Original Objective

$$\min_{\alpha_A \alpha_P \theta_1 \theta_2} (\alpha_A r_A + \alpha_P r_P) + \frac{1}{n_A n_P} \sum_{(i,j)\in M} (f(y_j^P \langle \theta, (\hat{x}_{i,j}, a_i^A)\rangle) + \max(y_j^P \langle \theta, (\hat{x}_{i,j}, a_i^A)\rangle - \alpha_P \kappa_P, 0) - \hat{\alpha}||x_i^A - x_j^P||)$$

Divide into two cases for $\hat{x}_{i,j}$ and $\hat{\alpha}$ where

$$\hat{x}_{i,j} = \begin{cases} x_j^P & \text{if } \alpha_A < \alpha_P \\ x_i^A & \end{cases}$$

$$\hat{\alpha} = \min(\alpha_A, \alpha_P)$$

The two objectives to optimize at the same time are

$$\begin{aligned}
&\Omega^A(\alpha_A, \alpha_P, \theta) \\
&= \min_{\alpha_A \alpha_P \theta_1 \theta_2} (\alpha_A r_A + \alpha_P r_P) + \frac{1}{n_A n_P} \sum_{(i,j)\in M} (f(y_j^P \langle \theta, (x_j^P, a_i^A)\rangle) + \max(y_j^P \langle \theta, (x_j^P, a_i^A)\rangle - \alpha_P \kappa_P, 0) - \alpha_A||x_i^A - x_j^P||) \\
&\Omega^P(\alpha_A, \alpha_P, \theta) \\
&= \min_{\alpha_A \alpha_P \theta_1 \theta_2} (\alpha_A r_A + \alpha_P r_P) + \frac{1}{n_A n_P} \sum_{(i,j)\in M} (f(y_j^P \langle \theta, (x_i^A, a_i^A)\rangle) + \max(y_j^P \langle \theta, (x_j^P, a_i^A)\rangle - \alpha_P \kappa_P, 0) - \alpha_P||x_i^A - x_j^P||)
\end{aligned}$$

Notation

The paper adopts notations like $f(t) = \log(1 + \exp(t))$ to avoid overhead of labels in the logistic loss. Here for multi-class settings, we use $g(x, a, \theta) : \{x, a\} \to y \in \mathbb{R}^C$ to denote our network function. Note that $g(x, a, \theta)$ only outputs the logits of each class and requires $\text{Softmax}(\cdot)$ to normalize the probability. We use $\text{CrossEntropy}(\cdot, y)$ to denote cross entropy loss with respect to the class label y. Then we can formulate our problem into

$$\begin{aligned}
&\Omega^A(\alpha_A, \alpha_P, \theta) \\
&= \min_{\alpha_A \alpha_P \theta_1 \theta_2} (\alpha_A r_A + \alpha_P r_P) + \frac{1}{n_A n_P} \sum_{(i,j)\in M} \big(\text{CrossEntropy}(\text{Softmax}(g(x_j^P, a_i^A, \theta)), y_j^P) + \max(g(x_j^P, a_i^A, \theta)_{y_j^P} - \alpha_P \kappa_P, 0) \\
&\qquad - \alpha_A||x_i^A - x_j^P||\big) \\
&\Omega^P(\alpha_A, \alpha_P, \theta) \\
&= \min_{\alpha_A \alpha_P \theta_1 \theta_2} (\alpha_A r_A + \alpha_P r_P) + \frac{1}{n_A n_P} \sum_{(i,j)\in M} \big(\text{CrossEntropy}(\text{Softmax}(g(x_i^A, a_i^A, \theta)), y_j^P) + \max(g(x_i^A, a_i^A, \theta)_{y_j^P} - \alpha_P \kappa_P, 0) \\
&\qquad - \alpha_P||x_i^A - x_j^P||\big)
\end{aligned}$$

Constrained by

$$C^A = \{(\alpha_A, \alpha_P, \theta) : ||\theta_1||_* \le \alpha_A + \alpha_P, ||\theta_2|| \le \kappa_A \alpha_A, \alpha_A < \alpha_P\}$$
$$C^P = \{(\alpha_A, \alpha_P, \theta) : ||\theta_1||_* \le \alpha_A + \alpha_P, ||\theta_2|| \le \kappa_A \alpha_A, \alpha_A > \alpha_P\}$$

A question not studied is the robustness of prediction with no auxiliary feature

Question

Why don't we just get a closed form solution from Lagrangian to solve the constraint problem?
Maybe try to start with the original form instead of multi-class form.

Note

The lagrangian in the paper that solves for projection is not used in the code base. The code base simply solves the optimization problem for projection with constraint requirements.