

CombinedBaselines

June 9, 2024

```
[8]: import numpy as np
import torch
from torch.utils.data import DataLoader
import matplotlib.pyplot as plt
from FunctionDataset import FunctionDataset
from LSGDBaseline import LSGDBaseline
from LeastSquaresBaseline import LeastSquaresBaseline
from AveragingBaseline import AveragingBaseline
from KNNBaseline import KNNBaseline

def generate_linear_prompts(num_samples, input_dim, noise_std=0.1):
    dataset = FunctionDataset(num_samples=num_samples, input_dim=input_dim,
    ↪function_class='linear', noise_std=noise_std)
    dataloader = DataLoader(dataset, batch_size=num_samples, shuffle=False)
    for data in dataloader:
        inputs, targets = data
        return inputs.numpy(), targets.numpy()

if __name__ == "__main__":
    # Generate data
    num_samples = 40
    input_dim = 20
    X, y = generate_linear_prompts(num_samples, input_dim)

    # List of models to test
    models = {
        'LSGDBaseline': LSGDBaseline(input_dim, learning_rate=0.02),
        'AveragingBaseline': AveragingBaseline(),
        'KNNBaseline': KNNBaseline(k=3),
        'LeaseSquaresBaseline': LeastSquaresBaseline(input_dim)
    }

    results = {}

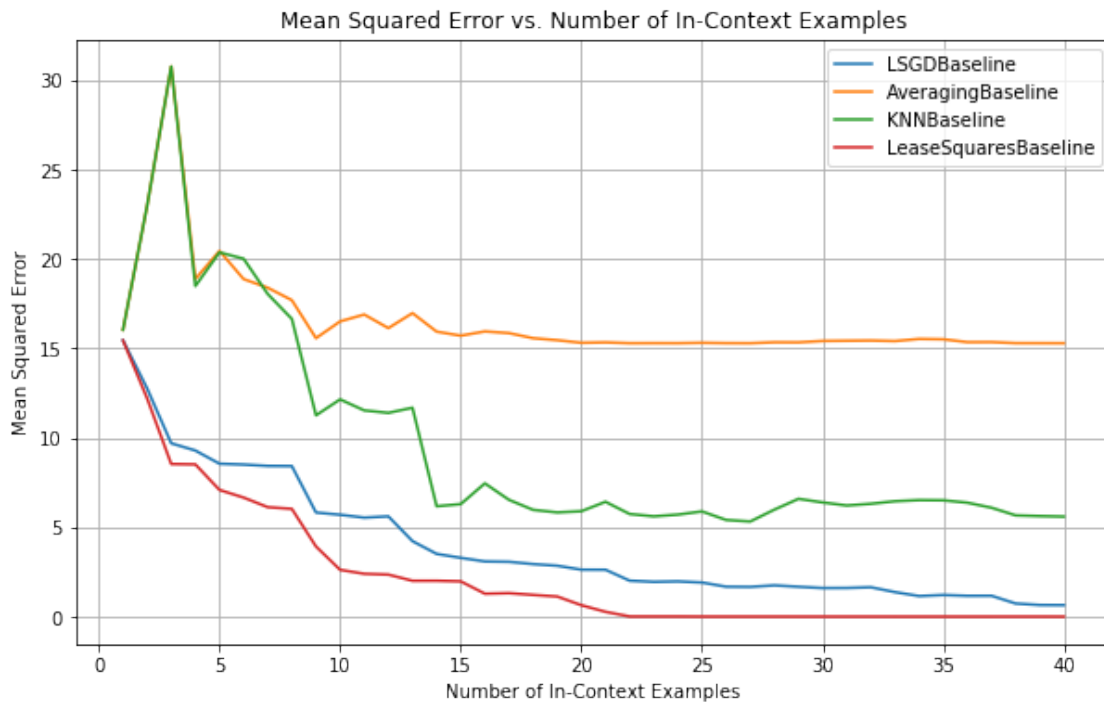
    for model_name, model in models.items():
        mse_list = []
        for in_context_examples in range(1, num_samples + 1):
```

```

        model.fit(X[:in_context_examples], y[:in_context_examples])
        mse = model.evaluate(X, y)
        mse_list.append(mse)
    results[model_name] = mse_list

    # Plotting squared error vs in-context examples
    plt.figure(figsize=(10, 6))
    for model_name, mse_list in results.items():
        plt.plot(range(1, num_samples + 1), mse_list, marker=None,
        ↪linestyle='-', label=model_name)
    plt.xlabel('Number of In-Context Examples')
    plt.ylabel('Mean Squared Error')
    plt.title('Mean Squared Error vs. Number of In-Context Examples')
    plt.legend()
    plt.grid(True)
    plt.show()

```



[]:

[]: