SpaceTreeRegressor and SpaceBoostingRegressor: Mathematical and Algorithmic Structure

These models provide innovative solutions to regression problems, with **SpaceTreeRegressor** using projection-based splitting logic and **SpaceBoostingRegressor** utilizing an ensemble learning approach. The mathematical details outline how the data is processed and how the split decisions are made.

1. SpaceTreeRegressor

How is the Projection Direction Determined?

`SpaceTreeRegressor` , unlike traditional decision trees, projects data along a specific direction for each split. This direction is computed as follows:

1. **Linear Regression**:
    - Given the data X (features matrix) and y (target values), a linear model that best fits the data is found:

    $$\hat{y} = X \cdot w$$

    where w are the linear regression coefficients that represent the direction for projecting the data.
2. **Data Projection**:
    - Each data point x_i is projected along the direction given by the coefficient vector w:

    $$p_i = x_i \cdot w$$

    - This projection maps the high-dimensional data space to a one-dimensional projection axis.

How is the Split Point Determined?

After projecting the data, the split point is optimized as follows:

1. **Sorting Projections**:
   - All projection values $\{p_1, p_2, \ldots, p_n\}$ are sorted in ascending order.
2. **Split Candidates**:
   - The sorted projections are used to generate candidate split positions. Possible split positions are selected by ensuring that the resulting left and right groups satisfy the minimum samples per leaf constraint. These split positions correspond to points between adjacent values in the sorted projections.
3. **Error Calculation (MSE)**:
   - For each split candidate, the data is divided into two groups: left (L) and right (R).
   - The mean squared error (MSE) for each group is computed:

$$\text{MSE}_L = \frac{1}{|L|} \sum_{i \in L} (y_i - \bar{y}_L)^2$$

$$\text{MSE}_R = \frac{1}{|R|} \sum_{i \in R} (y_i - \bar{y}_R)^2$$

   - The total MSE is the weighted average of the two groups:

$$\text{Total MSE} = \frac{|L|}{n} \text{MSE}_L + \frac{|R|}{n} \text{MSE}_R$$

4. **Selecting the Best Split**:
   - The split candidate that minimizes the total MSE is selected as the optimal threshold t^*.

How is the Tree Built?

1. **Data Split**:
   - Based on the chosen threshold t^*, the data is divided into two subgroups:

$$L = \{i \mid p_i \leq t^*\} \quad \text{and} \quad R = \{i \mid p_i > t^*\}$$

2. **Recursion**:
   - The same process is recursively applied to both subgroups.

- The process stops when the maximum depth is reached, or when the subgroup contains fewer than the minimum number of samples required to split further.

2. SpaceBoostingRegressor

Boosting Logic

`SpaceBoostingRegressor` is an ensemble model that sequentially uses multiple `SpaceTreeRegressor` models, aiming to reduce the error at each step. The key here is that the residuals are recalculated in each iteration, causing the projection direction to change with every new tree.

1. **Initial Model**:
   - The first prediction is the mean of the target values $\bar{y}$:

   $$f_0(x) = \bar{y}$$

2. **Residual Learning**:
   - In each iteration, the residuals (errors) are computed based on the current model's predictions:

   $$r_i = y_i - f_t(x_i)$$

   - These residuals become the new targets for the next iteration. A new `SpaceTreeRegressor` is trained on these residuals to learn the errors that the current model hasn't captured.
3. **Dynamic Projection Direction**:
   - In each boosting iteration, the residuals r_i are projected using linear regression to determine the new projection direction. This means that the projection direction w_t is different for each boosting step because the residuals change after each model update:

   $$p_i = x_i \cdot w_t$$

   - As a result, the linear regression coefficients w_t and the resulting projection direction change in each iteration.
4. **Model Update**:
   - The new model $h_t(x)$ (the tree built on the residuals) is added to the current model, weighted by the learning rate η:

   $$f_{t+1}(x) = f_t(x) + \eta h_t(x)$$

where $h_t(x)$ is the prediction of the `SpaceTreeRegressor` for the residuals.

5. **Iteration**:
    - The process is repeated for a predefined number of iterations or until the error drops below a specified threshold.

Final Model and Prediction

- After training for n iterations, the final model is a weighted sum of all individual trees:

$$f(x) = f_0(x) + \eta \sum_{t=1}^{n} h_t(x)$$

- The prediction for a new input x is computed as:

$$\hat{y}(x) = f(x)$$