Cloudy fundamentals

To generate a Cloudy model we usually supply the following :

1. An extragalactic UV background (EUVB) radiation model
2. Specify the redshift z
3. For a "cool" cloud ($T \sim 10^4$K), we supply the HI column density ($N_{\rm HI}$), the density ($n_{\rm H}$), the metallicity [X/H]

These inputs are used by the CLOUDY code to generate the ionization parameter ($U = n_\gamma / n_{\rm H}$) and the ionization states of all ions. That is, the output is:

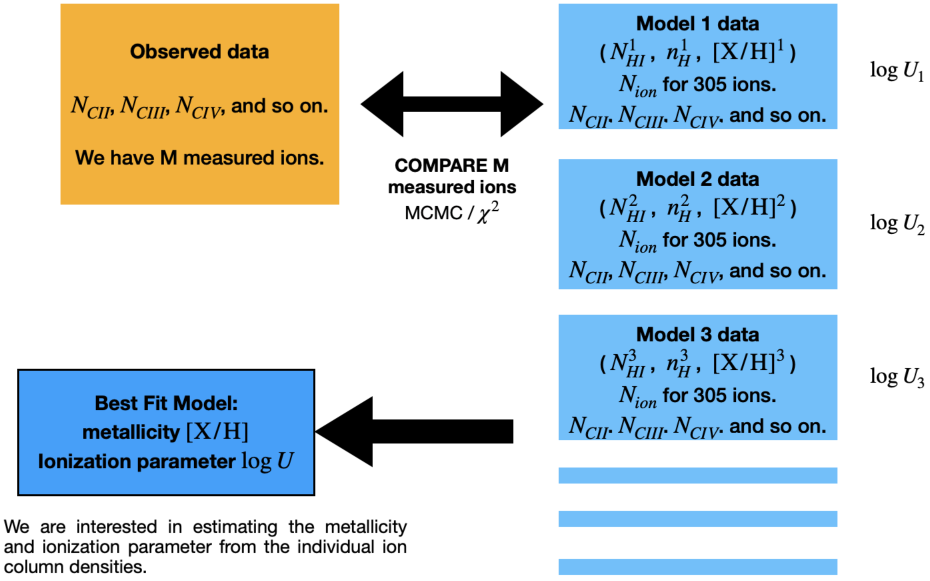
1. $\log U$
2. N_{ion} for > 300 ions

Say we identify each model by its unique $\log U$ value, we have models corresponding to $\log U_1$, $\log U_2$, $\log U_3$, ... , $\log U_K$.

Observations:

The observed quantities are $N_{\rm HI}$ and N_{ion} 's for as many ions as accessible in the observed spectrum. This is always less than the number of ions that Cloudy provides us column densities for.

Optimization or finding the best fit model:


Observed data
$N_{CII}, N_{CIII}, N_{CIV}$, and so on.
We have M measured ions.
COMPARE M measured ions
MCMC / χ^2
Model 1 data
(N_{HI}^1, n_H^1, $[{\rm X/H}]^1$)
N_{ion} for 305 ions.
$N_{CII}, N_{CIII}, N_{CIV}$, and so on.
$\log U_1$
Model 2 data
(N_{HI}^2, n_H^2, $[{\rm X/H}]^2$)
N_{ion} for 305 ions.
$N_{CII}, N_{CIII}, N_{CIV}$, and so on.
$\log U_2$
Model 3 data
(N_{HI}^3, n_H^3, $[{\rm X/H}]^3$)
N_{ion} for 305 ions.
$N_{CII}, N_{CIII}, N_{CIV}$, and so on.
$\log U_3$
Best Fit Model:
metallicity [X/H]
Ionization parameter $\log U$

We are interested in estimating the metallicity and ionization parameter from the individual ion column densities.

- So there is a 3-D parameter grid: $\{x_1, x_2, x_3\} = \{N_{\rm HI}, n_{\rm H}, [{\rm X/H}]\}$
- What we compare are the observables: $y_{obs} = \{N_i(observed)\}$ with $y_{model} = \{N_i(model)\}$
- We want to minimize: $\sum_i (y_{obs} - y_{model})^2$

Goals with ML / emulator:

Instead of the optimizer (MCMC code) comparing every model stored in memory to the observed data, we want it to use the ML or emulator tool to first learn the data and then predict values at each point in the MCMC step. That is, as the sampler takes small jumps in the parameter space of $N_{\rm HI}$, $n_{\rm H}$ and, [X/H], **instead** of :

1. evaluating the model at each point

2. OR using precomputed models to generate an interpolation for each ion that is being compared (often 30 or 40 ions are compared) ,

we use the ML model or emulator to predict values. This will allow for the following:

1. We can set the precision of sampling to be small (find more accurate best fit models) without computing models or interpolating on smaller denser grids.

2. **More importantly :** Speed up the sampling and consequtive evaluation of model values using interpolation.

Essentially we are replacing interpolation of ion column densities from different models with **creating a single "machine learned" model that can be used to predict the entire model at once (all column densities and** $\log U$.