Experimental optimization

Lecture 12: Bayesian optimization: Gaussian process regression

David Sweet

Review

Surrogate function in Bayesian optimization

- A *surrogate function* models the response surface, BM(parameters)
- BO uses Gaussian process regression (GPR) to create the model
- BO optimizes over surrogate:
    - Finds parameter value that maximizes the GPR estimate of BM

Gaussian process regression

Overview

- Estimates both **function value** (BM) and **model uncertainty**
    - uncertainty in its own estimate of BM
    - NOT uncertainty in your measurement, that's S.E.
- Forms estimates at parameter values where you haven't taken measurements
- Forms estimates directly from measurements; no fitting
    - So we call GPR *nonparametric*

GPR function estimates

Baseline

- BM aggregate measurements so far: y_i
- Call parameter values x_i, so $y_i(x_i)$ is BM measured at parameter x_i
- Goal: Estimate y for **any** x, call it $\hat{y}(x)$
- Start somewhere: Estimate everywhere by the average

$$\hat{y}(x) = \bar{y} = \big(y_1(x_1) + y_2(x_2) + y_3(x_3)\big)/3$$

- Not very precise, but unbiased. Good place to start for any model.

GPR function estimates

Standardize

- Subtract out mean and focus on deviations from it

$$y_i \to y_i - \bar{y}$$

- While we're at it, divide by stddev:

$$y_i \to \frac{y_i - \bar{y}}{\sigma_y}$$

- Transformed y_i have no units ($, clicks, minutes, etc.) and new y_i has mean 0 and stddev 1.

GPR function estimates

Standardize

- Now $\hat{y}(x) = 0$ for all x.
- N.B.: Recover original units by:

$$\sigma_y \hat{y}(x) + \bar{y}$$

GPR function estimates

Weighted average

- Model variation with x
- From average (which is now just 0)

$$\hat{y}(x) = \frac{\Sigma y_i}{n}$$

- to weighted average

$$\hat{y}(x) = \frac{\Sigma w_i y_i}{\Sigma w_i}$$

GPR function estimates

Weighted average

- Key: weights depend on distance from x to x_i
  - x - parameter at which we're forming an estimate
  - x_i - parameter where we've already measured
- So:

$$\hat{y}(x) = \frac{\Sigma w(x, x_i) y_i}{\Sigma w(x, x_i)}$$

- N.B.: y_i are fixed numbers, the measurements of BM

GPR function estimates

Weighted average: functional form of w

- Criteria:
    - $w(x, x_i)$ should be larger when x nearer to x_i b/c we assume the response surface, $y(x)$, varies reasonably smoothly
    - $w(x, x_i)$ should approach 0 when farther away from x_i
        - (when far from **all** x_i, $\hat{y}(x)$ will approach zero, the baseline value)
- $w(x, x_i)$ called a *kernel function*

GPR function estimates

Weighted average: functional form of w

- We'll use squared exponential: $w(x, x_i) = e^{-(x-x_i)^2/(2s^2)}$
  - nice and smooth, so we get smooth interpolations between measurements
  - a *universal kernel*, i.e., can be used in GPR to model any smooth function
  - N.B.: not calling it "gaussian kernel" in this context b/c confusing (i.e., it's not this kernel that puts the "G" in GPR)
- s is a hyperparameter; determines scale of smoothness
- tune s by LOOCV or other out-of-sample method

GPR function estimates

Weighted average: functional form of w

$$\hat{y}(x) = \Sigma w(x, x_i) y_i(x_i) \propto \Sigma e^{-(x-x_i)^2/(2s^2)} y_i(x_i)$$

- or, with $K = [w(x, x_1), w(x, x_2), \ldots]^T$ and $y = [y_1, y_2, \ldots]^T$

$$\hat{y}(x) \propto K^T y$$

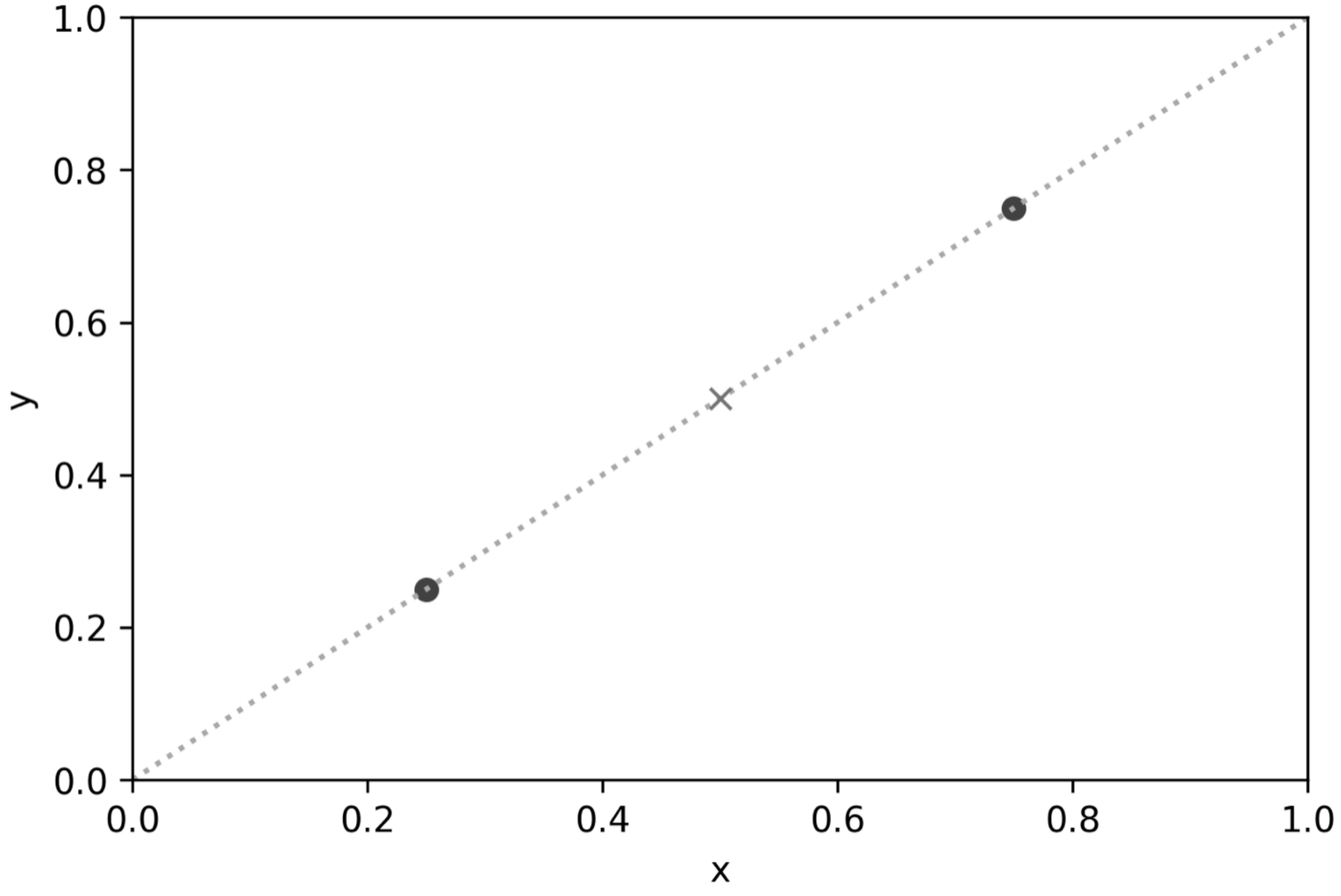
- a compact, matrix-vector form

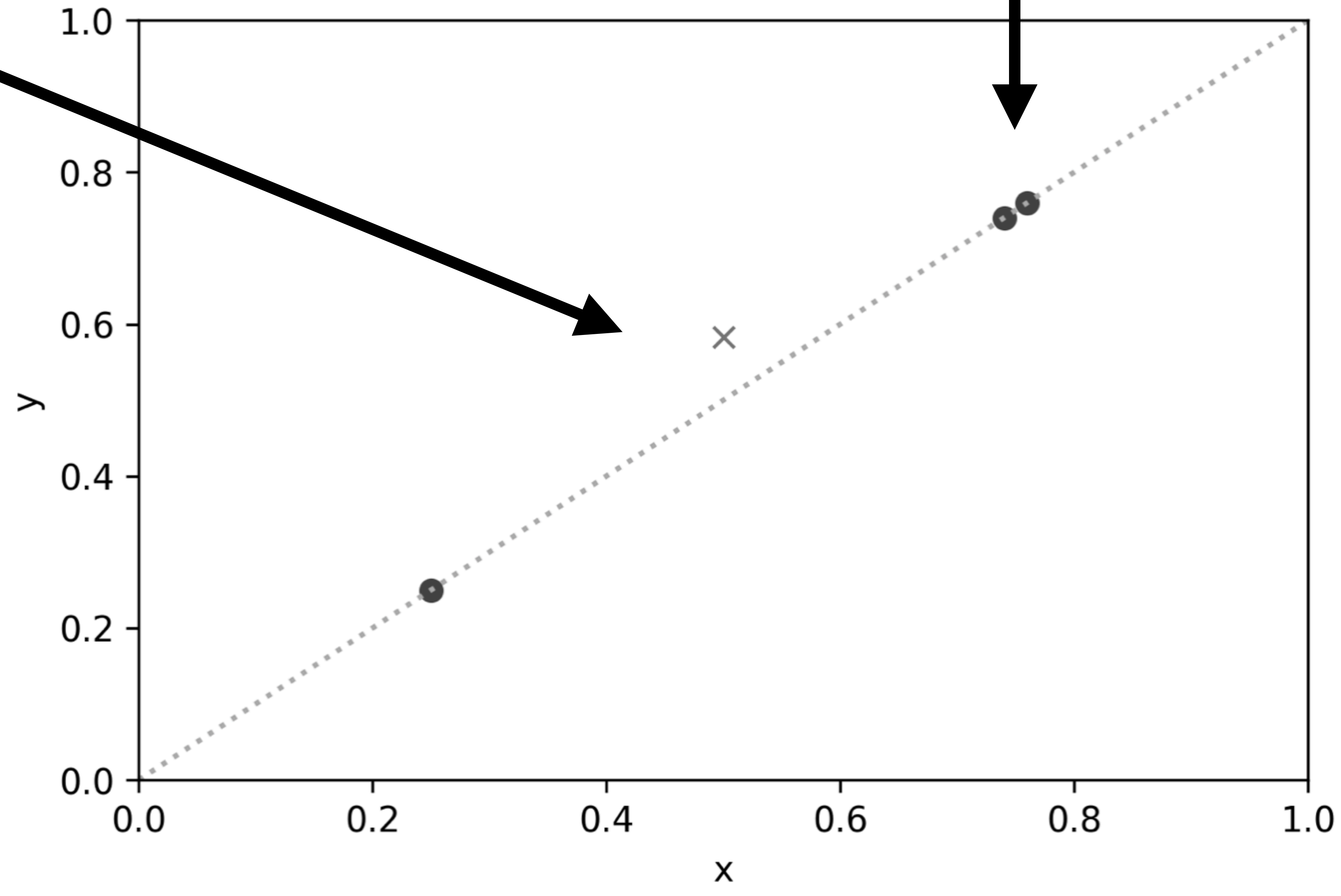
GPR function estimates

“Clustering”

Double weight
b/c two measurements

Too high





GPR function estimates

“Clustering”

- Soln: reduce weight based on nearness to other measurements
    - Use squared exponential still, but in denominator

$$K_{xx} = \begin{bmatrix} e^{-(x_1-x_1)^2/(2s^2)} & e^{-(x_1-x_2)^2/(2s^2)} & e^{-(x_1-x_3)^2/(2s^2)} & \cdots \\ e^{-(x_2-x_1)^2/(2s^2)} & e^{-(x_2-x_2)^2/(2s^2)} & e^{-(x_2-x_3)^2/(2s^2)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

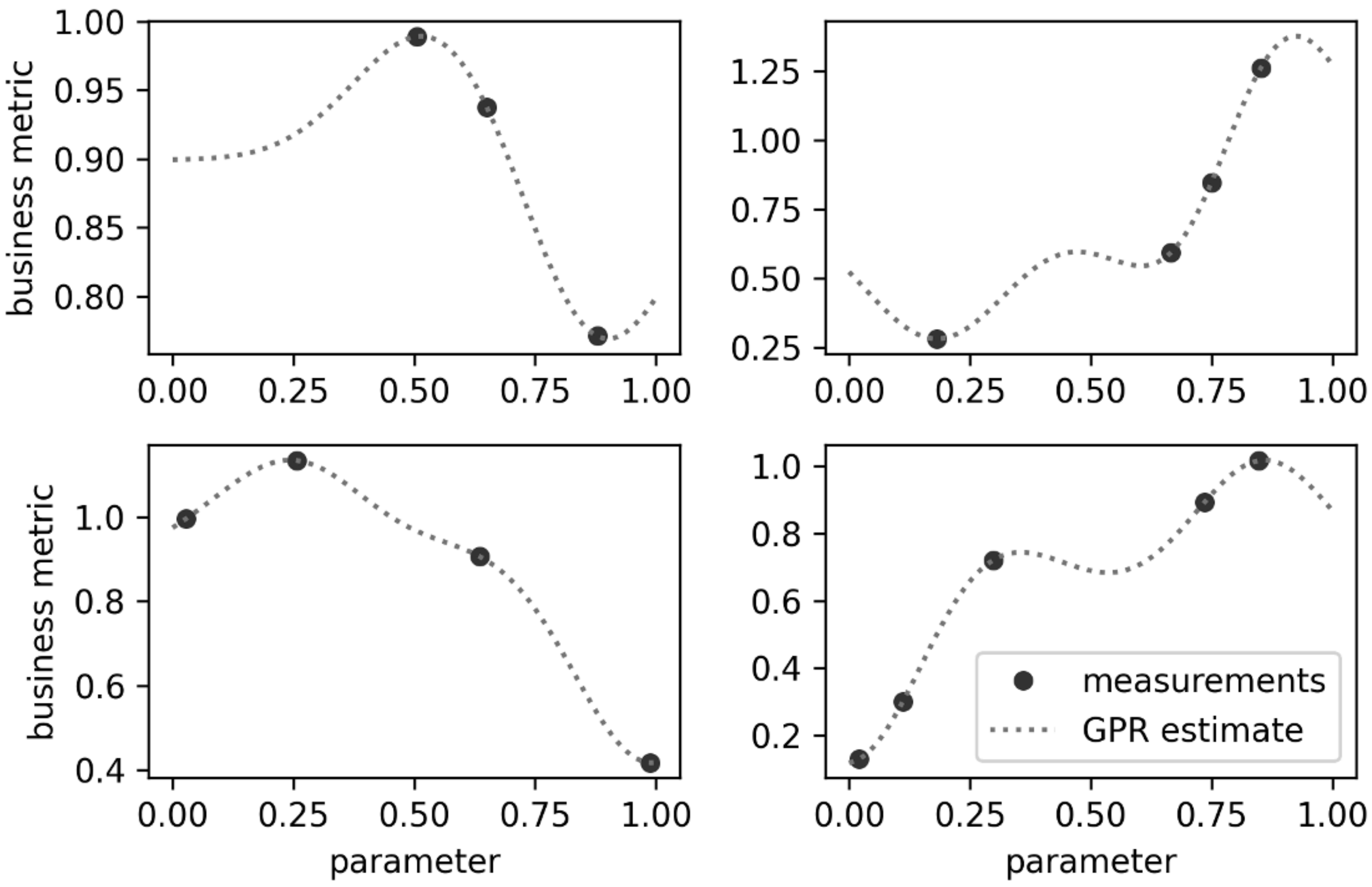
- Each element is $(K_{xx})_{i,j} = w(x_i, x_j)$ <== both x’s are measurements

GPR function estimates

Estimate y

$$\hat{y}(x) = K_x^T K_{xx}^{-1} y$$

- Estimate ($\hat{y}$) is a weighted average of the measurements (y)

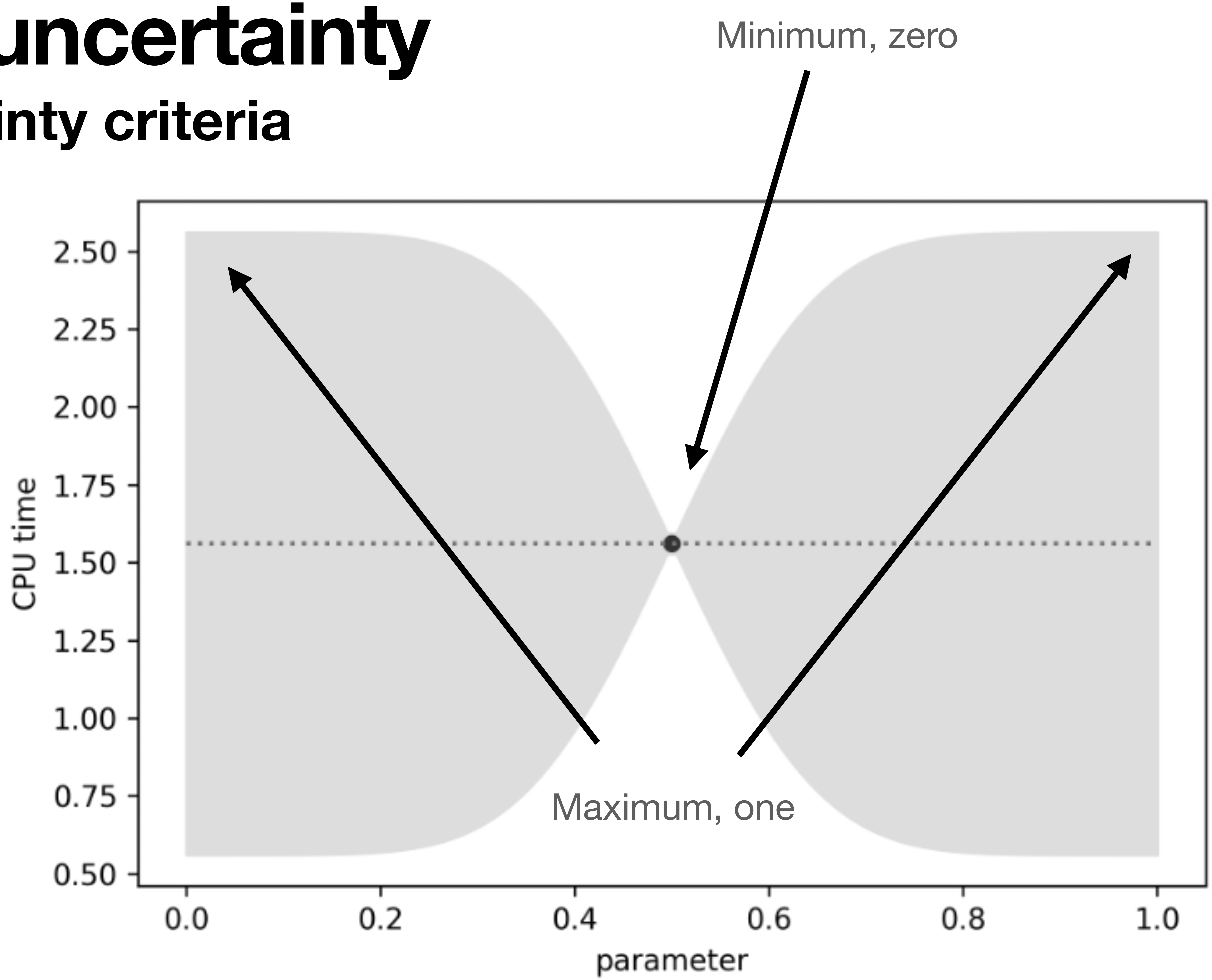


GPR uncertainty

Two types of uncertainty

- GPR reports uncertainty in its estimate, *model uncertainty*
- Contrast with SE of an aggregate measurement, *measurement uncertainty*
- Criteria:
  - Uncertainty is zero (minimum) at x_i, where you have a measured value
  - Uncertainty is one (maximum) far from any measurements

GPR uncertainty
Uncertainty criteria



GPR uncertainty

Model uncertainty

- Think “certainty” for a moment, “certainty = 1-uncertainty”:
  - 1 at measurement, x_i
  - 0 far from measurements
- What has this form? Squared exponential, K_x
- Interpolate between certainties the same as we interpolated between measurements: $K_x^T K_{xx}^{-1} K_x$
- Switch back to uncertainty: $1 - K_x^T K_{xx}^{-1} K_x$

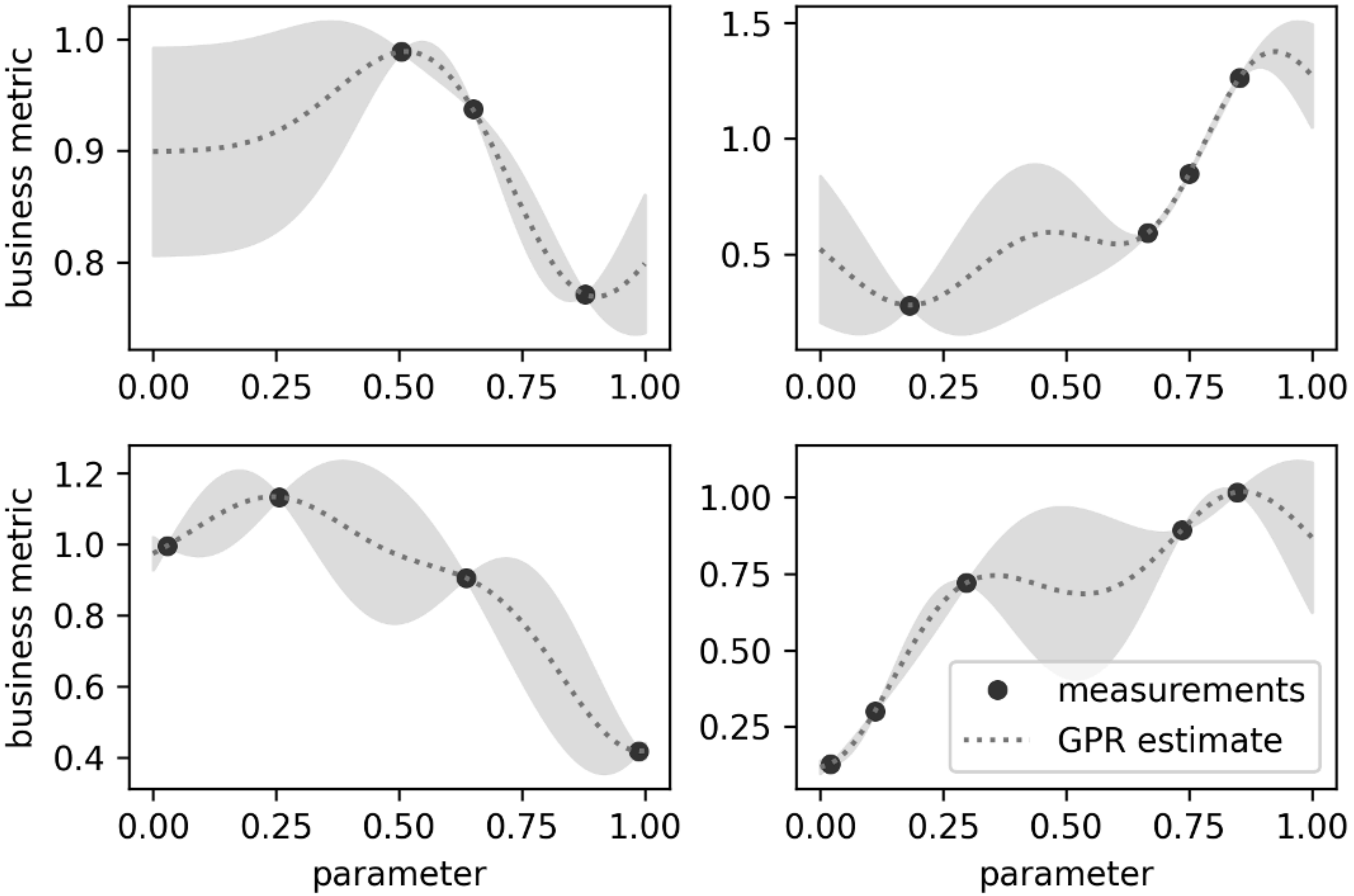
Full GPR

- $\hat{y}(x) = K_x^T K_{xx}^{-1} y$

  $\hat{\sigma}_y{}^2 = 1 - K_x^T K_{xx}^{-1} K_x$

- N.B.: Matrix inversion is O(n^3), which is slow

- This was an intuitive “reading” of the GPR equations. For a more precise presentation, see Appendix C.



What puts the G in GPR?

And how is it a "process"?

- Model each value $y(x)$ as a gaussian distribution
- Model any collection of $\{y(x)\}$ as a multivariate gaussian distribution
  - x is continuous, so really an infinite-dimension gaussian distribution
- First considered as $y(t)$, where t is time. A process is something that changes over time. A gaussian process is one where y has a gaussian distribution that changes over time. Ex: a Brownian motion (continuous random walk)
- Change t to x and you have a machine learning tool, GP regression

Summary

Gaussian process regression (GPR)

- GPR is *nonparametric*, forms estimates directly from measurements
- GPR reports estimates of BM and of *model uncertainty* in BM
- Used as *surrogate function* in Bayesian optimization
- Computation is kind of slow, $O(n^3)$

$$\hat{y}(x) = K_x^T K_{xx}^{-1} y$$

$$\hat{\sigma}_y{}^2 = 1 - K_x^T K_{xx}^{-1} K_x$$