Machine Learning
MSE FTP MachLe
Christoph Würsch

OST
Ostschweizer
Fachhochschule

Gaussian process regression (GPR) on Mauna Loa $\mathrm{CO_2}$ data.

This example is based on Section 5.4.3 of Gaussian Processes for Machine Learning.

- It illustrates an example of complex kernel engineering and **hyperparameter optimization using gradient ascent on the log-marginal-likelihood**.
- The data consists of the monthly average atmospheric $\mathrm{CO_2}$ concentrations (in parts per million by volume (ppmv)) collected at the **Mauna Loa Observatory in Hawaii**, between 1958 and 2001.

The objective is to model the CO2 concentration as a function of the time t.

In [1]:
```
%matplotlib inline
#%matplotlib notebook
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
```

The kernel is composed of several terms that are responsible for explaining different properties of the signal:

- **K1**: a long term, smooth rising trend is to be explained by an `RBF kernel`. The `RBF kernel` with a large length-scale enforces this component to be smooth; it is not enforced that the trend is rising which leaves this choice to the GP. The specific length-scale and the amplitude are free hyperparameters.

- **K2**: a seasonal component, which is to be explained by the periodic `ExpSineSquared kernel` with a fixed periodicity of 1 year. The length-scale of this periodic component, controlling its smoothness, is a free parameter. In order to allow decaying away from exact periodicity, the product with an RBF kernel is taken. The length-scale of this RBF component controls the decay time and is a further free parameter.

- **K3**: smaller, medium term irregularities are to be explained by a `RationalQuadratic` kernel component, whose length-scale and alpha parameter, which determines the diffuseness of the length-scales, are to be determined. According to RW2006, these irregularities can better be explained by a RationalQuadratic than an RBF kernel component, probably because it can accommodate several length-scales.

- **K4** :a "noise" term, consisting of an `RBF kernel` contribution, which shall explain the correlated noise components such as local weather phenomena, and a `WhiteKernel` contribution for the white noise. The relative amplitudes and the RBF's length scale are further free parameters.

Maximizing the log-marginal-likelihood after subtracting the target's mean yields the following kernel with an LML of -83.214::

34.4**2 * RBF(length_scale=41.8) + 3.27**2 * RBF(length_scale=180) * ExpSineSquared(length_scale=1.44, periodicity=1) + 0.446**2 * RationalQuadratic(alpha=17.7, length_scale=0.957) + 0.197**2 * RBF(length_scale=0.138) + WhiteKernel(noise_level=0.0336)

- Thus, most of the target signal (34.4ppm) is explained by a long-term rising trend (length-scale 41.8 years). The periodic component has an amplitude of 3.27ppm, a decay time of 180 years and a length-scale of 1.44.
- The long decay time indicates that we have a locally very close to periodic seasonal component.
- The correlated noise has an amplitude of 0.197ppm with a length scale of 0.138 years and a white-noise contribution of 0.197ppm. Thus, the overall noise level is very small, indicating that the data can be very well explained by the model.
- The figure shows also that the model makes very confident predictions until around 2015.

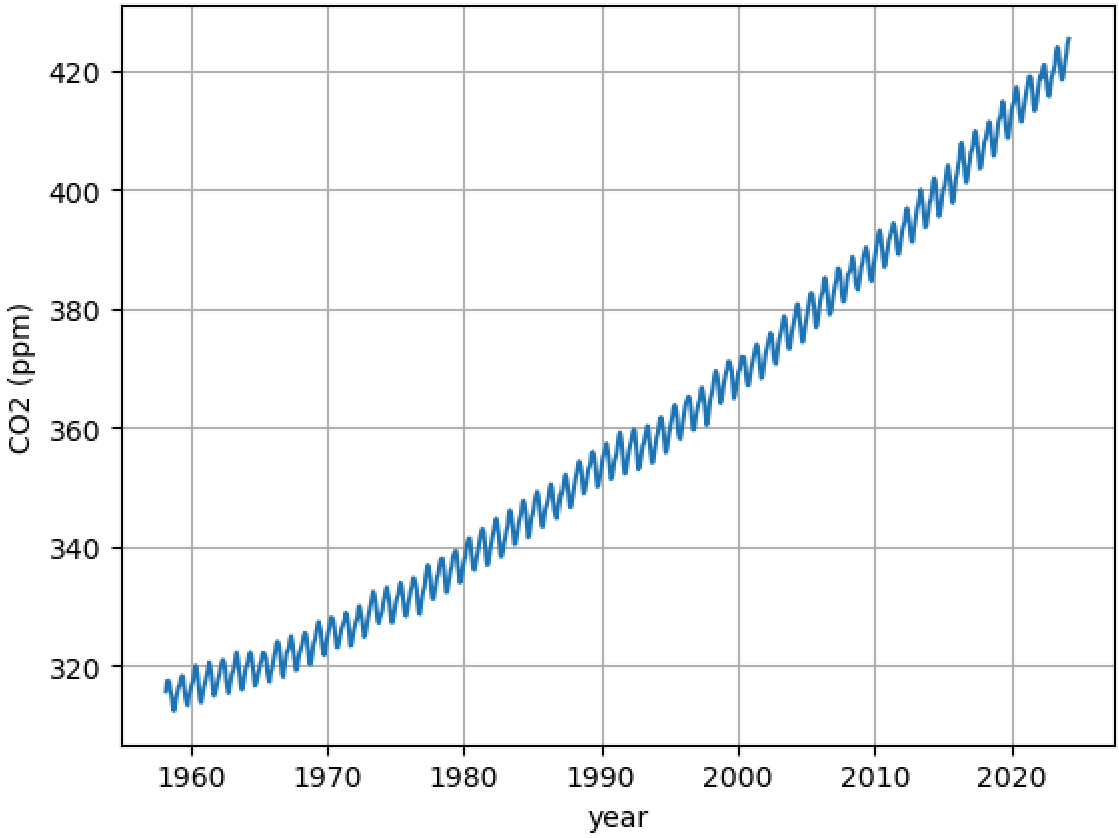
In [15]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Load the CSV file
df = pd.read_csv('co2_mm_mlo.csv', parse_dates=[0])

# Assuming the first column is the date column and you want it as the DataFrame inc
df.set_index(df.columns[0], inplace=True)

df.head()
```

Out[15]:

	month	decimal date	average	deseasonalized	ndays	sdev	unc
year							
1958-01-01	3	1958.2027	315.70	314.43	-1	-9.99	-0.99
1958-01-01	4	1958.2877	317.45	315.16	-1	-9.99	-0.99
1958-01-01	5	1958.3699	317.51	314.71	-1	-9.99	-0.99
1958-01-01	6	1958.4548	317.24	315.14	-1	-9.99	-0.99
1958-01-01	7	1958.5370	315.86	315.18	-1	-9.99	-0.99

In [21]:
```
X=df['decimal date'].values.reshape(-1,1)
y=df['average'].values.reshape(-1,1)

plt.plot(X,y)
plt.grid(True)
plt.xlabel('year')
plt.ylabel('CO2 (ppm)')
```

Out[21]:
```
Text(0, 0.5, 'CO2 (ppm)')
```



In [22]:
```
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF, WhiteKernel, RationalQuadratic, E
```

In [23]:
```
# Kernel with parameters given in GPML book
k1 = 66.0**2 * RBF(length_scale=67.0)  # long term smooth rising trend
k2 = 2.4**2 * RBF(length_scale=90.0) \
    * ExpSineSquared(length_scale=1.3, periodicity=1.0)  # seasonal component
# medium term irregularity
k3 = 0.66**2 \
    * RationalQuadratic(length_scale=1.2, alpha=0.78)
k4 = 0.18**2 * RBF(length_scale=0.134) \
    + WhiteKernel(noise_level=0.19**2)  # noise terms
kernel_gpml = k1 + k2 + k3 + k4
```

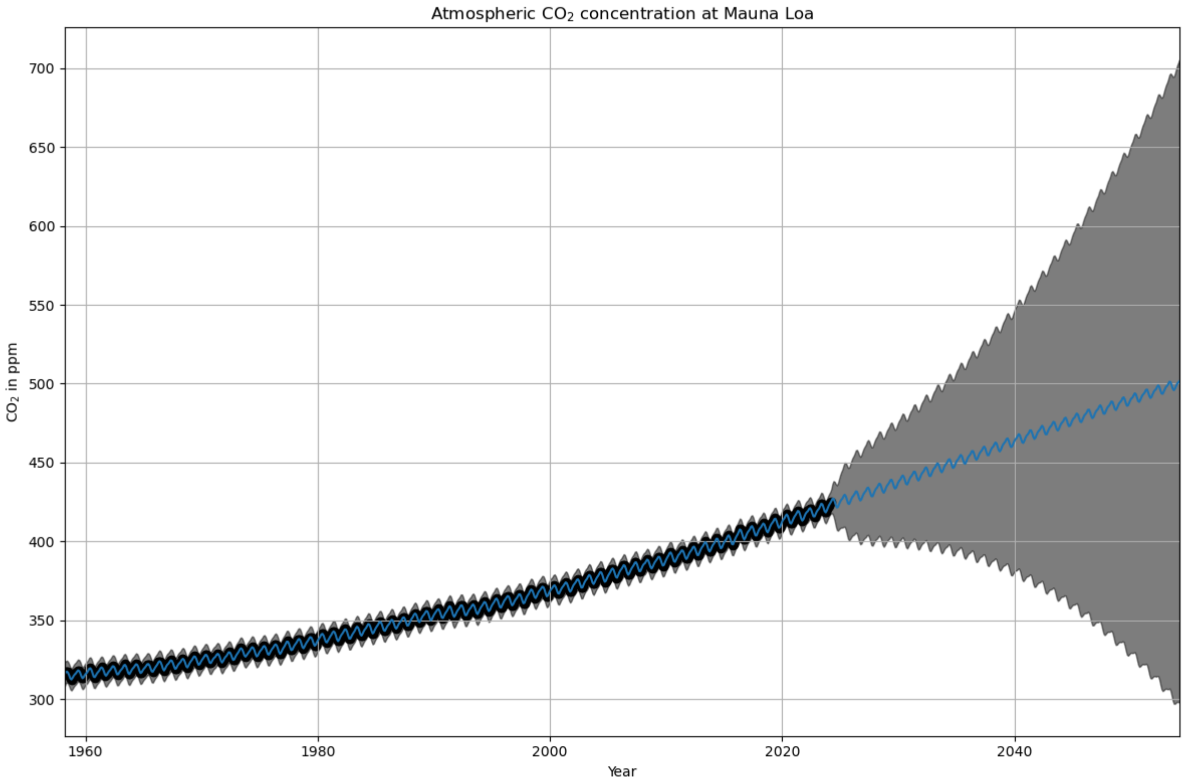
In [24]:
```
gp = GaussianProcessRegressor(kernel=kernel_gpml, alpha=0,
                              optimizer=None, normalize_y=True)
gp.fit(X, y)

print("GPML kernel: %s" % gp.kernel_)
print("Log-marginal-likelihood: %.3f"
      % gp.log_marginal_likelihood(gp.kernel_.theta))
```

```
GPML kernel: 66**2 * RBF(length_scale=67) + 2.4**2 * RBF(length_scale=90) * ExpSin
eSquared(length_scale=1.3, periodicity=1) + 0.66**2 * RationalQuadratic(alpha=0.7
8, length_scale=1.2) + 0.18**2 * RBF(length_scale=0.134) + WhiteKernel(noise_level
=0.0361)
Log-marginal-likelihood: 252.055
```

In [25]:
```
X_ = np.linspace(X.min(), X.max() + 30, 1000)[:, np.newaxis]
y_pred, y_std = gp.predict(X_, return_std=True)
```

In [26]:
```
# Illustration
plt.figure(figsize=(12,8))
plt.scatter(X, y, c='k')
plt.plot(X_, y_pred)
```

```
plt.fill_between(X_[:, 0], y_pred - y_std, y_pred + y_std, alpha=0.5, color='k')
plt.xlim(X_.min(), X_.max())
plt.xlabel("Year"); plt.ylabel(r"CO$_2$ in ppm")
plt.title(r"Atmospheric CO$_2$ concentration at Mauna Loa")
plt.tight_layout(); plt.grid(True); plt.show()
```



In [27]:

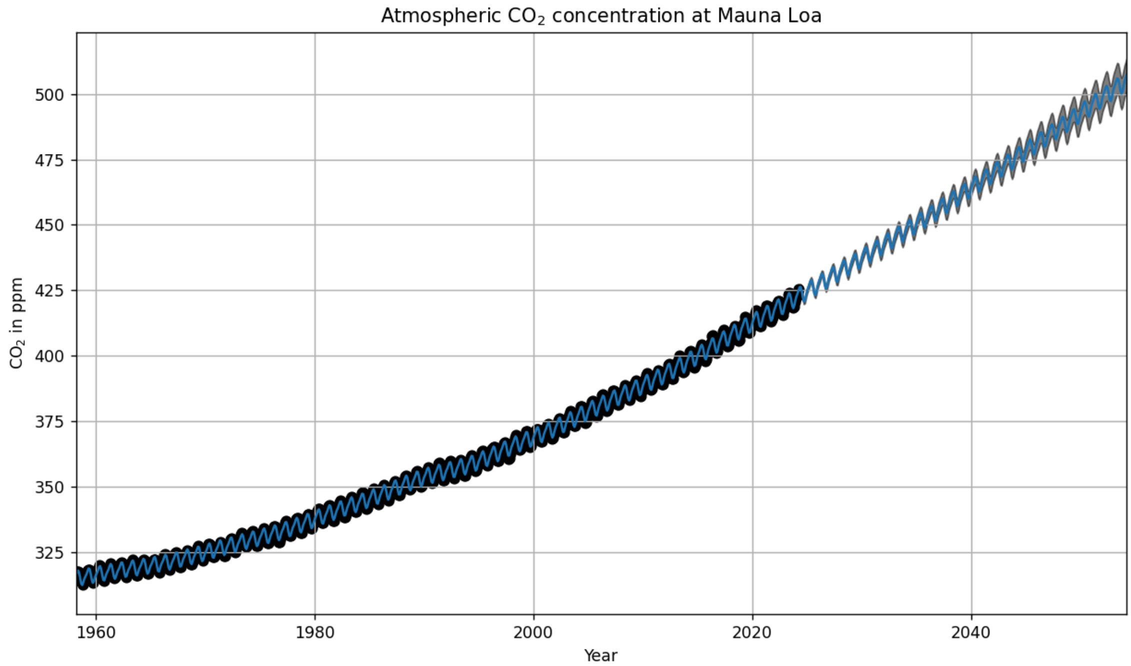
```
# Kernel with optimized parameters
k1 = 50.0**2 * RBF(length_scale=50.0)  # long term smooth rising trend
k2 = 2.0**2 * RBF(length_scale=100.0) \
    * ExpSineSquared(length_scale=1.0, periodicity=1.0,
                     periodicity_bounds="fixed")  # seasonal component
# medium term irregularities
k3 = 0.5**2 * RationalQuadratic(length_scale=1.0, alpha=1.0)
k4 = 0.1**2 * RBF(length_scale=0.1) \
    + WhiteKernel(noise_level=0.1**2,
                  noise_level_bounds=(1e-3, np.inf))  # noise terms
kernel = k1 + k2 + k3 + k4

gp = GaussianProcessRegressor(kernel=kernel, alpha=0,
                              normalize_y=True)
gp.fit(X, y)

print("\nLearned kernel: %s" % gp.kernel_)
print("Log-marginal-likelihood: %.3f"
      % gp.log_marginal_likelihood(gp.kernel_.theta))

X_ = np.linspace(X.min(), X.max() + 30, 1000)[:, np.newaxis]
y_pred, y_std = gp.predict(X_, return_std=True)
```

```
C:\Users\christoph.wuersch\.conda\envs\ML\lib\site-packages\sklearn\gaussian_proce
ss\kernels.py:420: ConvergenceWarning: The optimal value found for dimension 0 of
parameter k2__k2__noise_level is close to the specified lower bound 0.001. Decreas
ing the bound and calling fit again may find a better value.
  warnings.warn(
```

```
Learned kernel: 7.94**2 * RBF(length_scale=131) + 0.223**2 * RBF(length_scale=331)
* ExpSineSquared(length_scale=2.78, periodicity=1) + 0.104**2 * RationalQuadratic
(alpha=4.58, length_scale=133) + 0.0245**2 * RBF(length_scale=3.22) + WhiteKernel
(noise_level=0.001)
Log-marginal-likelihood: 1878.694
```

In [28]:

```
%matplotlib notebook

plt.figure(figsize=(10,6))
plt.scatter(X, y, c='k')
plt.plot(X_, y_pred)
plt.fill_between(X_[:, 0], y_pred - y_std, y_pred + y_std, alpha=0.5, color='k')
plt.xlim(X_.min(), X_.max()); plt.xlabel("Year")
plt.ylabel(r"CO$_2$ in ppm"); plt.title(r"Atmospheric CO$_2$ concentration at Mauna
plt.tight_layout(); plt.grid(True); plt.show()
```



In [ ]: