# PyLogit: Maximum Likelihood Estimation for Conditional "Logit-type" Models in Python

Timothy Brathwaite

March 21, 2016

## 1 Introduction

In many fields—including transportation, marketing, economics, finance, medicine, public health, and operations research—it is necessary to model the occurrence of unordered, discrete outcomes. Often, such outcomes represent an individual's choice from a set of alternatives. A simple way to model such an outcome is with what is ambiguously known as a "logit model." Across the many fields mentioned above, "logit models" are also referred to as:

1. ***binary logistic regression*** or simply ***logistic regression*** when there are only two alternatives and all covariates are individual specific. These names are often used in statistics, computer science, medicine, and public health.

2. ***binary logit*** when there are only two alternatives and covariates can vary across alternatives as well as across individuals. This name is often used in transportation, marketing, and economics.

3. ***polychotomous regression*** or ***multinomial logistic regression*** when there are two or more (typically more) alternatives and all covariates are individual specific. These names are often used in medicine and public health.

4. ***softmax regression*** when there are two or more (typically more) alternatives and all covariates are individual specific. This name is often used in computer science and machine learning.

5. ***conditional logit*** or ***multinomial logit*** when there are two or more (typically more) alternatives and covariates may vary across alternatives as well as across individuals. These names are often used in transportation, marketing, and economics.

Of the various logit model varieties listed above, conditional logit models are the most general. Polychotomous/softmax regression is a special case of conditional logit where the covariates only vary across individuals. Binary logit is a special case of conditional logit where there are only two alternatives. Lastly, (binary) logistic regression is a special case of conditional logit where there are only two alternatives and when covariates only vary across individuals.

In Python's scientific-computing ecosystem today, logit models are capable of being estimated by packages such as statsmodels and sci-kit-learn. However, these packages can only estimate binary logistic regression and polychotomous/softmax/multinomial regression where the covariates are individual specific. There are currently no Python packages on the Python Packaging Index (PyPi) that allow one to estimate conditional logit models. Note, the Python Biogeme software is able to estimate conditional logit models. However, Python Biogeme is not available on PyPi, so it cannot easily be installed. Moreover, Python Biogeme is largely a wrapper for routines written in C, and it does not fully accept all Python syntax.

Beyond Python, R is also heavily used for free and open-source scientific-computing. In R, the packages mlogit and mnlogit are capable of estimating conditional logit models. However, these packages cannot estimate conditional logit models when the choice set differs across individual. Additionally, these packages do not directly support the estimation of models where variables are associated with coefficients that vary only across subsets of alternatives. For instance, in a travel-mode choice context, mlogit and mnlogit do

not directly support models where the travel time variable has one coefficient in the index of a drive-alone alternative and one coefficient for both the bus and train alternatives.

The pylogit package that is being introduced is capable of

- estimating conditional logit models (as well as other conditional "logit-type" models to be defined later)
- estimating models using datasets where the choice set differs between individuals
- estimating models with very general model specifications where the coefficients for a given variable may be
  - completely alternative-specific (i.e. one coefficient per alternative, subject to identification of the coefficients),
  - subset-specific (i.e. one coefficient per subset of alternatives, where each alternative belongs to only one subset, and there are more than 1 but less than $J$ subsets, where $J$ is the maximum number of available alternatives in the dataset),
  - completely generic (i.e. one coefficient across all alternatives).

# 2 Logit-type Models

This section describes the type of models that are capable of being estimated with PyLogit, and in the process, it introduces some of the notation that will be used later in describing how the maximum likelihood estimation (MLE) is performed for this class of models.

"Logit-type" models are models where the probability of individual $i$ choosing alternative $j$ (or being associated with outcome $j$) is given by:

$$
\begin{aligned}
P\left(y_{ij} = 1 | x_{ij}, \beta, \tau, \gamma\right) &= \frac{\exp\left[S\left(V_{ij} \mid \tau_j, \gamma_j\right)\right]}{\sum_{\ell \in C_i} \exp\left[S\left(V_{i\ell} \mid \tau_\ell, \gamma_\ell\right)\right]} = \frac{\exp\left(S_{ij}\right)}{\sum_{\ell \in C_i} \exp\left(S_{i\ell}\right)} \\
\text{where } y_{ij} &= \text{a binary (0 or 1) indicator of whether individual } i \\
&\quad \text{is associated with outcome } j. \\
x_{ij} &= \text{a row vector} = r\left(z_j, \zeta_i\right) \\
r\left(\cdot\right) &= \text{a function that returns a row vector.} \\
z_j &= \text{attributes of alternative } j \text{ for individual } i. \\
\zeta_i &= \text{characteristics of individual } i. \\
\beta &= \text{a column vector of unknown population parameters.} \\
\tau &= \text{a 1-dimensional vector of constants, with one value} \\
&\quad \text{for each alternative in the dataset.} \\
\gamma &= \text{a 1-dimensional vector of "shape parameters", with} \\
&\quad \text{one value for each alternative in the dataset.} \\
\tau_j &= \text{a constant associated with alternative } j. \\
\gamma_j &= \text{a "shape parameter" associated with alternative } j. \\
V_{ij} &= x_{ij}\beta = \text{the } \textit{index} \text{ for alternative } j \\
S\left(\cdot\right) &= \text{a model-specific function of } V_{ij},\ \tau_j \text{ and } \gamma_j. \\
&\quad \text{It is monotonically increasing in } V_{ij}. \\
C_i &= \text{the choice set for individual } i.
\end{aligned}
\tag{1}
$$

All models estimated by PyLogit have this form. In a standard conditional logit model, $S\left(V_{ij} \mid \tau_j, \gamma_j\right) = \tau_j + V_{ij}$ where $\tau_j$ is the alternative specific constant or intercept term for alternative $j$ and $V_{ij}$ does not contain a constant of its own.

# 3 Computation

In performing the MLE of logit-type models, PyLogit evaluates as many operations as it can through the use of matrix dot products. In particular, the calculation of the log-likelihood, the gradient of the log-likelihood with respect to the estimated parameters, and the hessian of the log-likelihood of log-likelihood with respect to the estimated parameters are all calculated through the use of one or more matrix dot products. Often, to be able to use dot products to perform the desired calculations, we require the use of "mapping matrices": matrices of zeros and ones that map one quantity to another related quantity. More will be said about the mapping matrices later.

## 3.1 The log-likelihood

Maximum likelihood estimation works as follows to find a desired vector of parameters $\hat{\theta}$:

$$
\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} \prod_{i=1}^{N} \prod_{j \in C_i} P_{ij}^{y_{ij}} \\
&= \arg\max_{\theta} \sum_{i}^{N} \sum_{j \in C_i} y_{ij} \ln\left(P_{ij}\right) \\
&= \arg\max_{\theta} Y^T \ln\left(P\right) \\
&= \arg\max_{\theta} \mathcal{L}\left(\theta\right) \\
\text{where } \theta &= \left[\hat{\gamma}^T \mid \hat{\tau}^T \mid \hat{\beta}^T\right]^T = \text{a column vector created by stacking} \\
&\quad \hat{\gamma} \text{ on top of } \hat{\tau} \text{ on top of } \hat{\beta}. \text{ Note that terms with "hats"} \\
&\quad \text{on them are estimates of their corresponding, unknown} \\
&\quad \text{quantities.} \\
P_{ij} &= \text{the probability of individual } i \text{ choosing alternative } j \\
&\quad \text{or being associated with outcome } j. \text{ This probability is} \\
&\quad \text{given in Eqution 1.} \\
Y &= \text{a column vector in } \mathbb{R}^{N_r}, \text{ made by vertically stacking the} \\
&\quad y_{ij} \text{ for every available alternative for each individual.} \\
N_r &= \sum_{i} \parallel C_i \parallel \\
\parallel C_i \parallel &= \text{the size of the choice set for individual } i. \\
P &= \text{a column vector in } \mathbb{R}^{N_r}, \text{ made by vertically stacking the} \\
&\quad P_{ij} \text{ for every available alternative for every individual.} \\
\ln\left(P\right) &= \text{a column vector in } \mathbb{R}^{N_r}, \text{ made by taking the natural} \\
&\quad \text{logarithm, element-wise, of all the elements in } P. \\
\mathcal{L}\left(\theta\right) &= \text{the log-likelihood at the current value of } \theta. = Y^T \ln\left(P\right)
\end{aligned}
\tag{2}
$$

To compute the maximum-likelihood estimate, $\hat{\theta}$, we therefore need at minimum the vectors $Y$ and $P$. While introduced above through the notion of vertically stacking each of the individual elements, $Y$ and $P$ are not typically obtained this way by PyLogit. First, PyLogit requires that all data used to estimate its logit-type models be in so-called "long-format." Long-format data has one row for every available alternative for every individual. Moreover, the column used to indicate the choice or outcome for an individual is required to be a binary column—i.e. filled with zeros and ones. A "1" indicates that the alternative on the given row was chosen or associated with the corresponding individual for that row. In other words, $Y$ is available directly from the dataset provided to PyLogit.

A series of internal steps is used to efficiently calculate $P$. First, from the specification of the desired index coefficients, $\beta$, the indices those coefficients belong to, and the variables being multiplied by those coefficients, a design matrix $X$ is formed. Second, the vector $V$ of desired index values for each available alternative for each individual is formed through the dot product of $X$ and $\hat{\beta}$. An example of a set of desired index values is given in the "Main PyLogit Example" (← click me). Thirdly, the necessary transformations $S\left(V_{ij} \mid \tau_j, \gamma_j\right)$ are computed. As much as possible, such transformations are vectorized for speed using numpy and scipy. Finally, $P$ is formed through a set of elementwise operations and matrix dot products. Formally,

$$
\begin{aligned}
V = X\hat{\beta} \quad \text{and } P &= \frac{\exp\left(\vec{S}\right)}{\lambda\lambda^T \exp\left(\vec{S}\right)} \\
\text{where } X &\in \mathbb{R}^{N_r \times E} \\
X &= \text{The design matrix, formed by vertically stacking all} \\
&\quad \text{of the } X_{ij} \text{ on top of each other.} \\
X_{ij} &\in \mathbb{R}^{1 \times E} \\
N_r &= \text{the number of rows in X. Defined computationally in} \\
&\quad \text{Equation 2.} \\
E &= \text{the number of explanatory variables in the model.} \\
&\quad \text{I.e. the number of index coefficients being estimated.} \\
\vec{S} &\in \mathbb{R}^{N_r \times 1} \\
\vec{S} &= \text{a column vector formed by stacking all of the} \\
&\quad S\left(V_{ij} \mid \hat{\tau_j}, \hat{\gamma_j}\right) = S_{ij} \text{ on top of each other.} \\
\exp\left(\vec{S}\right) &= \text{a column vector in } \mathbb{R}^{N_r}\text{, made by raising} \\
&\quad \text{the natural number } e\text{, element-wise, to the power of the} \\
&\quad \text{elements in } \vec{S}. \\
\lambda &\in \mathbb{R}^{N_r \times N} \\
\lambda &= \text{a mapping matrix of zeros and ones that indicates the} \\
&\quad \text{observation (given on the columns) that each row of the} \\
&\quad \text{design matrix corresponds to.} \\
N &= \text{the number of observations in one's dataset.}
\end{aligned}
\tag{3}
$$

Note that in the formula given above for $P$, the fraction indicates element-wise division.

## 3.2 The Gradient, $\partial\mathcal{L}\left(\hat{\theta}\right)/\partial\hat{\theta}$

The gradient of the log-likelihood with respect to our estimates $\hat{\theta}$ is used by many of the optimization algorithms in Scipy to numerically find a local optima of our log-likelihood function. Therefore, we need an efficient way of calculating the gradient.

The gradient of the log-likelihood function with respect to $\hat{\theta}$ is defined as follows:

$$
\begin{aligned}
\nabla \mathcal{L}\left(\hat{\theta}\right) &= \frac{\partial \mathcal{L}\left(\hat{\theta}\right)}{\partial \hat{\theta}} \\
&= \frac{\partial \left[Y^T \ln\left(P\right)\right]}{\partial \hat{\theta}} \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} \ln\left(P_{ij}\right)\right], \quad i \in \{1, 2, ..., N\} \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} \left(\ln\left[\exp\left(S_{ij}\right)\right] - \ln\left[\sum_{\ell \in C_i} \exp\left(S_{i\ell}\right)\right]\right)\right] \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} \left(S_{ij} - \ln\left[\sum_{\ell \in C_i} \exp\left(S_{i\ell}\right)\right]\right)\right] \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} S_{ij} - y_{ij} \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} S_{ij} - \sum_i \sum_{j \in C_i} y_{ij} \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} S_{ij} - \sum_i \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right) \sum_{j \in C_i} y_{ij}\right] \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \sum_{j \in C_i} y_{ij} S_{ij} - \sum_i \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \frac{\partial}{\partial \hat{\theta}} \left[\sum_i \left(\sum_{j \in C_i} y_{ij} S_{ij}\right) - \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \sum_i \left[\frac{\partial}{\partial \hat{\theta}} \left(\sum_{j \in C_i} y_{ij} S_{ij}\right) - \frac{\partial}{\partial \hat{\theta}} \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \sum_i \left[\left(\sum_{j \in C_i} \frac{\partial}{\partial \hat{\theta}} \left[y_{ij} S_{ij}\right]\right) - \frac{\partial}{\partial \hat{\theta}} \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \sum_i \left[\left(\sum_{j \in C_i} \frac{\partial}{\partial S_{ij}} \left[y_{ij} S_{ij}\right] \frac{\partial S_{ij}}{\partial \hat{\theta}}\right) - \frac{\partial}{\partial \hat{\theta}} \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \sum_i \left[\left(\sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}}\right) - \frac{\partial}{\partial \hat{\theta}} \ln\left(\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]\right)\right] \\
&= \sum_i \left[\left(\sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}}\right) - \frac{\partial}{\partial \varphi} \left[\ln\left(\varphi\right)\right] \frac{\partial \varphi}{\partial \hat{\theta}}\right] \quad \text{where } \varphi = \sum_{\ell \in C_i} \exp\left[S_{i\ell}\right] \\
&= \sum_i \left[\left(\sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}}\right) - \frac{1}{\varphi} \frac{\partial \varphi}{\partial \hat{\theta}}\right]
\end{aligned}
$$

$$
\begin{aligned}
\nabla \mathcal{L}\left(\hat{\theta}\right) &= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \frac{1}{\varphi} \frac{\partial}{\partial \hat{\theta}} \left( \sum_{j \in C_i} \exp\left[S_{ij}\right] \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \frac{1}{\varphi} \left( \sum_{j \in C_i} \frac{\partial}{\partial \hat{\theta}} \exp\left[S_{ij}\right] \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \frac{1}{\varphi} \left( \sum_{j \in C_i} \frac{\partial}{\partial S_{ij}} \exp\left[S_{ij}\right] \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \frac{1}{\varphi} \left( \sum_{j \in C_i} \exp\left[S_{ij}\right] \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \left( \sum_{j \in C_i} \frac{\exp\left[S_{ij}\right]}{\varphi} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \left( \sum_{j \in C_i} \frac{\exp\left[S_{ij}\right]}{\sum_{\ell \in C_i} \exp\left[S_{i\ell}\right]} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) - \left( \sum_{j \in C_i} P_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) \right] \\
&= \sum_i \left[ \left( \sum_{j \in C_i} y_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} - P_{ij} \frac{\partial S_{ij}}{\partial \hat{\theta}} \right) \right] \qquad (4) \\
&= \sum_i \sum_{j \in C_i} \left[ \left( y_{ij} - P_{ij} \right) \frac{\partial S_{ij}}{\partial \hat{\theta}} \right] \\
&= \left( Y - P \right)^T \frac{\partial \vec{S}}{\partial \hat{\theta}} \\
&= \left( Y - P \right)^T \left[ \frac{\partial \vec{S}}{\partial \hat{\gamma}} \mid \frac{\partial \vec{S}}{\partial \hat{\tau}} \mid \frac{\partial \vec{S}}{\partial \hat{\beta}} \right] \\
&= \left( Y - P \right)^T \left[ \frac{\partial \vec{S}}{\partial \hat{\gamma}} \mid \frac{\partial \vec{S}}{\partial \hat{\tau}} \mid \frac{\partial \vec{S}}{\partial V} \frac{\partial V}{\partial \hat{\beta}} \right] \\
&= \left( Y - P \right)^T \left[ \frac{\partial \vec{S}}{\partial \hat{\gamma}} \mid \frac{\partial \vec{S}}{\partial \hat{\tau}} \mid \frac{\partial \vec{S}}{\partial V} X \right] \\
&= \left( Y - P \right)^T \left[ \frac{\partial \vec{S}}{\partial \hat{\gamma}} \mid \xi^{(-1)} \mid \frac{\partial \vec{S}}{\partial V} X \right]
\end{aligned}
$$

where $\xi =$ a mapping matrix of zeros and ones that indicates the alternative (given on the columns) that each row of the design matrix corresponds to.

$\xi^{(-1)} =$ the mapping matrix $\xi$, without the column that corresponds to the alternative whose intercept, $\tau_j$, is not being estimated.

Note that by the chain rule, $\dfrac{\partial \mathcal{L}\left(\hat{\theta}\right)}{\partial \hat{\theta}} = \dfrac{\partial \mathcal{L}\left(\hat{\theta}\right)}{\partial \vec{S}} \dfrac{\partial \vec{S}}{\partial \hat{\theta}}$. From above, this implies that $\dfrac{\partial \mathcal{L}\left(\hat{\theta}\right)}{\partial \vec{S}} = \left(Y - P\right)^T$.

In PyLogit, the calculation of the gradient is based on Equation 4. The vectors $Y$ and $P$ are readily available based on the provided data and Equation 3. The matrices $\frac{\partial \vec{S}}{\partial \hat{\gamma}}$ and $\frac{\partial \vec{S}}{\partial V}$ differ from model to model, and custom functions to calculate them must[1] be written for each model that PyLogit supports. $\xi^{(-1)}$ differs from specification to specification, but it can always be easily calculated once per model estimation. Note that in PyLogit, all of the mapping matrices are implemented as Scipy sparse matrices for fast matrix dot products.

## 3.3 The Hessian, $\frac{\partial^2 \mathcal{L}\left(\hat{\theta}\right)}{\partial \hat{\theta}^2}$

Although it is not required by all of the optimizers in scipy.optimize, being able to calculate the Hessian in closed-form is useful for optimizing non-concave log-likelihood functions. Below, the form of the Hessian used by PyLogit is derived for logit-type models.

The hessian is given by:

$$
\begin{aligned}
\text{Hessian} = \frac{\partial^2 \mathcal{L}\left(\hat{\theta}\right)}{\partial \hat{\theta}^2} &= \frac{\partial}{\partial \hat{\theta}}\left[\frac{\partial \left(Y^T \ln\left[P\right]\right)}{\partial \hat{\theta}}\right] \\
&= \frac{\partial}{\partial \hat{\theta}}\left[\frac{\partial}{\partial \hat{\gamma}}\left(Y^T \ln\left[P\right]\right) \mid \frac{\partial}{\partial \hat{\tau}}\left(Y^T \ln\left[P\right]\right) \mid \frac{\partial}{\partial \hat{\beta}}\left(Y^T \ln\left[P\right]\right)\right] \\
&= \begin{bmatrix} \frac{\partial}{\partial \hat{\theta}}\left(\frac{\partial}{\partial \hat{\gamma}}\left[Y^T \ln\left(P\right)\right]\right) \\ \frac{\partial}{\partial \hat{\theta}}\left(\frac{\partial}{\partial \hat{\tau}}\left[Y^T \ln\left(P\right)\right]\right) \\ \frac{\partial}{\partial \hat{\theta}}\left(\frac{\partial}{\partial \hat{\beta}}\left[Y^T \ln\left(P\right)\right]\right) \end{bmatrix} \\
&= \left[\begin{array}{c|c|c} \frac{\partial}{\partial \hat{\gamma}}\left(\frac{\partial}{\partial \hat{\gamma}}\left[Y^T \ln\left(P\right)\right]\right) & \frac{\partial}{\partial \hat{\tau}}\left(\frac{\partial}{\partial \hat{\gamma}}\left[Y^T \ln\left(P\right)\right]\right) & \frac{\partial}{\partial \hat{\beta}}\left(\frac{\partial}{\partial \hat{\gamma}}\left[Y^T \ln\left(P\right)\right]\right) \\ \frac{\partial}{\partial \hat{\gamma}}\left(\frac{\partial}{\partial \hat{\tau}}\left[Y^T \ln\left(P\right)\right]\right) & \frac{\partial}{\partial \hat{\tau}}\left(\frac{\partial}{\partial \hat{\tau}}\left[Y^T \ln\left(P\right)\right]\right) & \frac{\partial}{\partial \hat{\beta}}\left(\frac{\partial}{\partial \hat{\tau}}\left[Y^T \ln\left(P\right)\right]\right) \\ \frac{\partial}{\partial \hat{\gamma}}\left(\frac{\partial}{\partial \hat{\beta}}\left[Y^T \ln\left(P\right)\right]\right) & \frac{\partial}{\partial \hat{\tau}}\left(\frac{\partial}{\partial \hat{\beta}}\left[Y^T \ln\left(P\right)\right]\right) & \frac{\partial}{\partial \hat{\beta}}\left(\frac{\partial}{\partial \hat{\beta}}\left[Y^T \ln\left(P\right)\right]\right) \end{array}\right] \\
&= \left[\begin{array}{c|c|c} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{array}\right]
\end{aligned}
\tag{5}
$$

Based on Equation 5, we can see that the Hessian is actually a partitioned matrix with nine sub-matrices. In the cases where either $\gamma$ and/or $\tau$ do not exist in one's model, the Hessian matrix can be derived from Equation 5 by deleting the rows and columns that contain partial derivatives with respect to the parameters that are not present in $\theta$. In the simplest case, where there are no shape parameters ($\gamma$) and no intercept parameters ($\tau$) outside of the index, the Hessian reduces to the usual expression for conditional logit models: $\frac{\partial}{\partial \hat{\beta}}\left(\frac{\partial}{\partial \hat{\beta}}\left[Y^T \ln\left(P\right)\right]\right)$.

Below, we will derive expressions for the nine sub-matrices $H_{mn}$ where $m, n \in \{1, 2, 3\}$. Note however, that we do not actually need to derive all nine sub-matrices because mixed partial derivatives of vectors are transposes of each other. Stated more precisely, $\frac{\partial}{\partial A}\left[\frac{\partial \alpha\left(A, B\right)}{\partial B}\right] = \left(\frac{\partial}{\partial B}\left[\frac{\partial \alpha\left(A, B\right)}{\partial A}\right]\right)^T$ for a function $\alpha\left(\cdot\right)$ that maps input vectors $A$ and $B$ to scalar outputs.

[1] Automatic Differentiation or Finite Differences could be used to avoid having to manually write such functions, but the estimation times would increase greatly as a result.

### 3.3.1 $H_{11}$

$$
\begin{aligned}
H_{11} &= \frac{\partial}{\partial \hat{\gamma}} \left( \frac{\partial}{\partial \hat{\gamma}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\gamma}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \\
&= \frac{\partial}{\partial \hat{\gamma}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= - \left( \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}}
\end{aligned}
\tag{6}
$$

### 3.3.2 $H_{12}$

$$
\begin{aligned}
H_{12} &= \frac{\partial}{\partial \hat{\tau}} \left( \frac{\partial}{\partial \hat{\gamma}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\tau}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \\
&= \frac{\partial}{\partial \hat{\tau}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= - \left( \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right)^T \frac{\partial P}{\partial \vec{S}} \xi^{(-1)}
\end{aligned}
\tag{7}
$$

### 3.3.3 $H_{13}$

$$
\begin{aligned}
H_{13} &= \frac{\partial}{\partial \hat{\beta}} \left( \frac{\partial}{\partial \hat{\gamma}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\beta}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \\
&= \frac{\partial}{\partial \hat{\beta}} \left( \left[ Y - P \right]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( \left[ Y - P \right]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= \frac{\partial}{\partial \left( Y - P \right)} \left( \left[ Y - P \right]^T \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right) \frac{\partial \left( Y - P \right)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right)^T \frac{\partial \left( Y - P \right)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= - \left( \frac{\partial \vec{S}}{\partial \hat{\gamma}} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial V} X
\end{aligned}
\tag{8}
$$

### 3.3.4 $H_{21}$

$$
\begin{aligned}
H_{21} &= \frac{\partial}{\partial \hat{\gamma}} \left( \frac{\partial}{\partial \hat{\tau}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\gamma}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \\
&= \frac{\partial}{\partial \hat{\gamma}} \left( \left[ Y - P \right]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( \left[ Y - P \right]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= \frac{\partial}{\partial \left( Y - P \right)} \left( \left[ Y - P \right]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \frac{\partial \left( Y - P \right)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\tau}} \right)^T \frac{\partial \left( Y - P \right)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= - \left( \xi^{(-1)} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}}
\end{aligned}
\tag{9}
$$

### 3.3.5 $H_{22}$

$$
\begin{aligned}
H_{22} &= \frac{\partial}{\partial \hat{\tau}} \left( \frac{\partial}{\partial \hat{\tau}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\tau}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \\
&= \frac{\partial}{\partial \hat{\tau}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\tau}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= - \left( \xi^{(-1)} \right)^T \frac{\partial P}{\partial \vec{S}} \xi^{(-1)}
\end{aligned}
\tag{10}
$$

### 3.3.6 $H_{23}$

$$
\begin{aligned}
H_{23} &= \frac{\partial}{\partial \hat{\beta}} \left( \frac{\partial}{\partial \hat{\tau}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\beta}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \\
&= \frac{\partial}{\partial \hat{\beta}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\tau}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\tau}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= - \left( \xi^{(-1)} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= - \left( \xi^{(-1)} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial V} X
\end{aligned}
\tag{11}
$$

### 3.3.7 $H_{31}$

$$
\begin{aligned}
H_{31} &= \frac{\partial}{\partial \hat{\gamma}} \left( \frac{\partial}{\partial \hat{\beta}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\gamma}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \\
&= \frac{\partial}{\partial \hat{\gamma}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\beta}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= - \left( \frac{\partial \vec{S}}{\partial \hat{\beta}} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= - \left( \frac{\partial \vec{S}}{\partial V} X \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}} \\
&= -X^T \left( \frac{\partial \vec{S}}{\partial V} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\gamma}}
\end{aligned}
\tag{12}
$$

### 3.3.8 $H_{32}$

$$
\begin{aligned}
H_{32} &= \frac{\partial}{\partial \hat{\tau}} \left( \frac{\partial}{\partial \hat{\beta}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\tau}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \\
&= \frac{\partial}{\partial \hat{\tau}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\beta}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\tau}} \\
&= - \left( \frac{\partial \vec{S}}{\partial \hat{\beta}} \right)^T \frac{\partial P}{\partial \vec{S}} \xi^{(-1)} \\
&= - \left( \frac{\partial \vec{S}}{\partial V} X \right)^T \frac{\partial P}{\partial \vec{S}} \xi^{(-1)} \\
&= -X^T \left( \frac{\partial \vec{S}}{\partial V} \right)^T \frac{\partial P}{\partial \vec{S}} \xi^{(-1)}
\end{aligned}
\tag{13}
$$

### 3.3.9 $H_{33}$

$$
\begin{aligned}
H_{33} &= \frac{\partial}{\partial \hat{\beta}} \left( \frac{\partial}{\partial \hat{\beta}} \left[ Y^T \ln (P) \right] \right) \\
&= \frac{\partial}{\partial \hat{\beta}} \left( \frac{\partial}{\partial \vec{S}} \left[ Y^T \ln (P) \right] \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \\
&= \frac{\partial}{\partial \hat{\beta}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \\
&= \frac{\partial}{\partial \vec{S}} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= \frac{\partial}{\partial (Y - P)} \left( [Y - P]^T \frac{\partial \vec{S}}{\partial \hat{\beta}} \right) \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= \left( \frac{\partial \vec{S}}{\partial \hat{\beta}} \right)^T \frac{\partial (Y - P)}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial \hat{\beta}} \\
&= - \left( \frac{\partial \vec{S}}{\partial V} X \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial V} X \\
&= -X^T \left( \frac{\partial \vec{S}}{\partial V} \right)^T \frac{\partial P}{\partial \vec{S}} \frac{\partial \vec{S}}{\partial V} X
\end{aligned}
\tag{14}
$$

### 3.3.10 $\frac{\partial P}{\partial \vec{S}}$

The last nine sub-sections showed that it is possible to express all of the sub-matrices of the Hessian in terms of dot products of matrices that were used to form the gradient and one unknown matrix $\frac{\partial P}{\partial \vec{S}}$. Below, we derive an expression for $\frac{\partial P}{\partial \vec{S}}$, thereby allowing us to compute a closed-form solution for the Hessian of the logit-type models used in PyLogit.

First, note that we can write:

$$P = \begin{bmatrix} P_1 \\ P_2 \\ ... \\ P_N \end{bmatrix}$$

$$\text{where } P_i = \begin{bmatrix} P_{i1} \\ P_{i2} \\ ... \\ P_{ij} \end{bmatrix}, \quad j \in C_i$$

It then follows that

$$\frac{\partial P}{\partial \vec{S}} = \begin{bmatrix} \frac{\partial P_1}{\partial \vec{S}} \\ \frac{\partial P_2}{\partial \vec{S}} \\ ... \\ \frac{\partial P_N}{\partial \vec{S}} \end{bmatrix} \tag{15}$$

A generic partition, $\frac{\partial P_i}{\partial \vec{S}}$ of $\frac{\partial P}{\partial \vec{S}}$ can further be decomposed as follows:

$$\begin{aligned} \frac{\partial P_i}{\partial \vec{S}} &= \left[\frac{\partial P_i}{\partial \vec{S}_1} \mid \frac{\partial P_i}{\partial \vec{S}_2} \mid ... \mid \frac{\partial P_i}{\partial \vec{S}_N}\right] \\ \text{where } \frac{\partial P_i}{\partial \vec{S}_{i'}} &= \vec{0}_{i,i'}, \quad \forall i' \neq i \\ \vec{0}_{i,i'} &\in \mathbb{R}^{\|C_i\| \times \|C_{i'}\|} \\ i, i' &\in \{1, 2, ...N\} \\ \vec{S}_i &= [S_{i1} \mid S_{i2} \mid ... \mid S_{ij}]^T, \quad j \in C_i \end{aligned} \tag{16}$$

Because $\frac{\partial P_i}{\partial \vec{S}_{i'}} = \vec{0}_{i,i'}, \ \forall i' \neq i$, we can focus on computing $\frac{\partial P_i}{\partial \vec{S}_i}$.

$$\frac{\partial P_i}{\partial \vec{S}_i} = \begin{bmatrix} \frac{\partial P_{i1}}{\partial S_{i1}} & \frac{\partial P_{i1}}{\partial S_{i2}} & ... & \frac{\partial P_{i1}}{\partial S_{ij}} \\ \frac{\partial P_{i2}}{\partial S_{i1}} & \frac{\partial P_{i2}}{\partial S_{i2}} & ... & \frac{\partial P_{i2}}{\partial S_{ij}} \\ ... & ... & ... & ... \\ \frac{\partial P_{ij}}{\partial S_{i1}} & \frac{\partial P_{ij}}{\partial S_{i2}} & ... & \frac{\partial P_{ij}}{\partial S_{ij}} \end{bmatrix} \quad j \in C_i \tag{17}$$

The individual elements of $\frac{\partial P_i}{\partial \vec{S}_i}$ can be computed as follows:

$$\begin{aligned}
\frac{\partial P_{i\ell}}{\partial S_{i\ell}} &= \frac{\partial}{\partial S_{i\ell}}\left[\frac{\exp\left(S_{i\ell}\right)}{\sum_{k\in C_i}\exp\left(S_{ik}\right)}\right] \\
&= \frac{\left[\sum_{k\in C_i}\exp\left(S_{ik}\right)\right]\exp\left(S_{i\ell}\right) - \exp\left(S_{i\ell}\right)\exp\left(S_{i\ell}\right)}{\left[\sum_{k\in C_i}\exp\left(S_{ik}\right)\right]^2} \\
&= P_{i\ell} - \left(P_{i\ell}\right)^2 \\
\frac{\partial P_{i\ell}}{\partial S_{i\ell'}} &= \frac{\partial}{\partial S_{i\ell'}}\left[\frac{\exp\left(S_{i\ell}\right)}{\sum_{k\in C_i}\exp\left(S_{ik}\right)}\right] \quad \forall \ell' \neq \ell \\
&= \frac{\left[\sum_{k\in C_i}\exp\left(S_{ik}\right)\right] * 0 - \exp\left(S_{i\ell}\right)\exp\left(S_{i\ell'}\right)}{\left[\sum_{k\in C_i}\exp\left(S_{ik}\right)\right]^2} \\
&= -P_{i\ell}P_{i\ell'}
\end{aligned}$$

With these results in hand, $\frac{\partial P_i}{\partial \vec{S}_i}$ can be rewritten as follows:

$$\begin{aligned}
\frac{\partial P_i}{\partial \vec{S}_i} &= \begin{bmatrix} \frac{\partial P_{i1}}{\partial S_{i1}} & \frac{\partial P_{i1}}{\partial S_{i2}} & \cdots & \frac{\partial P_{i1}}{\partial S_{ij}} \\ \frac{\partial P_{i2}}{\partial S_{i1}} & \frac{\partial P_{i2}}{\partial S_{i2}} & \cdots & \frac{\partial P_{i2}}{\partial S_{ij}} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial P_{ij}}{\partial S_{i1}} & \frac{\partial P_{ij}}{\partial S_{i2}} & \cdots & \frac{\partial P_{ij}}{\partial S_{ij}} \end{bmatrix}, \quad j \in C_i \\
&= \begin{bmatrix} P_{i1} - \left(P_{i1}\right)^2 & -P_{i1}P_{i2} & \cdots & -P_{i1}P_{ij} \\ -P_{i1}P_{i2} & P_{i2} - \left(P_{i2}\right)^2 & \cdots & -P_{i2}P_{ij} \\ \cdots & \cdots & \cdots & \cdots \\ -P_{i1}P_{ij} & -P_{i2}P_{ij} & \cdots & P_{ij} - \left(P_{ij}\right)^2 \end{bmatrix} \\
&= \operatorname{diag}\left(P_i\right) - P_i {P_i}^T \\
&\text{where } \operatorname{diag}\left(\cdot\right) = \text{a diagonal matrix with the entries of the argument} \\
&\qquad\qquad \text{on the main diagonal.}
\end{aligned} \tag{18}$$

Returning finally to $\frac{\partial P}{\partial \vec{S}}$ we can convince ourselves that $\frac{\partial P}{\partial \vec{S}}$ is a block diagonal matrix formed by the submatricies $\frac{\partial P_i}{\partial \vec{S}_i}$ for $i \in \{1, 2, ..., N\}$.

$$\begin{aligned}
\frac{\partial P}{\partial \vec{S}} &= \begin{bmatrix} \frac{\partial P_1}{\partial \vec{S}} \\ \frac{\partial P_2}{\partial \vec{S}} \\ \cdots \\ \frac{\partial P_N}{\partial \vec{S}} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial P_1}{\partial \vec{S}_1} & \vec{0}_{1,2} & \cdots & \vec{0}_{1,N} \\ \vec{0}_{2,1} & \frac{\partial P_2}{\partial \vec{S}_2} & \cdots & \vec{0}_{2,N} \\ \cdots & \cdots & \ddots & \cdots \\ \vec{0}_{N,1} & \vec{0}_{N,1} & \cdots & \frac{\partial P_N}{\partial \vec{S}_N} \end{bmatrix}
\end{aligned} \tag{19}$$

PyLogit uses Equation 18 and 19 along with the diag function of Scipy.sparse in order to calculate $\frac{\partial P}{\partial \vec{S}}$.