BART as a Gaussian process

Giacomo Petrillo
2023-10-26
Department of Statistics, Computer Science, Applications (DISIA)
University of Florence

Hosted by UT Austin's SDS

Gaussian process regression

- Another Bayesian nonparametric method

Gaussian process regression

- Another Bayesian nonparametric method
- Gaussian process = Multivariate Normal in ∞ dimensions

Gaussian process regression

- Another Bayesian nonparametric method
- Gaussian process = Multivariate Normal in ∞ dimensions
- Finite marginals are Normal

Gaussian process regression

- Another Bayesian nonparametric method
- Gaussian process = Multivariate Normal in ∞ dimensions
- Finite marginals are Normal
- $\implies$ Analytical calculations

Gaussian process regression

- Another Bayesian nonparametric method
- Gaussian process = Multivariate Normal in ∞ dimensions
- Finite marginals are Normal
- $\implies$ Analytical calculations
- A priori $\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix} \right)$

GP—Inference

- We know $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) = \mathbf{y}$

GP—Inference

- We know $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) = \mathbf{y}$
- We want $f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_m^*)$

GP—Inference

- We know $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)) = \mathbf{y}$
- We want $f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_m^*)$
- $$\begin{pmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_n) \\ f(\mathbf{x}_1^*) \\ \vdots \\ f(\mathbf{x}_m^*) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \\ m(\mathbf{x}_1^*) \\ \vdots \\ m(\mathbf{x}_m^*) \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) & k(\mathbf{x}_1, \mathbf{x}_1^*) & \cdots & k(\mathbf{x}_1, \mathbf{x}_m^*) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) & k(\mathbf{x}_n, \mathbf{x}_1^*) & \cdots & k(\mathbf{x}_n, \mathbf{x}_m^*) \\ k(\mathbf{x}_1^*, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1^*, \mathbf{x}_n) & k(\mathbf{x}_1^*, \mathbf{x}_1^*) & \cdots & k(\mathbf{x}_1^*, \mathbf{x}_m^*) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_m^*, \mathbf{x}_1) & \cdots & k(\mathbf{x}_m^*, \mathbf{x}_n) & k(\mathbf{x}_m^*, \mathbf{x}_1^*) & \cdots & k(\mathbf{x}_m^*, \mathbf{x}_m^*) \end{pmatrix} \right)$$

GP—Inference

- Abbreviate $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)), \quad \mathbf{f}^* = (f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_m^*))$

GP—Inference

- Abbreviate $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)), \quad \mathbf{f}^* = (f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_m^*))$
- Abbreviate $\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{m}^* \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xx^*} \\ \Sigma_{x^*x} & \Sigma_{x^*x^*} \end{pmatrix} \right)$

GP—Inference

- Abbreviate $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)), \quad \mathbf{f}^* = (f(\mathbf{x}_1^*), \ldots, f(\mathbf{x}_m^*))$
- Abbreviate $\begin{pmatrix} \mathbf{f} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m} \\ \mathbf{m}^* \end{pmatrix}, \begin{pmatrix} \Sigma_{xx} & \Sigma_{xx^*} \\ \Sigma_{x^*x} & \Sigma_{x^*x^*} \end{pmatrix}\right)$
- $(\mathbf{f}^* \mid \mathbf{f} = \mathbf{y}) \sim \mathcal{N}(\mathbf{m}^* + \Sigma_{x^*x}\Sigma_{xx}^{+}(\mathbf{y} - \mathbf{m}), \quad \Sigma_{x^*x^*} - \Sigma_{x^*x}\Sigma_{xx}^{+}\Sigma_{xx^*})$

BART ≠ GP

- "Given its underlying tree structure, intuitively **BART may not have the flexibility** to capture the additional uncertainty in regions of poor overlap, **whereas** some other "smoother" Bayesian nonparametric models such as the **Gaussian Process may fare better**." (Hahn et al. 2020)

BART ≠ GP

- "Given its underlying tree structure, intuitively **BART may not have the flexibility** to capture the additional uncertainty in regions of poor overlap, **whereas** some other "smoother" Bayesian nonparametric models such as the **Gaussian Process may fare better**." (Hahn et al. 2020)

- "Similarly, while **Gaussian processes may induce smoothness** in the regression, it could be argued **BART-based models are easier to implement** in practice and work well off-the-shelf with minimal tuning." (Hahn et al. 2020)

BART ≠ GP

- "Given its underlying tree structure, intuitively **BART may not have the flexibility** to capture the additional uncertainty in regions of poor overlap, **whereas** some other "smoother" Bayesian nonparametric models such as the **Gaussian Process may fare better.**" (Hahn et al. 2020)

- "Similarly, while **Gaussian processes may induce smoothness** in the regression, it could be argued **BART-based models are easier to implement** in practice and work well off-the-shelf with minimal tuning." (Hahn et al. 2020)

- "Note how the **GP-estimated** expected outcomes tick up or down outside the range of the data **based on a handful of observations at the extremes**, as opposed to **BART** and the linear model which **extrapolate in predictable ways.**" (Hahn et al. 2020)

BART ≠ GP

- “Finally, several of the discussants proposed Gaussian process models with **limited discussion of the covariance function and how its parameters are set or inferred**. The covariance function is often pivotal to their success. Unsurprisingly, the squared exponential covariance function performs splendidly on very smooth response surfaces, but what happens when this strong assumption is violated? **By contrast, BART has a long track record of adapting successfully to a wide variety of unknown covariance structures and this robustness** is why we chose to design BCF around BART priors.” (Hahn et. al 2020)

BART ≠ GP

- "Although not widely appreciated, **BART actually is a Gaussian process**, conditional on the trees (integrating over Gaussian priors over the leaf parameters). Specifically, the trees define a covariance function where the correlation between points x and x′ are a function of the proportion of trees in the forest in which the two points occupy the same leaf. **As the number of trees is increased, this covariance function becomes increasingly smooth, although it is singular and nonstationary for a finite number of trees.**" (Hahn et al. 2020)

- ~~(N.B. there are technical errors here)~~ Correction: J. Murray has clarified to me what he meant, and I now agree I was misunderstanding him.

BART $\longrightarrow$ GP

- $y_i = \sum_{j=1}^{m} g(x_i; T_j, M_j) + \varepsilon_i$

BART $\longrightarrow$ GP

- $y_i = \sum_{j=1}^{m} g(x_i; T_j, M_j) + \varepsilon_i$

- $g(x; T_j, M_j)$ are a priori i.i.d.

BART $\longrightarrow$ GP

- $y_i = \sum_{j=1}^{m} g(x_i; T_j, M_j) + \varepsilon_i$

- $g(x; T_j, M_j)$ are a priori i.i.d.

- These are the hypotheses of the multivariate CLT

BART $\longrightarrow$ GP

- $y_i = \sum_{j=1}^{m} g(x_i; T_j, M_j) + \varepsilon_i$

- $g(x; T_j, M_j)$ are a priori i.i.d.

- These are the hypotheses of the multivariate CLT

- As $m \to \infty$: $\begin{pmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} k_{\text{BART}}(x_1, x_1) & \cdots & k_{\text{BART}}(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k_{\text{BART}}(x_n, x_1) & \cdots & k_{\text{BART}}(x_n, x_n) \end{pmatrix} \right)$

What is k_{BART}?

- Linero 2017:
- “[...] under some approximations [...] the associated kernel function [...] is [...] $k(x, x') \propto \exp(-\lambda \|x - x'\|_1)$.”

What is k_{BART}?

- Linero 2017:
- "[...] under some approximations [...] the associated kernel function [...] is [...] $k(x, x') \propto \exp(-\lambda \|x - x'\|_1)$."
- This is a bog-standard GP covariance function

What is k_{BART}?

- Linero 2017:
- "[...] under some approximations [...] the associated kernel function [...] is [...] $k(x, x') \propto \exp(-\lambda \|x - x'\|_1)$."
- This is a bog-standard GP covariance function
- But: "Furthermore, our experience is that the empirical performance of a minimally-tuned implementation of **BART is frequently better than Gaussian process regression using the equivalent kernel** [...] We conjecture that the reason for BART outperforming Gaussian process regression is that limiting the number of trees in the ensemble allows one to learn a data-adaptive notion of distance between points."

What is k_{BART}?

- O'Reilly 2022 (h/t S. Deshpande):
- $k(x, x') \propto \exp(-\lambda P_{\text{split}}(\{\text{hyperplanes separating the points}\})$

What is k_{BART}?

- O'Reilly 2022 (h/t S. Deshpande):
- $k(x, x') \propto \exp(-\lambda P_{\text{split}}(\{\text{hyperplanes separating the points}\})$
- I did not know about this when I did the calculation in 2022

What is k_{BART}?

- O'Reilly 2022 (h/t S. Deshpande):
- $k(x, x') \propto \exp(-\lambda P_{\text{split}}(\{\text{hyperplanes separating the points}\})$
- I did not know about this when I did the calculation in 2022
- But I don't see how to use it to do the specific BART calculation

My k_{BART}

- $k(x, x') = P(x$ and x' not separated$)$

My k_{BART}

- $k(x, x') = P(x \text{ and } x' \text{ not separated})$
- $$= \sum_{\text{non-separating trees}} P(\text{tree})$$

My k_{BART}

- $k(x, x') = P(x \text{ and } x' \text{ not separated})$
- $= \sum_{\text{non-separating trees}} P(\text{tree})$
- I write out the summation recursively for the BART prior

My k_{BART}

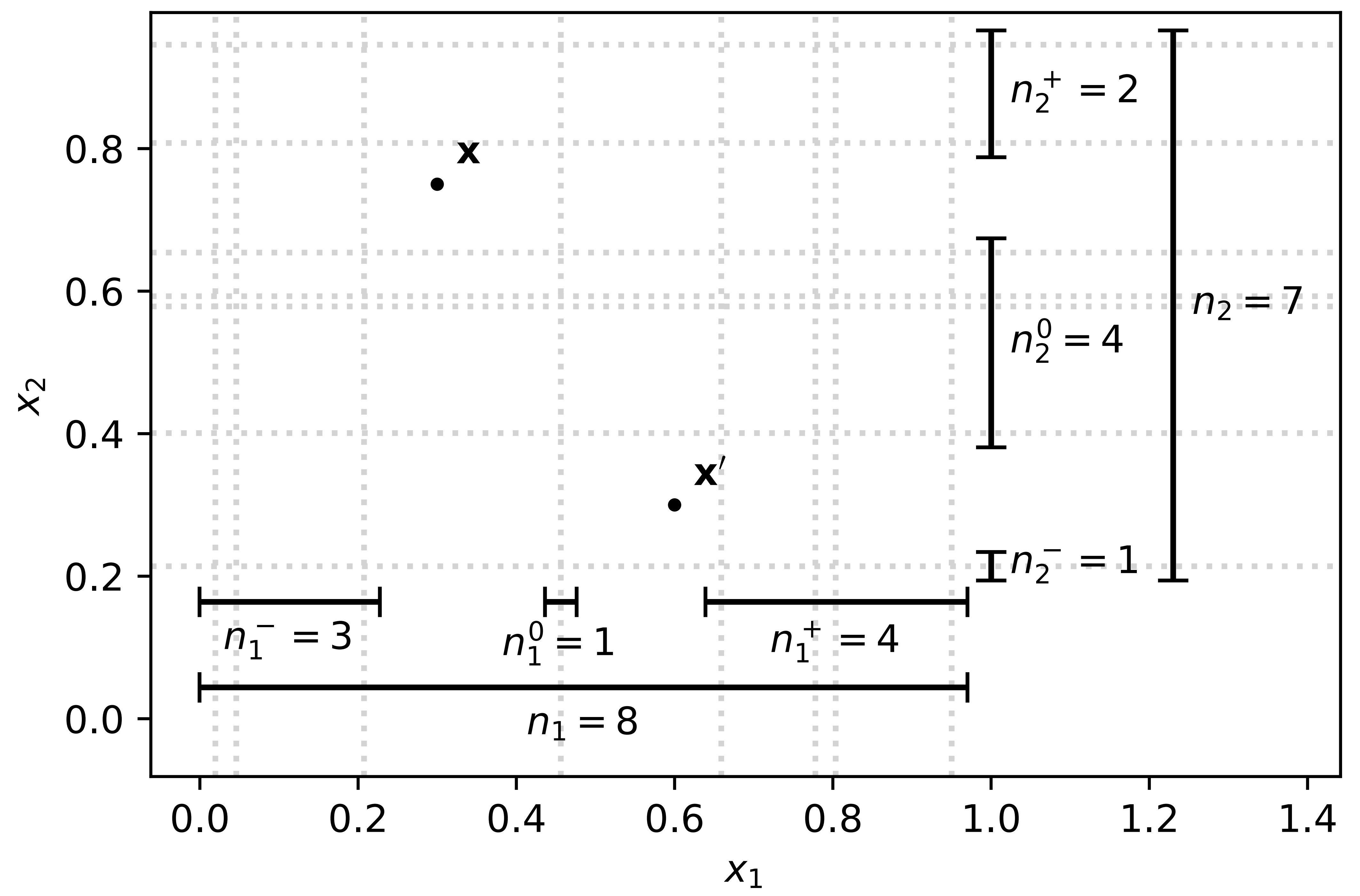
$$k(\mathbf{x}, \mathbf{x}') = k_0(\mathbf{n}^-, \mathbf{n}^0, \mathbf{n}^+), \qquad \mathbf{n} = \mathbf{n}^- + \mathbf{n}^0 + \mathbf{n}^+,$$

$$k_d(\mathbf{0}, \mathbf{0}, \mathbf{0}) = k_d((), (), ()) = 1,$$

$$k_d(\mathbf{n}^-, \mathbf{n}^0, \mathbf{n}^+) = 1 - P_d\left[1 - \frac{1}{W(\mathbf{n})} \sum_{\substack{i=1 \\ n_i \neq 0}}^{p} \frac{w_i}{n_i} \left(\sum_{k=0}^{n_i^- - 1} k_{d+1}(\mathbf{n}^-_{n_i^- = k}, \mathbf{n}^0, \mathbf{n}^+) + \sum_{k=0}^{n_i^+ - 1} k_{d+1}(\mathbf{n}^-, \mathbf{n}^0, \mathbf{n}^+_{n_i^+ = k}) \right)\right],$$

$$W(\mathbf{n}) = \sum_{\substack{i=1 \\ n_i \neq 0}}^{p} w_i, \qquad \mathbf{w} > 0, \qquad P_d = \frac{\alpha}{(1+d)^\beta},$$

Incomputable!



My k_{BART}

$$k^D_{D-2}(\mathbf{n}^-, \mathbf{0}, \mathbf{n}^+) = 1,$$

$$k^D_{D-2}(\mathbf{n}^-, \underbrace{\mathbf{n}^0}_{\neq \mathbf{0}}, \mathbf{n}^+) = 1 - P_{D-2}\Bigg[1 - \frac{1}{W(\mathbf{n})}\Bigg[(1-P_{D-1})S + P_{D-1}(1-(1-\gamma)P_D)\sum_{\substack{i=1\\ n_i\neq 0}}^{p} \frac{w_i}{n_i}\Bigg($$

$$\left(S + w_i\frac{n_i^0}{n_i}\right)\left(\frac{1}{W(\mathbf{n}_{n_i^-=0})} + \frac{1}{W(\mathbf{n}_{n_i^+=0})} + \frac{n_i^- + n_i^+ - 2}{W(\mathbf{n})}\right) +$$

$$+ \frac{w_i}{W(\mathbf{n}_{n_i^-=0})}\left(\left\{n_i^0 + n_i^+ \,\middle|\, \frac{n_i^+}{n_i^0 + n_i^+}\right\} - 1\right) +$$

$$+ \frac{w_i}{W(\mathbf{n}_{n_i^+=0})}\left(\left\{n_i^0 + n_i^- \,\middle|\, \frac{n_i^-}{n_i^0 + n_i^-}\right\} - 1\right) +$$

$$- \frac{w_i n_i^0}{W(\mathbf{n})}\left(2\psi(n_i) - \psi(1 + n_i^0 + n_i^-) - \psi(1 + n_i^0 + n_i^+)\right)\Bigg)\Bigg]\Bigg],$$

$$S = \sum_{\substack{i=1\\ n_i\neq 0}}^{p} w_i\left(1 - \frac{n_i^0}{n_i}\right),$$

$$\{x \mid E\} = \begin{cases} E & x > 0, \\ 0 & x = 0, \text{ even if } E \text{ is not well defined,} \end{cases}$$

Computable approximate formula (first stage)

This is exact for depth ≤ 2. Then I do some tricks to "repeat" it without actually doing the recursion.

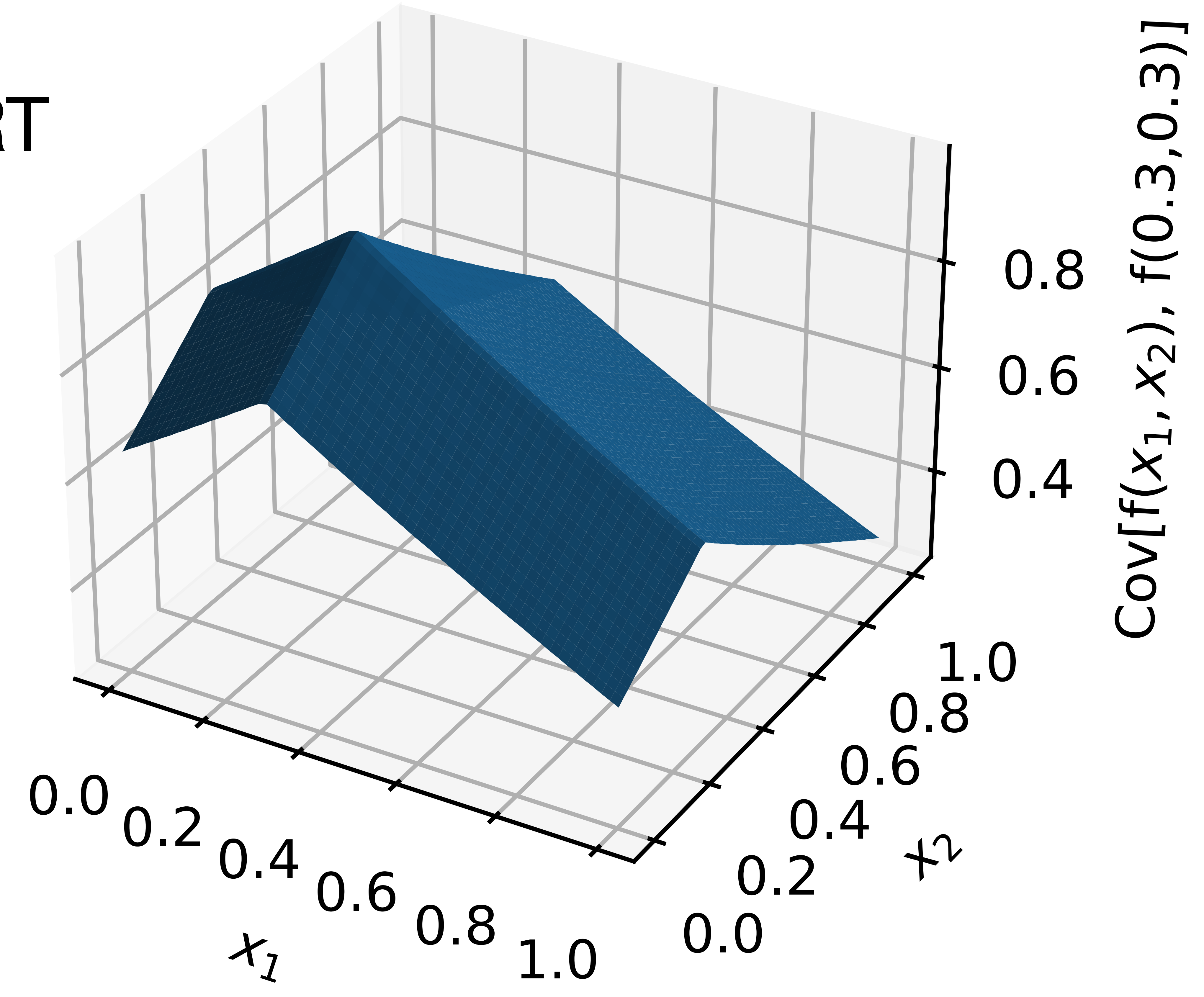
k_{BART} vs. $e^{-\lambda \|x - x'\|_1 / p}$

- Are they similar enough?

k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Are they similar enough?

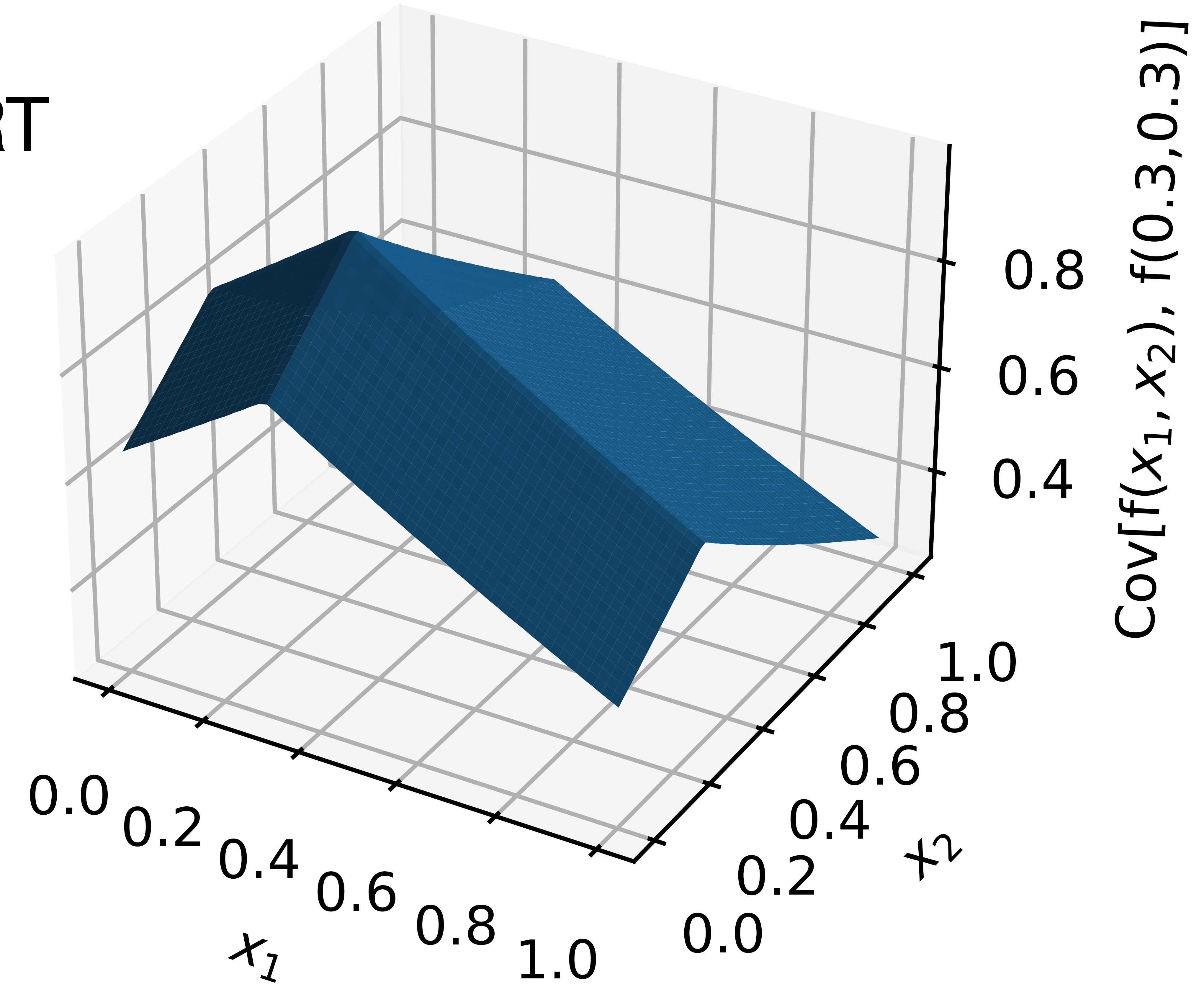
BART



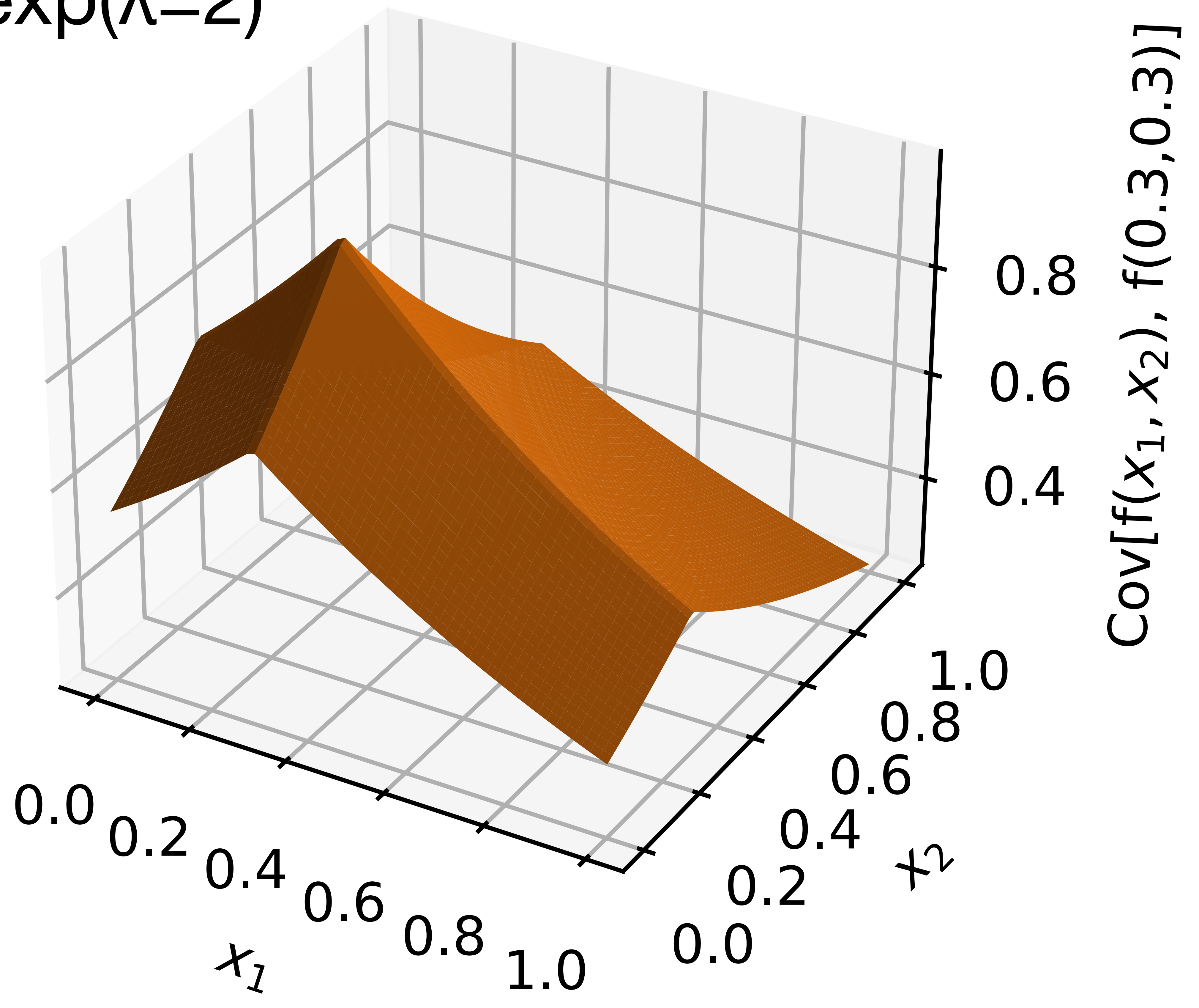
k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Are they similar enough?

BART



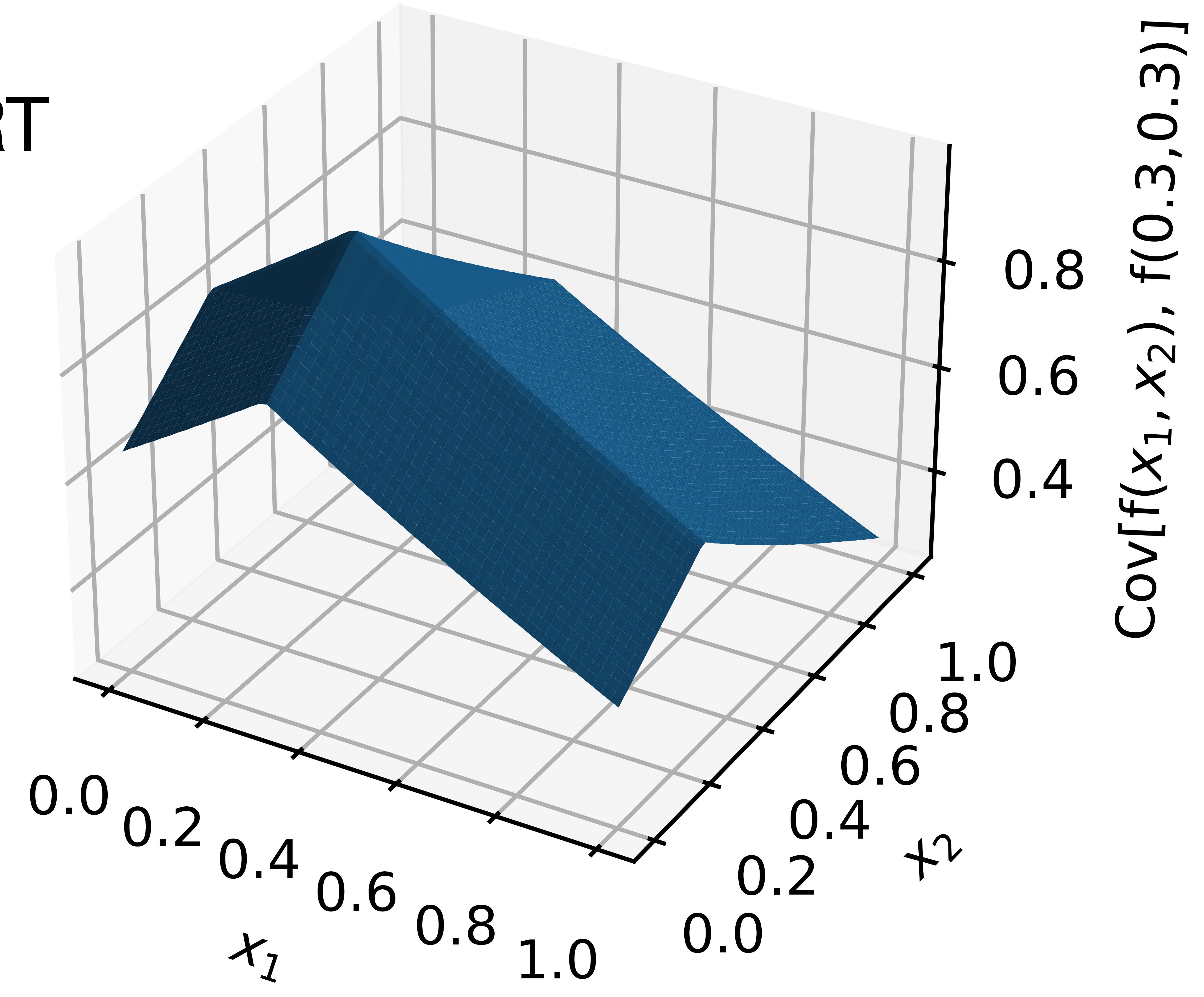
exp(λ=2)



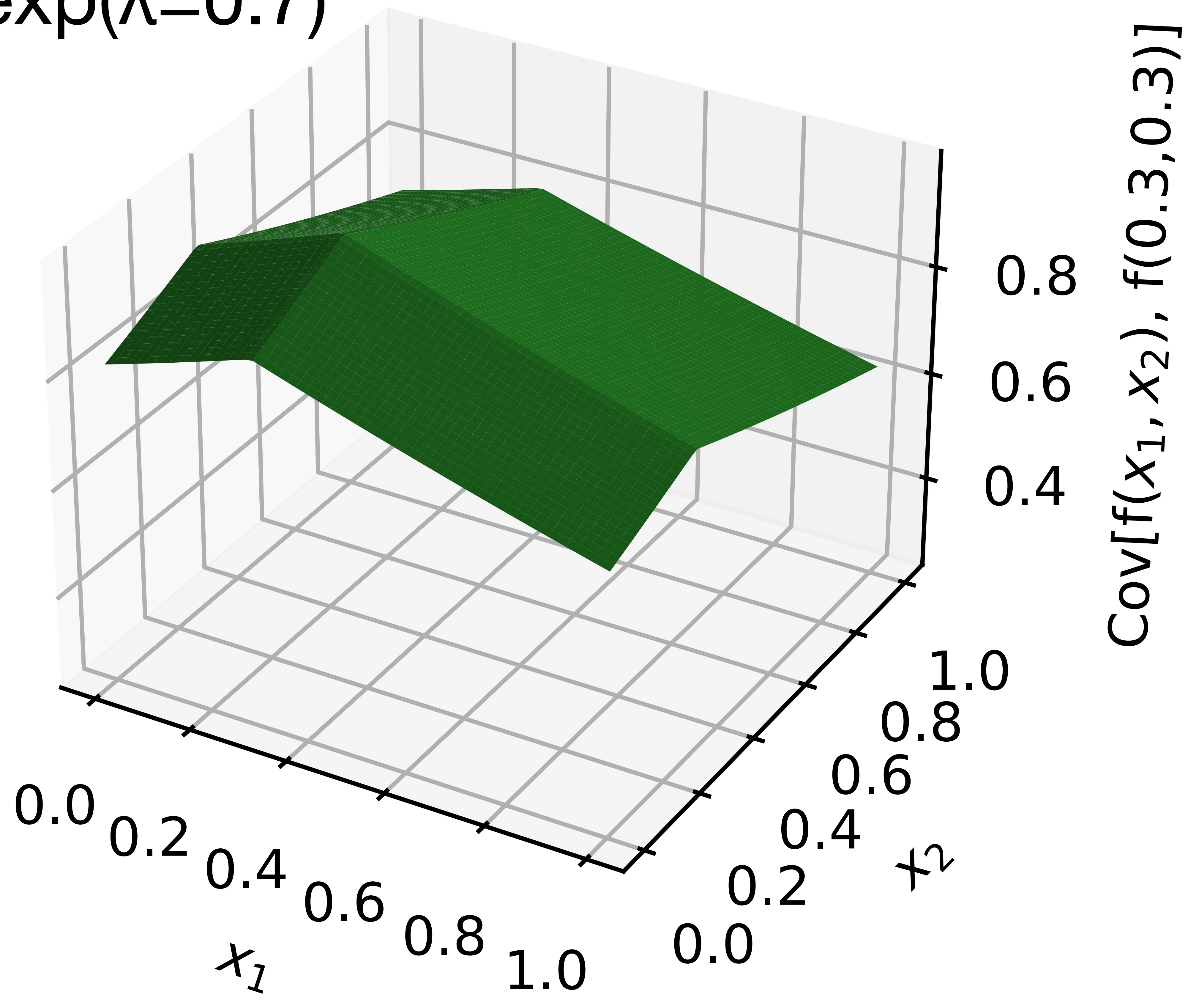
k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Are they similar enough?

BART



exp(λ=0.7)



k_{BART} vs. $e^{-\lambda \|x - x'\|_1 / p}$

- Problem:

k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Problem:
- $k_{\text{BART}}(\mathbf{x}, \mathbf{x}') \approx 1 - \frac{\|\mathbf{x} - \mathbf{x}'\|_1}{p}$

k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Problem:
- $k_{\mathrm{BART}}(\mathbf{x},\mathbf{x}') \approx 1 - \dfrac{\|\mathbf{x}-\mathbf{x}'\|_1}{p}$
- $e^{-\lambda\|\mathbf{x}-\mathbf{x}'\|_1/p} \approx 1 - \lambda\,|x_1 - x_1'| - \lambda\,|x_2 - x_2'|$ if $\lambda \to 0$

k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Problem:
- $k_{\text{BART}}(\mathbf{x}, \mathbf{x}') \approx 1 - \dfrac{\|\mathbf{x} - \mathbf{x}'\|_1}{p}$
- $e^{-\lambda\|\mathbf{x}-\mathbf{x}'\|_1/p} \approx 1 - \lambda\,|x_1 - x_1'| - \lambda\,|x_2 - x_2'|$ if $\lambda \to 0$
- Either it's not separable, or the intercept prior variance is large

k_{BART} vs. $e^{-\lambda\|x-x'\|_1/p}$

- Problem:
- $k_{\text{BART}}(\mathbf{x}, \mathbf{x}') \approx 1 - \dfrac{\|\mathbf{x} - \mathbf{x}'\|_1}{p}$
- $e^{-\lambda\|\mathbf{x}-\mathbf{x}'\|_1/p} \approx 1 - \lambda\,|\,x_1 - x_1'\,| - \lambda\,|\,x_2 - x_2'\,|$ if $\lambda \to 0$
- Either it's not separable, or the intercept prior variance is large
- Speculative solution: $\exp(-\lambda\|x - x'\|_1/p) - e^{-\lambda}$, which is p.s.d. although not widely known

$$\exp(-\lambda\|x - x'\|_1/p) - e^{-\lambda}$$

- Proof of positivity:

$\exp(-\lambda\|x - x'\|_1/p) - e^{-\lambda}$

- Proof of positivity:
- $e^{\lambda k} = \sum_{n=0}^{\infty} \frac{(\lambda k)^n}{n!}$

$\exp(-\lambda \|x - x'\|_1 / p) - e^{-\lambda}$

- Proof of positivity:
- $e^{\lambda k} = \sum_{n=0}^{\infty} \frac{(\lambda k)^n}{n!}$ $\qquad$ $e^{\lambda k} - 1 = \sum_{n=1}^{\infty} \frac{(\lambda k)^n}{n!}$
- so $\exp(\lambda k(x, x')) - 1$ is a valid covariance function for any k

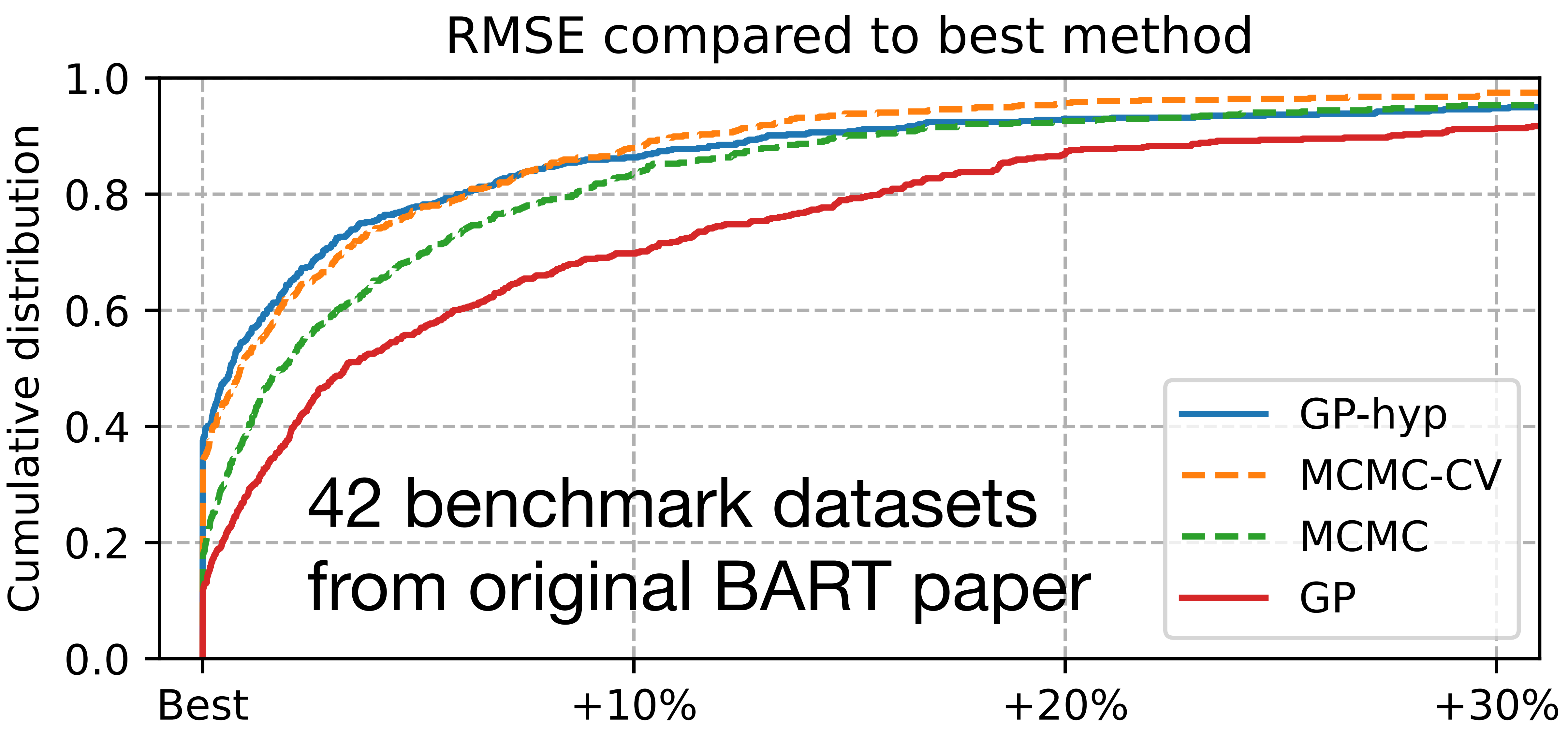
$\exp(-\lambda \|x - x'\|_1/p) - e^{-\lambda}$

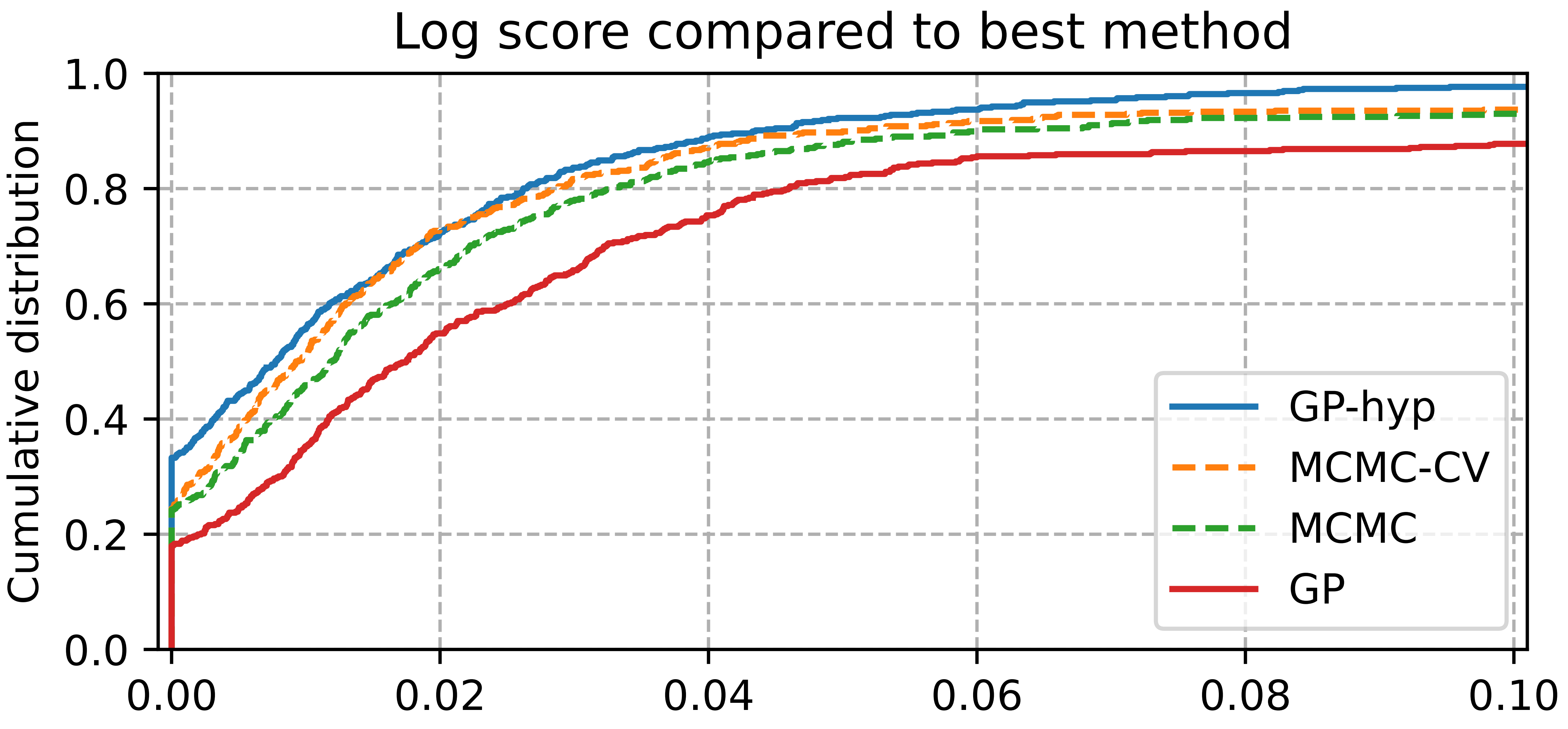
- Proof of positivity:
- $e^{\lambda k} = \sum_{n=0}^{\infty} \frac{(\lambda k)^n}{n!}$ $\qquad e^{\lambda k} - 1 = \sum_{n=1}^{\infty} \frac{(\lambda k)^n}{n!}$
- so $\exp(\lambda k(x, x')) - 1$ is a valid covariance function for any k
- plug $k(x, x') = \frac{1}{p} \sum_{i=1}^{p} (1 - |x_i - x_i'|)$

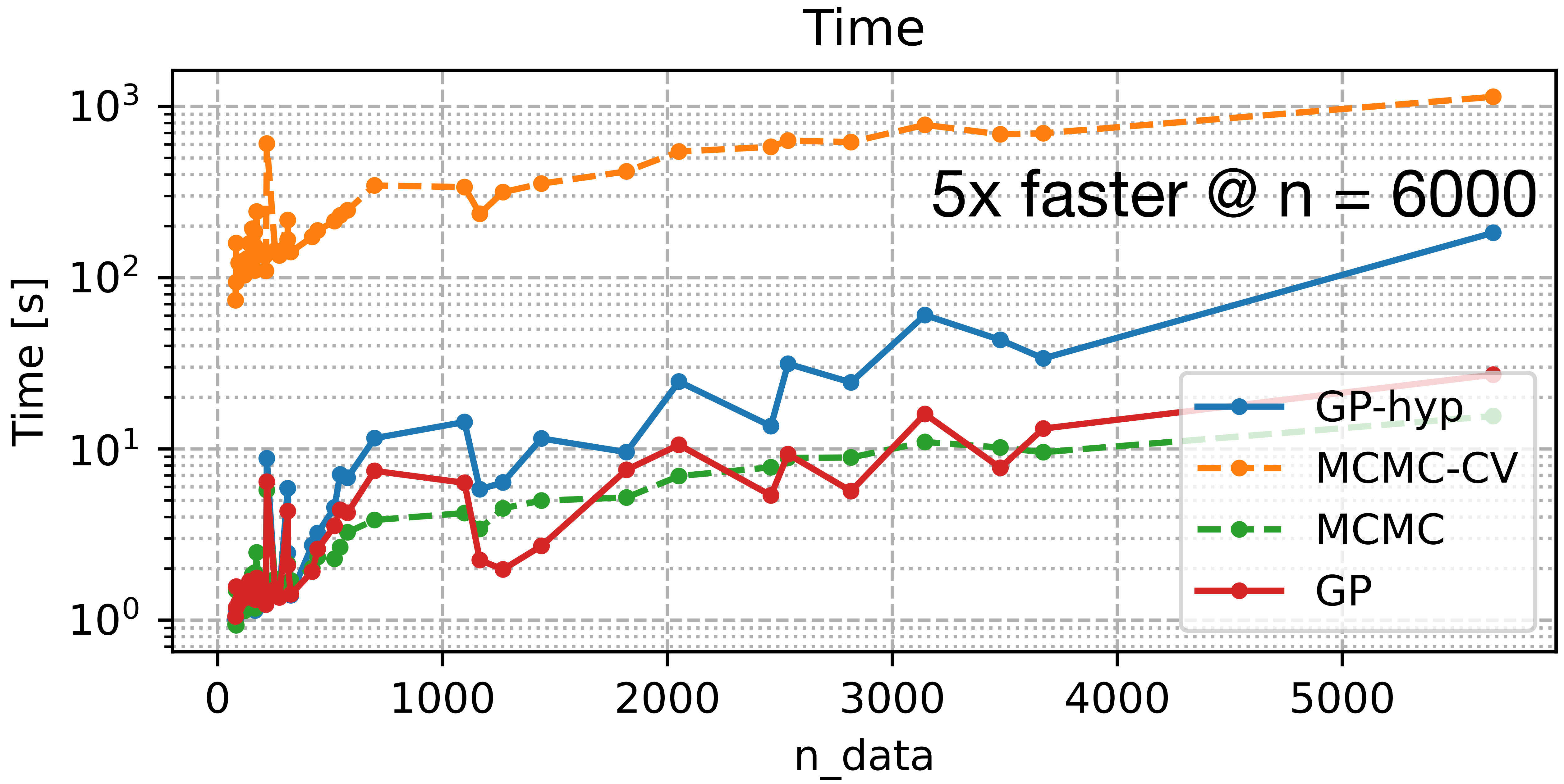
$\exp(-\lambda \|x - x'\|_1 / p) - e^{-\lambda}$

- Proof of positivity:
- $e^{\lambda k} = \sum_{n=0}^{\infty} \frac{(\lambda k)^n}{n!}$ $\qquad$ $e^{\lambda k} - 1 = \sum_{n=1}^{\infty} \frac{(\lambda k)^n}{n!}$
- so $\exp(\lambda k(x, x')) - 1$ is a valid covariance function for any k
- plug $k(x, x') = \frac{1}{p} \sum_{i=1}^{p} (1 - |x_i - x_i'|)$ (triangular covariance function)

BART MCMC vs. BART GP







At fixed hypers, MCMC > GP

At free hypers, GP > MCMC

Can't explore all hypers with MCMC because trees must be shallow, and needs CV

Whither BART GP?

Many possible further directions:

Whither BART GP?

Many possible further directions:

1. Could we bypass MCMC hyperparameter restrictions by combining BART with something simpler similar to deep trees?

Whither BART GP?

Many possible further directions:

1. Could we bypass MCMC hyperparameter restrictions by combining BART with something simpler similar to deep trees?

2. I benchmarked BART packages on CRAN and picked the fastest; what about flexBART? (should be faster)

Whither BART GP?

Many possible further directions:

1. Could we bypass MCMC hyperparameter restrictions by combining BART with something simpler similar to deep trees?

2. I benchmarked BART packages on CRAN and picked the fastest; what about flexBART? (should be faster)

3. GP versions of BART variants (doable but tedious)

Whither BART GP?

Many possible further directions:

1. Could we bypass MCMC hyperparameter restrictions by combining BART with something simpler similar to deep trees?

2. I benchmarked BART packages on CRAN and picked the fastest; what about flexBART? (should be faster)

3. GP versions of BART variants (doable but tedious)

4. Trying GP techniques to scale to large datasets

Whither BART GP?

Many possible further directions:

1. Could we bypass MCMC hyperparameter restrictions by combining BART with something simpler similar to deep trees?
2. I benchmarked BART packages on CRAN and picked the fastest; what about flexBART? (should be faster)
3. GP versions of BART variants (doable but tedious)
4. Trying GP techniques to scale to large datasets
5. Make up GP kernels similar to the BART kernel

Conclusions

I learned:

Conclusions

I learned:

- What you can do with BART you can do with GP

Conclusions

I learned:

- What you can do with BART you can do with GP
- Covariance matrices are very sensitive

Conclusions

I learned:

- What you can do with BART you can do with GP
- Covariance matrices are very sensitive
- Choice of kernel is very important with GPs, I have the impression there's too much defaulting

Conclusions

I learned:

- What you can do with BART you can do with GP
- Covariance matrices are very sensitive
- Choice of kernel is very important with GPs, I have the impression there's too much defaulting
- (e.g. exponential quadratic $e^{-\|x-x'\|^2}$, weird guy)

Code

- My GP Python package: https://github.com/Gattocrucco/lsqfitgp
- Implements the BART kernel
- And ready to use functions for BART or BCF GP regression