Gaussian Process

By: Cameron Scolari, Jack Seymour, & Maria Dominguez

Learning Objectives

1	What is a Gaussian Process (GP)?
2	Advantages & Disadvantages of GPs
3	Kernels
4	Gaussian Process Regression (GPR)
5	Gaussian Process Regression (GPC)

What is a Gaussian Process?

Gaussian Processes are a type of supervised machine learning algorithm that are used to solve regression and probabilistic classification problems.

Some characteristics of GPs are:

- probabilistic predictions
- non-parametric nature
- smoothing and interpolation
- marginalization of parameters

Strengths and Limitations

Advantages	Disadvantages
The prediction interpolates the observations	They use the whole samples to perform the prediction, so the implementation is not sparse
The prediction is probabilistic, so you can calculate the empirical confidence intervals and decide whether to refit the prediction in a specific region	These algorithms lose efficiency in high dimensional spaces
Different kernels can be specified, so it is versatile	Understanding exact relationship between input and output can be difficult with complex kernel functions

Kernels

- Kernels define an "infinite" number of prior functions by describing the covariance between any pair of points in the input space (allows for a continuous function to be modeled with a high degree of flexibility across the entire domain)
- GPR uses kernels (e.g., RBF, Matern) to measure similarity between points
- The kernel choice heavily influences model performance and the smoothness of predictions
- Kernel hyperparameters are optimized during training

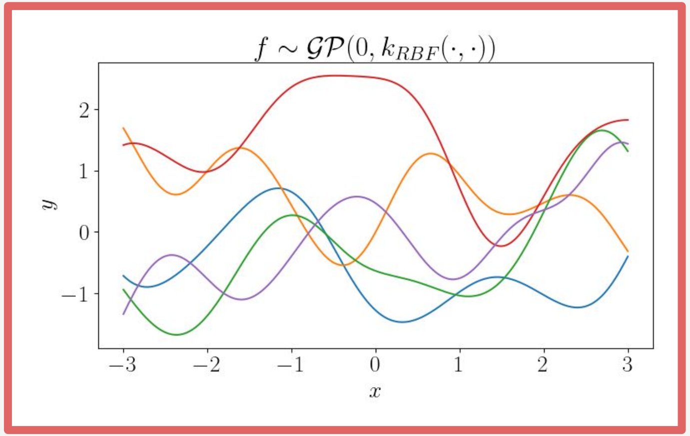
Kernel Comparison

Additional parameter v, controls smoothness

Infinitely smooth

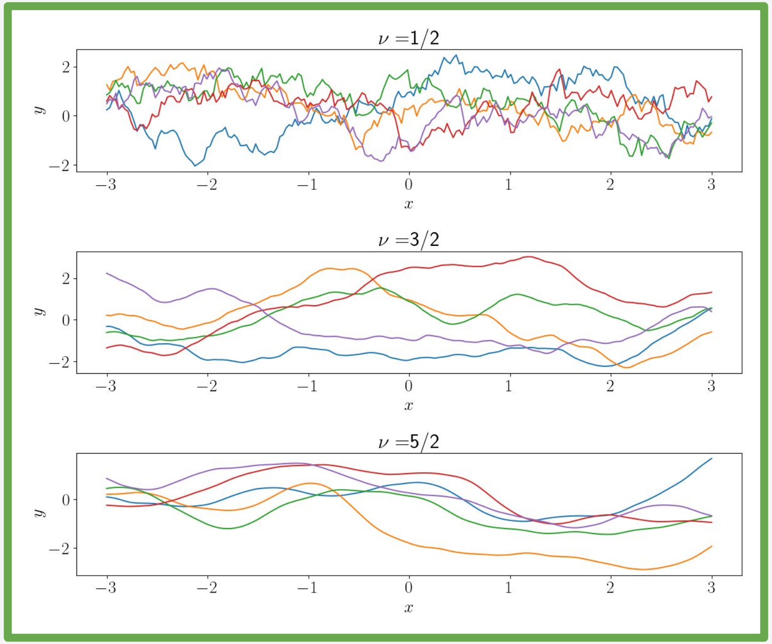
For any n >= 0 , the n-th derivative of the RBF kernel with respect to x exists and is continuous.

RBF



$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$$

Matern

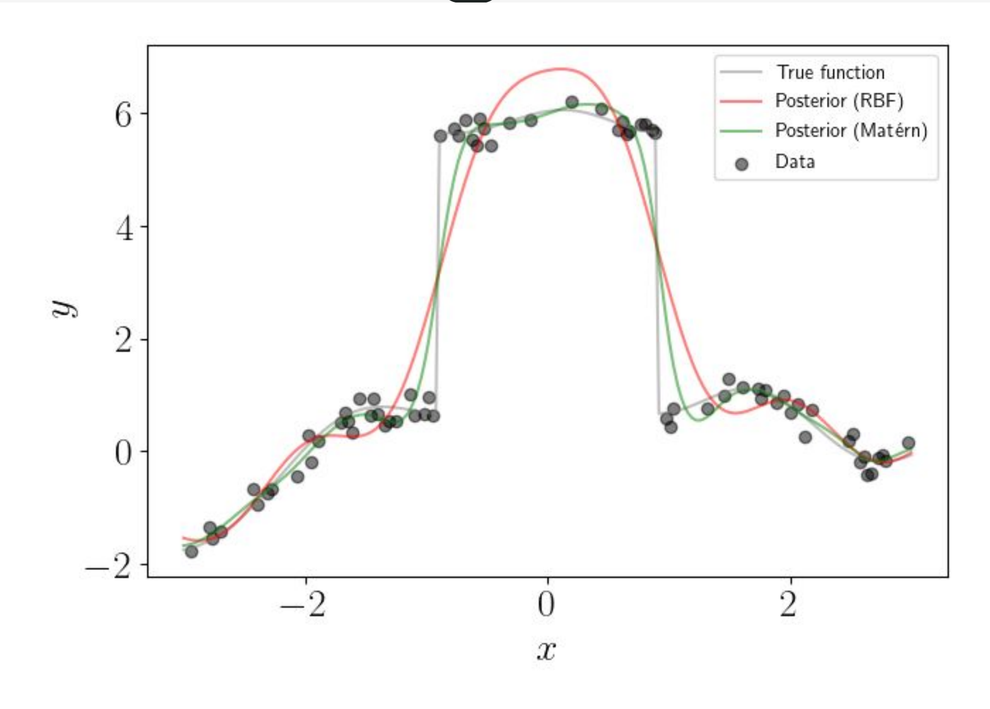


Kernel Smoothing

True

RBF

Matern



Gaussian Process Regression

- Non-parametric model: Does not know functional relationship between the dependent and independent variables.
- Prior Assumption: Any smooth function could represent relationship between data
- Noise optimization: Optimizes noise hyperparameter during training using MLE
- Fitting the data: After seeing data, GPR fits the points but remains smooth, even in regions with noise.
- Probabilistic predictions: Provides the mean and standard deviation, capturing uncertainty

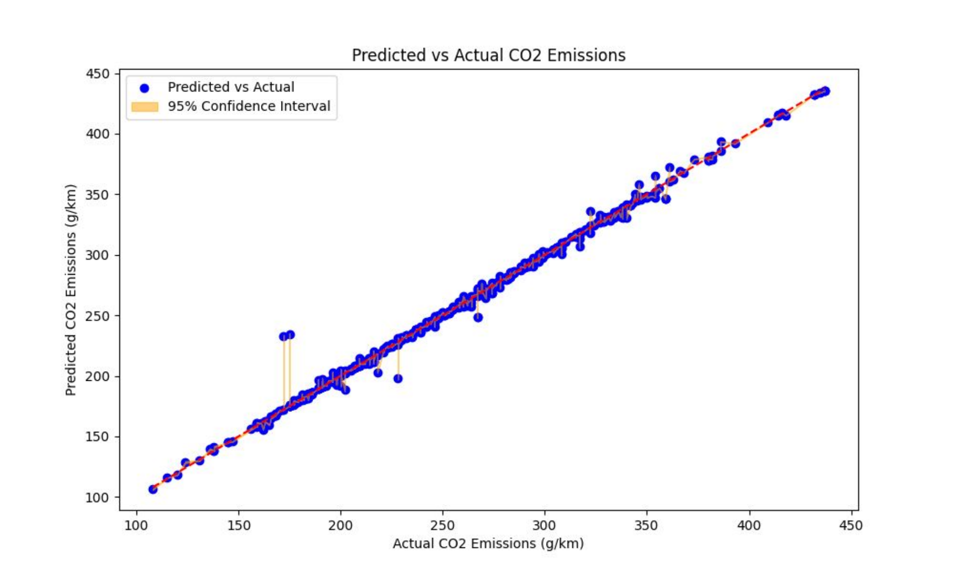
```
param_distributions = {
    'kernel': [RBF(), Matern()],
    'alpha': [1, 1e-1, 1e-2, 1e-3, 1e-4]
}

random_search = RandomizedSearchCV(
    estimator=gpr,
    param_distributions=param_distributions,
    n_iter=10,
    cv=3,
    random_state=42,
    scoring='r2'
)

random_search.fit(X_train, y_train)

gpr = random_search.best_estimator_
```

Kernel and Alpha Optimization

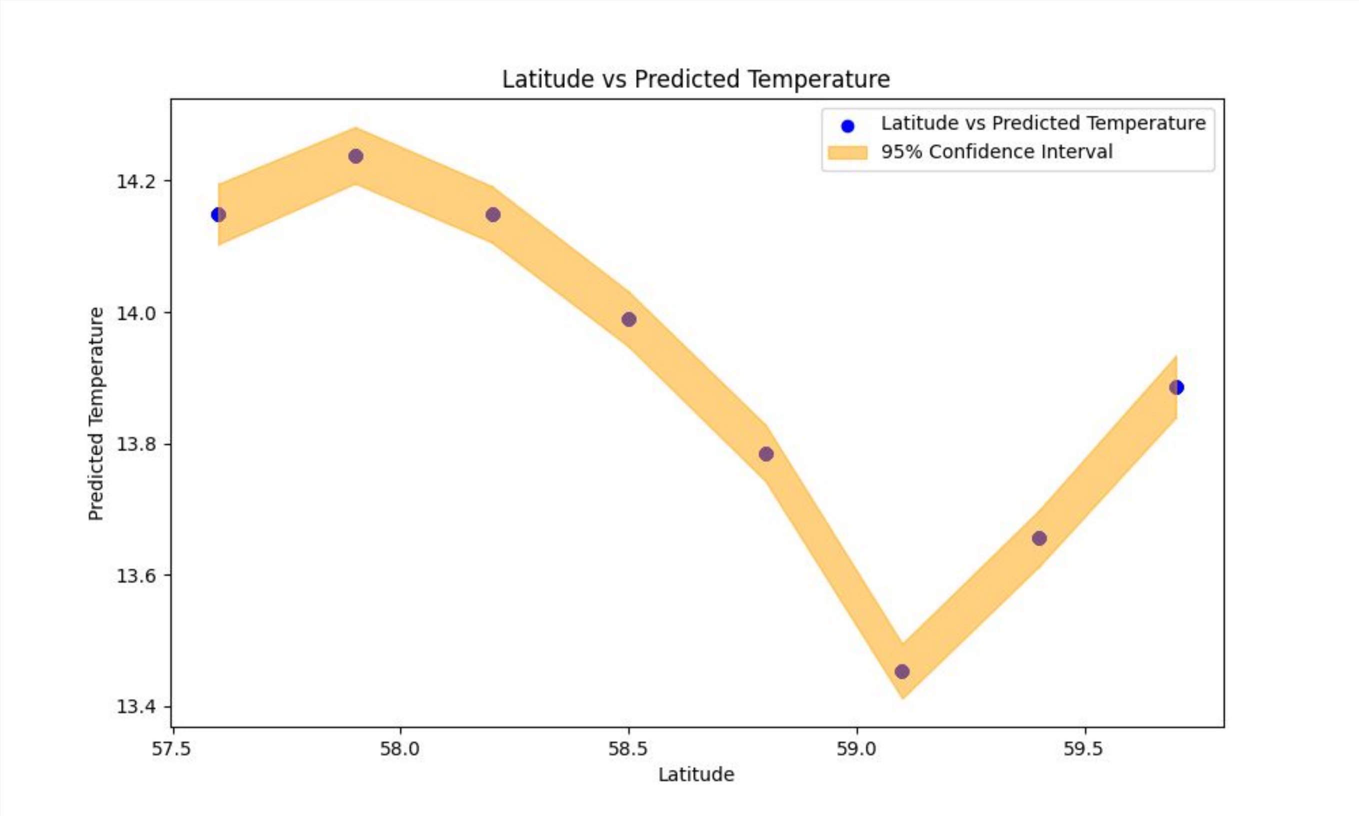
- RBF assumes smooth true function
- Matern allows for less smooth true function
- Both balances smoothness and flexibility

GPR Example on Data

- Kaggle Dataset of Observations of Cars and CO2 emissions
  - Includes features like make, model, vehicle class, engine size, fuel consumption
  - 10,000 observations but we sampled only 2,000
- 99.45% R^2 value on test data but only 9% of values fall within 95% CI
- This data has categorical variables so GPR may not be the best



Better Graph



Gaussian Process Classification

Uncertainty Quantification

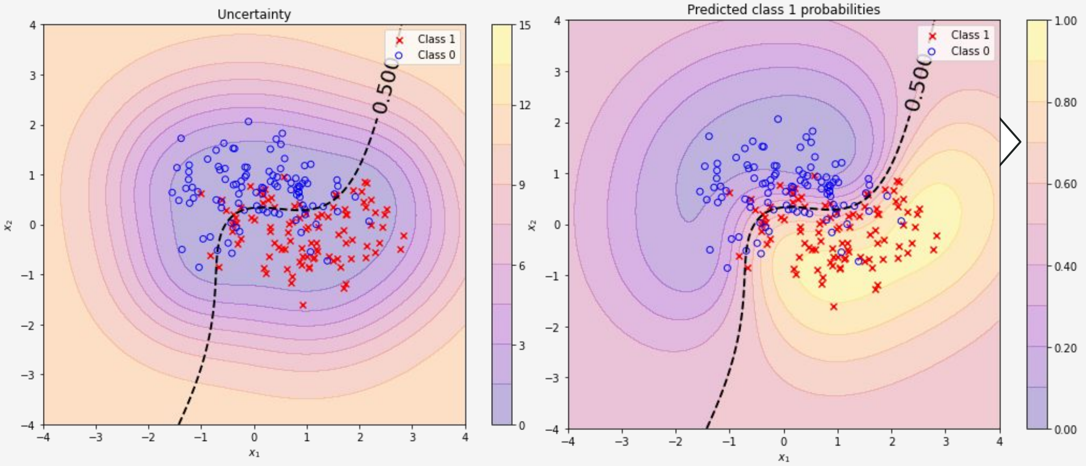
GPC provides uncertainty estimates for predictions, helping identify low-certainty classifications that may need further review.

Allows Flexibility

With GPC, a direct classification is not made. This allows for further flexibility without feature engineering.

How is output interpreted?

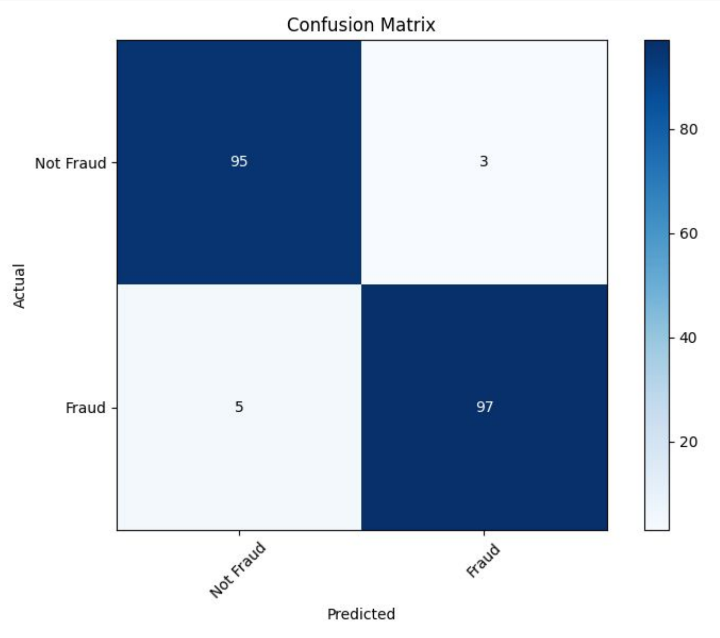
With GPC, curves drawn are thought of as “guesses” based on uncertainty.



GPC Example: Kaggle Credit Card Fraud Detection Dataset

.csv of:

- **V1-V28:** Anonymized features representing various transaction attributes (e.g., time, location, etc.)
- **Amount:** The transaction amount
- **Class:** Binary label indicating whether the transaction is fraudulent (1) or not (0)



Challenge: fraud data is imbalanced. In our data set had 284,807 total transactions, 492 were fraudulent

In credit card fraud detection,
False positives can be inconvenient to customers
E.g. manual verification from bank

False negatives can be costly to the credit card user
E.g. customer/bank loses large sum of money

Threshold Trade Off

Precision VS Recall

Precision

Out of all flagged transactions, how many are truly fraudulent?

High Precision **reduces false positives**, but may miss some fraudulent cases

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

Out of all actual fraudulent transactions, how many were detected?

High recall **reduces false negatives** but may increase false positives

$$\text{Recall} = \frac{TP}{TP + FN}$$

For credit card fraud example, it may be advantageous to focus on <u>Recall</u>

Probabilistic Outputs

With a given input *x*, it computes:

P(y=1 | x, data): Probability that the transaction is fraudulent

***P(y=0 | x, data)*:** Probability that the transaction is legitimate

<u>Example thresholds by applying this model:</u>

High Certainty: *P(y = 1) > 0.8*: High confidence fraud; immediately block transaction

Medium Certainty: *0.5 < P(y=1) < 0.8*: Medium confidence fraud; flag for manual review

Low Certainty: *P(y=1) < 0.5*: Likely legitimate; do not block transaction. Log if p(y=1) > 0.3.

Even if classified as legitimate, a low certainty score might warrant further review. The model can be **fine-tuned** for individuals more susceptible to fraud, considering factors like identity theft history or high-profile status.

Example

P(y=1)	Possible classification	Certainty
0.1	No	High
0.4	No (log)	Medium
0.95	Yes	High
0.52	Yes	Low

Flagged as Fraud with low certainty
Bank can use this information & further investigate

Bibliography

- David Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012. http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/200620.pdf
- https://scikit-learn.org/1.5/modules/gaussian_process.html
- https://www.geeksforgeeks.org/gaussian-process-regression-gpr/
- https://andrewcharlesjones.github.io/journal/matern-kernels.html

Thanks!

Any questions?