Gaussian Processes and Kernel Ridge Regression

Theoretical Connections, Hyperparameter Tuning, and Experiments

Team 10: Leonardo Cartesegna, Chandrasekhara Devarakonda, Giulia Scagliarini

MATH-412: Statistical Machine Learning

December 9, 2025

Outline

Table of Contents

Context and Motivation

- Focus is on understanding the link between **Gaussian process (GP) regression** and **kernel ridge regression (KRR)**.
- Both methods use a positive definite kernel $k(\cdot, \cdot)$ and lead to very similar prediction formulas.
- However:
  - GP regression is a **Bayesian, probabilistic** model.
  - KRR is a **regularization / optimization** approach in a RKHS.

Project Objectives

- **Theoretical comparison**
  - Show how GP regression and KRR lead to (almost) the same predictor.
  - Explain the functional / RKHS viewpoint behind this equivalence.
- **Hyperparameter selection**
  - GP: **marginal likelihood** (type-II ML).
  - KRR: **cross-validation** (e.g. K-fold).
  - Discuss whether cross-validation also makes sense for GPs.
- **Empirical study**
  - Use a **static financial dataset** (portfolio performance).
  - Compare GP and KRR on the same kernel and same data.
  - Stay in an **interpolation** regime, avoid unjustified time extrapolation.

Table of Contents

Supervised Regression Setup

Following Rasmussen & Williams:

- Training set

$$\mathcal{D} = \{(x_i, y_i) \mid i = 1, \ldots, n\}, \quad x_i \in \mathbb{R}^D, \; y_i \in \mathbb{R}.$$

- Inputs (covariates):
  - collected in the $D \times n$ design matrix $X = [x_1, \ldots, x_n]$.
- Targets:

$$\mathrm{y} = (y_1, \ldots, y_n)^\top.$$

- Goal: for a new input x_*, infer its output y_*

Observation generation model

- Typically we assume the observations are generated from the latent function subject to an additive gaussian noise:

$$y_i = f(x_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_n^2) \text{ i.i.d.}$$

- So, conditional on $f(x_i)$, the observation y_i is distributed as:

$$y_i \mid f(x_i) \sim \mathcal{N}(f(x_i), \sigma_n^2).$$

- We are primarily interested in learning f from $\mathcal{D}$.
- Two perspectives:
  - **Gaussian processes:** Learns a posterior distribution over functions f, by assuming a prior on the function space - **Bayesian Approach**.
  - **Kernel ridge regression:** Learns a particular function f that minimizes the regularized empirical risk in an RKHS ($\mathcal{H}_k$) - **Risk Minimization Approach**.

Table of Contents

Gaussian Process Prior

- A Gaussian process is a collection of random variables such that any finite subset is jointly Gaussian:

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

- For simplicity, we assume a **zero mean function**, i.e., $m(x) = 0$. Note that GPs are not limited to this and can incorporate arbitrary mean functions.
- The kernel function is denoted by $k_\theta(x, x')$ with hyperparameters θ (e.g. length-scales, variances).
- Prior over function values at training inputs:

$$\mathrm{f} = (f(x_1), \ldots, f(x_n))^\top \sim \mathcal{N}(0, K),$$

where $K_{ij} = k(x_i, x_j)$.

Likelihood and Joint Distribution

- Noise model:
$$p(\mathrm{y} \mid \mathrm{f}) = \mathcal{N}(\mathrm{f}, \sigma_n^2 I).$$
- Marginalizing out f gives
$$\mathrm{y} \sim \mathcal{N}(0, K_y), \quad K_y = K + \sigma_n^2 I.$$
- For a test input x_*:
$$f_* = f(x_*)$$
with joint prior
$$\begin{bmatrix} \mathrm{f} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K & \mathrm{k}_* \\ \mathrm{k}_*^\top & k_{**} \end{bmatrix} \right),$$
where $\mathrm{k}_* = k(X, x_*)$, $k_{**} = k(x_*, x_*)$.

GP Posterior and Predictions

Conditional on y, the posterior over f_* is Gaussian:

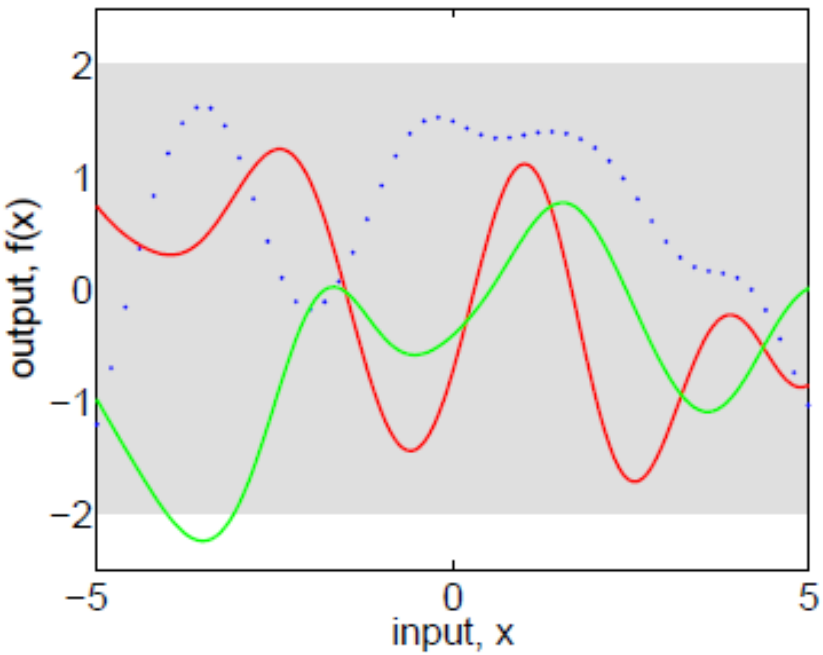
$$f_* \mid x_*, X, \mathrm{y} \sim \mathcal{N}\big(m(x_*), \mathrm{cov}(x_*)\big),$$

with

$$\begin{aligned} m(x_*) &= \mathrm{k}_*^\top K_y^{-1} \mathrm{y}, \\ \mathrm{cov}(x_*) &= k_{**} - \mathrm{k}_*^\top K_y^{-1} \mathrm{k}_*. \end{aligned}$$

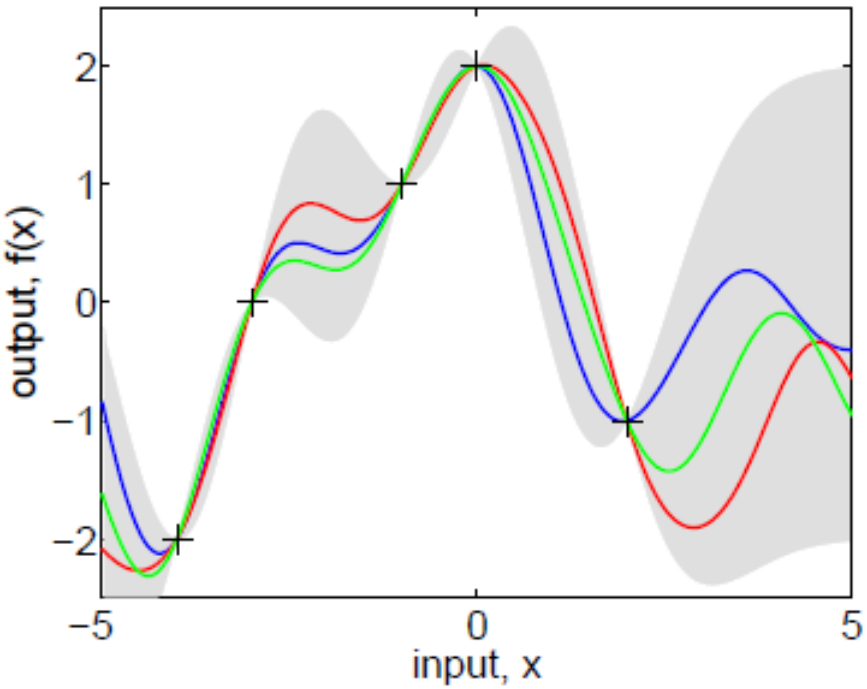
- **Posterior mean** $m(x_*)$ is the mean latent function value at x_* given y
- **Posterior variance** $\mathrm{cov}(x_*)$ quantifies uncertainty in f_* given y.
  - very small near training points ($\because k_* \to 1$).
  - reverts to prior variance (k_{**}) far from data ($\because k_* \to 0$).

GP Prior and Posterior



Prior

Before seeing any data, the GP prior represents a wide range of plausible smooth functions drawn from the covariance structure.



Posterior

After conditioning on observed data, the GP posterior concentrates around the training points and uncertainty shrinks locally.

Inference with the GP Posterior

- Unlike Empirical Risk Minimization (ERM), which yields a single estimate $\hat{f}$, the GP gives a full posterior distribution $p(f_* \mid \mathrm{y})$ of true latent value f_*
- We can choose a point prediction based on a loss function $\mathcal{L}(f_*, \hat{f})$:
  - **Squared Error Loss**: Minimizing $\mathbb{E}[(f_* - \hat{f})^2]$ yields the posterior mean.
  - **Absolute Error Loss**: Minimizing $\mathbb{E}[|f_* - \hat{f}|]$ yields the posterior median.
- For a Gaussian posterior, the mean and median coincide.
- This framework allows for decision-making under arbitrary asymmetric loss functions using the full uncertainty information.

Table of Contents

Kernel Ridge Regression Objective

- Let $k(\cdot, \cdot)$ be a positive definite kernel with associated RKHS $\mathcal{H}_k$.
- Kernel ridge regression solves the following objective:

$$\min_{f \in \mathcal{H}_k} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \frac{\lambda}{2} \|f\|_{\mathcal{H}_k}^2 \right\}.$$

- Since $\mathcal{H}_k$ is potentially an infinite-dimensional space, direct optimization is challenging.
- The squared norm $\|f\|_{\mathcal{H}_k}^2$ quantifies the **roughness** of the function f. A larger norm implies a less smooth function. The hyperparameter $\lambda > 0$ balances the trade-off between minimizing the empirical risk (data fit) and controlling the function's smoothness.

Representer Theorem and Finite-Dimensional Form

- Since f lies in RKHS $\mathcal{H}_k$, by the representer theorem, the minimizer satisfies

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x).$$

- Let $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$. Then

$$f(x_j) = \sum_{i=1}^{n} \alpha_i k(x_i, x_j),$$

so in vector form

$$\mathsf{f} = K\boldsymbol{\alpha}.$$

- Plugging into the objective, one obtains the finite-dimensional problem

$$\min_{\boldsymbol{\alpha}} \frac{1}{2n} \|K\boldsymbol{\alpha} - \mathsf{y}\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}.$$

Solution of Kernel Ridge Regression

- Differentiating with respect to $\boldsymbol{\alpha}$ and setting to zero yields

$$\boldsymbol{\alpha}^* = (K + \lambda n I)^{-1} \mathrm{y}.$$

- The predictor at a test point x_* is

$$\begin{aligned} f_{\mathrm{KRR}}(x_*) &= \sum_{i=1}^{n} \alpha_i^* k(x_i, x_*) \\ &= \mathrm{k}_*^\top \boldsymbol{\alpha}^* \\ &= \mathrm{k}_*^\top (K + \lambda n I)^{-1} \mathrm{y}. \end{aligned}$$

- Compare with the GP predictive estimate under squared error loss (posterior mean):

$$m(x_*) = \mathrm{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathrm{y}.$$

Table of Contents

What the representer theorem does (and does not) explain

- From the KRR objective, the representer theorem guarantees the minimizer has the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i \, k(x_i, \cdot),$$

  so the problem reduces to a finite-dimensional optimization over $\boldsymbol{\alpha} \in \mathbb{R}^n$.
- However, this alone does not explain *why* the specific penalty $\|f\|_{\mathcal{H}_k}^2$ is the "right" regularizer, nor why it matches GP regression.
- The missing principle is: **KRR is the MAP estimator under a GP prior with Gaussian noise.**

MAP for latent training values

Let $\mathsf{f} = (f(x_1), \ldots, f(x_n))^\top$ and $K_{ij} = k(x_i, x_j)$. Under the GP model:

$$p(\mathsf{y} \mid \mathsf{f}) = \mathcal{N}(\mathsf{y} \mid \mathsf{f}, \sigma_n^2 I), \qquad p(\mathsf{f}) = \mathcal{N}(\mathsf{f} \mid 0, K).$$

Bayes' rule gives $p(\mathsf{f} \mid X, \mathsf{y}) \propto p(\mathsf{y} \mid \mathsf{f})\, p(\mathsf{f})$. Dropping constants, the negative log-posterior is

$$-\log p(\mathsf{f} \mid X, \mathsf{y}) = \frac{1}{2\sigma_n^2} \|\mathsf{y} - \mathsf{f}\|_2^2 + \frac{1}{2} \mathsf{f}^\top K^{-1} \mathsf{f}.$$

Therefore the MAP estimator solves the strictly convex quadratic program

$$\mathsf{f}_{\mathrm{MAP}} = \arg\min_{\mathsf{f}} \left\{ \frac{1}{2\sigma_n^2} \|\mathsf{y} - \mathsf{f}\|_2^2 + \frac{1}{2} \mathsf{f}^\top K^{-1} \mathsf{f} \right\}.$$

Posterior Gaussian $\Rightarrow$ MAP = posterior mean (and the same linear system)

- Since the likelihood and prior are Gaussian, the posterior $p(\mathrm{f} \mid X, \mathrm{y})$ is Gaussian.
- For a Gaussian distribution, **mode = mean**. Hence

$$\mathrm{f}_{\mathrm{MAP}} = \mathbb{E}[\mathrm{f} \mid X, \mathrm{y}].$$

- First-order optimality condition for the MAP objective:

$$\frac{1}{\sigma_n^2}(\mathrm{f} - \mathrm{y}) + K^{-1}\mathrm{f} = 0 \quad \Longrightarrow \quad (K + \sigma_n^2 I)\mathrm{f} = K\mathrm{y}.$$

- Solving gives

$$\mathrm{f}_{\mathrm{MAP}} = K(K + \sigma_n^2 I)^{-1}\mathrm{y},$$

which matches the GP posterior mean evaluated at training inputs.

From the GP quadratic penalty to the RKHS norm penalty

- Consider $f(\cdot) = \sum_{i=1}^{n} \alpha_i \, k(x_i, \cdot)$ with $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\mathrm{f} = K\boldsymbol{\alpha}$.
- In the RKHS $\mathcal{H}_k$, this expansion satisfies

$$\|f\|_{\mathcal{H}_k}^2 = \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}.$$

- If K is invertible, $\boldsymbol{\alpha} = K^{-1}\mathrm{f}$ and therefore

$$\boldsymbol{\alpha}^\top K \boldsymbol{\alpha} = \mathrm{f}^\top K^{-1} \mathrm{f}.$$

- Substituting $\mathrm{f} = K\boldsymbol{\alpha}$ into the MAP objective yields

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2\sigma_n^2} \|\mathrm{y} - K\boldsymbol{\alpha}\|_2^2 + \frac{1}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha} \right\}.$$

KRR emerges by rescaling: parameter identification

- Multiply the MAP objective by σ_n^2/n (this does not change the minimizer):

$$\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2n} \|\mathrm{y} - K\boldsymbol{\alpha}\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top K \boldsymbol{\alpha} \right\}, \qquad \lambda = \frac{\sigma_n^2}{n}.$$

- This is exactly the finite-dimensional KRR problem (as derived earlier from the RKHS objective).
- Consequently, the linear system defining the KRR coefficients,

$$(K + \lambda n I)\boldsymbol{\alpha} = \mathrm{y},$$

coincides with the GP system under the identification $\sigma_n^2 = \lambda n$.

Conclusion: same predictor, different interpretation

- Under $\sigma_n^2 = \lambda n$, the KRR predictor and the GP posterior mean are identical:

$$f_{\mathrm{KRR}}(x_*) = \mathrm{k}_*^\top (K + \lambda n I)^{-1} \mathrm{y} = \mathrm{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathrm{y} = m(x_*).$$

- Interpretation:
  - **KRR:** point estimator (regularized risk minimization), equivalently a **MAP** estimate under the GP model.
  - **GP:** full posterior over functions; beyond the mean, it provides uncertainty via the posterior covariance.

Table of Contents

Parameters/Hyperparameters

- The kernel function and mean function are the design variables of the GP model, which are characterized by hyperparameters.
- Hyperparameters:
  - Kernel parameters (θ): length-scale ℓ, signal variance σ_f^2, etc.
  - Noise variance σ_n^2.
  - For KRR: regularization λ.
- Question: **How do we choose them?**
  - GP: maximize the **marginal likelihood** $p(\mathrm{y} \mid X, \theta)$.
  - KRR: choose λ (and kernel parameters) via **cross-validation**.

Example: Squared Exponential (SE) Kernel

- Also known as Radial Basis Function (RBF) or Gaussian kernel.
- **Formula:**

$$k(\mathrm{x}_i, \mathrm{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathrm{x}_i - \mathrm{x}_j\|^2\right)$$

- **Hyperparameters:**
  - σ_f^2 (signal variance): Controls the overall amplitude or vertical scale of the function. A larger σ_f^2 allows for larger deviations from the mean.
  - ℓ (length-scale): Determines how quickly the function varies with input distance. A small ℓ leads to a very wiggly function, while a large ℓ results in a smooth function.

GP: Marginal Likelihood

- Under the GP model, we have

$$\mathrm{y} \sim \mathcal{N}(0, K_y).$$

- The log marginal likelihood is

$$\log p(\mathrm{y} \mid X, \theta) = -\frac{1}{2}\mathrm{y}^\top K_y^{-1}\mathrm{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi.$$

- Interpretation:
  - First term: data fit (quadratic form).
  - Second term: complexity penalty (Occam's razor).
  - Third term: normalization.
- We choose θ by (local) maximization of this likelihood, just like a usual Maximum Likelihood Estimation (MLE) problem.

KRR: Cross-Validation

- In GP, since we have an explicit probability distribution, we were able to obtain the optimal hyperparameters by maximizing the marginal likelihood.
- For KRR, we can only treat λ (and other kernel parameters) as tuning parameters.
- Standard approach: **K-fold cross-validation**.
  - Split the data into K folds.
  - For each candidate λ,
    - train on K-1 folds,
    - evaluate prediction error (e.g. MSE) on the held-out fold.
  - Choose λ that minimizes average validation error.

Does Cross-Validation Make Sense for GPs?

- Yes: we can also tune GP hyperparameters by cross-validation:
  - e.g. minimize validation **mean squared error**,
  - or minimize **negative log predictive density**.
- But:
  - CV is used with KRR because it helps prevent overfitting.
  - For GPs, hyperparameters are selected using the marginal likelihood since the GP models a full probability distribution, which naturally restricts overfitting. The marginal likelihood also provides an inherent trade-off between data fit and model complexity.
  - Also, cross-validation is generally avoided for GPs because each GP fit is computationally expensive and requires inversions of large covariance matrices.
- In this project:
  - We use **marginal likelihood** as the primary criterion for GPs.
  - We also experiment with **CV for GPs** and compare to KRR + CV.

Table of Contents

Interpolation vs Extrapolation for GPs

- For stationary kernels (e.g. squared exponential) and zero mean, as x_* moves far from the training inputs:
  - the covariance vector $\mathrm{k}_* \to 0$,
  - the predictive mean $m(x_*) \to 0$ (prior mean),
  - the predictive variance $\mathrm{cov}(x_*) \to k_{**}$ (prior variance).
- So, far outside the data region, GP predictions are essentially just the prior and without strong prior knowledge such extrapolations are unreliable.
- In applications like long-term financial forecasting, this can be highly misleading unless the kernel encodes very strong and correct structure.

Implications for This Project

- We therefore **avoid time extrapolation** tasks such as:
  - predicting future stock prices far beyond observed dates,
  - extrapolating volatility surfaces far outside observed maturities.
- Instead, our dataset is **static and cross-sectional**:
  - Inputs: portfolio construction features at a fixed period.
  - Output: performance measure (normalized annual return).
- The GP and KRR are used for **interpolation** in the space of portfolio weights, where the model is better supported by data.

Table of Contents

Dataset: Portfolio Performance

- Each example corresponds to a **portfolio rule**.
- Total of $n = 63$ portfolios, input dimension $D = 6$.
- Input $x_i \in \mathbb{R}^6$:
  - weights on six stock-selection concepts (e.g. book value, profitability, sales value, momentum, size, market risk).
- Output y_i:
  - a scalar performance measure: **normalized annual return**.

Preprocessing and Splits

- Preprocessing:
  - Standardize the six input features using the training set.
  - Center and scale the target y using the training mean and standard deviation.
- Train / test split:
  - 44 training portfolios and 19 test portfolios (roughly 70% / 30%).
- Performance metrics:
  - Test mean squared error (MSE) on the original target scale.
  - Mean negative log predictive density (NLPD) for GP models.

Models Compared

- All models use the same **RBF kernel**

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right).$$

- **GP-ML**:
  - GP regression with RBF kernel,
  - length-scale ℓ, signal std. σ_f and noise std. σ_n fitted by marginal likelihood.
- **GP-CV**:
  - Same GP model,
  - hyperparameters selected by 5-fold CV on a grid (minimizing validation MSE in the standardized space).
- **KRR-CV**:
  - Kernel ridge regression with the same RBF kernel,
  - regularization λ, ℓ and σ_f tuned by 5-fold CV.

Table of Contents

Learned Hyperparameters (Standardized Space)

Model	ℓ	σ_f	σ_n / λ
GP-ML	2.81	1.26	$\sigma_n \approx 0.088$
GP-CV	3.31	1.28	$\sigma_n \approx 0.039$
KRR-CV	3.31	1.26	$\lambda \approx 4.0 \times 10^{-5}$

- GP-CV and KRR-CV select almost identical kernel parameters (ℓ, σ_f).
- For GP-CV, $\sigma_n^2 \approx 1.49 \times 10^{-3}$, so

$$\frac{\sigma_n^2}{n_{\text{train}}} \approx 3.4 \times 10^{-5} \approx \lambda_{\text{KRR}},$$

confirming the theoretical mapping $\sigma_n^2 \approx \lambda n$.

Quantitative Results on Test Set

Model	Test MSE	Mean NLPD	CV MSE (scaled y)
GP-ML	1.82×10^{-3}	-1.36	–
GP-CV	1.81×10^{-3}	1.58	1.70×10^{-1}
KRR-CV	1.76×10^{-3}	–	1.70×10^{-1}

- All three methods achieve **very similar** test MSE ($\approx 1.8 \times 10^{-3}$).
- GP-ML and GP-CV have almost identical MSE, but very different NLPD.
- KRR-CV matches the GP-CV CV error; it has no notion of predictive density.

Calibration: GP-ML vs GP-CV

- GP-ML:
  - $\sigma_n \approx 0.088$, mean NLPD ≈ -1.36.
  - Predictive variances are reasonably aligned with residuals.
- GP-CV:
  - CV (based on MSE only) pushes σ_n down to ≈ 0.039.
  - Mean NLPD rises to ≈ 1.58: the model is **overconfident**.
- **Takeaway:**
  - Cross-validation can give good point predictions for GPs,
  - but it may severely miscalibrate predictive uncertainties.
  - Marginal likelihood explicitly balances fit and complexity and yields better-calibrated GPs on this dataset.

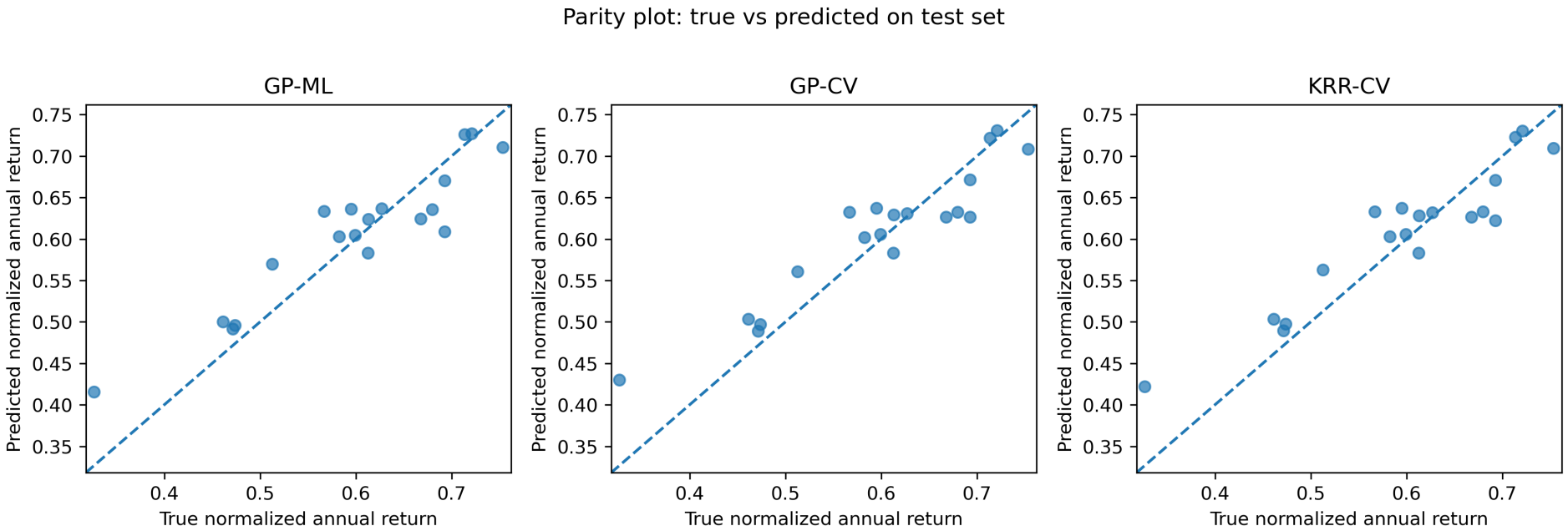
GP vs KRR Predictive Means

- Compare GP-ML posterior mean and KRR-CV predictor on the test set (standardized target):

$$\text{mean}\,|\Delta| \approx 2.17 \times 10^{-2},$$
$$\max|\Delta| \approx 9.31 \times 10^{-2}.$$

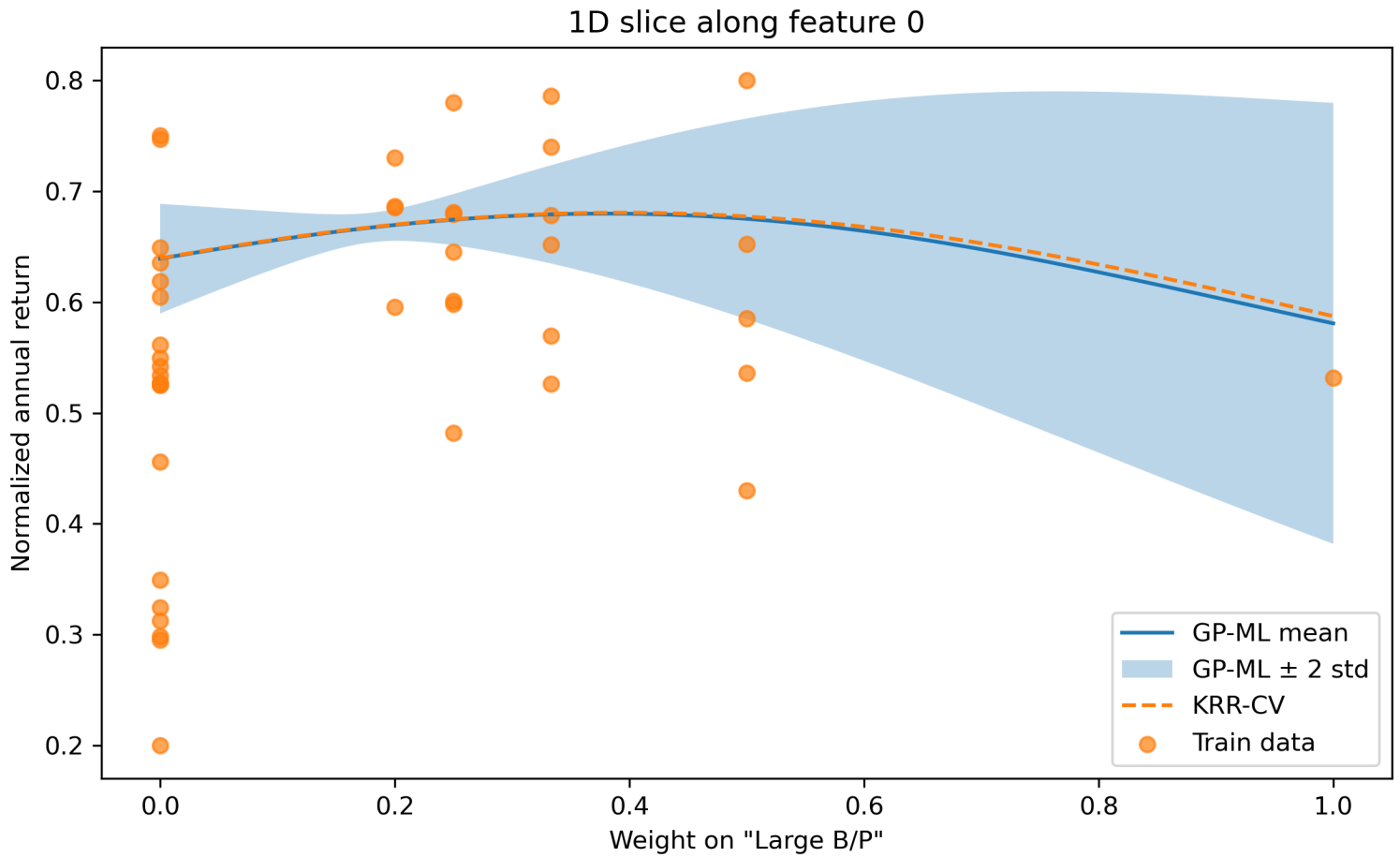
- Given that hyperparameters were tuned with different criteria, this difference is very small.
- When we plug the GP-CV noise level into the mapping $\lambda \approx \sigma_n^2/n$, KRR and GP-CV become almost **numerically indistinguishable**.

Parity Plot on Test Set



- True vs predicted normalized annual return for GP-ML, GP-CV, and KRR-CV.
- All models lie close to the diagonal, confirming similar test MSE.

1D Slice Along a Portfolio Weight



- One weight is varied; all other portfolio features are fixed at their mean.
- GP-ML mean (solid) and KRR-CV (dashed) almost coincide.
- Only the GP provides uncertainty bands, which widen away from dense data regions.

Table of Contents

Discussion

- **Predictive equivalence:**
  - GP posterior mean and KRR solution coincide up to the mapping $\sigma_n^2 \approx \lambda n$.
  - In our experiment, GP-CV and KRR-CV chose nearly identical ℓ, σ_f and consistent σ_n^2, λ.
- **Interpretation difference:**
  - GP: full probabilistic model over functions, uncertainty quantification, marginal likelihood.
  - KRR: deterministic regularization method, no built-in uncertainty.
- **Hyperparameters:**
  - GP-ML chooses a larger noise level and achieves much better NLPD than GP-CV, while keeping MSE essentially unchanged.
  - KRR-CV attains slightly smaller MSE but cannot assess calibration.

Conclusion

- We compared **Gaussian process regression** and **kernel ridge regression**:
  - same kernel, same data,
  - theoretically equivalent predictors (posterior mean vs KRR solution),
  - different viewpoints and hyperparameter selection strategies.
- Main takeaways from the experiment:
  - All three models achieve almost identical test MSE on the portfolio dataset.
  - GP-ML provides substantially better-calibrated predictive uncertainties than GP-CV.
  - KRR-CV chooses a regularization parameter consistent with the GP noise level via $\lambda \approx \sigma_n^2/n$, empirically confirming the theory.
  - GP regression can therefore be seen as **Bayesian KRR** with the added benefit of uncertainty quantification and a principled marginal-likelihood criterion for tuning.

References

C. E. Rasmussen and C. K. I. Williams,
Gaussian Processes for Machine Learning,
MIT Press, 2006.

Course notes,
MATH-412: Kernel Methods (Lecture 7b).