Scaling Comparisons for the filtering algorithms

Evan Tate Paterson Hughes

Table of contents

Currently, the Kalman filter, the Information filter, the Square-root filter, and the square-root-information filter are all implemented.

We will use a simulated data set, from cosine basis functions over 100 frequencies and invariant kernel. See figure (?) for the 6 time points of the simulated process.

We then filter this data, and time the implemented filters to compute the marginal log likelihood. We run the alogrithms with a sequence of model parameters that are close to the ones used to simulate the data, with added noise to avoid the memoisation the JAX uses.

We simulate $n = 300$ observations per time point in total, and redact observations in order to show the compute times for each n between 50 and 300. These are computed in 64 bit for stability, and the results are in figure (?)

Additionally, to demonstrate the square root filters stability, we also fix n and progressively increase n. The data is simulated from a high-dimensional process basis, and then filtered using models with a lower r, increasing in square between 5^2 and 15^2. The results are shown in figure (?).

```
import jax
import jax.numpy as jnp
import pandas as pd
import plotly.express as px

df = pd.read_csv('data/varying_n_32.csv')

fig = px.line(
    df,
    x="n",  # X-axis: values of n
    y=df.columns[3:7],  # Y-axis: all time columns
    labels={"n": "number of observations", "value": "Time (seconds)"},
    title="Filters Comparison with Varying n (32 bit)"
)

# Show the plot
fig.write_image("figure/scale_test_n_32.png")
fig.show()
```

```
Unable to display output for mime type(s): text/html
```

```
Unable to display output for mime type(s): text/html
```

```
import jax
import jax.numpy as jnp
import pandas as pd
import plotly.express as px

df = pd.read_csv('data/varying_n_64.csv')

fig = px.line(
    df,
    x="n",
    y=df.columns[3:7],
    labels={"n": "number of observations", "value": "Time (seconds)"},
    title="Average compute time Filters with Varying n (64 bit)"
)

# Show the plot
fig.write_image("figure/scale_test_n_64.png")
fig.show()
```

```
Unable to display output for mime type(s): text/html
```

```
import jax
import jax.numpy as jnp
import pandas as pd
import plotly.express as px

df = pd.read_csv('data/varying_r_32.csv')

fig = px.line(
    df,
    x="r",
    y=df.columns[7:],
    labels={"n": "number of observations", "value": "Time (seconds)"},
    title="Compuile Time of Filters Varying r (32 bit)"
)

# Show the plot
fig.write_image("figure/scale_test_r_32.png")
fig.show()
```

```
Unable to display output for mime type(s): text/html
```