Smart Allocation Algorithm for Patient-Therapist Matching

For Therapists (t):

- t_spec: One-hot encoded vector of specializations
- t_exp: Years of experience (normalized)
- t_avail: Availability score (e.g., total available hours per week, normalized)
- t_load: Current patient load (normalized inverse, so higher is better)
- t_rating: Average rating (normalized)

For Patients (p):

- p_diag: One-hot encoded vector of diagnoses
- p_pref: One-hot encoded vector of preferred therapy types
- p_sev: Severity score (normalized)

Similarity Calculation

cosine similarity to calculate how well a therapist matches a patient's needs:

sim(t, p) = cosine_similarity(t_spec, p_diag) * w1 + cosine_similarity(t_spec, p_pref) * w2

w1 and w2 are weights that can be tuned.

Therapist Scoring

For each therapist, calculate a score for a given patient:

score(t, p) = sim(t, p) * (1 + log(t_exp + 1)) * t_avail * (1 - t_load) * t_rating * (1 + log(p_sev + 1))

Allocation Algorithm

For each patient: a. Calculate similarity and initial score for all available therapists b. ML model to get a refined ranking score c. Sort therapists by the ranking score

Continuous Learning

feedback loop where the outcomes of therapist-patient matches (e.g., patient progress, session ratings) are used to continuously train and improve the ML model.

Notation:

Let T be the set of all therapists and P be the set of all patients.

For each $t \in T$ and $p \in P$, we want to maximize:

$\Sigma(t,p) \in M$ [ranking_score(t, p)]

subject to:

- $\forall t \in T$: $\Sigma(p:(t,p) \in M)\ 1 \leq$ max_load(t)
- $\forall p \in P$: $\Sigma(t:(t,p) \in M)\ 1 \leq 1$

*M is the set of matched therapist-patient pairs.