

形式知化による座席配置 の最適化・自動化

～「感覚」から「数値」へ。生徒と講師の最高の相性を導き出す～

はじめに、

- 座席決めをプログラムで自動化しようとかそういうことではありません。
- 人間の相性という極めて複雑な題材に対してデータサイエンスを活用してみようという試みを説明いたします。
- 注意しなければならないのは、これから出てくることは一つの方法であって正解ではありません。
- また、次に出てくる「座席配置の最適化・自動化システム」というのは、本研修のために作成したものです。あくまで「感覚で行っていたものをデータ化したらどうなるのだろう」という趣旨のものです。

概要

座席配置の最適化・自動化システムの概要

理論：人間を「ベクトル（矢印）」で捉える

- 生徒や講師を「文系・理系」といった単純なラベルではなく、「多次元のベクトル」として定義しました。

1. 4つの学力特性

従来の5教科を、脳の使い方や考え方の特性で4つに再分類しました。

- ✓ 数理・論理：数学、算数、プログラミング（論理的思考力）
- ✓ 英語・語学：英語全般（語学センス）
- ✓ 自然科学：物理、化学、生物（現象理解＋暗記）
- ✓ 文脈・教養：国語、社会（文脈読解、背景知識）

2. 3つの心理・行動特性

成績だけでなく、性格的な相性を測るための指標です。

- 外向性（Extraversion）：おしゃべりか、無口か。
- 受容力/繊細さ（Sensitivity）：共感的か、ドライ（論理的）か。
- 独立性（Independence）：手がかかるか、自走できるか。

3. 「位置」と「矢印」の概念

- このシステムは、これらの数値を複雑な計算（主成分分析）にかけ、6次元の情報を「3次元空間」に変換して扱います。
- 位置（Position）：その人の「大まかなキャラクター」。近くにいる人は似たタイプです。
- 矢印（Vector）：その人の「隠れたこだわり・特性」。

重要：位置が近くても、矢印が逆を向いているペアは、「一見合いそうだが、深い部分で噛み合わない」可能性があります。

導入ツール：3つのアプリケーション

座席配置の最適化・自動化をするために、Pythonというプログラミング言語を用いて3つのアプリケーションを開発しました。

1. データ管理アプリ (data.py)

講師や生徒の情報を入力・管理をするデータベースです。

- 機能：直感的なGUIで、誰でも簡単に「新規登録」「修正」「削除」が可能です。
- 運用：新しい生徒が入塾した際や、講師の特徴が変わった際にここで数値を更新します。データはsample.cvdというファイルに蓄積されます。

アプリケーション例

Compatibility Vector Data Editor

登録

講師A (理系エース)	名前:	講師A (理系エース)
講師B (英語特化)	Student/Teacher:	Teacher
講師C (文系オール)	数理・論理 (数学全般) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.9
講師D (数学・物理)	英語・語学 (英語全般) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.6
講師E (心理・生物)	自然科学 (物理・化学・生物) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.9
生徒1 (理系志望)	文脈・教養 (国語・社会) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.2
生徒2 (英語苦手)	外向性 0.0(無口・内向) ~ 1.0(おしゃべり・外向):	0.5
生徒3 (国語のみ得意)	傷つきやすさ/受容力 0.0(ドライ・論理的) ~ 1.0(共感的・繊細):	0.3
生徒4 (全教科平均)	独立性 0.0(依存的・手がかる) ~ 1.0(自律的・放置OK):	0.8
生徒5 (難関国立)	性別 (0:女性, 1:男性):	1.0
生徒6 (勉強嫌い)		

更新

追加 削除

保存

Compatibility Vector Data Editor

登録

講師A (理系エース)	名前:	生徒3 (国語のみ得意)
講師B (英語特化)	Student/Teacher:	Student
講師C (文系オール)	数理・論理 (数学全般) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.1
講師D (数学・物理)	英語・語学 (英語全般) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.3
講師E (心理・生物)	自然科学 (物理・化学・生物) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.1
生徒1 (理系志望)	文脈・教養 (国語・社会) (0.0: 不可/苦手 ~ 1.0: 指導可能/得意):	0.9
生徒2 (英語苦手)	外向性 0.0(無口・内向) ~ 1.0(おしゃべり・外向):	0.4
生徒3 (国語のみ得意)	傷つきやすさ/受容力 0.0(ドライ・論理的) ~ 1.0(共感的・繊細):	0.9
生徒4 (全教科平均)	独立性 0.0(依存的・手がかる) ~ 1.0(自律的・放置OK):	0.4
生徒5 (難関国立)	性別 (0:女性, 1:男性):	0.0
生徒6 (勉強嫌い)		

更新

追加 削除

保存

2. 3D可視化ビューワー (main.py)

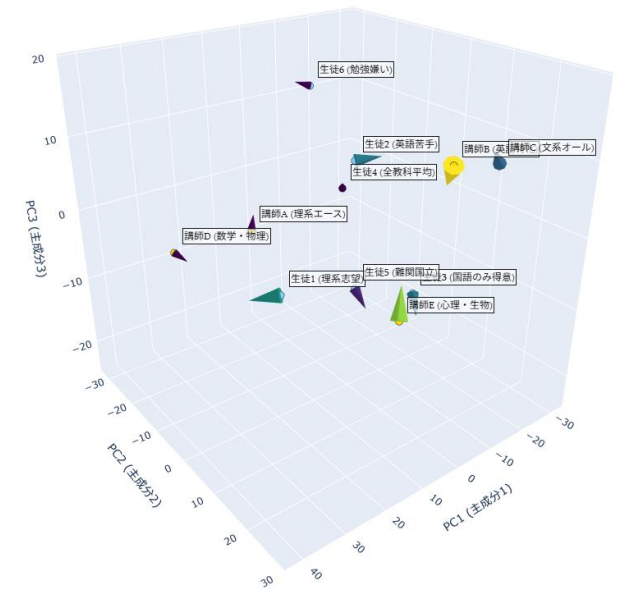
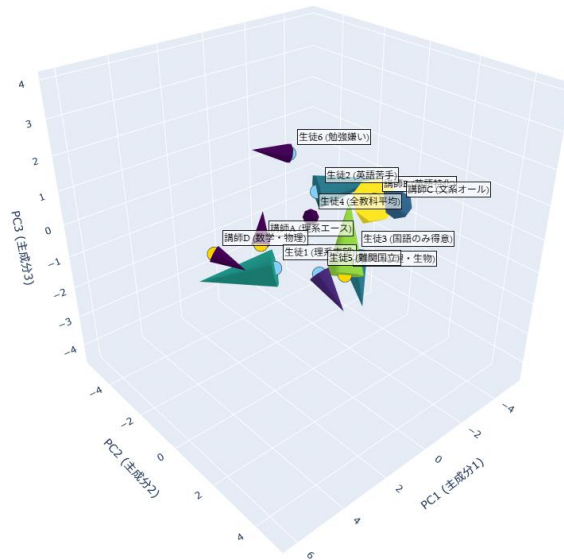
- 校舎全体の「人間関係地図」を立体的に表示します。
- 3Dマップ：マウスでグリグリ動かして、誰と誰が近いかを確認できます。
- 10x (Wide) モード：人が密集して見づらい場合、ボタン1つで空間を10倍広げて詳細を確認できます。
- 活用：引継ぎの際、「この生徒はこのエリアにいるから、あの先生たちが合いそうだね」といった視覚的な共有に使います。

アプリケーション例

Normal (1x) Wide (10x)



● Teacher
● Student



3. マッチングシステム (match.py)

- 特定の生徒に対して、数学的に相性が良い講師を瞬時に提案します。
- ベスト3提案：指定した生徒に対し、「空間的な距離」と「矢印の向き」を計算し、相性の良い順にランキングを表示します。
- α (アルファ) 値の調整：スライダーを動かすことで、「大まかな相性」を重視するか、「細部のこだわり (ベクトル)」を重視するかを調整できます。
- 実際の座席決めの際、迷ったときの意思決定サポートとして使用します。

アプリケーション例

AI Compatibility Matcher

検索条件

対象の生徒を選択: 生徒5 (雑関国立)

隠れた特性(ベクトル)の重視度 α : 1.0

AI推奨講師ベスト3

順位	講師名	総合距離(相性)	位置差	ベクトル差	得意科目
1位	講師A (理系工-ス)	3.4064	2.1419	1.2645	数
2位	講師D (数学・物理)	4.0338	3.6477	0.3861	数
3位	講師C (文系オール)	5.1283	3.8200	1.3083	社国
4位	講師E (心理・生物)	5.6470	3.6923	1.9547	理
5位	講師B (英語特化)	6.0257	4.2021	1.8236	英

※ 総合距離が 0 に近いほど「AI判定による相性」が良いことを示します。
※ 位置差: 大まかなキャラの一致度
※ ベクトル差: こざわりや細部の感性の一致度

AI Compatibility Matcher

検索条件

対象の生徒を選択: 生徒5 (雑関国立)

隠れた特性(ベクトル)の重視度 α : 5.0

AI推奨講師ベスト3

順位	講師名	総合距離(相性)	位置差	ベクトル差	得意科目
1位	講師D (数学・物理)	5.5780	3.6477	0.3861	数
2位	講師A (理系工-ス)	8.4644	2.1419	1.2645	数
3位	講師C (文系オール)	10.3616	3.8200	1.3083	社国
4位	講師B (英語特化)	13.3201	4.2021	1.8236	英
5位	講師E (心理・生物)	13.4658	3.6923	1.9547	理

※ 総合距離が 0 に近いほど「AI判定による相性」が良いことを示します。
※ 位置差: 大まかなキャラの一致度
※ ベクトル差: こざわりや細部の感性の一致度

実際の運用フロー

データ入力 (input)

- 面談や初回授業終了後、担当講師が生徒の特性（4つの学力＋3つの性格）をdata.pyに入力

可視化・共有 (Visualize)

- ミーティング時などにmain.pyを開き、校舎全体のバランスや、孤立している（対応が難しい）生徒がいないかを確認する。

マッチング実行 (Match)

- 座席を組む際、判断に迷う生徒がいればmatch.pyを起動。
- 提案された上位の講師の中から、シフトや科目（ハード制約）が合う講師を選定する。

フィードバック (Feedback)

- 「システムの提案通り組んだらうまくいった」「逆にうまくいかなかった」という結果をもとに、データの数値を微調整し、精度を高めていく。

ここからが本題

大規模相性モデルの数学的仕様書

1. 人間のベクトル空間定義 (Vector Representation)

講師及び生徒という「実体」を、 n 次元のユークリッド空間上の点（ベクトル）として定義する。

ある人物 P の特性ベクトル x を以下のように定義する。

$$x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$$

ここで、本システムにおける初期次元数 $n = 8$ であり、各成分は以下の要素に対応する。

x_1 : 数理・論理スキル (Score_Math_Logic)

x_2 : 英語・語学スキル (Score_English)

x_3 : 自然科学スキル (Score_Science)

x_4 : 文脈・教養スキル (Score_Humanities)

x_5 : 外向性 (Extraversion)

x_6 : 受容性 (Sensitivity)

x_7 : 独立性 (Independence)

x_8 : 性別コード (Gender_Code)

1. 式の解説

$$\boldsymbol{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$$

1. \boldsymbol{x} (ベクトル)

特定の1人の人間を表すデータそのものです。数学では太字の小文字でベクトルを表すのが通例です。

2. $[x_1, x_2, \dots, x_n]^T$ (成分と転置)

x_1 から x_n までの n 個の数値が並んでいることを示します。

右上の T は「転置 (Transpose)」を意味します。通常、論文や数式ではベクトルを「縦長の列」として扱いますが、文章中では書きにくいため、「横に書いてから T で縦にする」という表記法を使います。

3. $\in \mathbb{R}^n$ (空間の所属)

\in : 「～に含まれる (所属する)」という記号。

\mathbb{R} : 「実数 (Real numbers)」の集合。0.1, -5.2, $\sqrt{2}$ などを含む連続した数です。

n : 「次元数」。今回は $n = 8$ です。

つまり、「この人間データは、8個の実数で構成される8次元空間の中の1点である」と定義しています。

2. 前処理：標準化（Z-score Normalization）

各特徴量のスケールや分散の偏りを排除し、等価に扱うために、標準化（Z-score normalization）を行う。データセット全体の平均を μ 、標準偏差を σ としたとき、正規化されたベクトル z は以下で与えられる。

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \quad (i = 1, \dots, n)$$

これにより、すべての次元は平均0、分散1の正規分布に近似され、距離計算における各パラメータの寄与度が公平化される。

2. 式の解説

$$z_i = \frac{x_i - \mu_i}{\sigma_i}$$

1. x_i (Raw Value)

個人の数値です。例：「講師Aの数学スキル 0.9」。

2. μ_i (Mean)

その集団（全講師・全生徒）における平均値。例：「塾全体の数学スキルの平均が 0.5」。

3. σ_i (Standard Deviation)

標準偏差（データのばらつき具合）。例：「みんな大体0.5付近なら σ は小さい」「0.1の人も0.9の人もたくさんいるなら σ は大きい」。

4. z_i (Z-score)

標準化された値（標準得点）。

3. 次元圧縮：主成分分析（PCA）

$n = 8$ 次元の情報を、可視化および「主要因」と「微細因」に分類するため、主成分分析（PCA）を用いて $k = 6$ 次元へ写像する。

共分散行列 Σ の固有値分解を行い、固有値の大きい順に並べた固有ベクトル（主成分）への射影を行う。

$$w = V^T z \in \mathbb{R}^6$$

この6次元ベクトル w を、次の2つの3次元ベクトル空間に分割して解釈する。

3. 1. 位置空間 (Position Space) p

第1～第3主成分。データの分散（情報のばらつき）の大部分を説明する次元。これを「個人の
大まかな特性（キャラクター）」と定義する。

$$p = [w_1, w_2, w_3]^T$$

3. 2. 特性ベクトル空間 (Trait Vector Space) v

第4～第6主成分。主要因では説明しきれない「残差」や「微細なこだわり」。これを個人の内
部的な指向性（矢印）と定義する。

$$v = [w_4, w_5, w_6]^T$$

これにより、人物 P は、空間上の座標 p と、そこから伸びる矢印 v を持つ点としてモデル化される。

4. 相性評価関数 (Compatibility Metric Function)

生徒 S と講師 T の相性の良さは、この高次元空間における「距離の近さ」として定義される。距離が近いほど相性が良い（コストが低い）。

本モデルでは、以下の損失関数 (Loss Function) $D(S, T)$ を最小化することを目的とする。

$$D(S, T) = \|\mathbf{p}_S - \mathbf{p}_T\|_2 + \alpha \|\mathbf{v}_S - \mathbf{v}_T\|_2$$

4. 式の解説

$$D(S, T) = \|\mathbf{p}_S - \mathbf{p}_T\|_2 + \alpha \|\mathbf{v}_S - \mathbf{v}_T\|_2$$

位置項 $\|\mathbf{p}_S - \mathbf{p}_T\|_2$

- 3次元座標間のユークリッド距離。
- 「大まかなタイプの一貫度」を測る。ここが遠いと、根本的にキャラが合わない。

ベクトル項 $\|\mathbf{v}_S - \mathbf{v}_T\|_2$

- 微細特性（矢印）間のユークリッド距離。
- 「細部の感性やこだわりの一貫度」を測る。

ハイパーパラメータ α

- 「隠れた特性」をどれだけ重視するかを決定する重み係数。
- $\alpha = 0$ のとき：表面的な相性のみを見る。
- α 大のとき：細部の感性の一致を強く要求する（こだわりが強い生徒向け）。
- 実装済みのアプリケーション（match.py）では $0.0 \leq \alpha \leq 5.0$ で調整可能。

5. 最適化問題としての定式化 (Optimization Formulation)

塾全体の座席決定は、以下の制約付き最適化問題に帰着される。

- 定数 :

生徒集合 $S = \{1, \dots, N_S\}$

講師集合 $T = \{1, \dots, N_T\}$

コスト行列 $C_{ij} = D(S_i, T_j)$

- 決定変数 :

$x_{ij} \in \{0, 1\}$ (生徒 i を講師 j に割り当てるなら 1)

- 目的関数 :

$$\text{Minimize } Z = \sum_{i \in S} \sum_{j \in T} C_{ij} x_{ij}$$

- 制約条件 (Constraints) :

1. 全ての生徒は必ず1人の講師に割り当てられる : $\sum_j x_{ij} = 1 \quad (\forall i)$

2. 講師1人当たりの生徒数は最大2名まで : $\sum_i x_{ij} \leq 2 \quad (\forall j)$

まとめ

本システムは「正解」ではなく「視点」を与えるパートナー

「暗黙知」から「形式知」へ

- これまでベテラン講師の頭の中にしかなかった「相性の感覚（暗黙知）」をデータ化（形式知化）してみました。
- これにより、経験の浅い講師でも高精度なマッチング検討が可能になり、塾全体の資産となります。

システムが「計算」し、人間が「決断」する

- このシステムは講師の仕事を奪うものではありません。
- システムは膨大な組み合わせ計算を担当する「コンパス」です。
- 実際に生徒の表情を見て、最終的な座席を決定するのは、私たち講師「キャプテン」です。

システムを育てるのは「現場の声」

- このモデルは未完成です。
- 「システムの提案通りにして良かった」「逆に合わなかった」という皆さんのフィードバックこそが、システムの精度を高める教師データになります。

最後に、

- 今回は、「座席配置の最適化・自動化システム」というものを題材に、感覚で管理するデータを数学を用いて形式知化して管理する方法について説明しました。
- 今回扱った、大規模相性モデルというのは私が作ったものなので、全く同じ数理モデルは現時点では存在しません。しかし、「人間の好みをベクトルに変換し、数学的な距離でマッチングする」という技術自体は「レコメンデーションシステム（推薦システム）」や「マッチングアルゴリズム」と呼ばれ、世界中で広く活用されています。
- 本研修用に作成したPythonプログラム（data.py・main.py・match.py）については、以下のサイトで公開しております。ぜひ実際に試してみてください。
- リンク：https://slrte.com/slrte/investigation/HumanEducation/HE_00002.html

