Gaussian Processes and Kernel Ridge Regression

Theoretical Connections, Hyperparameter Tuning, and Experiments

Leonardo Cartesegna, Chandrasekhara Devarakonda, Giulia Scagliarini

MATH-412: Statistical Machine Learning

December 9, 2025

Outline

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

Table of Contents

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

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

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

Supervised Regression Setup

Following Rasmussen & Williams:

- Training set

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

- Inputs (covariates):
  - collected in the $D \times n$ design matrix X.
- Targets:

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

- Goal: for a new input x_*, infer its output y_* or latent function value $f(x_*)$.

Noise Model

- Latent function f and noisy observations:

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

- We are primarily interested in learning f (or $f(x_*)$) from $\mathcal{D}$.
- Two perspectives:
  - **Gaussian processes:** place a prior distribution over functions f.
  - **Kernel ridge regression:** minimize a regularized empirical risk in an RKHS.

Table of Contents

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

Gaussian Process Prior

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

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

- In this project, we use zero mean:

$$m(x) = 0,$$

and a kernel $k_\theta(x, x')$ 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:
$$p(\mathrm{y} \mid \mathrm{f}) = \mathcal{N}(\mathrm{y} \mid \mathrm{f}, \sigma_n^2 I).$$
- 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

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

- **Predictive mean** $m(x_*)$ is our best point prediction under squared loss.
- **Predictive variance** $\mathrm{cov}(x_*)$ quantifies uncertainty:
  - small near many training points,
  - reverts to prior variance far from data.

Table of Contents

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

Regularization in RKHS

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

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

- $\lambda > 0$ controls the balance between data fit and function smoothness (complexity).

Representer Theorem and Finite-Dimensional Form

- 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

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

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

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

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 mean:

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

Table of Contents

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. **Theoretical Connection**
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

Equivalence of Predictors

- If we identify

$$\sigma_n^2 \;\leftrightarrow\; \lambda n,$$

  then the GP predictive mean and the KRR predictor have the **same functional form**:

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

- Interpretation:
  - GP regression: Bayesian posterior mean under a Gaussian process prior.
  - KRR: unique minimizer of a regularized empirical risk in $\mathcal{H}_k$.
- Thus GP regression with Gaussian noise is equivalent to KRR at the level of the predictive mean.

Functional View of GP Regression

- GP regression can be characterized as the minimizer of

$$J[f] = \frac{1}{2}\|f\|^2_{\mathcal{H}_k} + \frac{1}{2\sigma_n^2}\sum_{i=1}^{n}(y_i - f(x_i))^2.$$

- This is exactly the same functional as KRR, with

$$\lambda = \frac{1}{\sigma_n^2}.$$

- Difference:
    - In KRR we only care about the *optimizer f*.
    - In GPs we also keep the *full posterior distribution* over functions, including predictive variances.

Table of Contents

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

Hyperparameters

- Hyperparameters (denote generically by θ):
  - Kernel parameters: length-scales, signal variance, 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**.

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 quantity.

KRR: Cross-Validation

- For KRR, we treat λ (and possibly 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.
- We use the same grid-search CV procedure for GPs (fixing θ on a grid and selecting the minimum validation MSE) to compare with marginal likelihood.

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:
  - GPs already have a natural **Bayesian objective**: the marginal likelihood.
  - Marginal likelihood explicitly trades off fit and model complexity.
- 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

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

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.
- 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 primarily for **interpolation** in the space of portfolio weights, where the model is better supported by data.

Table of Contents

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

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. value, profitability, size, past returns, market value, 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

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

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.

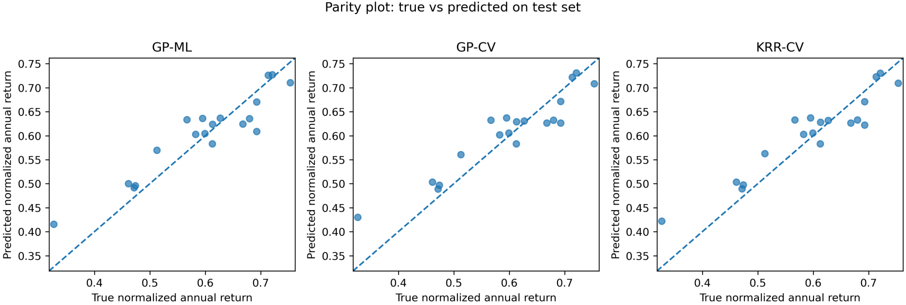
GP vs KRR Predictive Means

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

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

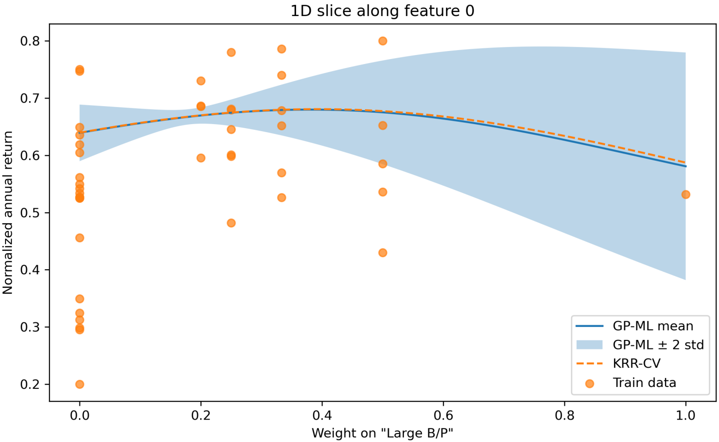
- Given that hyperparameters were tuned with different criteria, this difference is very small.
- When we plug the GP-CV noise level into the mapping $\lambda \approx \sigma_n^2/n$, KRR and GP-CV become almost **numerically indistinguishable**.

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

1. Motivation and Project Goals
2. Regression Setup and Notation
3. Gaussian Process Regression
4. Kernel Ridge Regression
5. Theoretical Connection
6. Hyperparameter Selection
7. Interpolation vs Extrapolation
8. Data and Experimental Design
9. Results
10. Discussion and Conclusion

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.

Interpolation vs Extrapolation Revisited

- This project **respects the GP assumptions**:
  - Static, cross-sectional regression,
  - Inputs primarily in a region supported by data.
- For extrapolation (e.g. far outside observed portfolio weights or long-term future):
  - GP predictions revert to prior assumptions encoded in the kernel,
  - without strong prior knowledge, such extrapolations are unreliable.
- Our results therefore illustrate GPs and KRR in a regime where they are actually designed to work well: **interpolation**.

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).