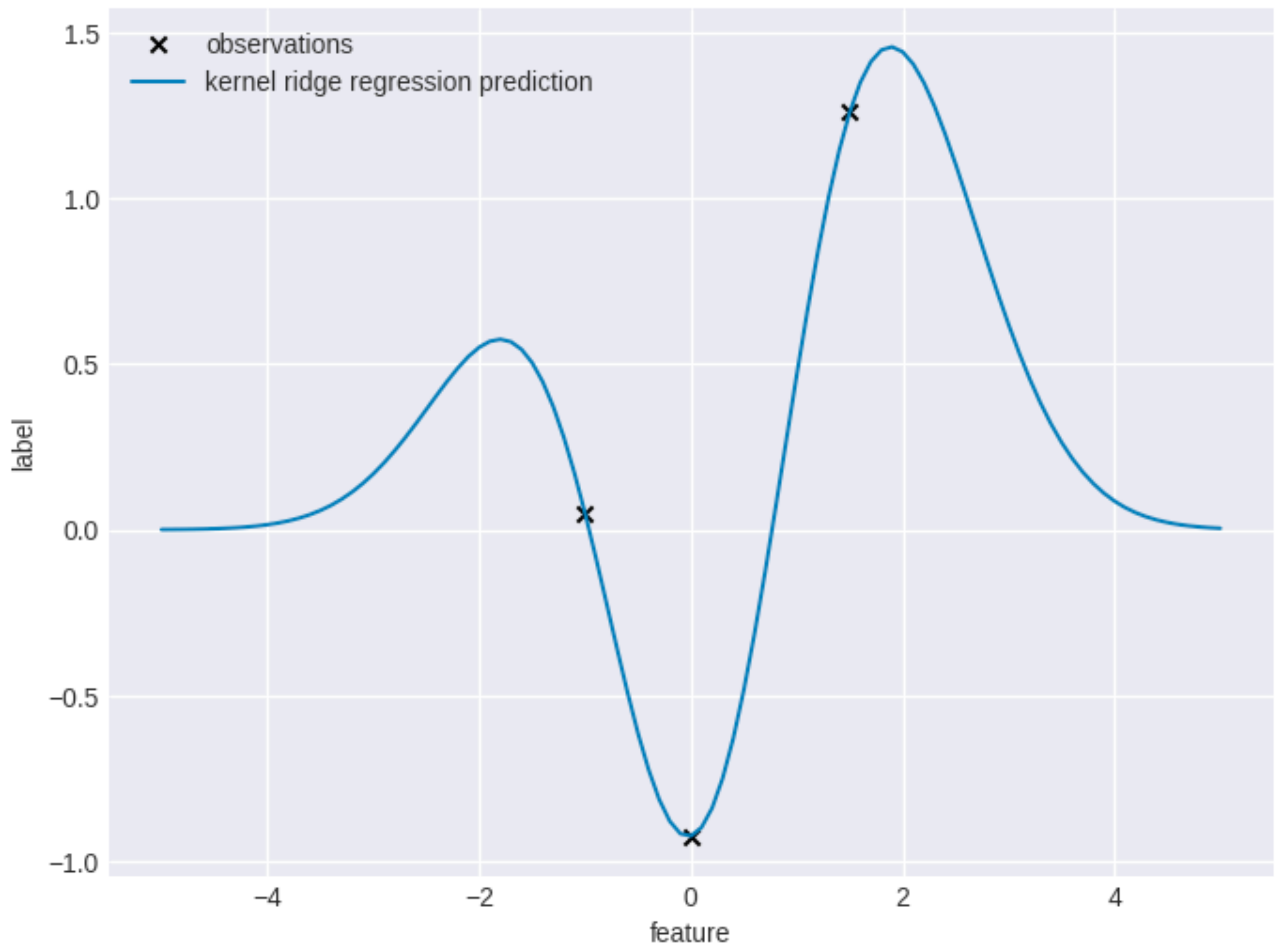
krisnguyen135.github.io

github.com/KrisNguyen135/Talks

@the_subtrahend

Interpolation vs. extrapolation

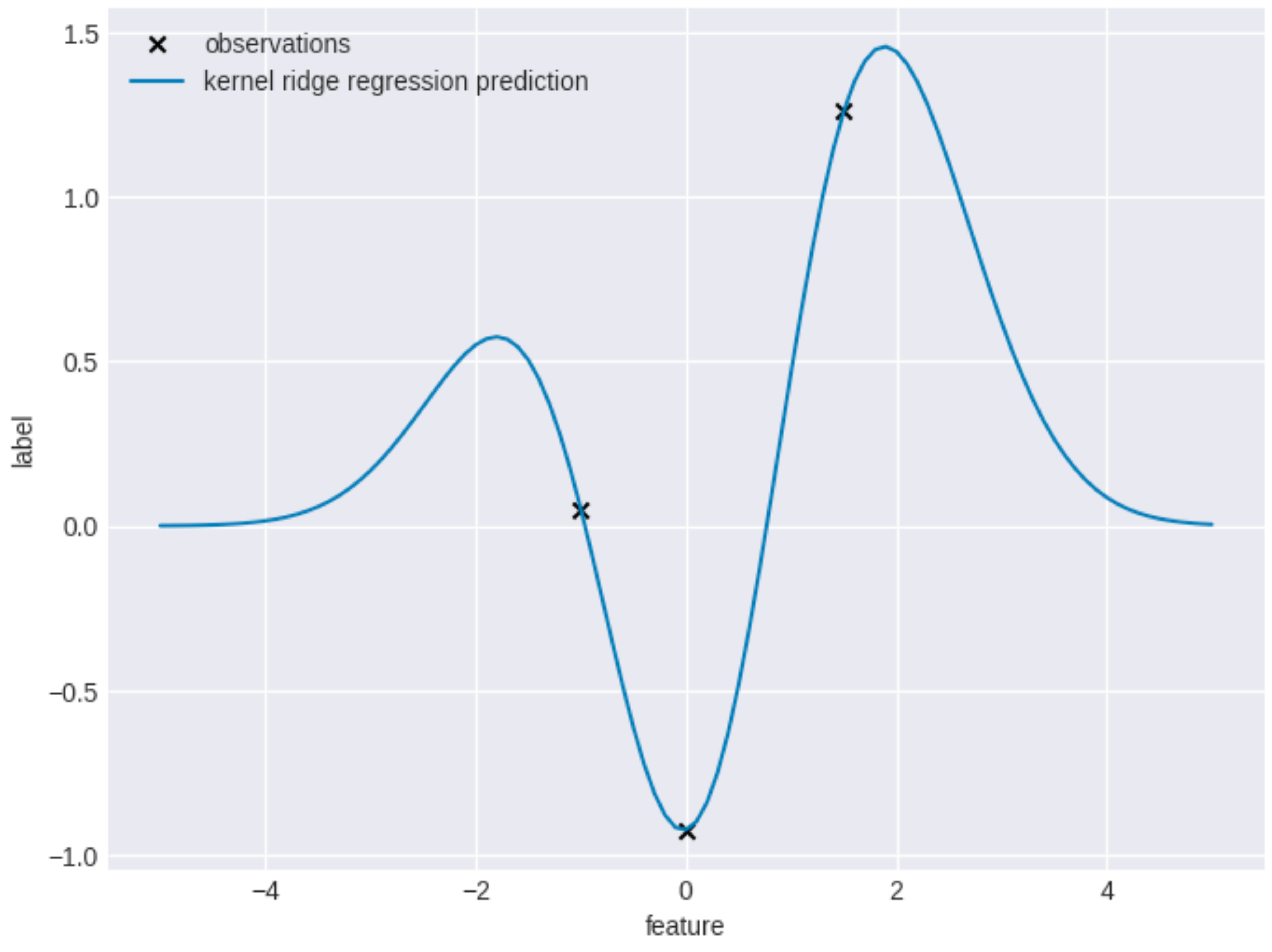
quantifying what we do and don’t know



Interpolation vs. extrapolation

quantifying what we do and don’t know

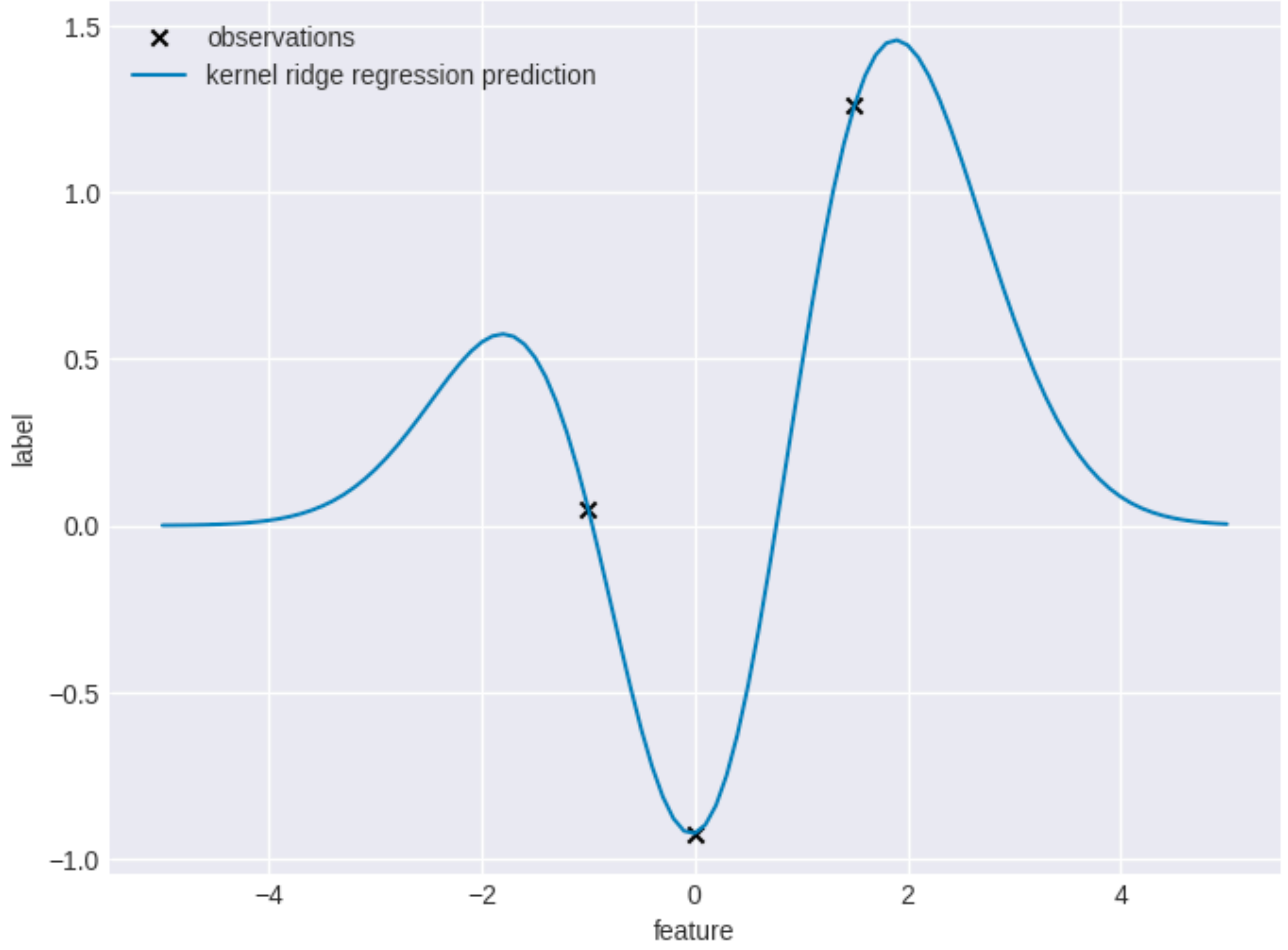
- **Typical ML models**: *single-valued* numbers as predictions



Interpolation vs. extrapolation

quantifying what we do and don’t know

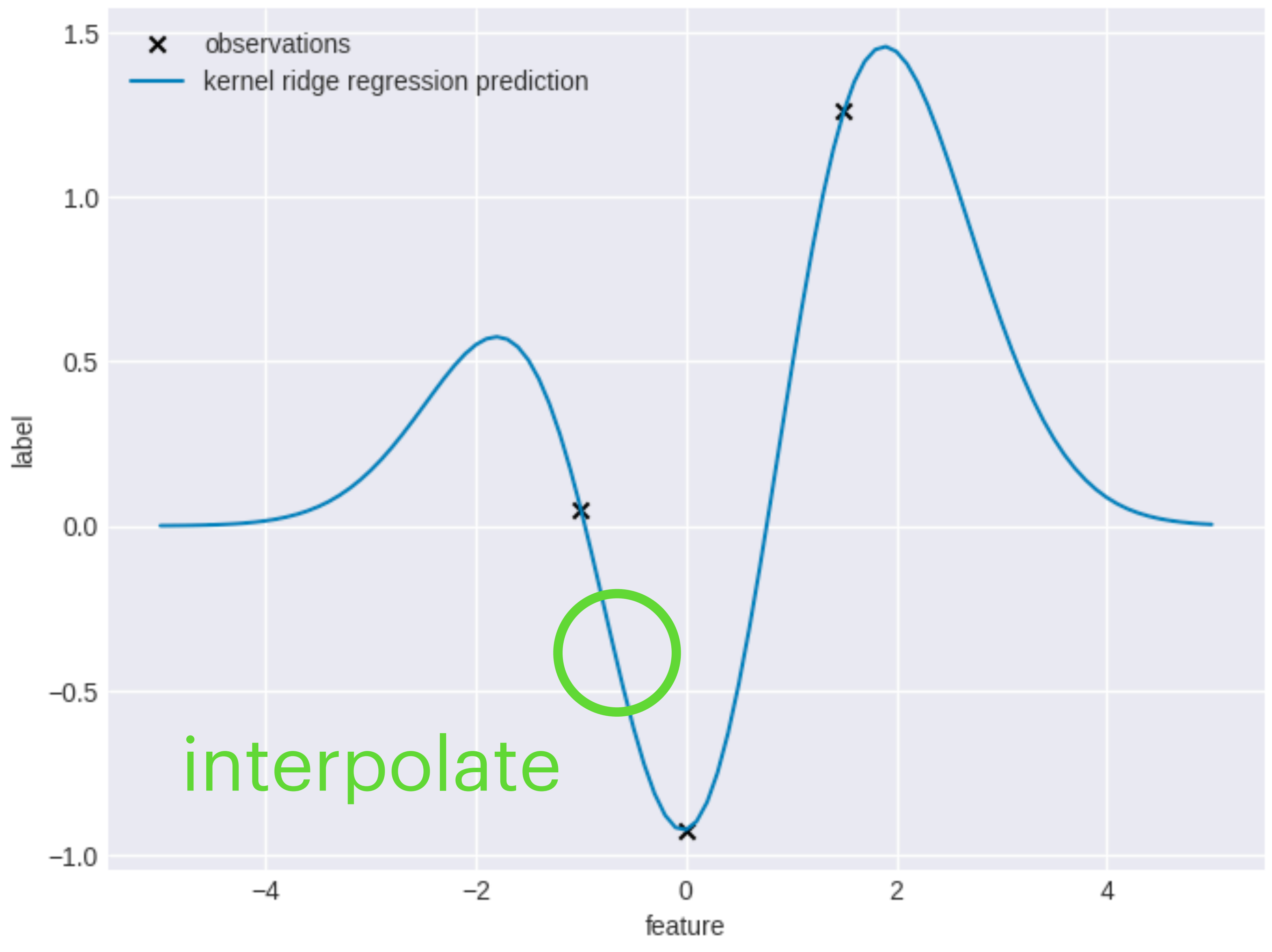
- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user



Interpolation vs. extrapolation

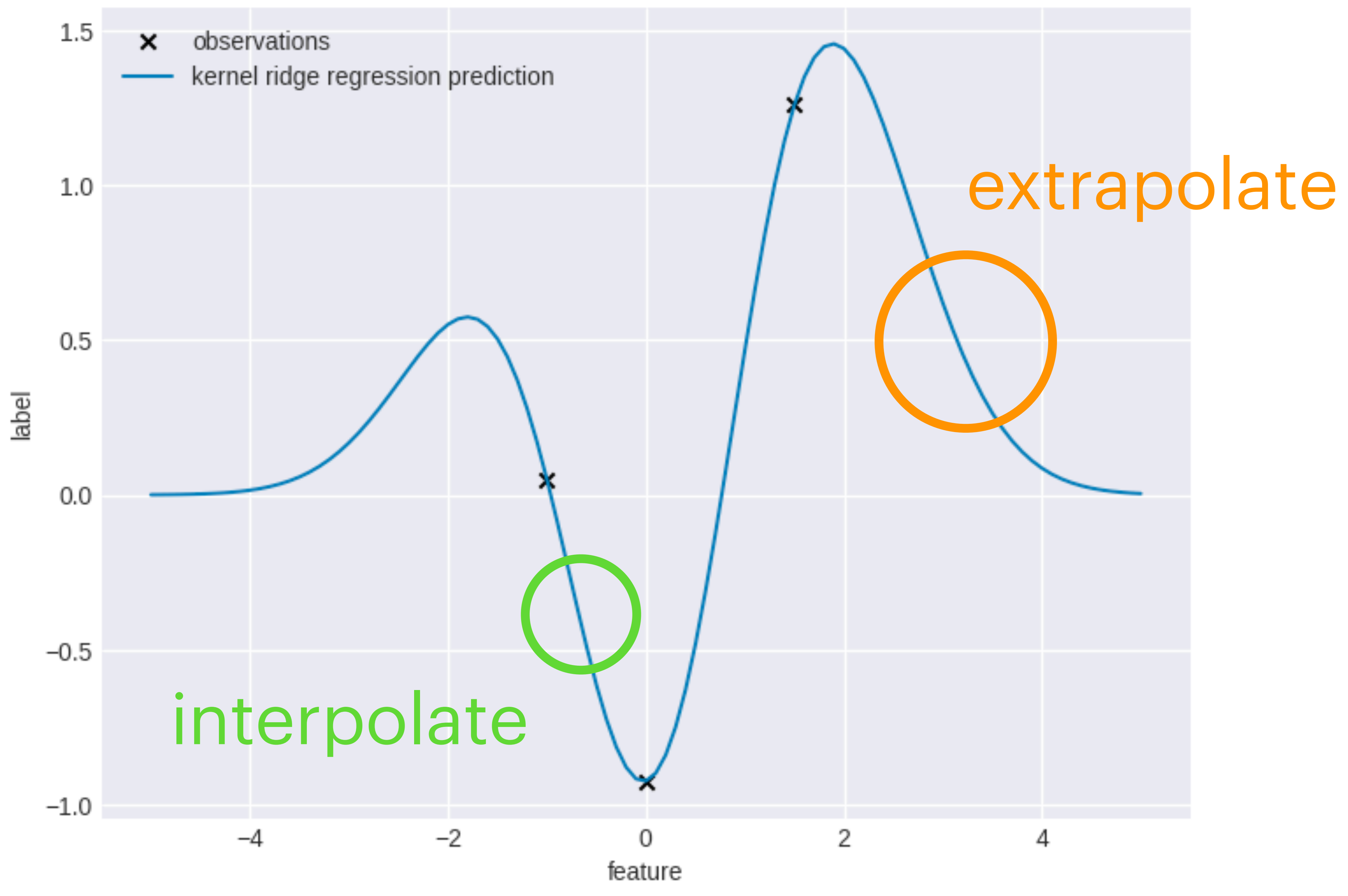
quantifying what we do and don't know

- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user



Interpolation vs. extrapolation
quantifying what we do and don't know

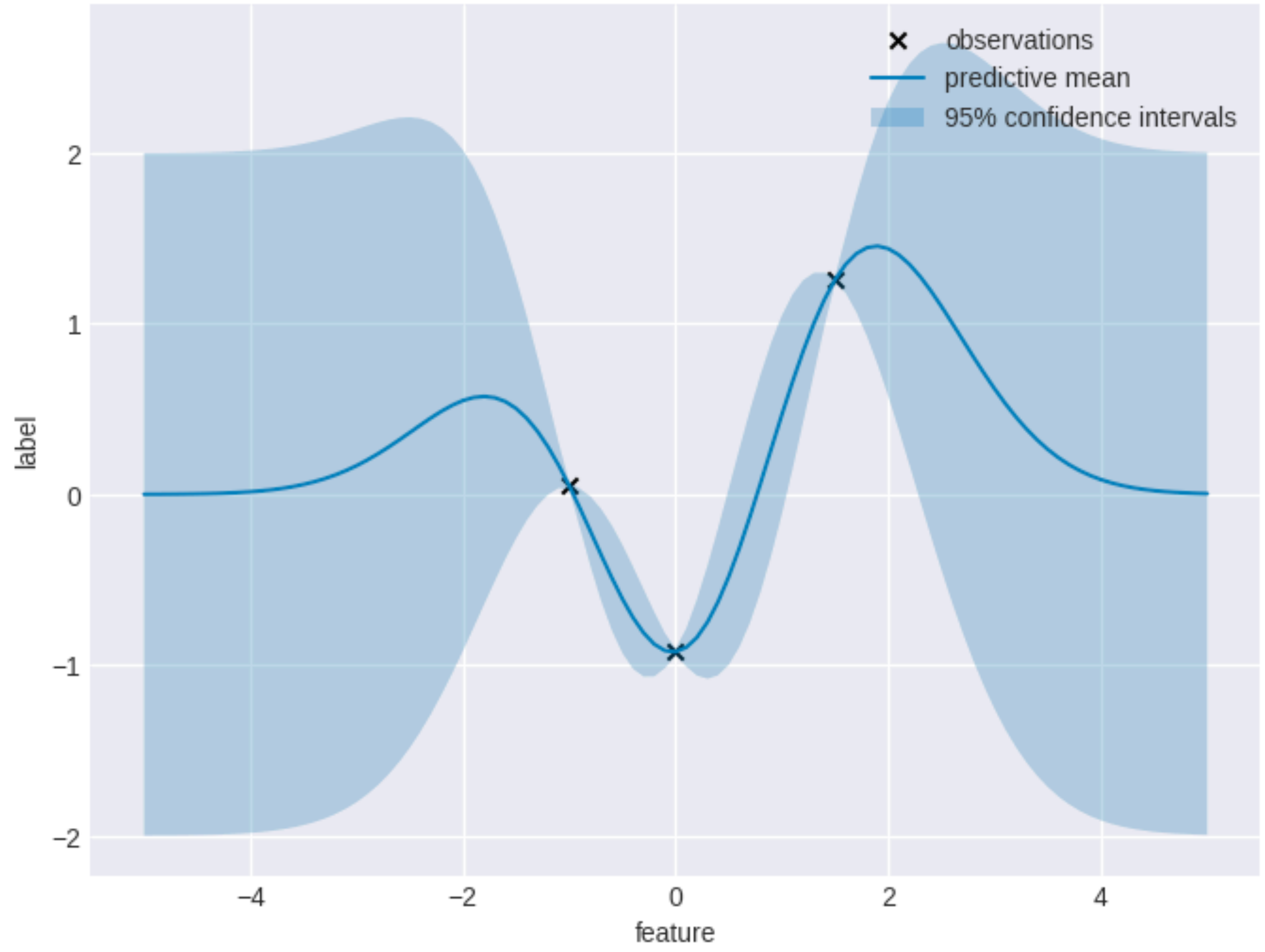
- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user



Interpolation vs. extrapolation

quantifying what we do and don't know

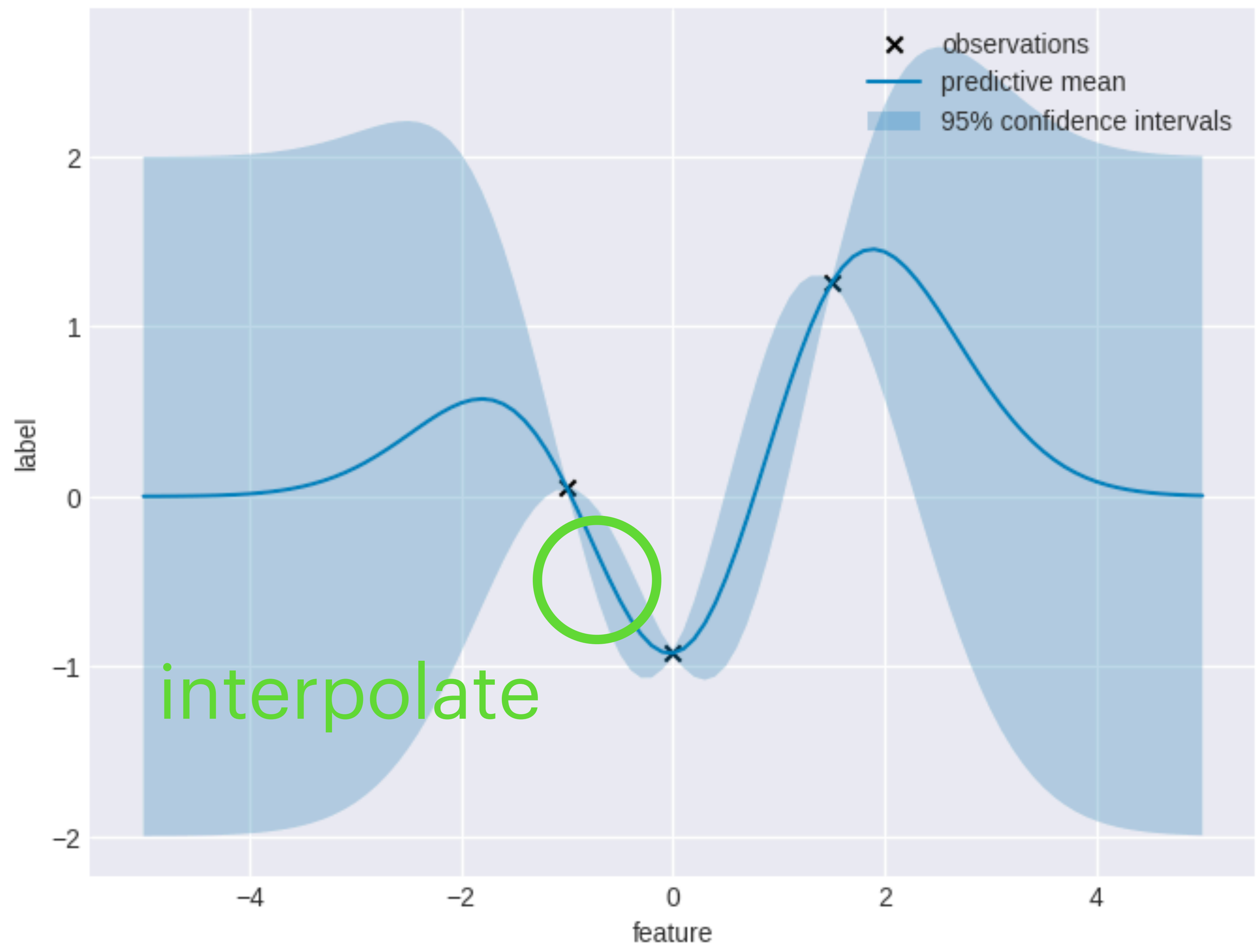
- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user
- **Introducing the Gaussian process**: account for how similar data points are



Interpolation vs. extrapolation

quantifying what we do and don’t know

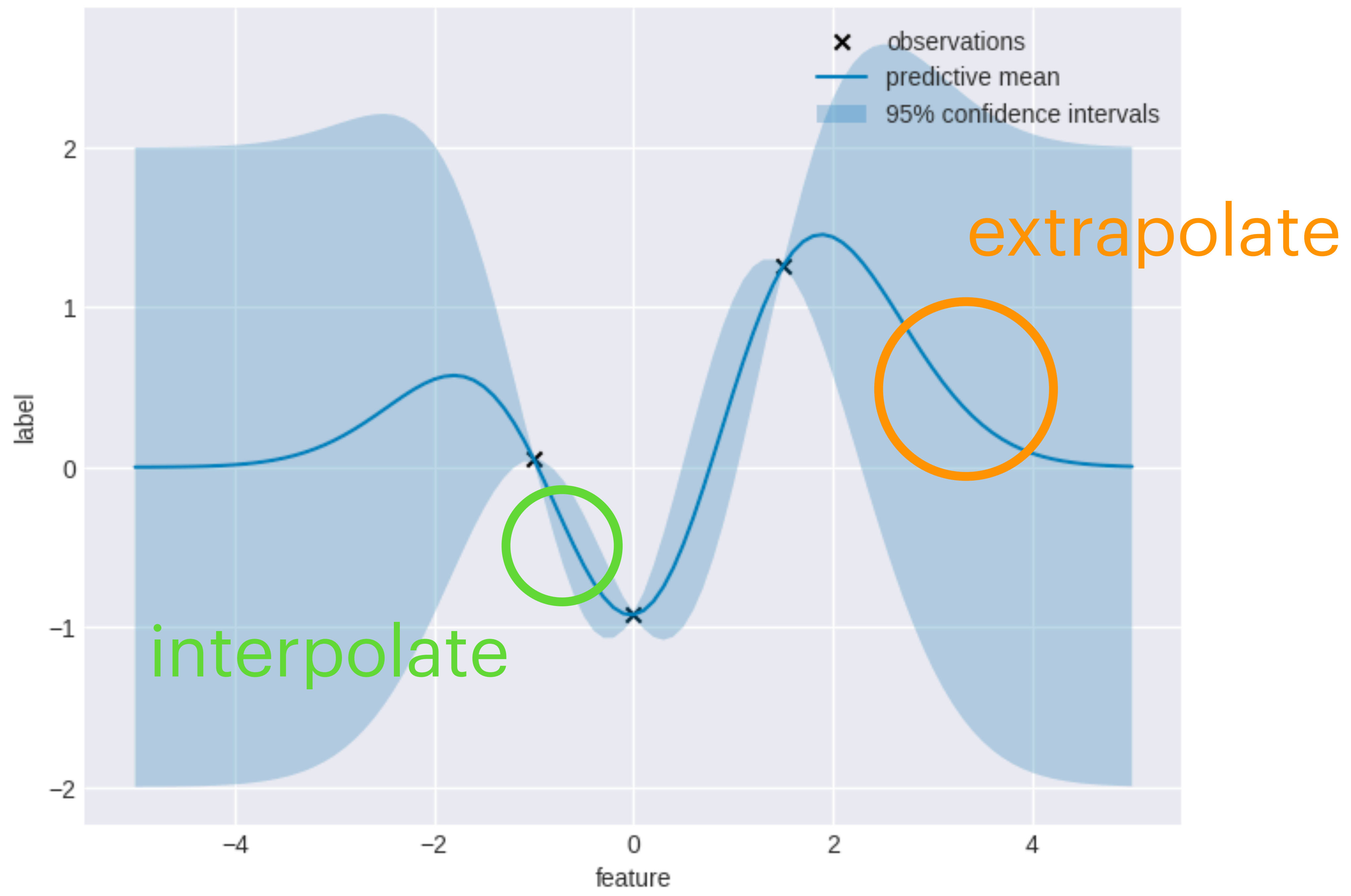
- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user
- **Introducing the Gaussian process**: account for how similar data points are
  - high confidence for test points similar to training points



Interpolation vs. extrapolation

quantifying what we do and don’t know

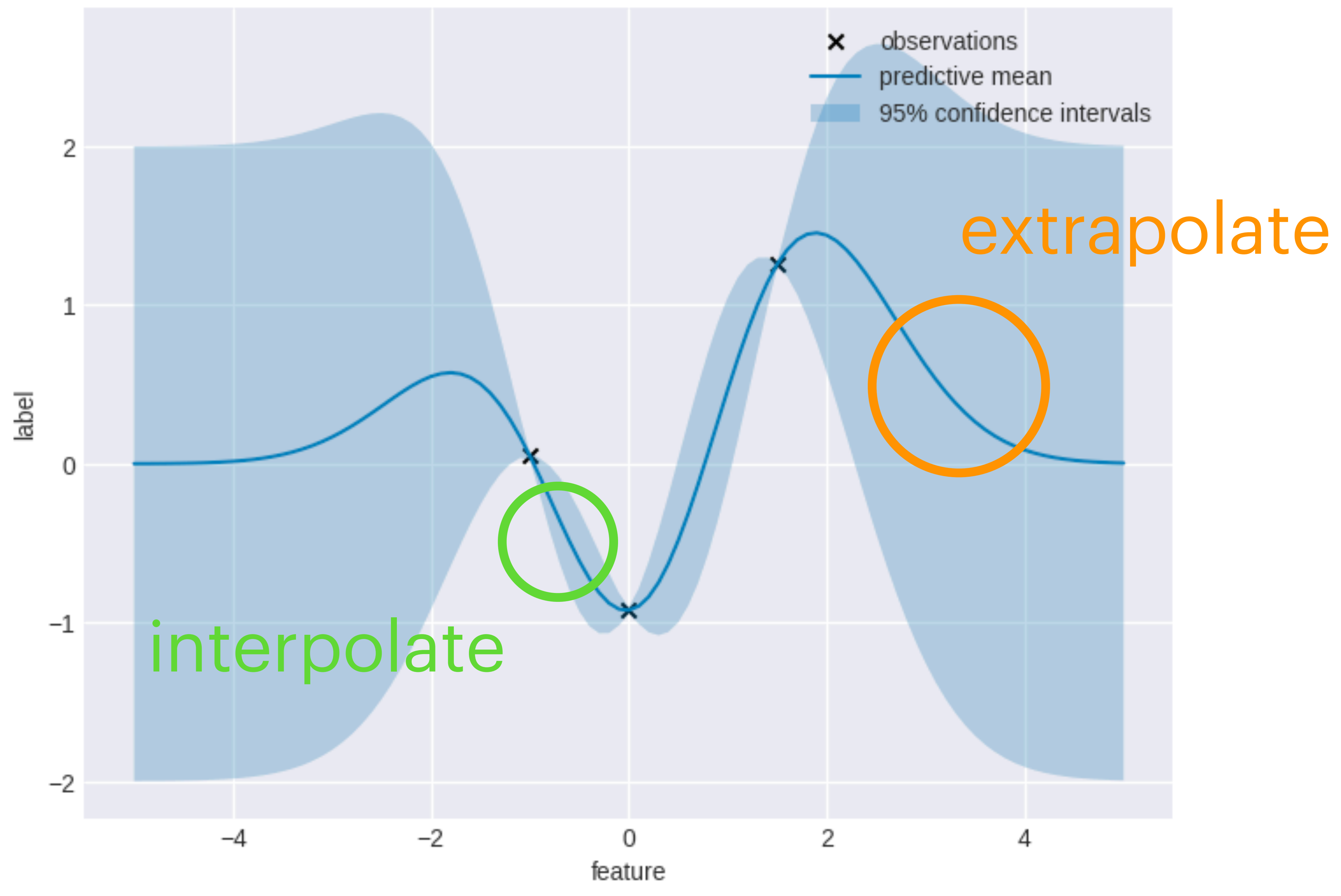
- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user
- **Introducing the Gaussian process**: account for how similar data points are
  - high confidence for test points similar to training points
  - low confidence for unseen test points



Interpolation vs. extrapolation

quantifying what we do and don't know

- **Typical ML models**: *single-valued* numbers as predictions
- **Challenge**: no error bars – lack of trust from user
- **Introducing the Gaussian process**: account for how similar data points are
  - high confidence for test points similar to training points
  - low confidence for unseen test points
  - good for decision making under uncertainty



Multivariate Gaussian distributions

a statistician’ way of
accounting for
similarity & uncertainty

Looking to sell your house?

see how much others are selling theirs for

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between $100k and $300k, but I'm not sure exactly how much... I need a reference point.

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between \$100k and \$300k, but I'm not sure exactly how much… I need a reference point.

Alice sold her house for \$250k.

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between $100k and $300k, but I'm not sure exactly how much… I need a reference point.

Alice sold her house for $250k.

Alix in SF? Very surprising for a house in California!
But that doesn't help me narrow the $100k–$300k range down.

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between $100k and $300k, but I'm not sure exactly how much… I need a reference point.

Alice sold her house for $250k.

Alix in SF? Very surprising for a house in California!
But that doesn't help me narrow the $100k–$300k range down.

No, Alice your neighbor!

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between $100k and $300k, but I'm not sure exactly how much… I need a reference point.

Alice sold her house for $250k.

Alix in SF? Very surprising for a house in California! But that doesn't help me narrow the $100k–$300k range down.

No, Alice your neighbor!

Ah, that's helpful, since our houses are very similar! My house should be around $190k–$270k then.

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between \$100k and \$300k, but I'm not sure exactly how much... I need a reference point.

test point: price of our house

Alice sold her house for \$250k.

Alix in SF? Very surprising for a house in California!
But that doesn't help me narrow the \$100k–\$300k range down.

No, Alice your neighbor!

Ah, that's helpful, since our houses are very similar! My house should be around \$190k–\$270k then.

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between \$100k and \$300k, but I'm not sure exactly how much... I need a reference point.

test point: price of our house

Alice sold her house for \$250k.

Alix in SF? Very surprising for a house in California!
But that doesn't help me narrow the \$100k–\$300k range down.

training point 1: price of SF house
not helpful

No, Alice your neighbor!

Ah, that's helpful, since our houses are very similar! My house should be around \$190k–\$270k then.

Looking to sell your house?

see how much others are selling theirs for

I guess my house is worth somewhere between \$100k and \$300k, but I'm not sure exactly how much... I need a reference point.

test point: price of our house

Alice sold her house for \$250k.

Alix in SF? Very surprising for a house in California! But that doesn't help me narrow the \$100k–\$300k range down.

training point 1: price of SF house
not helpful

No, Alice your neighbor!

Ah, that's helpful, since our houses are very similar! My house should be around \$190k–\$270k then.

training point 2: price of neighbor house
very helpful

Multivariate Gaussian distributions

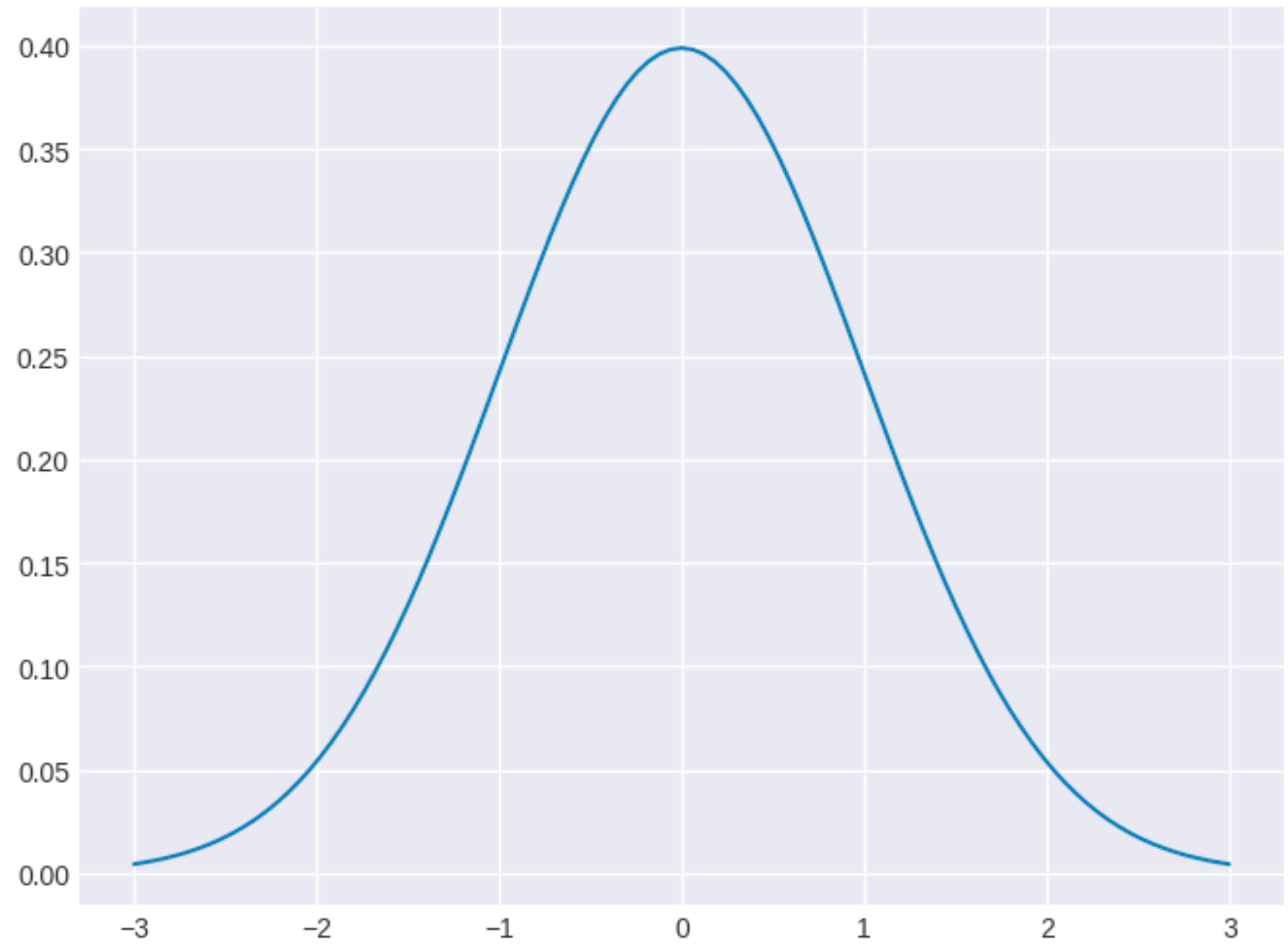
the high-dimensional bell curve

Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve

$$X \sim N(\mu, \sigma^2)$$



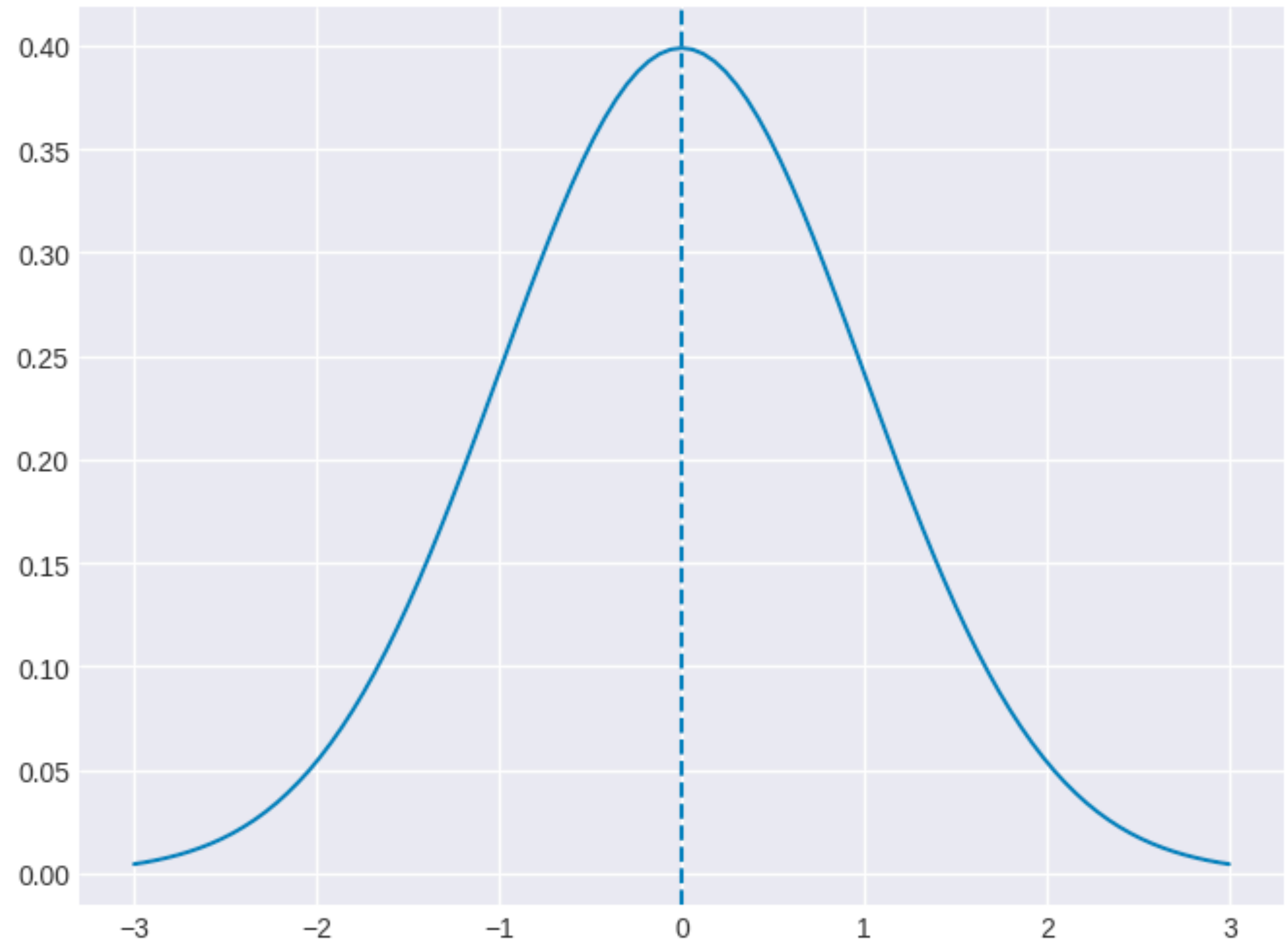
Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
    - mean μ and variance σ^2

$$X \sim N(\mu, \sigma^2)$$

μ

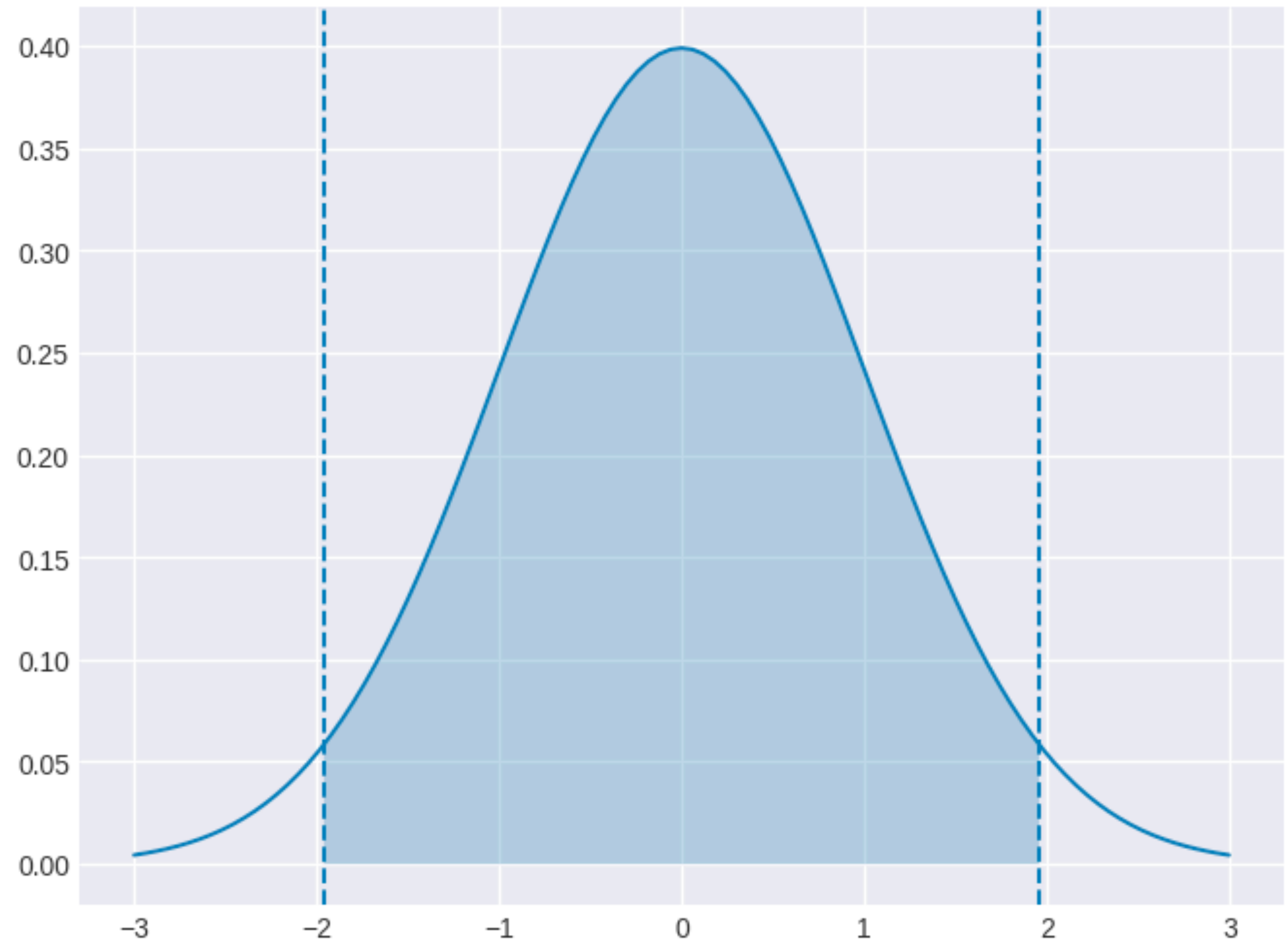


Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$

$$X \sim N(\mu, \sigma^2)$$



$$\mu \pm 1.96\,\sigma$$

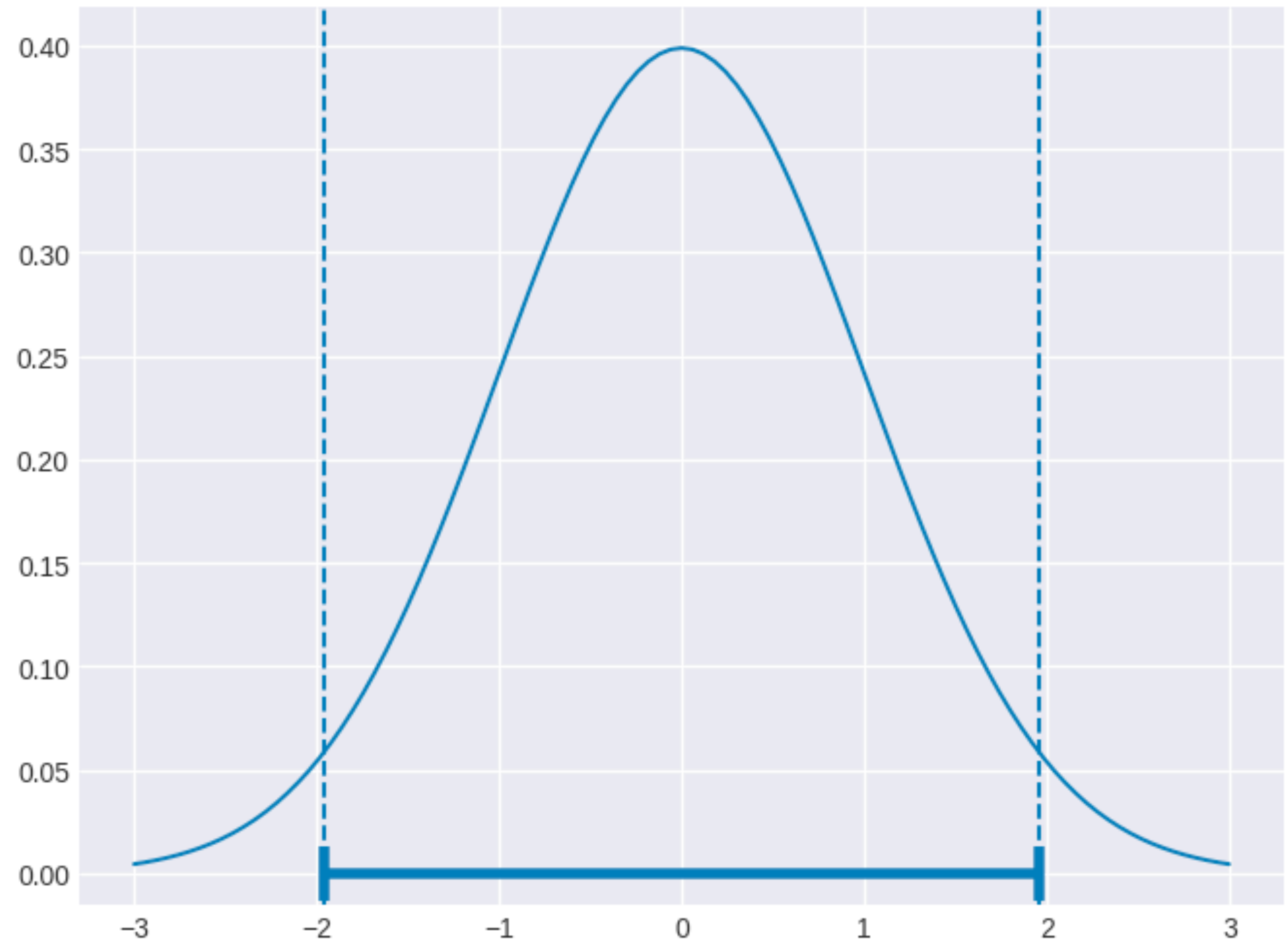
Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars)

    $\mu \pm 1.96\,\sigma$

$$X \sim N(\mu, \sigma^2)$$



$\mu \pm 1.96\,\sigma$

Multivariate Gaussian distributions

the high-dimensional bell curve

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix} = \mathbf{X} \sim N(\boldsymbol{\mu}, \Sigma)$$

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations

Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix} = \mathbf{X} \sim N\left(\boldsymbol{\mu}, \Sigma\right)$$

Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars)
    $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_d \end{bmatrix} = \mathbf{X} \sim N\left(\boldsymbol{\mu}, \Sigma\right)$$

- diagonal Σ_{ii}: variance
- off-diagonal Σ_{ij}: covariance

Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

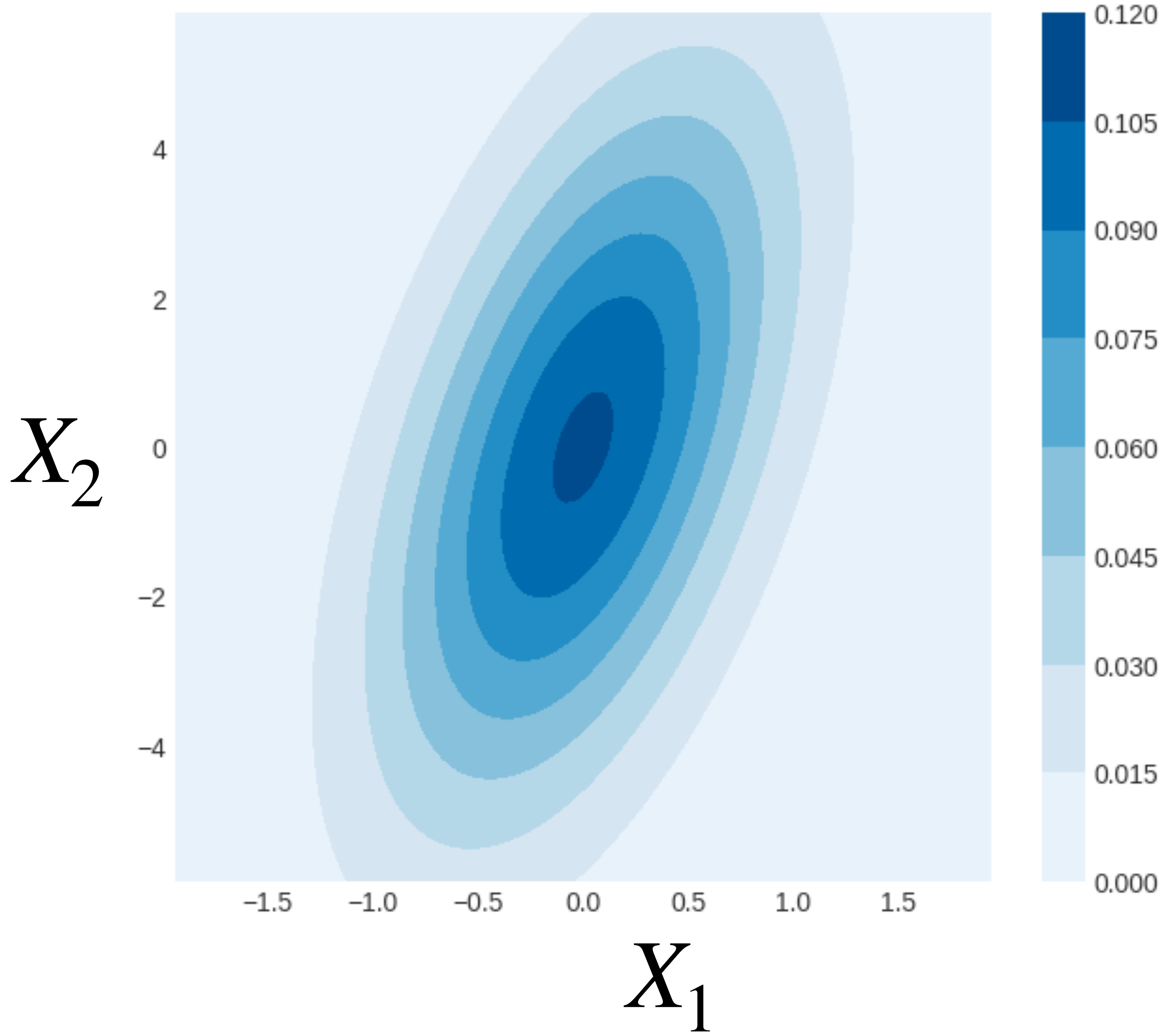
$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix} \right)$$

Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix} \right)$$

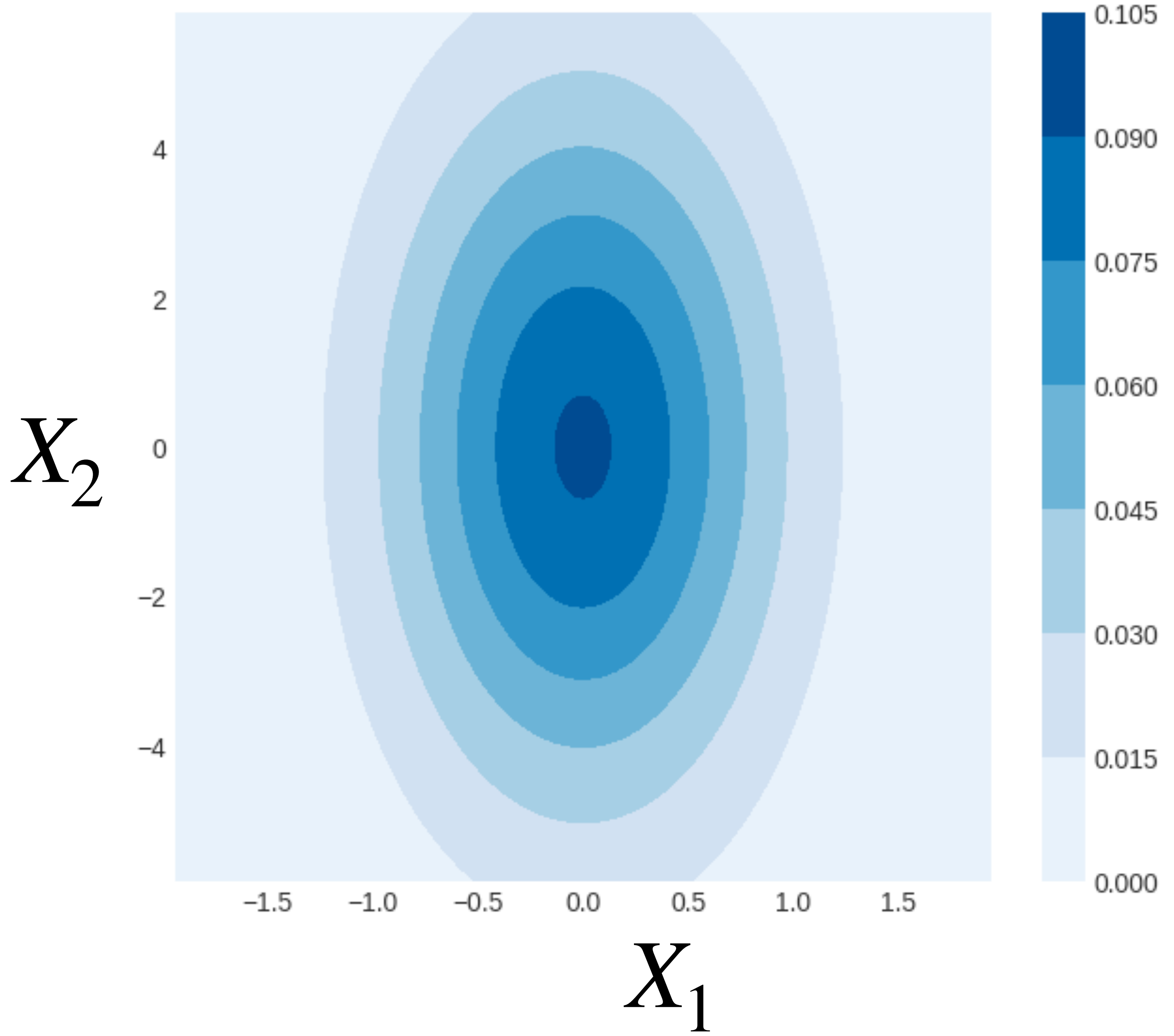


Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix}\right)$$



Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

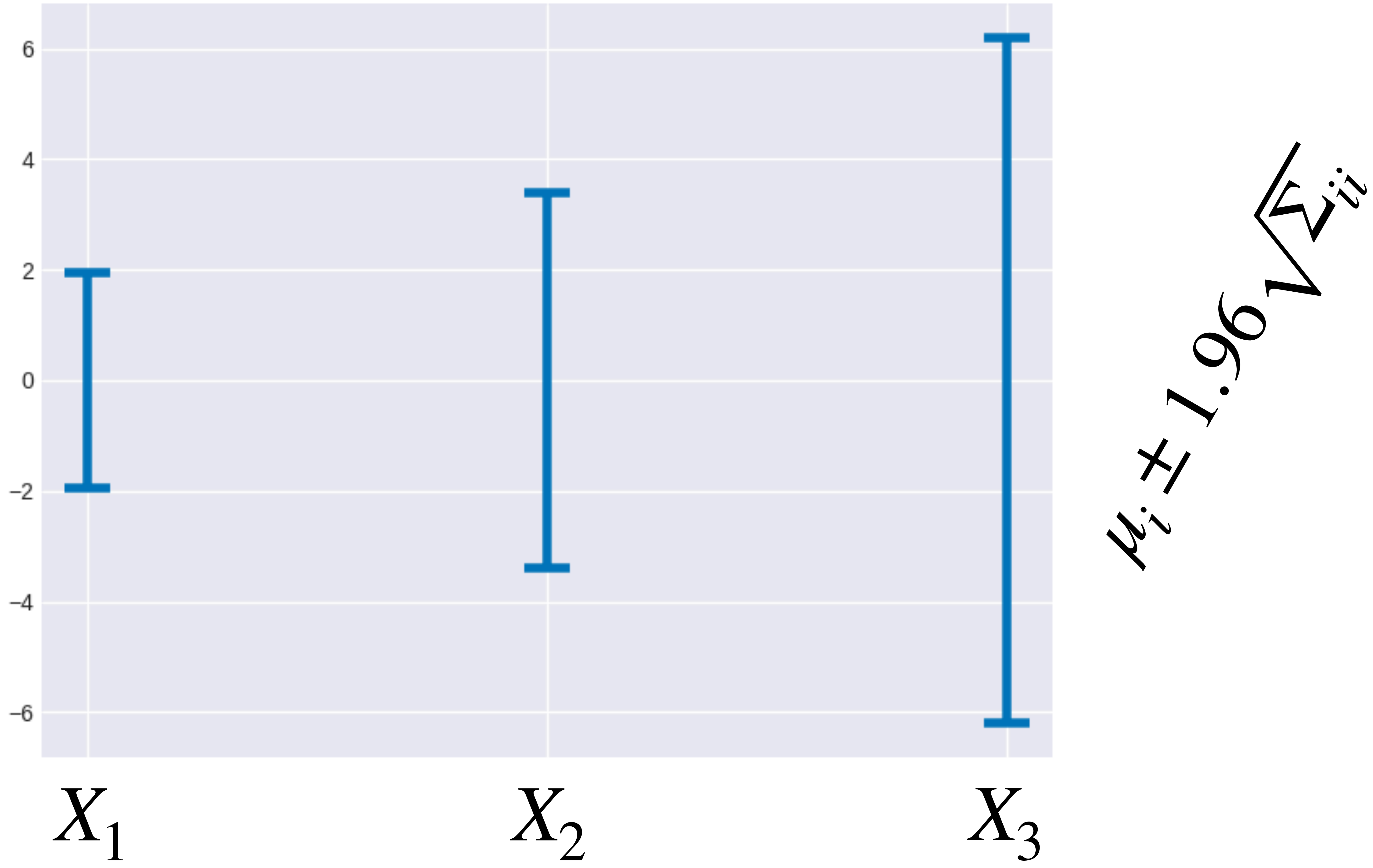
$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 3 & 0 \\ 0 & 0 & 10 \end{bmatrix} \right)$$

Multivariate Gaussian distributions

the high-dimensional bell curve

- **Gaussian distribution**: the familiar bell curve
  - mean μ and variance σ^2
  - 95% confidence interval (error bars) $\mu \pm 1.96\,\sigma$
- **Multivariate Gaussian**: multiple Gaussian variables with correlations
  - mean vector (normalized) $\boldsymbol{\mu}$ and covariance matrix Σ

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 3 & 0 \\ 0 & 0 & 10 \end{bmatrix} \right)$$



Updating Gaussian distributions

refining our beliefs with Bayes' rule

Updating Gaussian distributions

refining our beliefs with Bayes' rule

Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \ldots\, X_d]^\top$

Updating Gaussian distributions

refining our beliefs with Bayes' rule

Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \ldots\, X_d]^\top$

- observed the values of some X_i

Updating Gaussian distributions

refining our beliefs with Bayes' rule

Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \ldots\, X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

Updating Gaussian distributions

refining our beliefs with Bayes' rule

Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \ldots\, X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

A: still a multivariate Gaussian (with updated mean and covariance)!

Updating Gaussian distributions

refining our beliefs with Bayes' rule

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix}\right)$$

Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \ldots\, X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

A: still a multivariate Gaussian (with updated mean and covariance)!

Updating Gaussian distributions

refining our beliefs with Bayes' rule

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix}\right)$$

Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \ldots\, X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

A: still a multivariate Gaussian (with updated mean and covariance)!

X_1 X_2

$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

Updating Gaussian distributions

refining our beliefs with Bayes' rule

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix}\right)$$

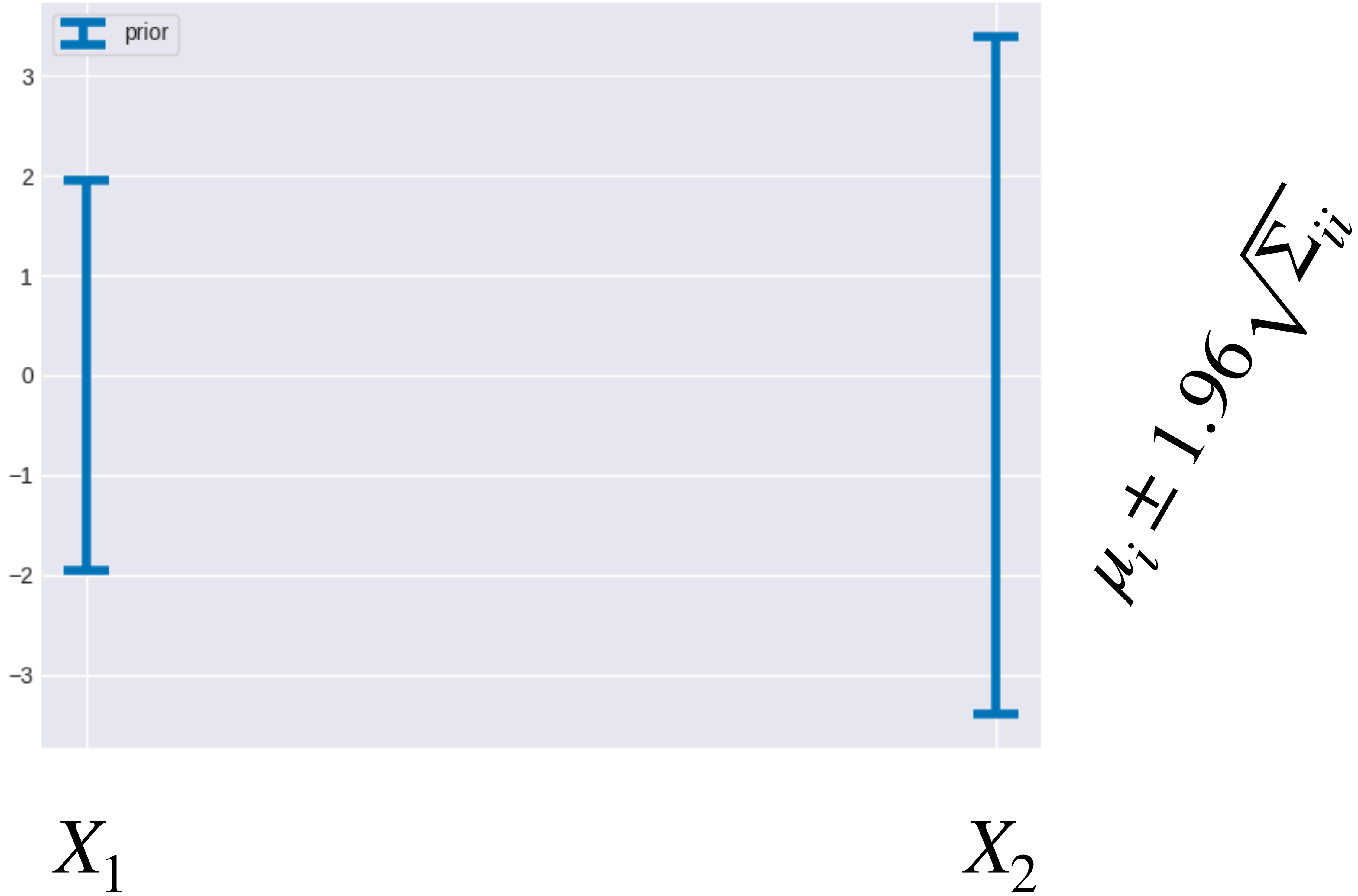
Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\, \dots\, X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

A: still a multivariate Gaussian (with updated mean and covariance)!

observation $X_2 = 3$



Updating Gaussian distributions

refining our beliefs with Bayes’ rule

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix} \right)$$

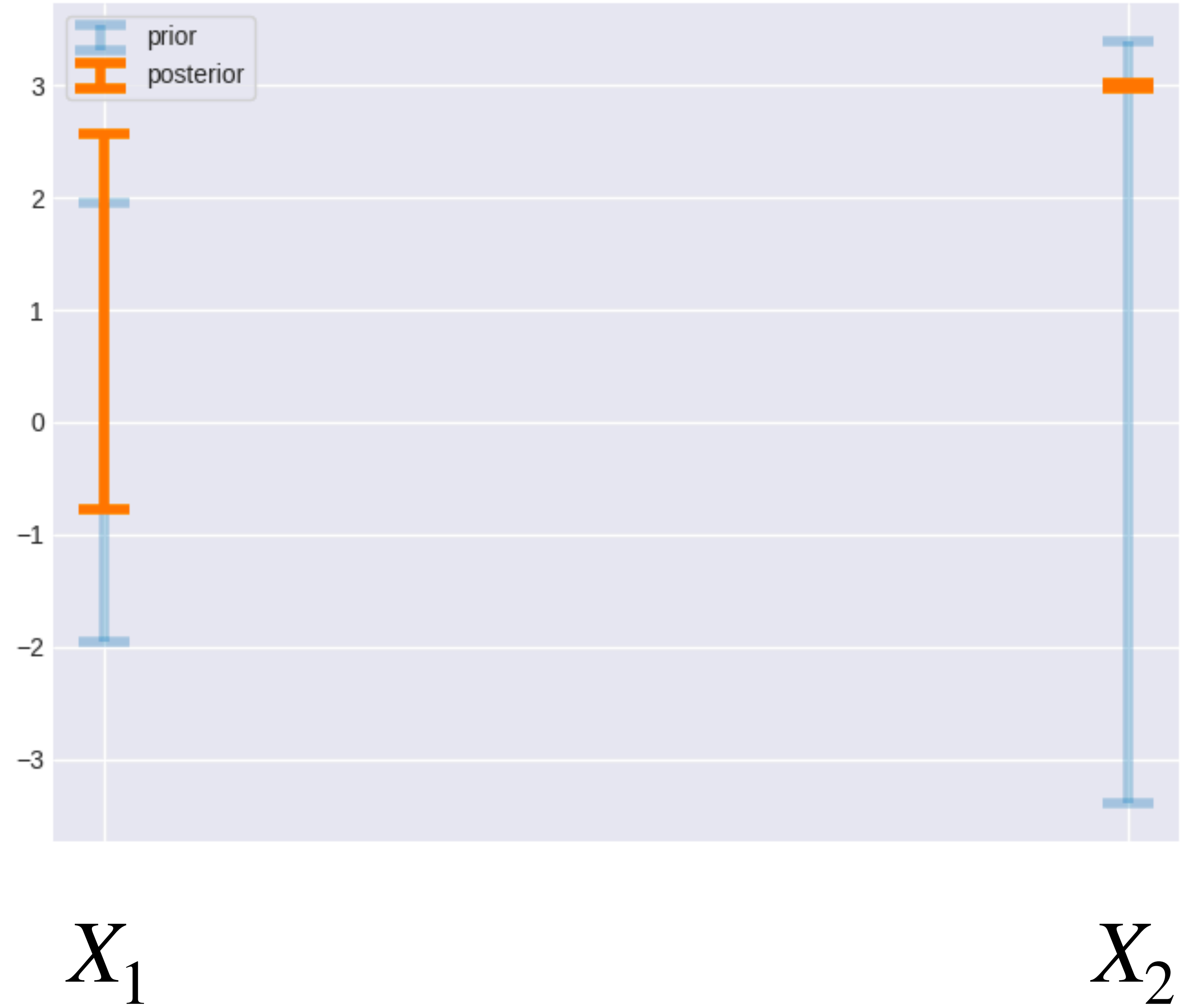
Q: multivariate Gaussian variable

$\mathbf{X} = [X_1\, X_2\ \dots\ X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j’s?

A: still a multivariate Gaussian (with updated mean and covariance)!

observation $X_2 = 3$



$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

Updating Gaussian distributions

refining our beliefs with Bayes' rule

Q: multivariate Gaussian variable

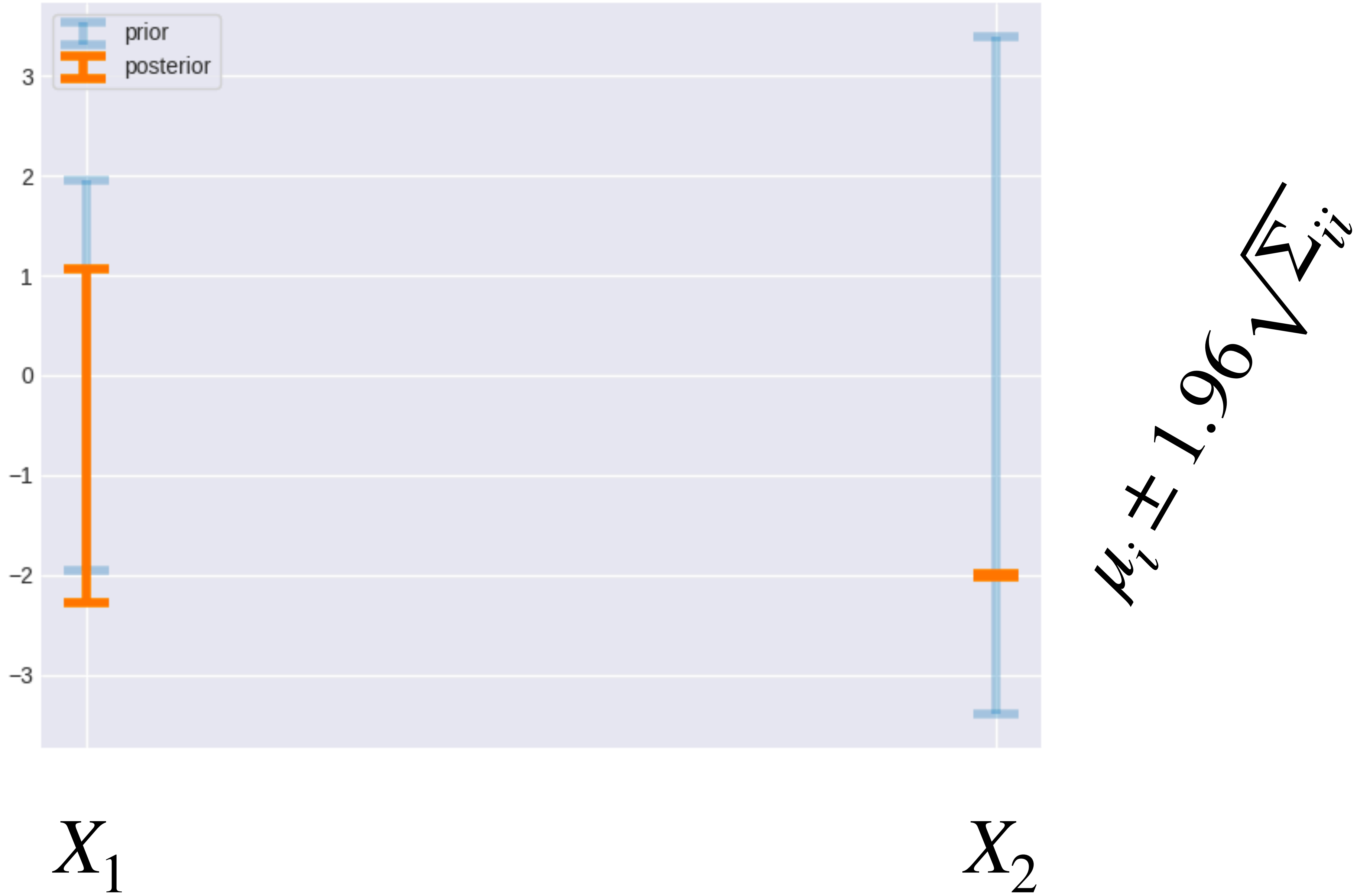
$\mathbf{X} = [X_1\, X_2\ \ldots\ X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

A: still a multivariate Gaussian (with updated mean and covariance)!

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 3 \end{bmatrix} \right)$$

observation $X_2 = -2$



Updating Gaussian distributions

refining our beliefs with Bayes' rule

Q: multivariate Gaussian variable

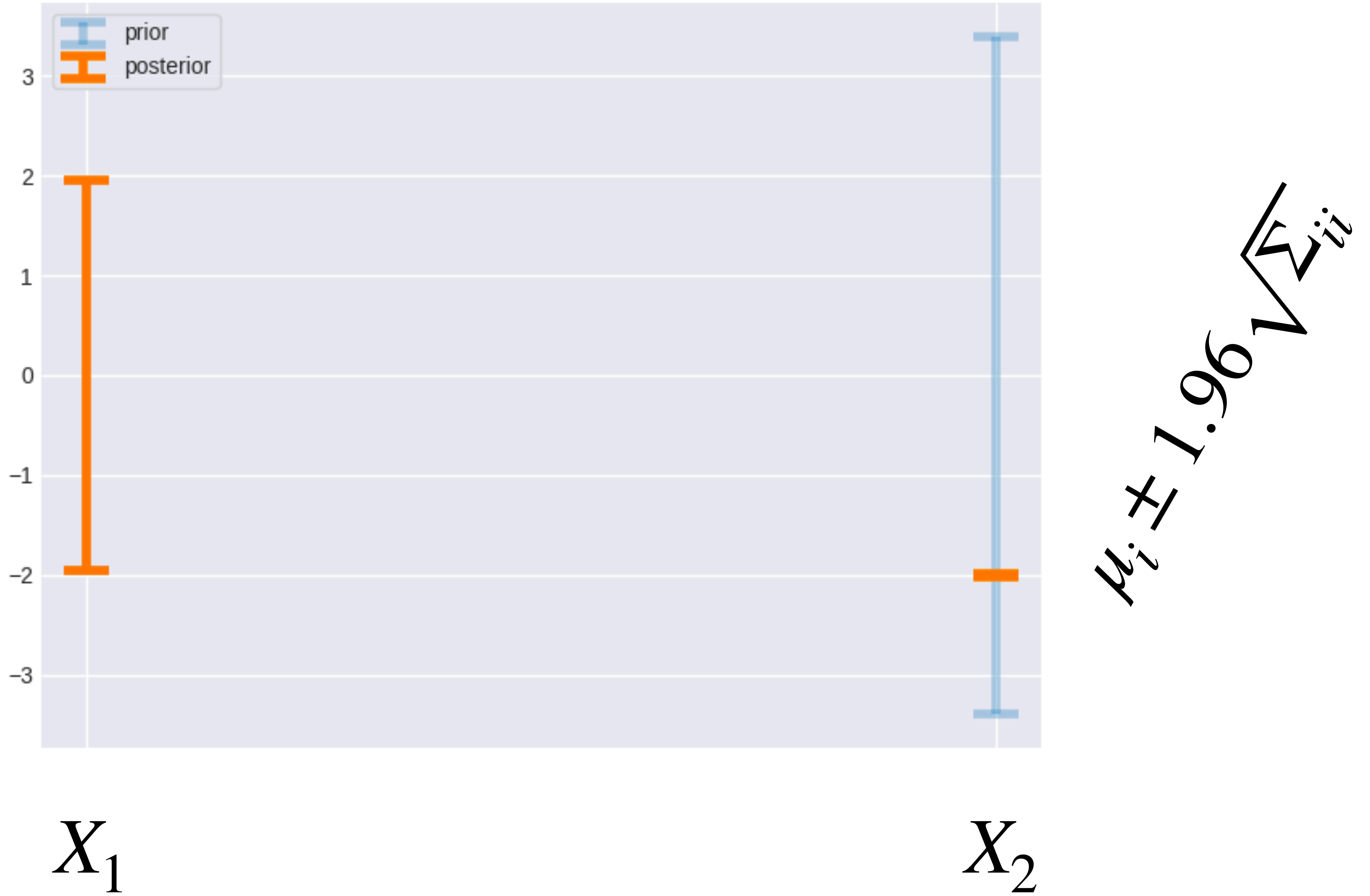
$\mathbf{X} = [X_1\, X_2\, \dots\, X_d]^\top$

- observed the values of some X_i
- what is the *posterior distribution* of remaining X_j's?

A: still a multivariate Gaussian (with updated mean and covariance)!

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \mathbf{X} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \right)$$

observation $X_2 = -2$



Setting house prices, the Bayesian way

updating our Gaussian beliefs

Setting house prices, the Bayesian way

updating our Gaussian beliefs

Model unknown house prices as

random variables

Setting house prices, the Bayesian way

updating our Gaussian beliefs

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances

Setting house prices, the Bayesian way

updating our Gaussian beliefs

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

Setting house prices, the Bayesian way

updating our Gaussian beliefs

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

$$\begin{bmatrix} \text{ours in MO} \\ \text{Alice's in MO} \\ \text{Alix's in CA} \end{bmatrix} \sim N\left(\begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.3 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \right)$$

x $100,000

Setting house prices, the Bayesian way

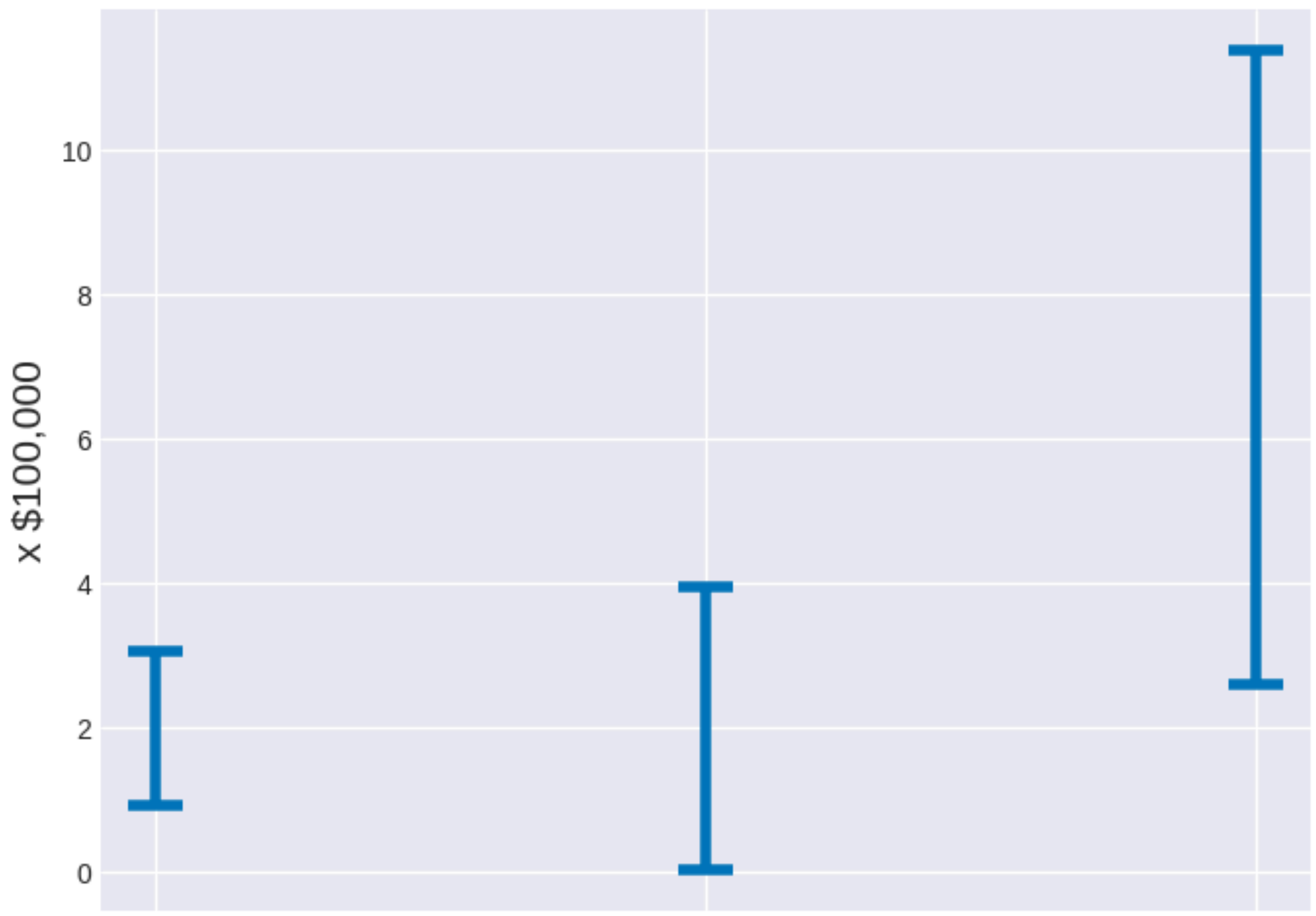
updating our Gaussian beliefs

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

$$\begin{bmatrix} \text{ours in MO} \\ \text{Alice's in MO} \\ \text{Alix's in CA} \end{bmatrix} \sim N\left(\begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.3 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \right)$$

x $100,000



$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

ours in MO Alice’s in MO Alix’s in CA

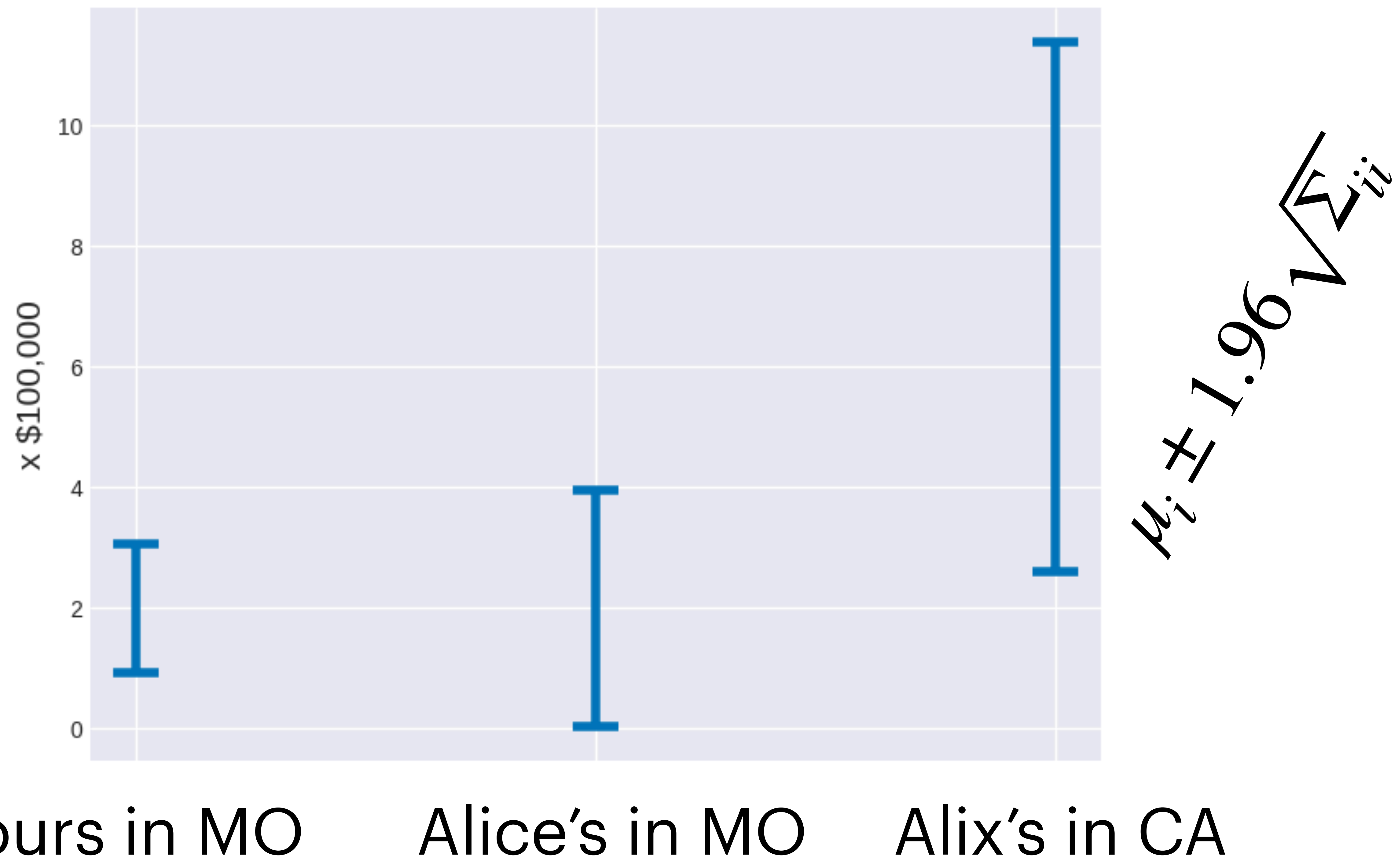
Setting house prices, the Bayesian way
updating our Gaussian beliefs

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

$$\begin{bmatrix} \text{ours in MO} \\ \text{Alice's in MO} \\ \text{Alix's in CA} \end{bmatrix} \sim N\left(\begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.3 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \right)$$

x $100,000



Alix in SF sold her house for $250k? Very surprising! But that doesn't help me narrow the $100k–$300k range down.

Setting house prices, the Bayesian way

updating our Gaussian beliefs

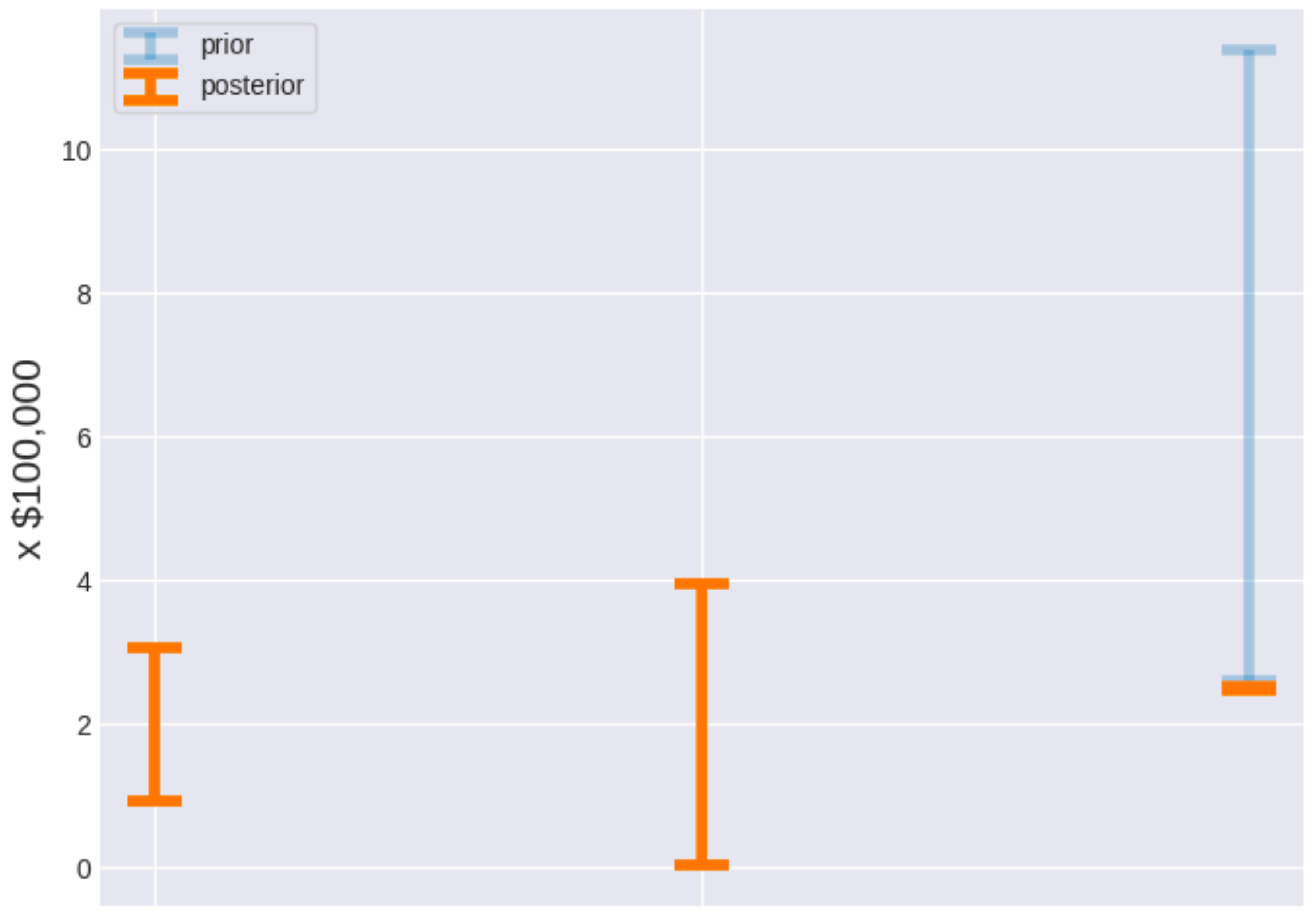
$$\begin{bmatrix} \text{ours in MO} \\ \text{Alice's in MO} \\ \text{Alix's in CA} \end{bmatrix} \sim N\left(\begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.3 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \right)$$

x $100,000

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

Alix in SF sold her house for $250k?
Very surprising!
But that doesn't help me narrow the $100k–$300k range down.



ours in MO | Alice's in MO | Alix's in CA

$$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$$

Setting house prices, the Bayesian way

updating our Gaussian beliefs

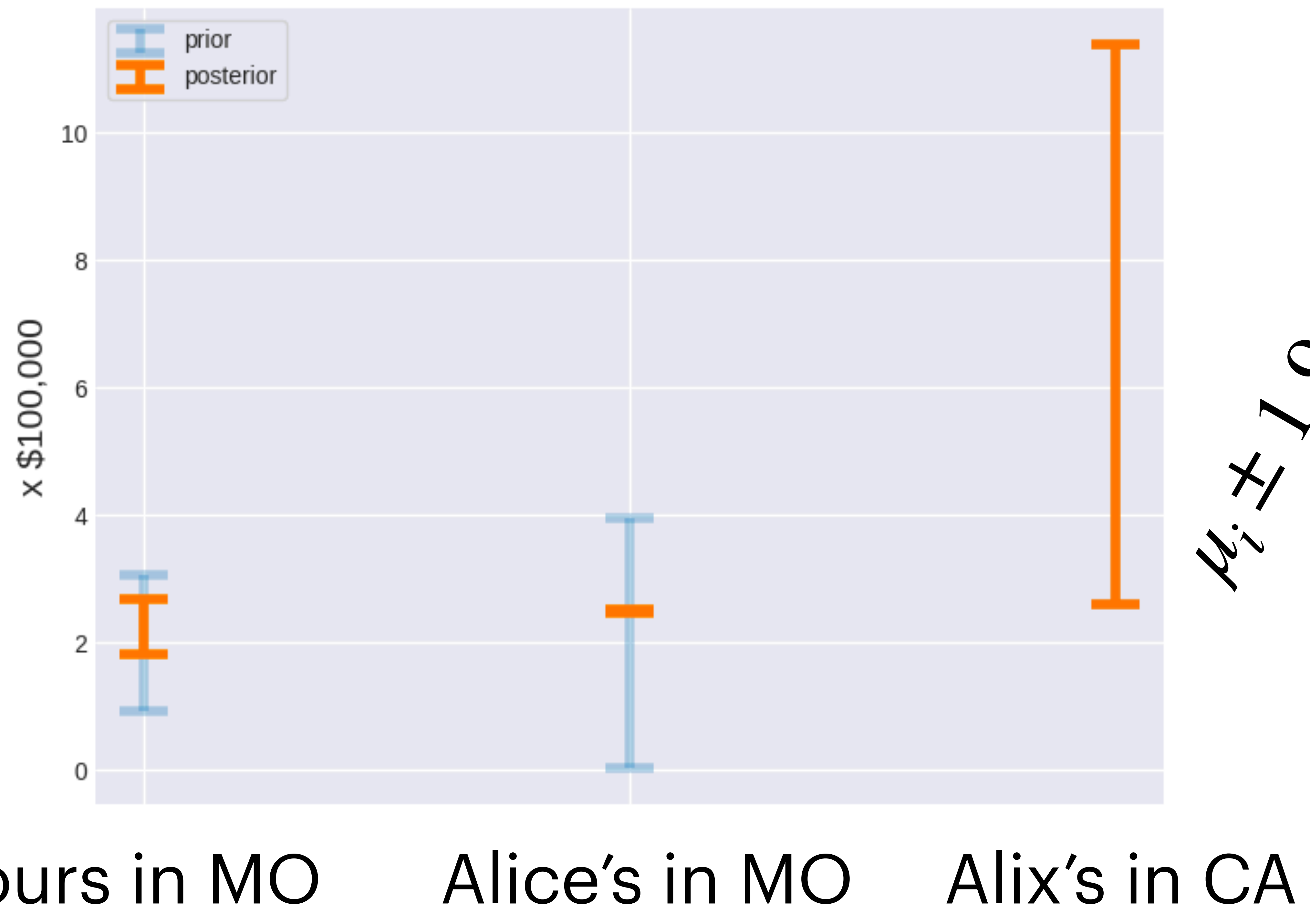
Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

Knowing Alice's house is worth $250k is helpful, since our houses are very similar! My house should be around $190k–$270k then.

$$\begin{bmatrix} \text{ours in MO} \\ \text{Alice's in MO} \\ \text{Alix's in CA} \end{bmatrix} \sim N\left(\begin{bmatrix} 2 \\ 2 \\ 5 \end{bmatrix}, \begin{bmatrix} 0.3 & 0.5 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0 & 5 \end{bmatrix} \right)$$

x $100,000



$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

Setting many house prices

updating a multivariate Gaussian distribution

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

Setting many house prices

updating a multivariate Gaussian distribution

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

20 houses on a block

- nearby houses have similar prices
- far-away houses have uncorrelated prices

Setting many house prices

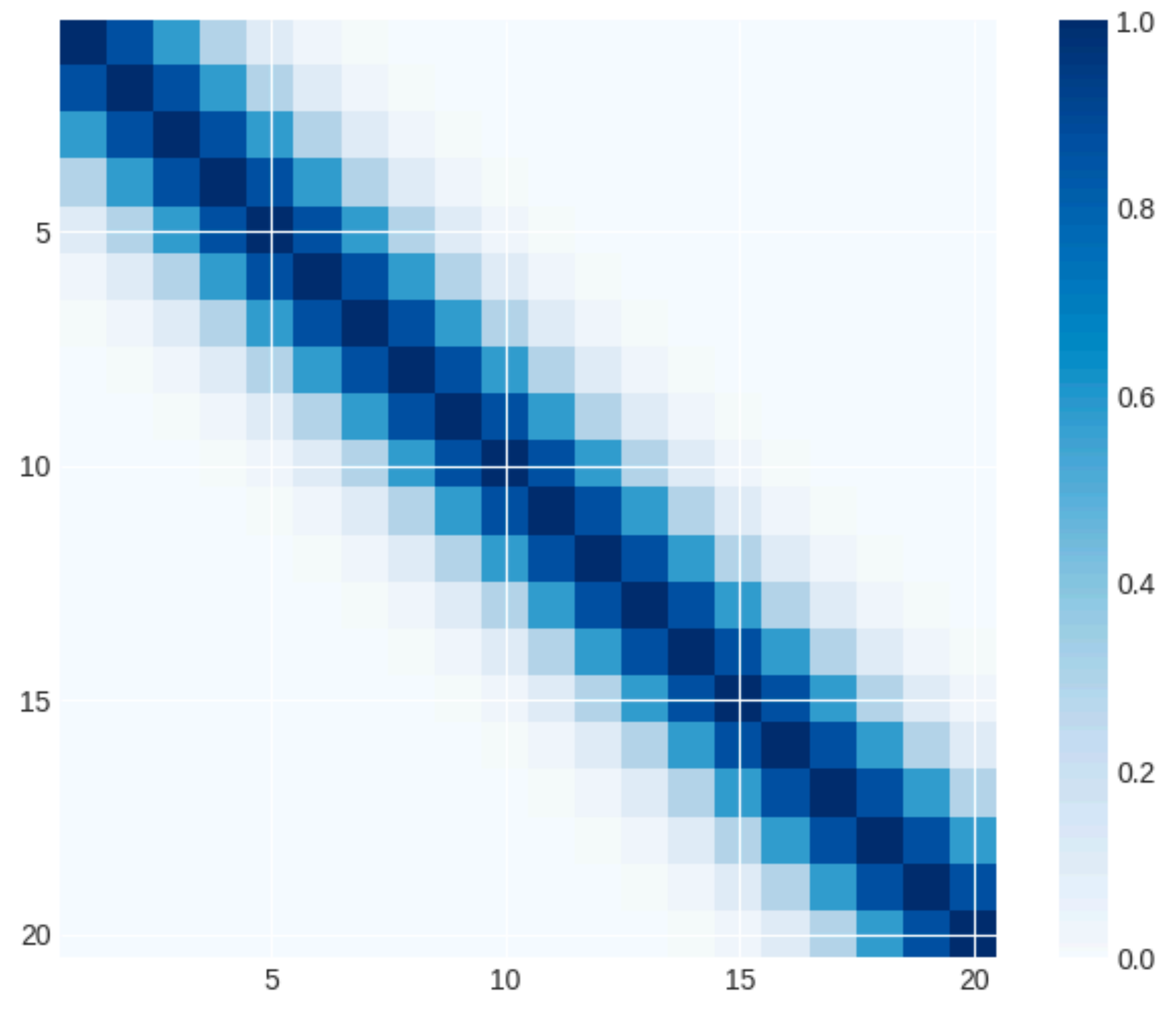
updating a multivariate Gaussian distribution

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

20 houses on a block

- nearby houses have similar prices
- far-away houses have uncorrelated prices



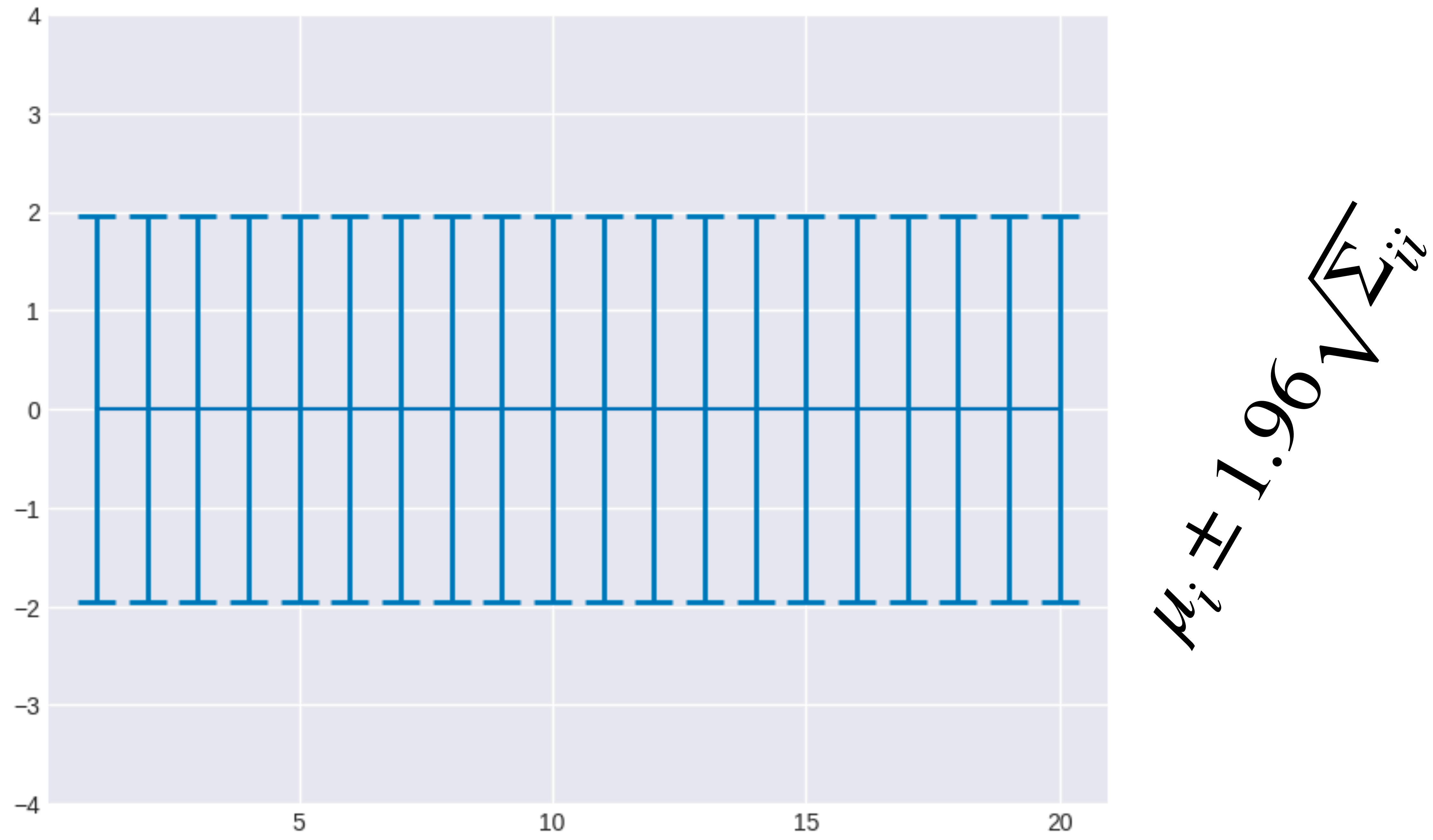
Setting many house prices
updating a multivariate Gaussian distribution

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

20 houses on a block

- nearby houses have similar prices
- far-away houses have uncorrelated prices



Setting many house prices

updating a multivariate Gaussian distribution

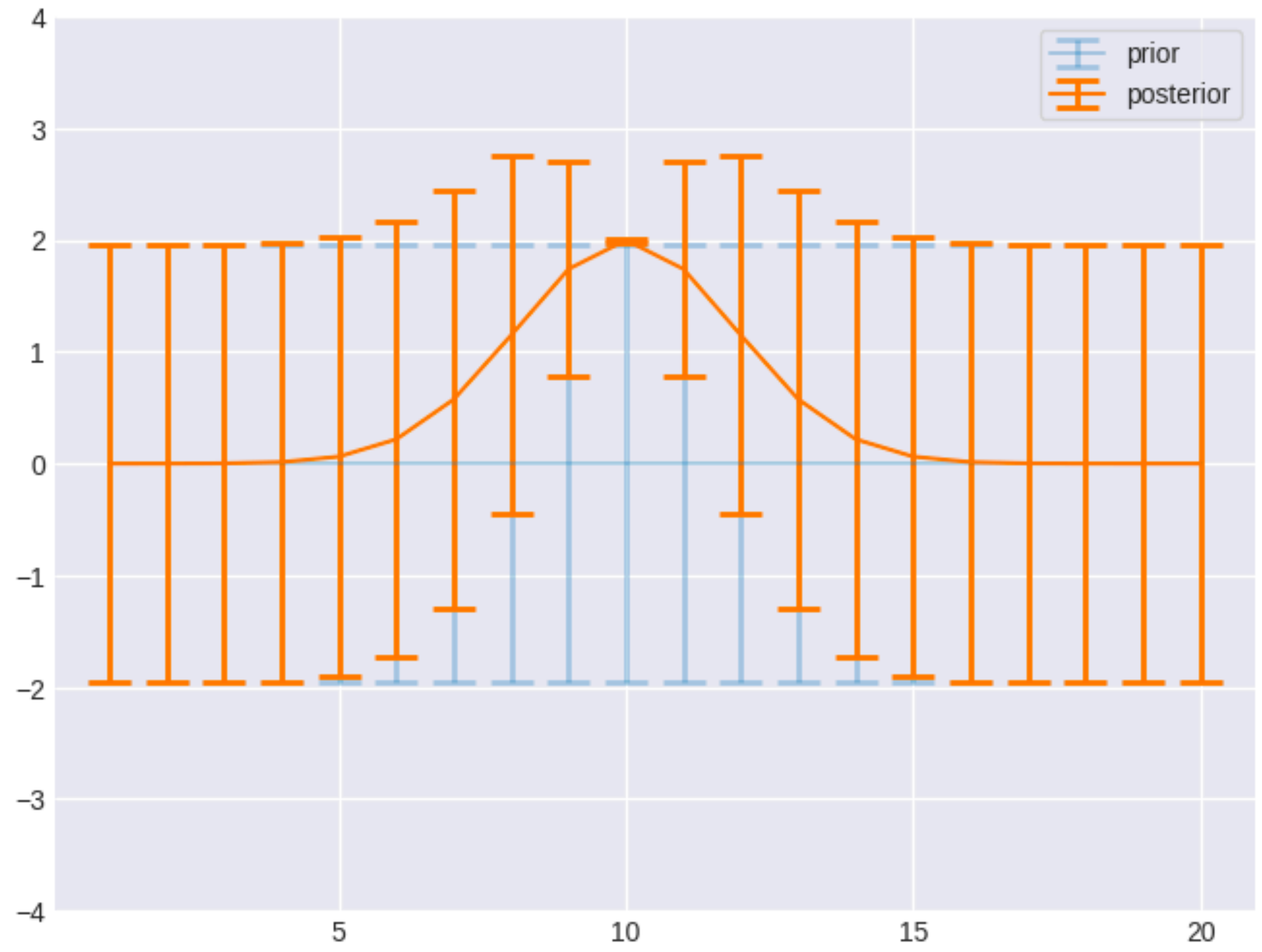
Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

20 houses on a block

- nearby houses have similar prices
- far-away houses have uncorrelated prices

observation $X_{10} = 2$



$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

Setting many house prices

updating a multivariate Gaussian distribution

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

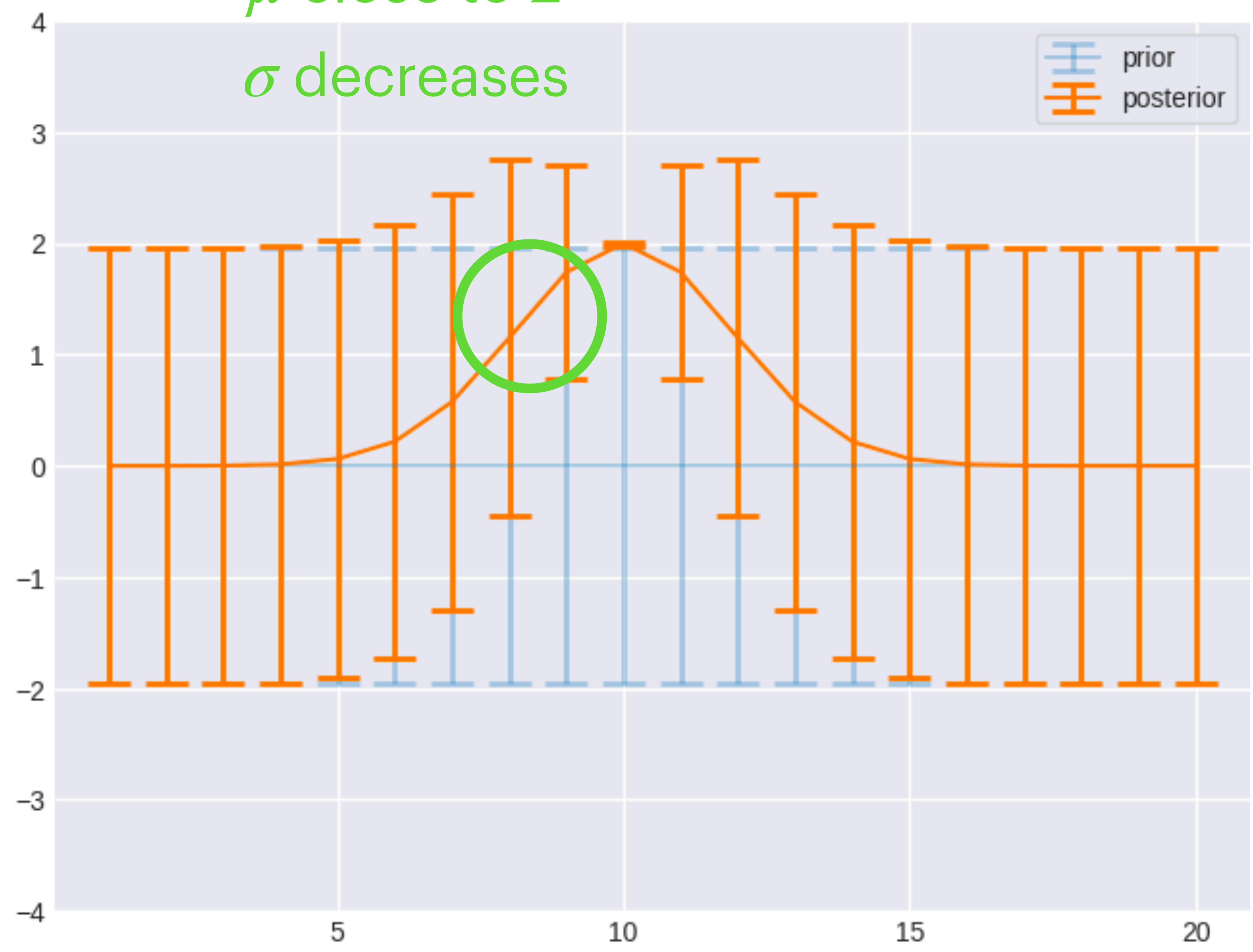
20 houses on a block

- nearby houses have similar prices
- far-away houses have uncorrelated prices

observation $X_{10} = 2$

μ close to 2

σ decreases



$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

Setting many house prices

updating a multivariate Gaussian distribution

Model unknown house prices as *random variables*

- multivariate Gaussian with appropriate covariances
- once the price of one house is observed, other prices are *updated*

20 houses on a block

- nearby houses have similar prices
- far-away houses have uncorrelated prices

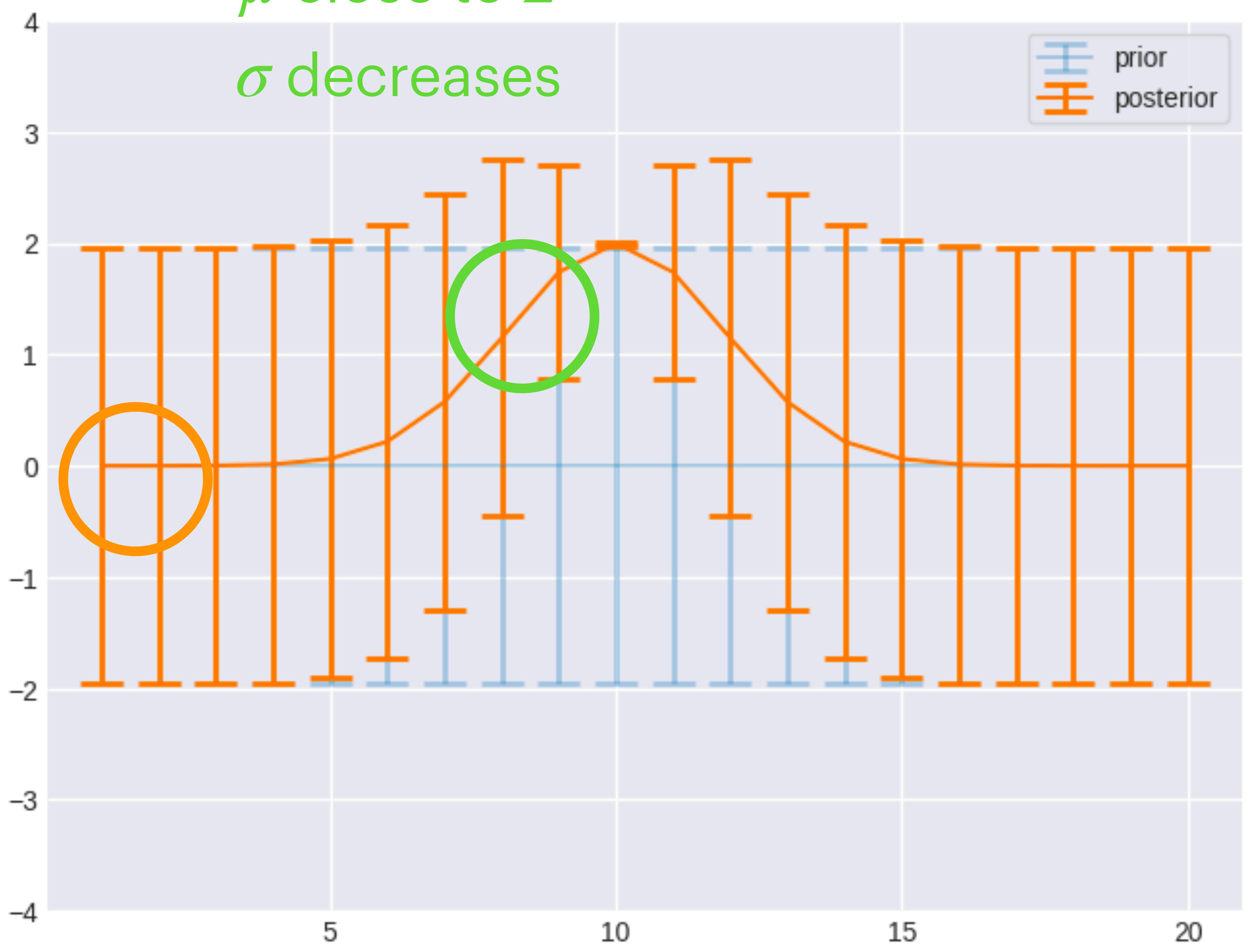
observation $X_{10} = 2$

μ close to 2

σ decreases

μ and σ

stay the same



$\mu_i \pm 1.96\sqrt{\Sigma_{ii}}$

Setting infinitely(!) many house prices

updating a Gaussian process

Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Setting infinitely(!) many house prices

updating a Gaussian process

Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

Setting infinitely(!) many house prices

updating a Gaussian process

Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

Setting infinitely(!) many house prices

updating a Gaussian process

Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

mean $\mu(x) = 0$

RBF covariance $K(x_1, x_2) = \exp\left(-(x_1 - x_2)^2/2\right)$

Setting infinitely(!) many house prices

updating a Gaussian process

Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

mean $\mu(x) = 0$

RBF covariance $K(x_1, x_2) = \exp\left(-(x_1 - x_2)^2/2\right)$

Setting infinitely(!) many house prices

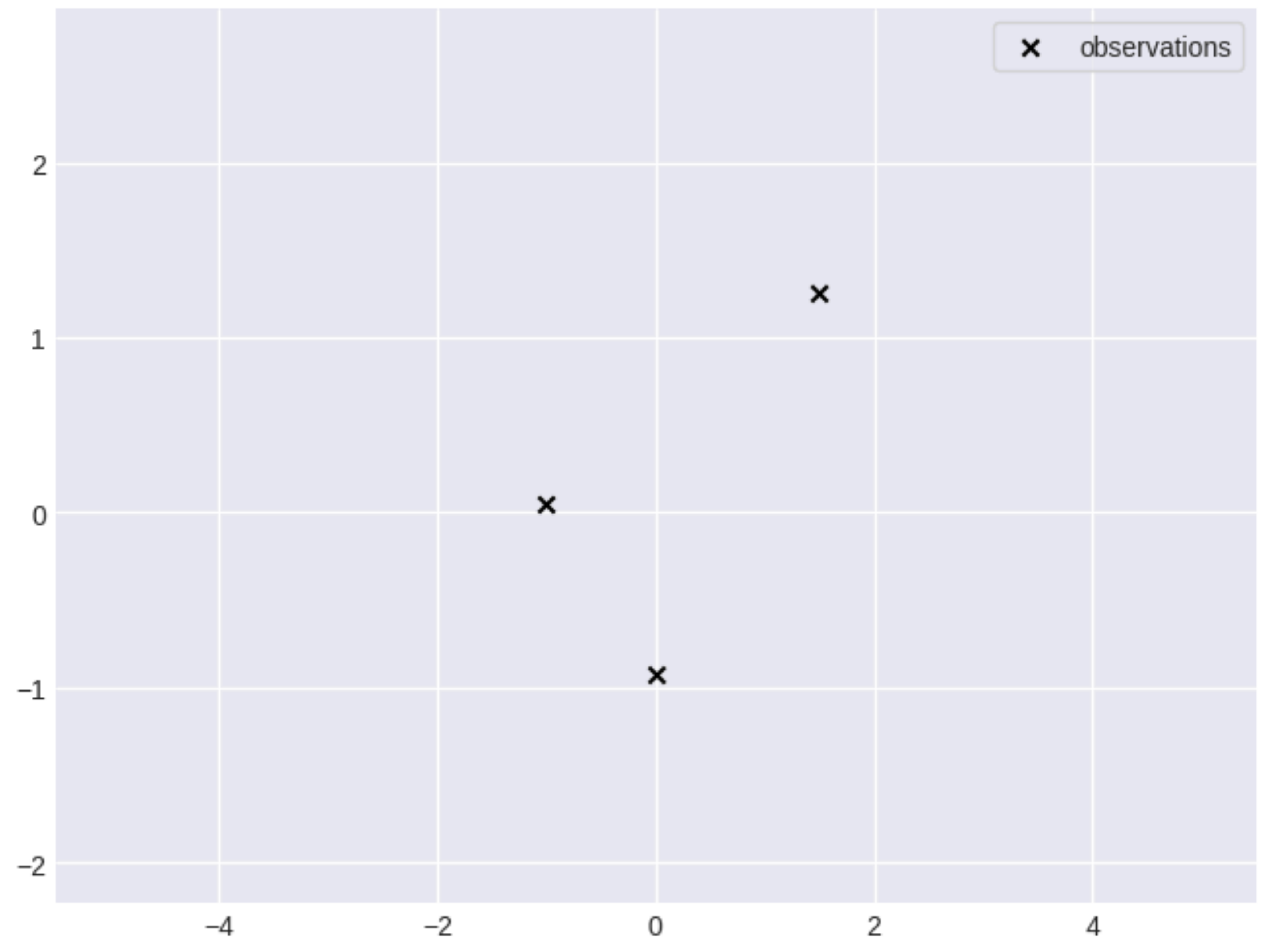
updating a Gaussian process

Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$



Setting infinitely(!) many house prices

updating a Gaussian process

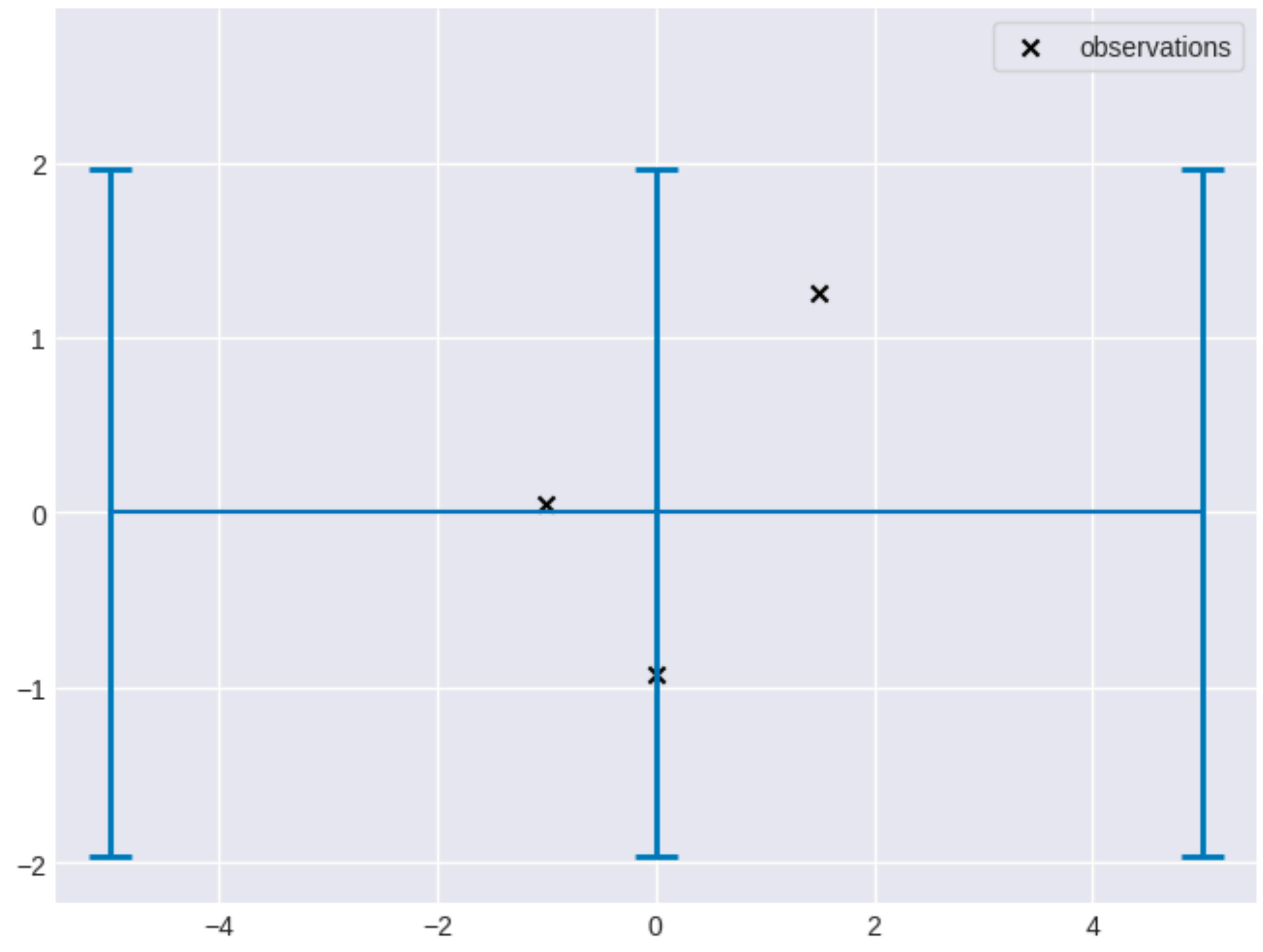
Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

target variable: $[f(-5) \quad f(0) \quad f(5)]^\top$



Setting infinitely(!) many house prices

updating a Gaussian process

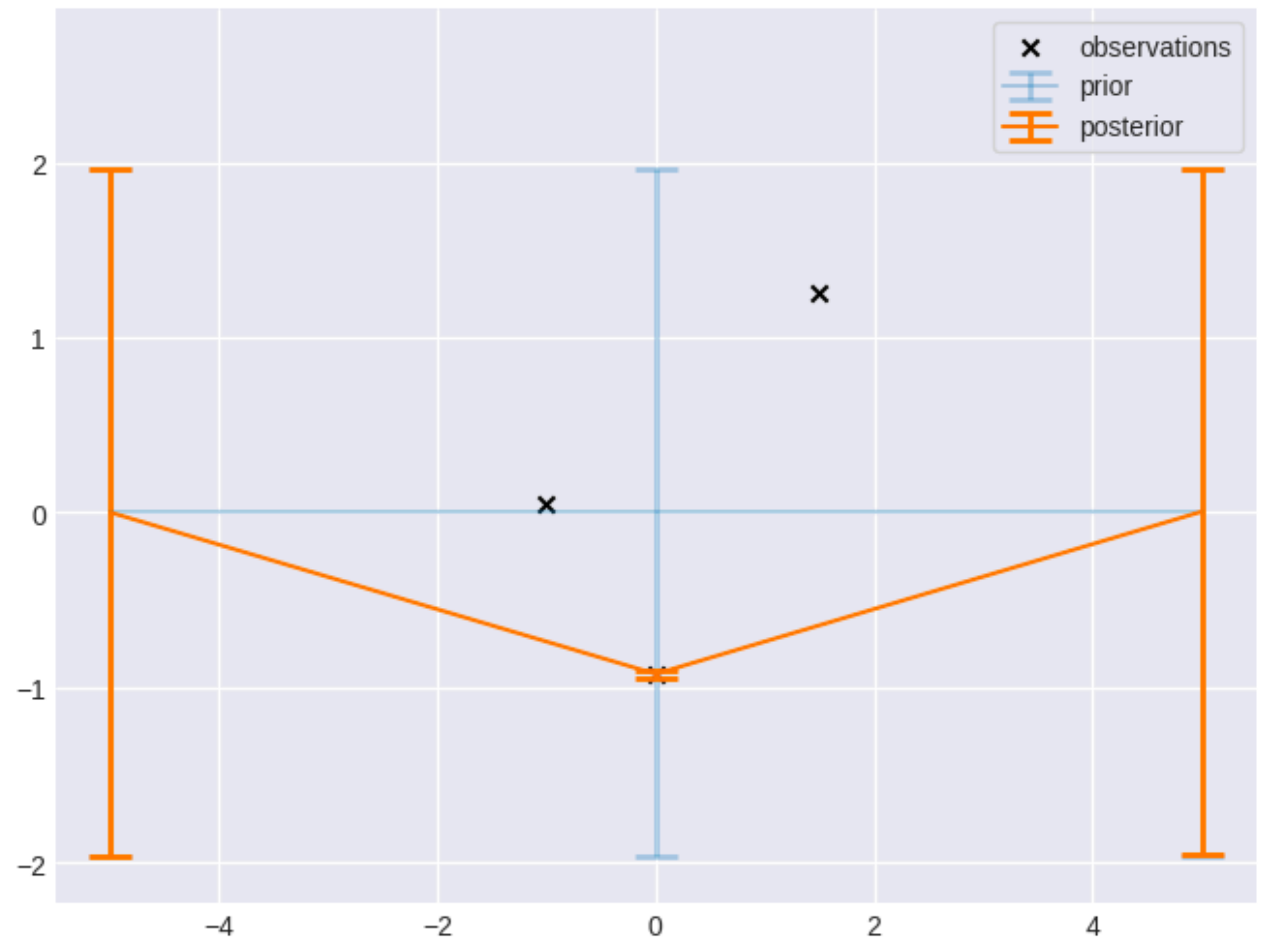
Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

target variable: $[f(-5) \quad f(0) \quad f(5)]^\top$



Setting infinitely(!) many house prices

updating a Gaussian process

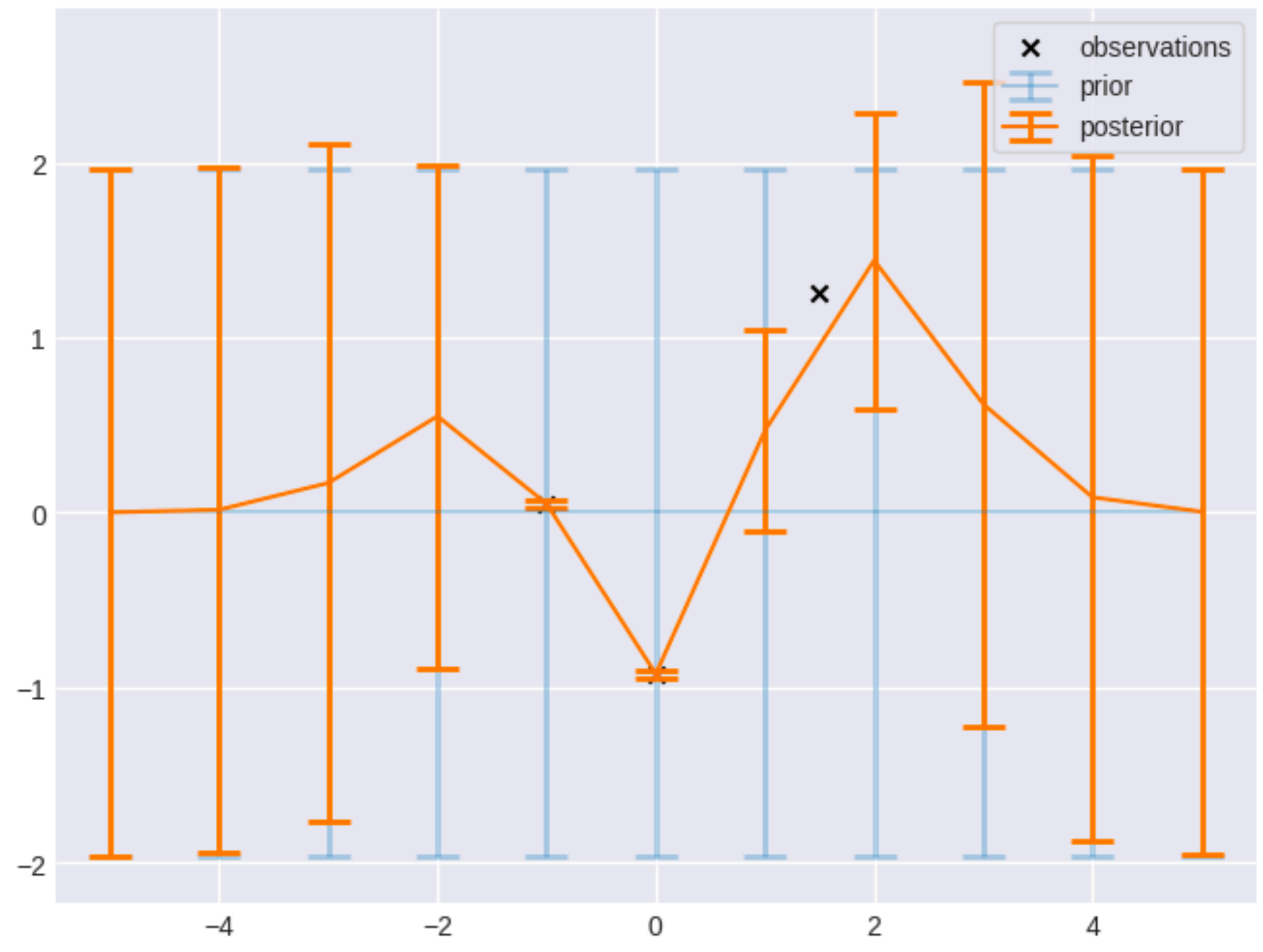
Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

target variable: $[f(-5), f(-4), \ldots, f(5)]^\top$



Setting infinitely(!) many house prices

updating a Gaussian process

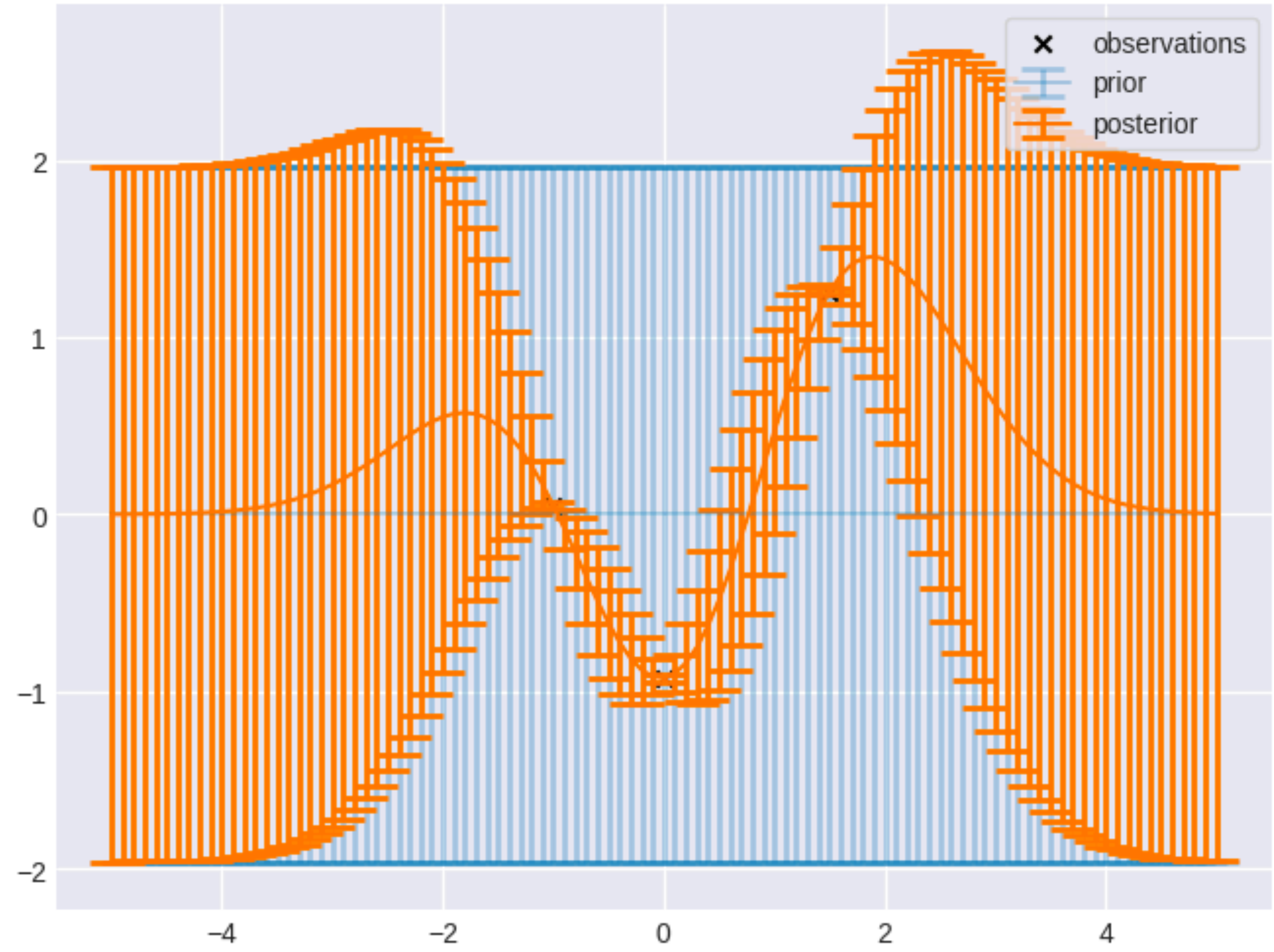
Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

target variable: $[f(-5), f(-4.9), f(-4.8), \ldots, f(5)]^\top$



Setting infinitely(!) many house prices

updating a Gaussian process

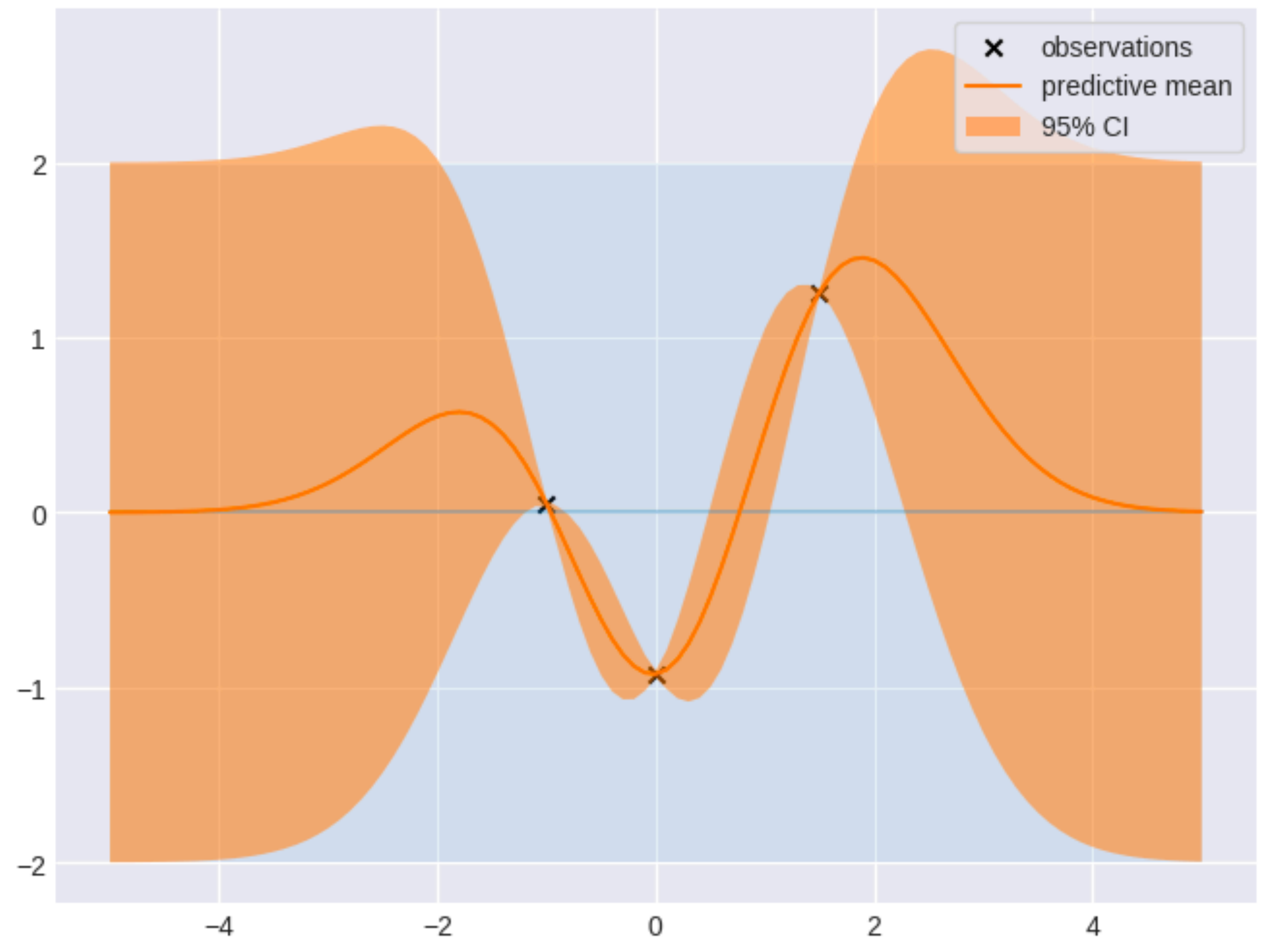
Infinitely many unknown quantities $f(x)$ for all x as random variables

- nearby x_1 and x_2: similar $f(x_1)$ and $f(x_2)$
- far-away x_1 and x_2: uncorrelated $f(x_1)$ and $f(x_2)$

Gaussian process: infinite-variate Gaussian distribution

- mean function $\mu(x)$
- covariance function $K(x_1, x_2)$

target variable: $f(x)$ for all x



Implementing Gaussian processes in Python

using state-of-the-art software

GPyTorch for Gaussian process modeling

modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

GPyTorch for Gaussian process modeling

modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

```
class GPModel(ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super().__init__(train_x, train_y, likelihood)
        self.mean_module = ConstantMean()
        self.covar_module = RBFKernel()

    def forward(self, x):
        mean_x = self.mean_module(x)
        covar_x = self.covar_module(x)
        return MultivariateNormal(mean_x, covar_x)
```

GPyTorch for Gaussian process modeling

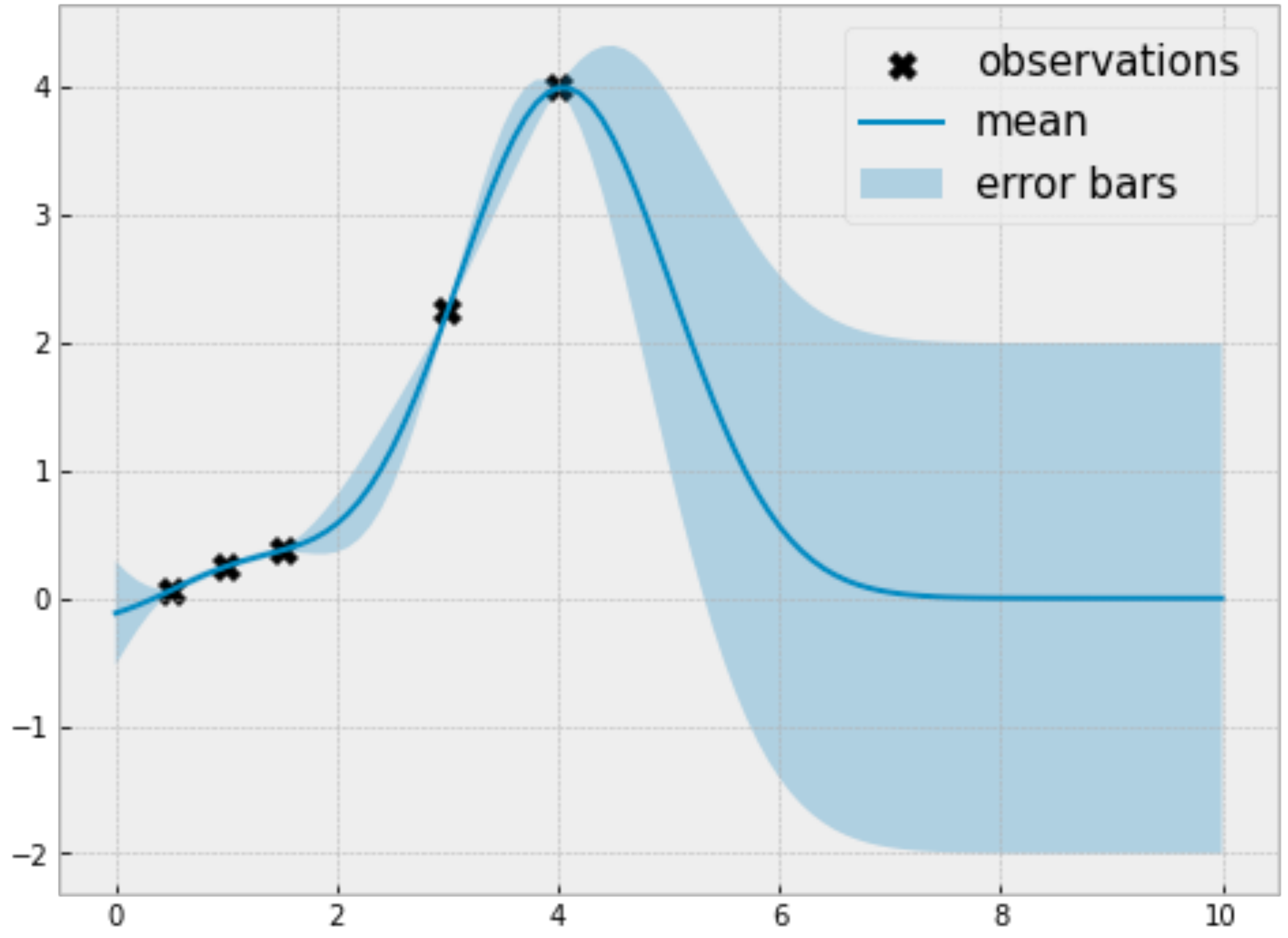
modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

```
class GPModel(ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super().__init__(train_x, train_y, likelihood)
        self.mean_module = ConstantMean()
        self.covar_module = RBFKernel()

    def forward(self, x):
        mean_x = self.mean_module(x)
        covar_x = self.covar_module(x)
        return MultivariateNormal(mean_x, covar_x)
```



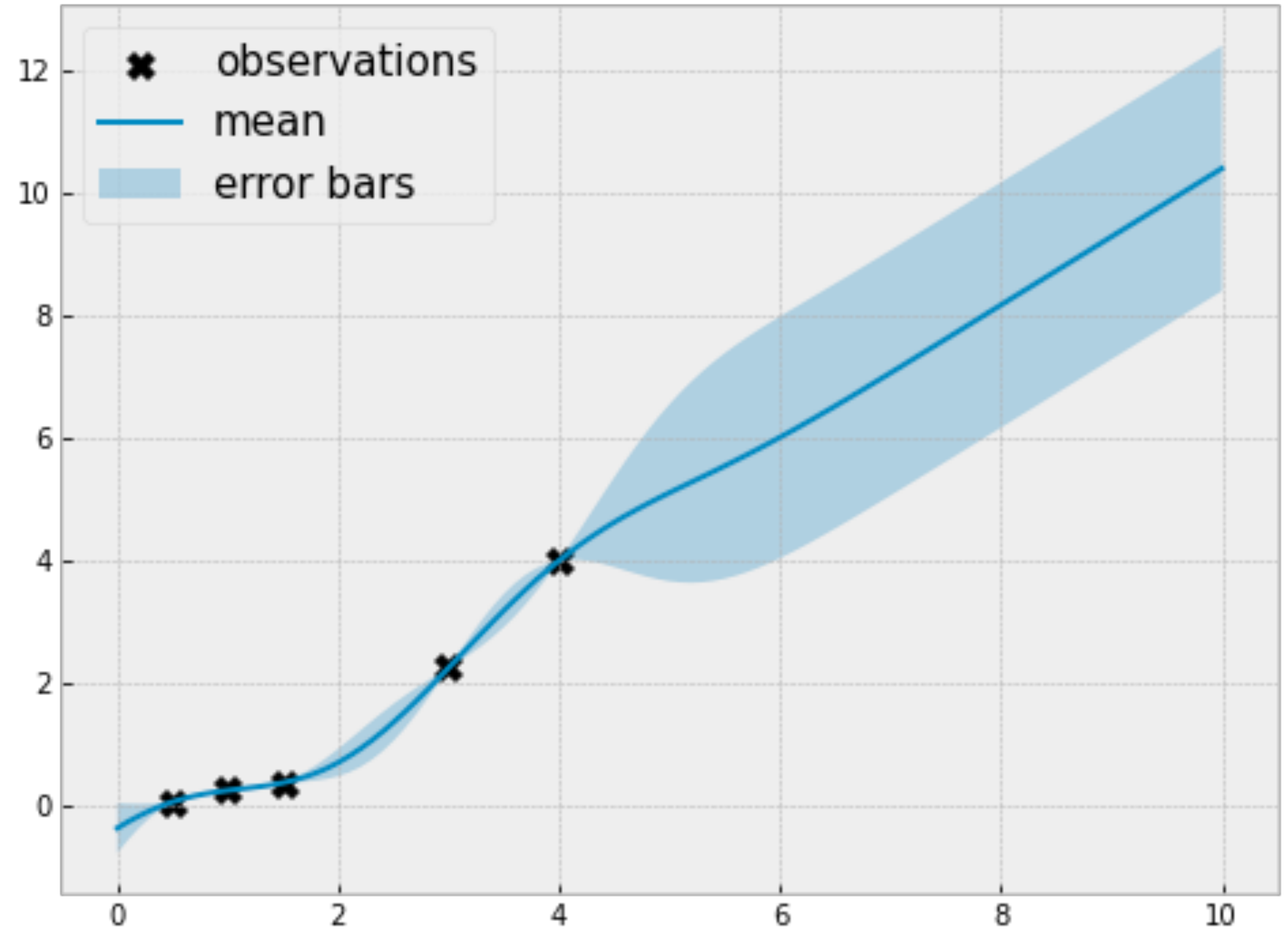
GPyTorch for Gaussian process modeling

modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

```
class GPModel(ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super().__init__(train_x, train_y, likelihood)
        self.mean_module = LinearMean(1)
        self.covar_module = RBFKernel()
```



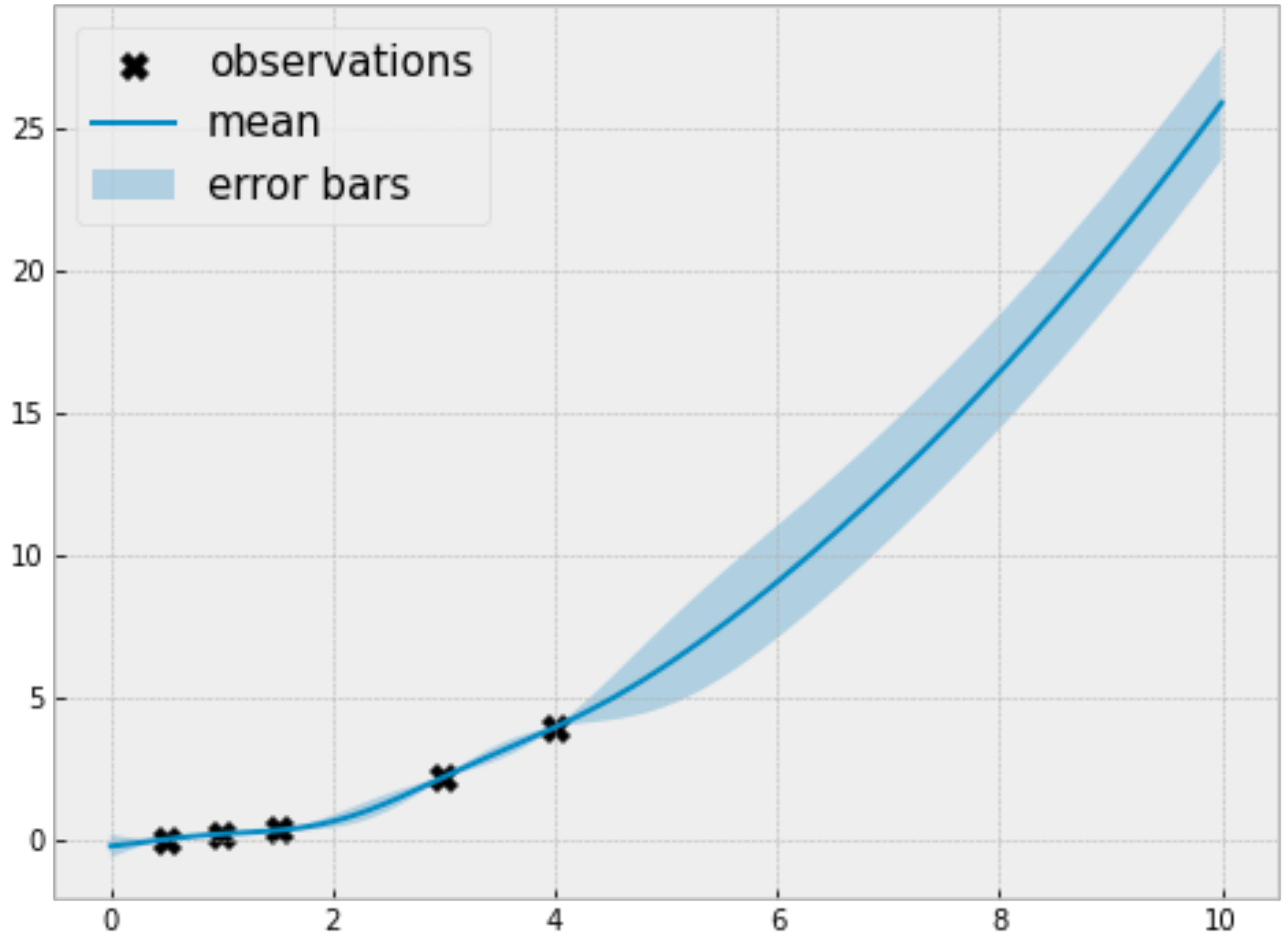
GPyTorch for Gaussian process modeling

modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

```
class QuadraticMean(Mean):
    def forward(self, x):
        res = (
            x.pow(2).matmul(self.second)
            + x.matmul(self.first)
            + self.bias
        )
        return res
```



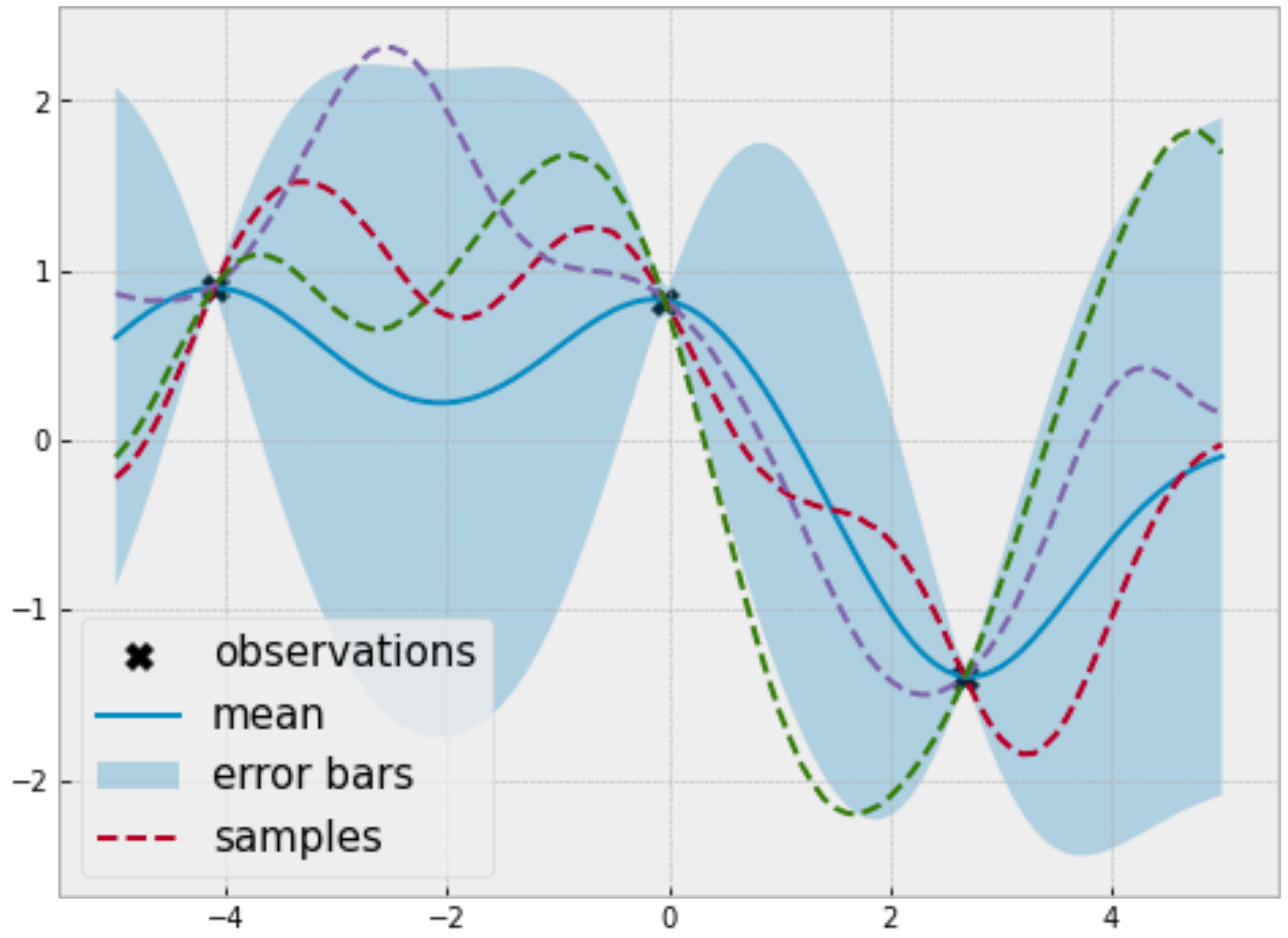
GPyTorch for Gaussian process modeling

modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

```
class GPModel(ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super().__init__(train_x, train_y, likelihood)
        self.mean_module = ConstantMean()
        self.covar_module = RBFKernel()
```



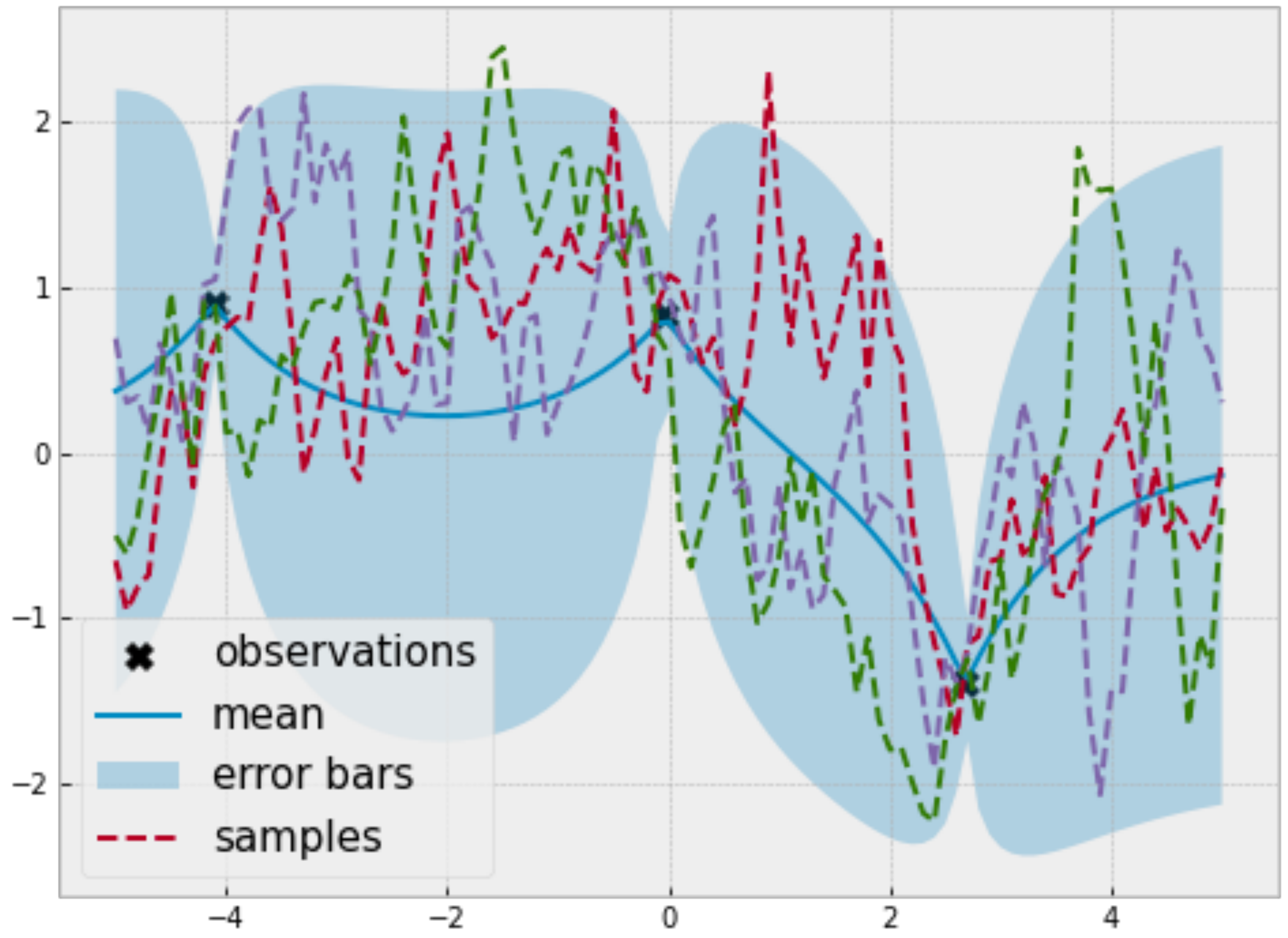
GPyTorch for Gaussian process modeling

modular implementation of Gaussian processes

Need to declare two components:

- **Mean function** models the *expected trend*
- **Kernel** models *smoothness*

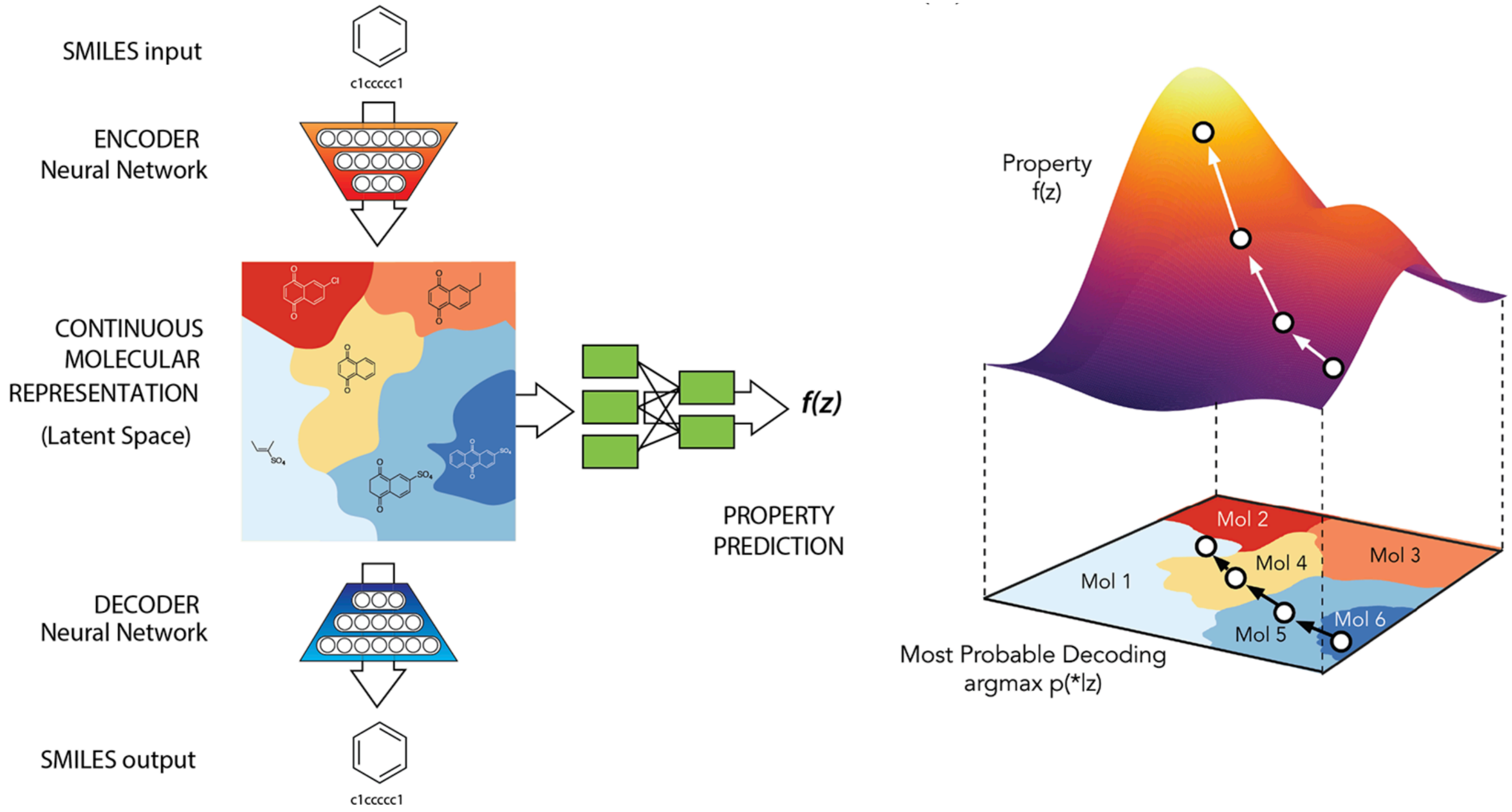
```
class GPModel(ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super().__init__(train_x, train_y, likelihood)
        self.mean_module = ConstantMean()
        self.covar_module = MaternKernel(nu=0.5)
```



Advanced Gaussian process models

taking Gaussian processes to the next level

- Combining neural networks and Gaussian processes

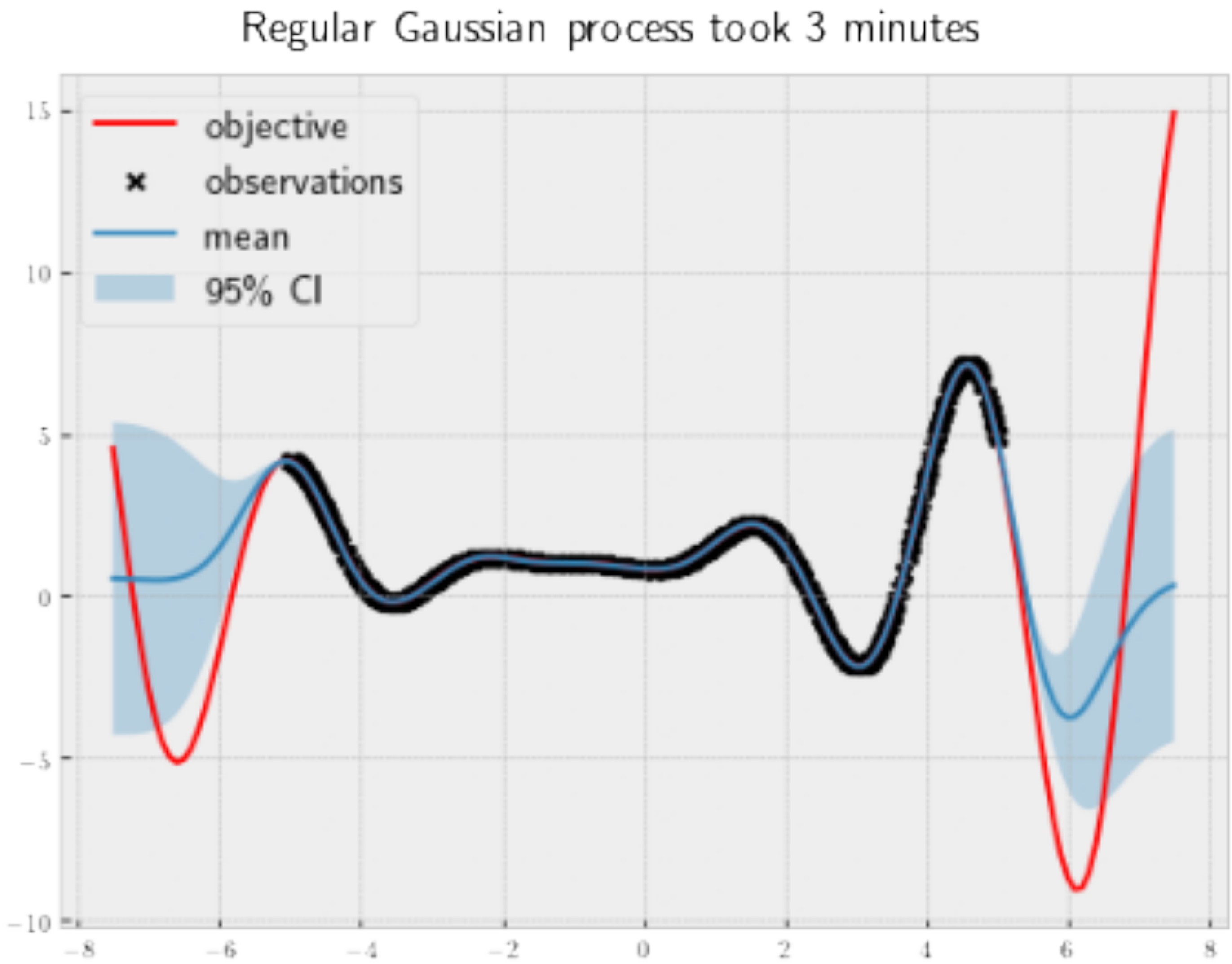


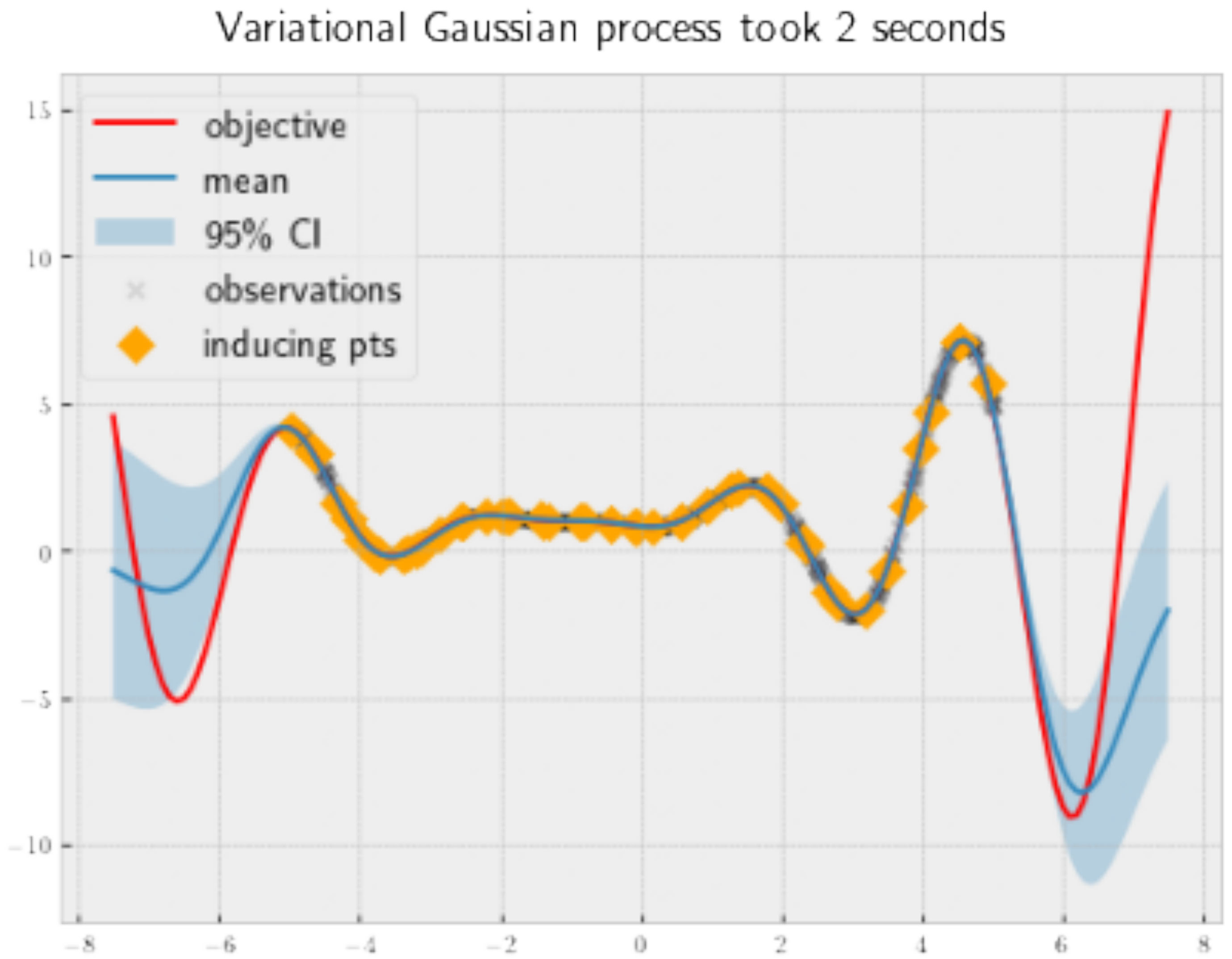
Gomez-Bombarelli *et al. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules*. ACS Central Science, 2018.

Advanced Gaussian process models

taking Gaussian processes to the next level

- Combining neural networks and Gaussian processes
- Scaling Gaussian processes to large data sets using representative data

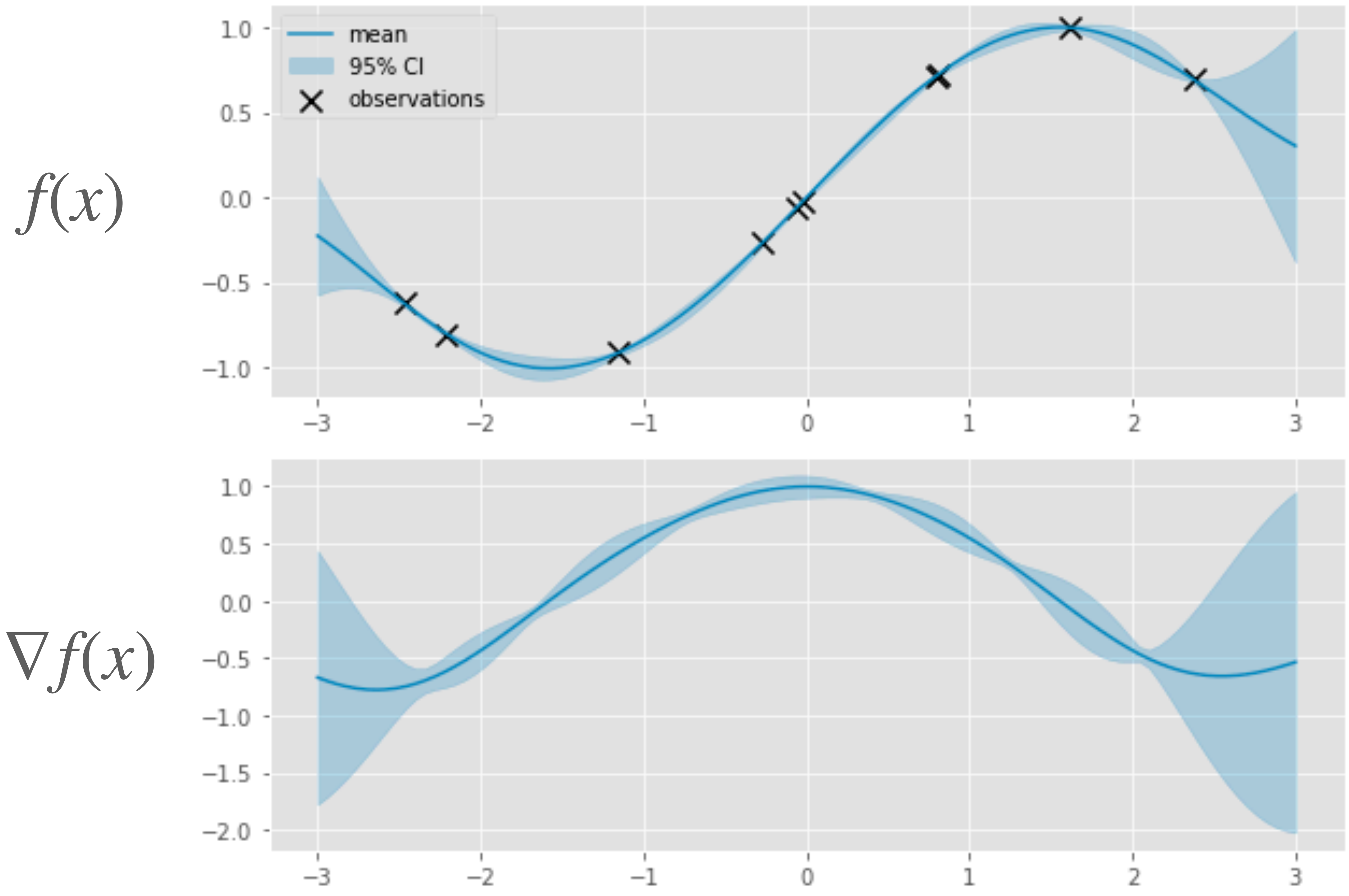




Advanced Gaussian process models

taking Gaussian processes to the next level

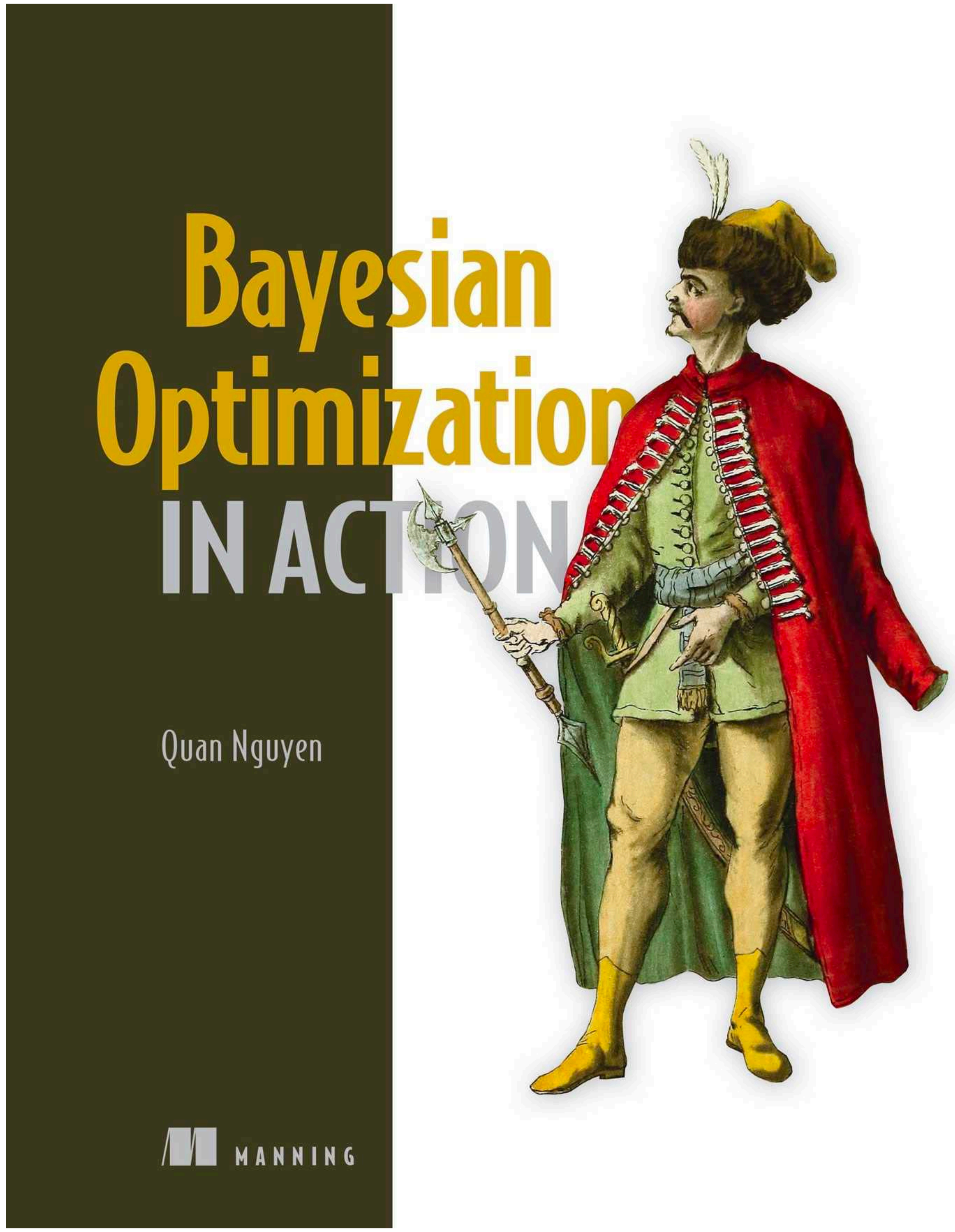
- Combining neural networks and Gaussian processes
- Scaling Gaussian processes to large data sets using representative data
- Learn from and/or predicting derivatives – derivative follows a GP



Bayesian optimization

blackbox optimization with Gaussian processes

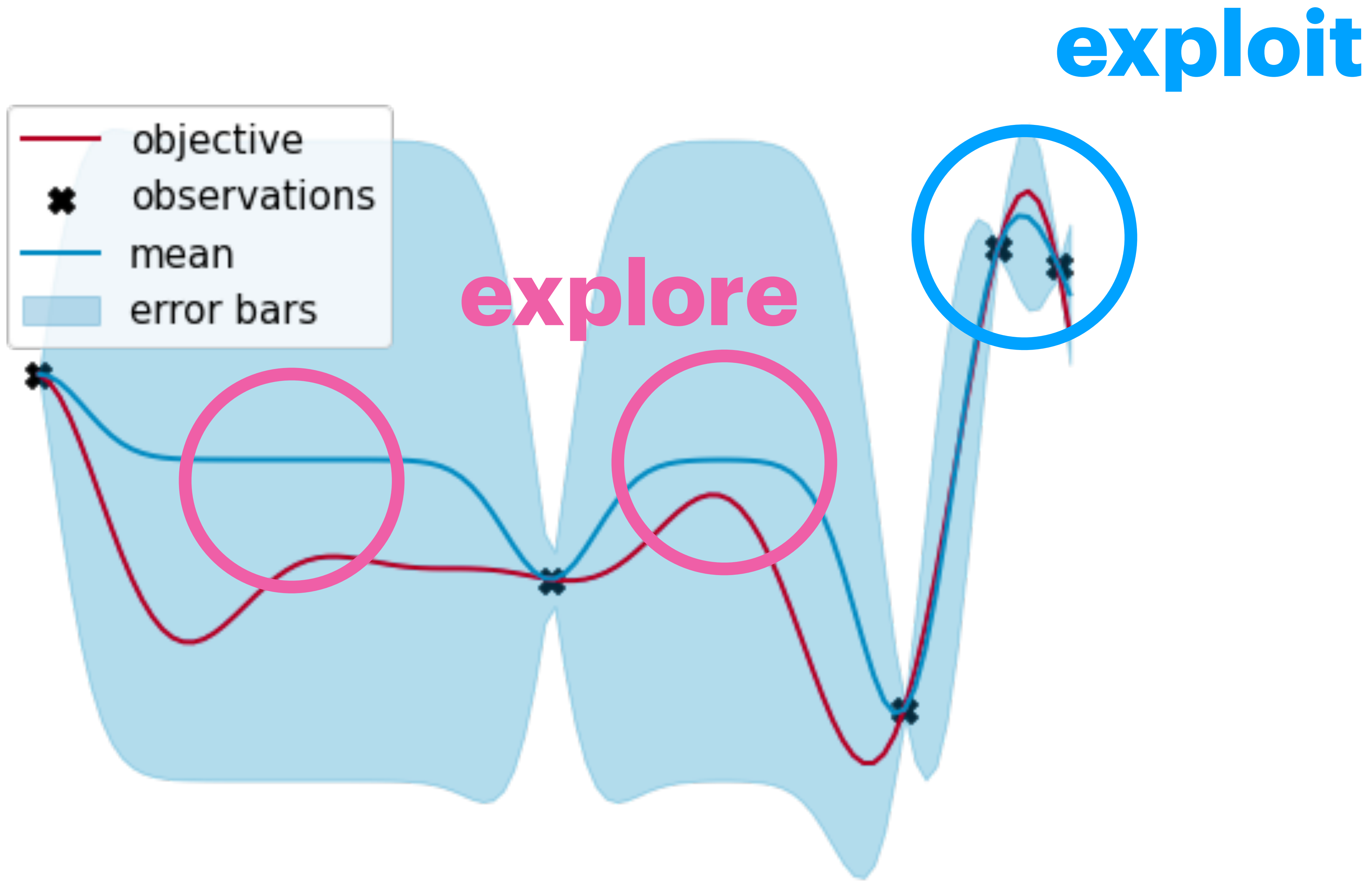
- Last year's PyData Global talk
- https://www.manning.com/books/bayesian-optimization-in-action
    - http://mng.bz/M9Qo
- Discount code: **ctwpydata45**



Bayesian optimization

blackbox optimization with Gaussian processes

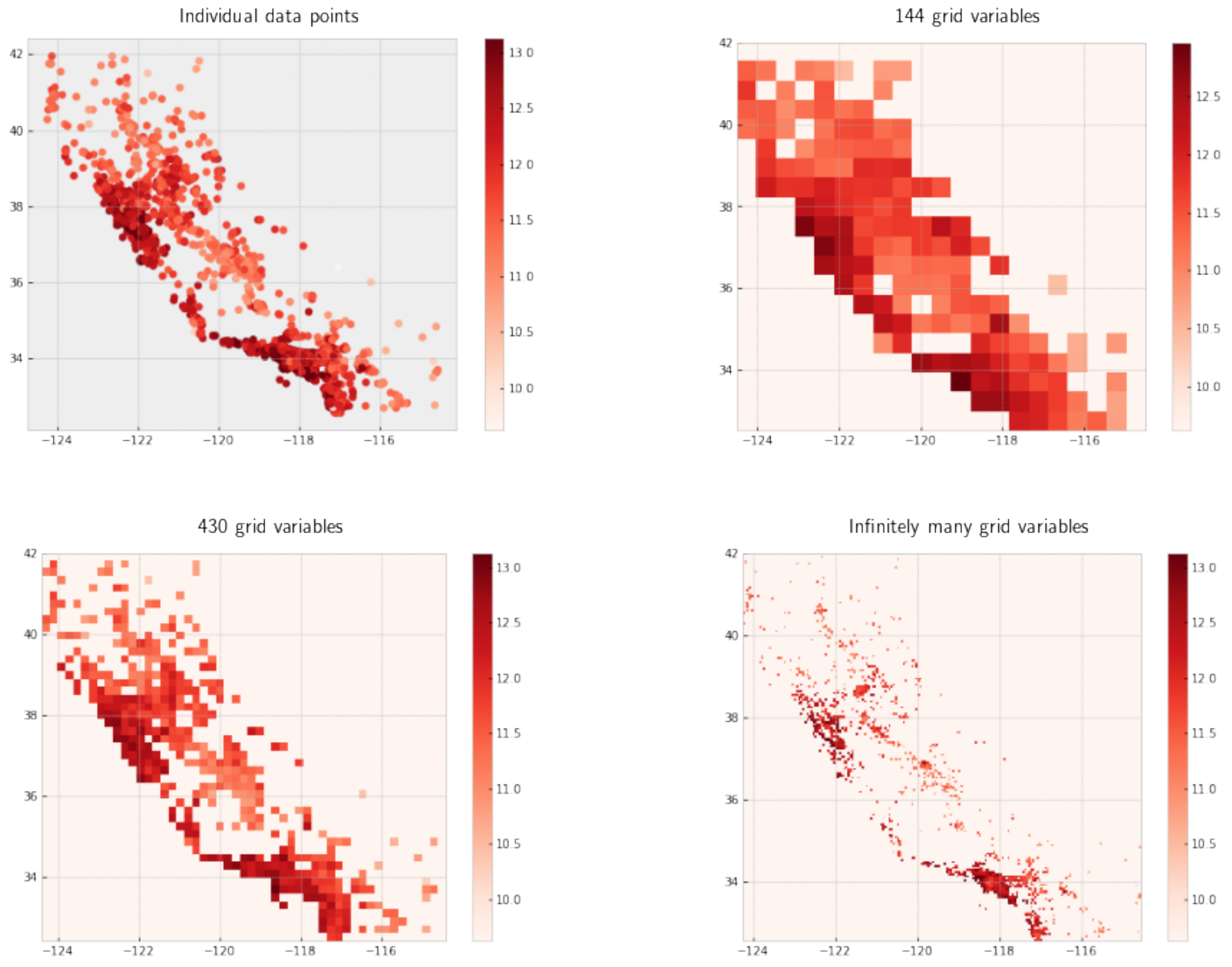
- Last year's [PyData Global talk]
- https://www.manning.com/books/bayesian-optimization-in-action
  - http://mng.bz/M9Qo
- Discount code: **ctwpydata45**



Bayesian optimization

blackbox optimization with Gaussian processes

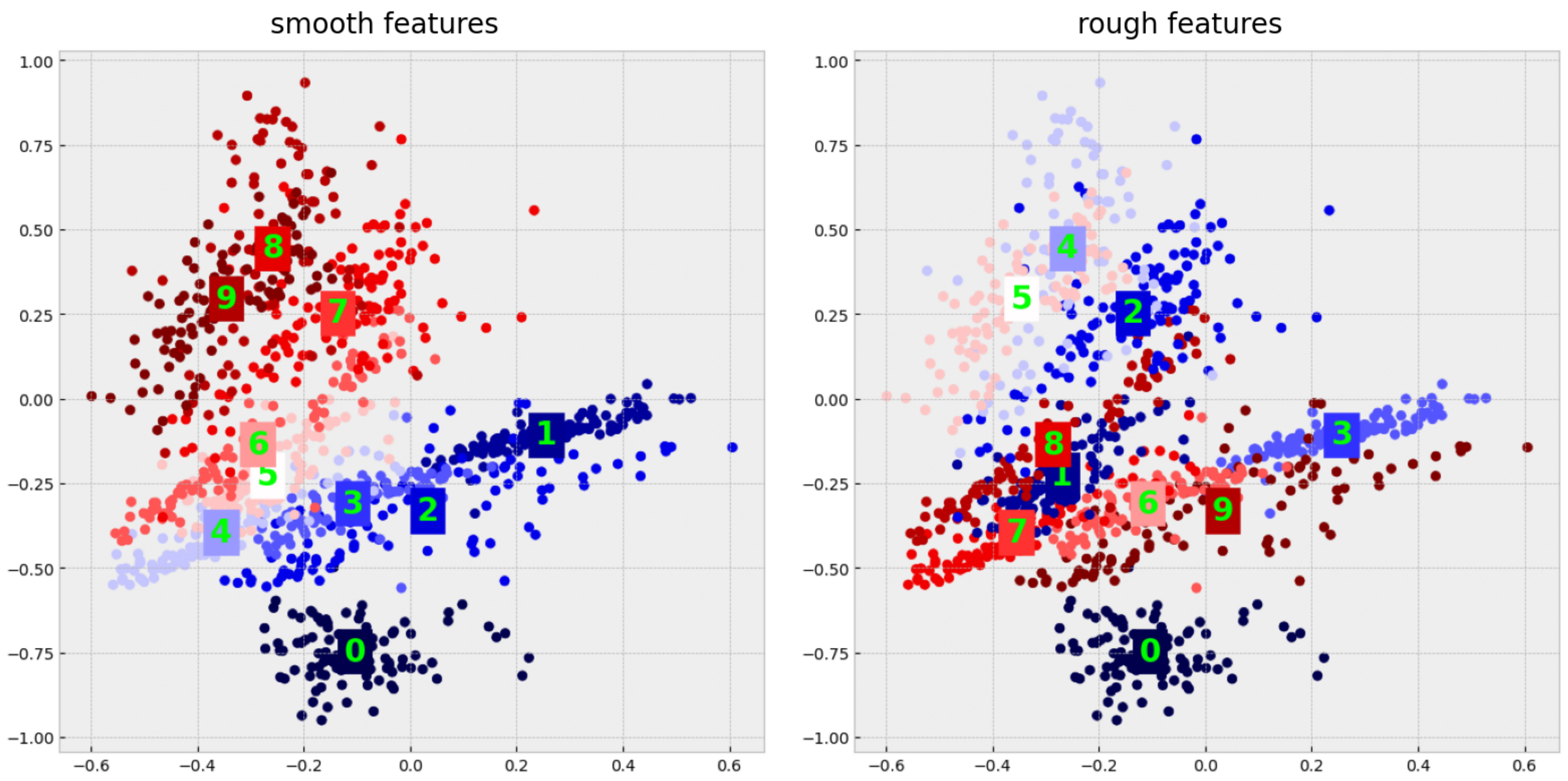
- Last year's [PyData Global talk]()
- https://www.manning.com/books/bayesian-optimization-in-action
    - http://mng.bz/M9Qo
- Discount code: **ctwpydata45**



Bayesian optimization

blackbox optimization with Gaussian processes

- Last year's [PyData Global talk]
- [https://www.manning.com/books/bayesian-optimization-in-action]
  - [http://mng.bz/M9Qo]
- Discount code: **ctwpydata45**



Bayesian optimization

blackbox optimization with Gaussian processes

- Last year's [PyData Global talk]
- [https://www.manning.com/books/bayesian-optimization-in-action]
  - [http://mng.bz/M9Qo]
- Discount code: **ctwpydata45**