kaggle.com

Identifying the Transit Interval

For identifying the time location of the transit, the wavelength-averaged signal was used. After estimating the approximate time position using a rule-based algorithm, the exact time was identified with a fitting algorithm.

Gain Drift

Gain drift refers to the variation in the detector's gain over time and across different wavelengths. These drifts can introduce systematic errors in the observed signal and must be corrected to accurately estimate the transit dip.

ExoSim2 models the gain drift as:

$$(1 + f(t) \cdot g(\lambda))$$

where $f(t)$ and $g(\lambda)$ are polynomial functions.

It is notable that the final function form is different from a two-variable polynomial $h(t, \lambda)$, because while the wavelength and time components can be separated in the former, they cannot in the latter.

By using this functional form directly in the fitting described below, we were able to achieve high performance in the dip estimation.

Dip Estimation with Gain Drift Fitting

The function used for fitting is as follows:

$$y_{\text{pred}} = I(\lambda) \times \text{Box}(\lambda) \times (1 + f(t) \cdot g(\lambda))$$

where $I(\lambda)$ is the spectrum of the star, and $\text{Box}(\lambda)$ describes the dip; it is 1 outside the transit and $1 - d_\lambda$ inside the transit.

The number of fitting parameters are as follows:

- $f(t)$: 5 parameters
- $g(\lambda)$: 5 parameters

It is worth mentioning that the optimal $I(\lambda)$ and the optimal dip used in $\text{Box}(\lambda)$, which minimize the mean squared error, can be found analytically if the other parameters are fixed, and they don't need to be considered by the fitting algorithm.

To stabilize the fitting process, a two-stage fitting was used. In the first stage, $I(\lambda)$ was directly estimated from the raw signal with temporal averaging and fixed during the fitting. In the second stage, $I(\lambda)$ was not fixed and was set optimally for each step during the fitting, while for the fitting parameters, the result from the first stage was used as the initial parameter. The error for each data point used in the fitting was estimated from the variation of the signal at each wavelength.

Dip Error Estimation with Bootstrapping

The errors in the dip estimation at each wavelength differ due to the varying signal-to-noise ratios associated with each wavelength. Bootstrapping was used to overcome this problem and to estimate the dip error of each wavelength.

Further details are available in our code.

Dip Estimation Considering Wavelength Correlations

Three models were used: Gaussian Process Regression, AutoEncoder, and Non-negative Matrix Factorization (NMF). They were ensembled with the ratio 6:2:2.

Gaussian Process Regression

Nothing fancy, unlike the second-place solution.

A simple kernel composed of RBF and Matern kernels was employed, and the errors calculated by bootstrapping were passed to `sklearn.gaussian_process.GaussianProcessRegressor` so that the model can consider the uncertainty of each data point.

AutoEncoder

We applied an autoencoder to capture relationships in the data.

Unlike PCA, which captures linear relationships, autoencoders can model more complex and nonlinear patterns.

Also, we expected that by training the model with MSE loss, the autoencoder model could take noise into account and recover the "optimal" spectrum.

Important points were to normalize the data for each exoplanet and to take the moving median of the dip spectrum to smooth the input dip spectrum.

The model we used is as below, with 4 nodes in the hidden layer:

```
input_data = Input(shape=(input_dim,))
encoded = Dense(encoding_dim, activation='relu')(input_data)
decoded = Dense(input_dim, activation='linear')(encoded)

autoencoder = Model(input_data, decoded)
```

NMF

Very similar to the autoencoder, but we added it to enhance the diversity.

Unlike with the autoencoder, the number of ranks was set to 5 for NMF.

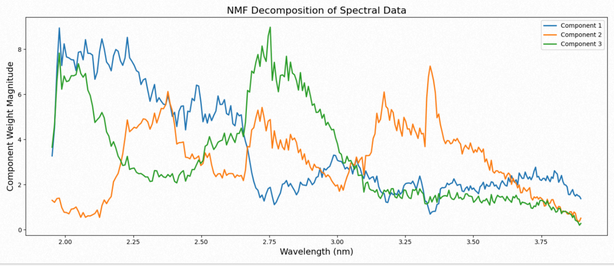
Spectral Components Identified by NMF

The plot below shows the identified spectral components by NMF for the training data.

It can be seen that the main contributing gases of each component are:

- **Component 1**: CO_2
- **Component 2**: CH_4
- **Component 3**: H_2O

This signifies NMF's ability to consider the correlation of the spectrum unsupervised.



Sigma

We took the weighted average of the following components:

1. **Constant value** (planet and wavelength independent)
2. **Standard deviation of the smoothed predicted dip spectrum** (planet dependent, wavelength independent)
3. **Uncertainty predicted by Gaussian Process Regression** (planet and wavelength dependent)

The constant value played an important role in cases where the dip spectrum was almost constant but had some bias.

The Contribution of Each Component

Since our solution incorporates a variety of ideas, we examined the contributions for those that seem important with late submissions.

Method	Public	Private	Public Loss	Private Loss
Final Submission	0.7330321	0.7420624	0.0000000	0.0000000
Only Gaussian Process Regression	0.7221480	0.7343485	-0.0108841	-0.0077139
Only AutoEncoder	0.7078056	0.7181137	-0.0252265	-0.0239487
Only NMF	0.7017943	0.7122631	-0.0312378	-0.0297993
With Hot Pixel Processing	0.7021653	0.7224989	-0.0308668	-0.0195635
Without Foreground Processing with *1.008 to prediction	0.7225193	0.7298121	-0.0105128	-0.0122503