LLM 間と問題間の類似度制約を加えた LLM の性能推定

田村拓也　矢野太郎　榎本昌文　小山田昌史
NEC データサイエンスラボラトリー
{tamura-takuya,taro_yano,masafumi-enomoto,oyamada}@nec.com

概要

本研究では、LLM が与えられた問題を適切に解けるかを推定するタスクにおいて、推定モデルの学習に含まれない新規の LLM や問題に対しても高精度に推定する手法を提案する。提案手法では、従来の行列分解に基づくアプローチに加えて、問題文の埋め込みやモデルの来歴などの補助情報から得られる類似度を考慮した**類似度制約項**を損失関数に導入する。その結果、提案手法では学習時に利用した既知の LLM による新規問題の平均的な解決性能の推定において 20.2%、新規 LLM の既知の問題に対する平均的な性能の推定において 15.6%の誤差軽減を達成した。

1　はじめに

大規模言語モデル (LLM) の性能推定タスクは、所与の LLM が特定の問題集合に対してどの程度の性能を発揮するかを推定するタスクである。本タスクは有用な応用先を多く持つ。例えば、LLM のベンチマーク評価において少数サンプルから全体スコアを推定する手法として利用されている [1, 2]。また、複数の LLM から最も良い回答を得るために、性能を事前推定し、最適な LLM にタスクを割り当てる Routing[3] にも応用される。さらに LLM の開発過程において、訓練データやハイパーパラメータの異なる多数の候補モデルから効率的に高性能なモデルを選択する際にも有用である。

この問題に対する既存研究として、主に二つのアプローチが提案されている。一つ目は、複数の LLM の評価結果から LLM と問題の特徴量を算出し、IRT モデル [4] 等を仮定して性能を推定する手法である [1, 2]。IRT モデルでは、観測済みの評価スコアのみから特徴量を学習する。そのため全く評価スコアが得られていない新規 LLM(図 1 テスト C,D) や新規の問題 (図 1 テスト B,D) については、特徴量を学習することができず推定性能が低くなるという課題がある。二つ目は、個別の LLM ごとに過去に評価した類似問題における性能を参照し、性能を推定する手法である [3]。この手法は新規の問題に対しても性能推定が可能であるものの、予め十分な量の問題について評価スコアが得られていない場合に適切な類似問題が見つからない課題がある。いずれの手法も LLM の評価実績のみに基づいて予測を行うため、評価値が十分に得られない問題・LLM については十分な推定精度が得られない。本研究では、新しい問題や新しい LLM を含む場合であっても、LLM に追加で問題を解かせることなく高精度な性能推定を実現することを目的とする。これを実現するためには、評価スコアが得られない新規問題・LLM について、特徴量の学習を助ける教師シグナルが必要である。そこで提案手法では、問題のプロンプトや LLM の属性や系統情報から算出した類似度を活用する。具体的には、特徴量間の類似度が、算出された類似度に近づくように類似度制約項を推定モデルの損失関数に導入する。

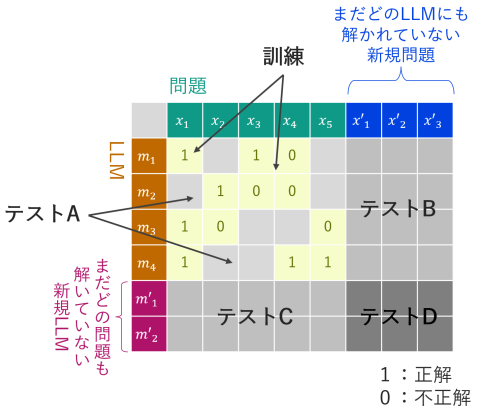

図 1　データセットの分割方法（観測済みの LLM-問題ペアを訓練データとし、未観測ペアを A,B,C,D に分割してテストデータとする）

RQ-1: LLM の性能推定タスクにおいて、LLM や問題についての補助情報から得られた類似度情報を利用することで推定性能を向上できるか？　通常の行列分解による推定手法に比べ、補助情報に基づく

類似度制約項を利用する場合には、既知 LLM が新規問題集合を解く際 (図 1 テスト B) の平均的な性能推定において 20.2%、新規 LLM 集合が既知の問題を解く場合 (図 1 テスト C) において 15.6%の誤差低減を達成した。一方、新規の LLM が新規の問題を解く場合 (図 1 テスト D) に対しては、性能の向上が見られなかった。

RQ-2: 理想的な問題/LLM 類似度が入手できた場合に、どの程度性能推定の向上を行えるか？ 理想的な問題/LLM 類似度を評価スコアの類似度と定めて実験を行ったところ、通常の行列分解と比較して、既知 LLM が新規問題を解く場合 (図 1 テスト B) に 26.7%、新規 LLM が既知問題を解く場合 (図 1 テスト C) に 34.0%改善することを確認した。これは現状の補助情報から構築された類似度行列に改善の余地があることを示唆している。

2 手法

本研究では、性能評価が未実施の LLM と問題のペアに対して事前に性能を推定する手法を提案する。本節ではまず問題設定を形式的に定義し、続いて提案手法の詳細を説明する。

2.1 問題設定

LLM の集合を $\mathcal{M}$、問題の集合を $\mathcal{X}$ とする。問題 $k \in \mathcal{X}$ を LLM$i \in \mathcal{M}$ が解いた時の評価スコアを $z_{i,k} \in \{0, 1\}$ と定義する。ここで、評価スコアは問題が正しく解けた場合は 1、そうでない場合は 0 となる二値変数である。一部の LLM と問題のペアについては既に真の評価スコアが得られており、これらのデータを用いて未評価のペアの評価スコアを高精度に推定することが本研究の目的である。

2.2 行列分解による特徴量算出と評価スコアの推定

本研究では、既存の IRT モデルや行列分解アプローチに倣い、評価済みペアのスコアを用いて LLM や問題の潜在特徴量を算出したのち、それらに基づいて未評価ペアのスコアを推定する二段階アプローチを採用する。LLM i の特徴量を $m_i \in \mathbf{R}^d$、問題 k の特徴量を $x_k \in \mathbf{R}^d$ とする。これらの特徴量と評価スコアの関係を $z_{i,k} = f(m_i, x_k)$ と仮定する。ここで、f は特徴量から評価スコアを予測する関数である。LLM 特徴量行列を $M = [..., m_i, ...]^T \in \mathbf{R}^{|\mathcal{M}|\times d}$、問題特徴量行列を $X = [..., x_k, ...]^T \in \mathbf{R}^{|\mathcal{X}|\times d}$ と定義する。また、評価スコア行列を $Z \in \mathbf{R}^{|\mathcal{M}|\times|\mathcal{X}|}$ とする。本研究では $Z = MX^T$ となる行列分解アプローチをベース手法として位置付けて、補助情報に基づく制約項の効果を検証する。そのため、f を特徴量の内積 $f(m_i, x_k) = m_i x_k^T$ として定義する。最適化問題の目的関数として、評価済みペアに対する BCE 損失 L_{BCE} と L2 正則化項 L_{reg}、問題と LLM の補助情報に基づく正則化項 L_M, L_X を組み合わせた以下の損失関数を定義する：

$$L = L_{\text{BCE}} + \lambda_{\text{reg}} L_{\text{reg}} + \lambda_X L_X + \lambda_M L_M \tag{1}$$

$$= \sum_{(i,k)\in\Omega} \text{BCE}(z_{ik}, \hat{z}_{ik}) + \lambda_{\text{reg}}(|M|_F^2 + |X|_F^2) \tag{2}$$

$$+ \lambda_X L_X + \lambda_M L_M \tag{3}$$

ここで、Ω は評価済みペアの集合を表す。

2.3 補助情報を活用した制約

評価スコアの推定精度向上のため、問題と LLM の補助情報を活用した制約を導入する。問題に関しては問題文や情報源などの付随情報が、LLM に関してはモデルのアーキテクチャ、パラメータ数、事前学習データ、ファインチューニング手法などの技術的特性が存在する。これらの補助情報は評価スコアと密接な関係があると考えられる。例えば、類似した問題文を持つ問題同士は難易度も近く、また同じベースモデルから派生した LLM 同士は問題解決能力においても類似していると考えられる。この仮説に基づき、**LLM や問題の特徴量の類似度が補助情報の類似度と近くなるような損失**を導入する。具体的には、問題間の類似度行列 S_X と LLM 間の類似度行列 S_M を定義する。ここで、問題 k と問題 l の類似度を S_X の (k, l) 成分に、LLM i と LLM j の類似度を S_M の (i, j) 成分とする。これらの類似度行列を用いて、次のように損失関数を定義する：

$$L_X = \|XX^T - \sqrt{S_X}\|_F^2 \tag{4}$$

$$L_M = \|MM^T - \sqrt{S_M}\|_F^2 \tag{5}$$

ここで、$\|\cdot\|_F$ はフロベニウスノルムを、$\sqrt{S_X}$ と $\sqrt{S_M}$ はそれぞれ S_X と S_M の主平方根を表す。

理想的な類似度行列 工学的な有用性を考慮すると、類似度行列を使って定義された損失関数を最適化することで得られた M, X を使えば、全ての未観測ペアのスコアを正しく予測できることが理想である。そこで、理想的な損失関数 L_X, L_M および理想的な類似度行列 S_X, S_M を、X, M に関して損失を最小化した際に $Z = MX^T$ と再構成できるものと定義

する。ここで、$S_X = Z^T Z, S_M = ZZ^T$ と置くとこの定義を満たす[1]ため、本研究ではこれらを理想的な類似度行列として扱って分析を行う。この意味において提案法の損失関数の最適化は、**LLM や問題の特徴量の類似度を評価スコアの類似度と近づける**との解釈ができる。実際の運用では、LLM や問題の補助情報から工学的に類似度行列 S_X, S_M を構築することとなるが、$S_M \sim (ZZ^T)^2, S_X \sim (Z^T Z)^2$ となるものを利用すればよい。

3 実験

3.1 実験設定

本研究では、Hugging Face Open LLM Leaderboard[5] から収集した評価データを使用した。対象は 1,704 の LLM に対し、BBH[6], GPQA[7], MATH[8], MMLU-Pro[9], MuSR[10] の 5 つのデータセットからなる 21,065 インスタンスである。訓練用としてランダムに 1,000 モデルと 1,000 問題を抽出し、図 1 に示すように一部のペアを訓練データ、残りをテストデータ A とした。訓練データには各 LLM および各問題に対して均等に T のペアを割り当て、T^2 のペアを利用した。学習データの数を変えた時の性能を比較するため $T = 30, 100, 300, 1000$ のパターンについてそれぞれ実験を行った。さらに、残りの 1,604 モデルと 20,065 問題からテストデータ B（既知 LLM ×新規問題）、C（新規 LLM ×既知問題）、D（新規 LLM ×新規問題）を構築し、それぞれ 1,000,000 ペアずつ無作為抽出して合計最大 4,000,000 ペアのデータとした（表 1）。

データ	構成	ペア数
訓練データ	-	T^2
テストデータ A	既知 LLM× 既知問題	1,000,000 - T^2
テストデータ B	既知 LLM× 新規問題	1,000,000
テストデータ C	新規 LLM× 既知問題	1,000,000
テストデータ D	新規 LLM× 新規問題	1,000,000

表 1　データセットの構成

3.2 評価指標

本研究では、真のスコア行列 Z と推定スコア行列 $\hat{Z}$ の間の RMSE を用いて評価を行った。ここで、J は LLM の数、K は問題数、z_{ik} は i 番目の LLM $(i = 1, \ldots, J)$ の k 番目の問題 $(k = 1, \ldots, K)$ に対する真のスコア、$\hat{z}_{ik}$ は推定スコアを表す。具体的には、以下の 3 つの観点から評価を実施した。

ペアワイズ RMSE　個々の LLM と問題の組み合わせごとのスコア予測精度を測る指標：

$$\mathrm{RMSE_p} = \sqrt{\frac{1}{JK}\sum_{i=1}^{J}\sum_{k=1}^{K}(z_{ik} - \hat{z}_{ik})^2} \tag{6}$$

この指標は、すべての LLM と問題ペアにおける平均誤差を測ることで、細かい単位での予測精度を評価する。

モデル平均の RMSE　各問題について、複数の LLM の平均性能をどれだけ正確に予測できたかを測る指標：

$$\mathrm{RMSE_m} = \sqrt{\frac{1}{J}\sum_{i=1}^{J}\left(\frac{1}{K}\sum_{k=1}^{K}(z_{ik} - \hat{z}_{ik})\right)^2} \tag{7}$$

この指標では、各問題固有の誤差は先に平均化されるため軽減され、各 LLM 固有の誤差は二乗誤差により強調される。したがって、概ね LLM 特徴量の質を捉える指標となっている。

問題平均の RMSE　各 LLM について、複数の問題に対する平均性能がどれだけ正確に推定されたかを測る指標：

$$\mathrm{RMSE_x} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{J}\sum_{i=1}^{J}(z_{ik} - \hat{z}_{ik})\right)^2} \tag{8}$$

同様に、概ね問題特徴量の質を捉える指標である。

3.3 補助情報から類似度行列を作成する手順

本研究では、問題の類似度と LLM の類似度という 2 つの補助情報を活用する。問題の類似度計算には、Multilingual E5 モデル [11] を用いて問題のプロンプト文を埋め込みベクトルに変換し、それらのコサイン類似度を計算することで問題間の類似度を定量化した[2]。LLM の類似度は、HuggingFace 上のモデルカードに記載された情報を基に、特徴ベースと系統関係ベースの 2 つの観点からそれぞれ計算を行い、それらを統合することで最終的な類似度行列を得た。詳細な特徴量設計と系統関係の定量化については付録 A に記述する。

1）Z を特異値分解したものを $Z = U\Sigma V^T$ とし、全ての特異値が相異なる値をとるとする。ここで、$S_X = Z^T Z$ かつ $S_M = ZZ^T$ のときに $L_X = L_M = 0$ となるのは $X = U\Sigma^{1/2} Q_X, M = V\Sigma^{1/2} Q_M$（ただし、$Q$ は任意の直交行列）のみであるため、BCE 損失により $Q_M Q_X^T = I$ となるものを見つけられれば、$MX^T = Z$ より行列 Z を完全に復元することができる。

2）embaas/sentence-transformers-multilingual-e5-base モデルを使用した。

		A（既知 LLM ×既知問題）			B（既知 LLM ×新規問題）			C（新規 LLM ×既知問題）			D（新規 LLM ×新規問題）		
T	推定手法	$RMSE_p$	$RMSE_m$	$RMSE_x$	$RMSE_p$	$RMSE_m$	$RMSE_x$	$RMSE_p$	$RMSE_m$	$RMSE_x$	$RMSE_p$	$RMSE_m$	$RMSE_x$
30	Mean Score	0.4137	0.0670	0.1385	0.4749	0.0972	0.2724	0.4791	**0.1010**	0.2713	**0.4769**	**0.0939**	**0.2786**
30	MF	**0.3945**	**0.0567**	**0.0982**	0.4694	0.0654	0.2734	0.4147	0.1039	0.1266	0.4784	0.1013	0.2812
30	SimRMF	0.3988	0.0591	0.1014	**0.4687**	**0.0597**	**0.2723**	**0.4086**	0.1013	**0.1048**	0.4778	0.0976	0.2798
30	OracleSimRMF	0.3778	0.0520	0.0839	0.4277	0.0578	0.2176	0.3833	0.0768	0.0863	0.4338	0.0785	0.2241
100	Mean Score	0.4123	0.0555	0.1306	0.4765	0.0988	**0.2745**	0.4732	0.0897	0.2600	**0.4751**	**0.0928**	**0.2758**
100	MF	**0.3891**	**0.0425**	**0.0823**	0.4696	0.0576	0.2758	0.4080	0.0897	0.1007	0.4759	0.0971	0.2772
100	SimRMF	0.3974	0.0450	0.0883	**0.4690**	**0.0507**	0.2751	**0.4049**	0.0897	**0.0871**	0.4758	0.0959	0.2767
100	OracleSimRMF	0.3773	0.0388	0.0697	0.4284	0.0484	0.2213	0.3813	0.0688	0.0691	0.4333	0.0781	0.2235
300	Mean Score	0.4072	0.0503	0.1363	0.4755	0.0953	**0.2747**	0.4775	**0.0954**	0.2723	**0.4767**	**0.0987**	**0.2751**
300	MF	**0.3824**	**0.0354**	**0.0827**	0.4681	0.0461	0.2755	0.4059	0.0963	0.1046	0.4777	0.1036	0.2768
300	SimRMF	0.3902	0.0379	0.0899	**0.4676**	**0.0389**	0.2750	**0.4020**	0.0955	**0.0883**	0.4775	0.1021	0.2763
300	OracleSimRMF	0.3693	0.0311	0.0706	0.4252	0.0370	0.2182	0.3773	0.0719	0.0691	0.4326	0.0811	0.2195
1000	Mean Score	-	-	-	0.4763	0.1001	0.2740	0.4820	**0.0971**	0.2848	0.4760	0.0959	0.2774
1000	MF	-	-	-	0.4669	0.0341	0.2729	0.4047	0.0977	0.1122	**0.4755**	0.0935	0.2765
1000	SimRMF	-	-	-	**0.4665**	**0.0272**	**0.2728**	**0.3995**	**0.0971**	**0.0915**	0.4756	**0.0931**	**0.2763**
1000	OracleSimRMF	-	-	-	0.4240	0.0250	0.2158	0.3737	0.0730	0.0709	0.4309	0.0723	0.2195

表 2　各手法の性能推定精度の比較。T は訓練データにおいて各 LLM が解いた問題数。MeanScore は観測済ペアの平均スコアを推定値とする手法を、MF は BCE 損失にのみ基づく行列分解手法を、SimRMF は類似度制約を含む行列分解手法を、OracleSimRMF は理想的な類似度行列が入手できた場合を表す。また、$T = 1000$ では既知モデル×既知問題について全て訓練データとして利用されているためテストデータ A は存在しない。

3.4　実験結果と考察

表 2 に実験結果を示す。

類似度制約項の有効性　提案手法 SimRMF の性能を BEC 損失のみに基づく行列分解手法 MF や評価済ペアの平均スコアによる推定手法 MeanScore と比較すると、個々の LLM と問題のペアに対する推定誤差 RMSEp では、わずかながら改善が見られた。一方で、以下の 2 つの状況において顕著な性能向上が確認された：(1) 既知 LLM に対する新規問題の組み合わせ（テスト B）における $RMSE_m$ では、すべての訓練サイズ T において一貫して改善が見られ、特に T=1000 のケースでは従来手法の 0.0341 から 0.0272 へと 20.2%の誤差を軽減した。一方で、$RMSE_x$ ではわずかな改善しか見られない。これは、各既知 LLM の特徴量の質は大きく向上したが、各新規問題特徴量の質は依然として低いことを示唆している。(2) 新規 LLM に対する既知問題の組み合わせ（テスト C）における $RMSE_x$ でも同様に、観測数の増加とともに改善の幅が大きくなり、T=300 のケースで従来手法の 0.1046 から 0.0883 へと 15.6%の改善を示した。一方で、$RMSE_m$ ではわずかな改善しか見られない。これは、各既知問題の特徴量の質は大きく向上したが、各新規 LLM 特徴量の質は依然として低いことを示唆している。また、新規の LLM と新規の問題の組み合わせ（テスト D）では、他のテストデータと比較して性能向上が限定的であった。テスト D に対応するモデル・問題の特徴量は、類似度行列のみに基づいて学習を行うしかない。そのため、正しく特徴量を推測することが困難であった。

理想的な類似度の効果　実際のパフォーマンスから得られる理想的な類似度を用いた OracleSimRMF は、全てのテストデータとメトリクスにおいて SimRMF を上回る性能を示した。特に、SimRMF では限定的な改善に留まった RMSEp においても、テスト B では 0.4665 から 0.4240 へ、テスト C では 0.3995 から 0.3737 へと改善が見られた。また、SimRMF で効果の高かった指標ではさらなる改善が確認され、テスト B の RMSEm では MF の 0.0341 から OracleSimRMF の 0.0250 へ改善を、テスト C の $RMSE_x$ では 0.1046 から 0.0691 へと改善した。これらの結果は、現状の補助情報から作成する類似度行列の質に改善の余地があることを示している。

4　おわりに

本研究では、LLM の性能推定において、補助情報に基づく類似度制約項を導入した行列分解手法 SimRMF を提案した。実験の結果、既知の LLM に対する新規の問題（テスト B）のモデル平均 RMSE と、既知の問題に対する新規の LLM（テスト C）の問題平均 RMSE において大きな改善が確認された。一方で、新規の LLM と新規の問題の組み合わせ（テスト D）では性能向上の幅は限定的であった。

参考文献

[1] Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinybenchmarks: evaluating llms with fewer examples. **ArXiv**, Vol. abs/2402.14992, , 2024.

[2] Alex Kipnis, Konstantinos Voudouris, Luca M. Schulze Buschoff, and Eric Schulz. metabench - a sparse benchmark to measure general ability in large language models. **ArXiv**, Vol. abs/2407.12844, , 2024.

[3] Tal Shnitzer, Anthony Ou, M'irian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. **ArXiv**, Vol. abs/2309.15789, , 2023.

[4] William W. Rozeboom, Frederic M. Lord, Melvin R. Novick, and Allan Birnbaum. Statistical theories of mental test scores. **American Educational Research Journal**, Vol. 6, p. 112, 1969.

[5] Clémentine Fourrier, Nathan Habib, Alina Lozovskaya, Konrad Szafer, and Thomas Wolf. Open llm leaderboard v2. https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard, 2024.

[6] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them, 2022.

[7] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof qa benchmark, 2023.

[8] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[9] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024.

[10] Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning, 2024.

[11] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. **ArXiv**, Vol. abs/2402.05672, , 2024.

A　LLM 間類似度の計算手法

本研究では、LLM 間の類似度を特徴ベースと系統関係ベースの 2 つの観点から計算し、それらを組み合わせた総合的な類似度指標を採用した。

A.1　特徴ベースの類似度

特徴ベースの類似度では、Hugging Face 上で公開されている各 LLM のタグ情報から以下の属性情報を取得し、数値ベクトル化ののちベクトル間のコサイン類似度を計算する。

- **パラメータ数：**LLM の規模を示す指標として、総パラメータ数を対数変換（$\log_{10}(\max(\text{param}, 1))/12$ ）し、スケーリングして特徴量に含める。
- **アーキテクチャ：**各 LLM が属するアーキテクチャファミリ（例：LLaMA 系、GPT 系、BERT 系など）をバイナリベクトルで表す。
- **モデルの目的：**各 LLM が想定する主要なタスク（例：Causal LM, Masked LM, Seq2Seq など）について、該当するものをバイナリベクトル化する。
- **タグ情報：**モデルに付与された各種タグ（例：`finetuned`, `merged`, `english`, `legal` など）をグルーピングし、該当すれば 1、なければ 0 とする形で特徴量化する。

こうして得られた特徴ベクトルをモデル数分並べ、正規化（L2 ノルムなど）を行ったうえで、コサイン類似度を計算する。すなわち、ある 2 つのモデルの特徴ベクトルを $\mathbf{f}_1, \mathbf{f}_2$ とした場合、

$$\mathrm{Sim}_{\text{feature}}(\mathbf{f}_1, \mathbf{f}_2) = \frac{\mathbf{f}_1 \cdot \mathbf{f}_2}{\|\mathbf{f}_1\| \, \|\mathbf{f}_2\|}.$$

これにより、全モデル間の特徴ベースの類似度行列 S_{feature} を得る。

A.2　系統関係ベースの類似度

次に、モデル同士の系統関係を考慮した類似度を定義する。具体的には、以下のような関係を想定し、対応する重みを割り当てる。

- **Fine-tuning 関係（w_{ft}）：**あるモデルが他のモデルをファインチューニングした場合や、ファインチューニング元・先が一致している場合を示す。
- **同一ベースモデル（$w_{\text{same-base}}$）：**複数のモデルが同一のベースモデルを起源としている場合を示す。
- **マージ関係（$w_{\text{merge-source}}$）：**複数のベースモデルを合成（マージ）して新たなモデルが生成された場合、その元となったモデル群との関連性を示す。
- **同一マージ元（$w_{\text{same-merge-source}}$）：**同じマージ元モデルを共有している複数モデル間の関係を示す。

各モデルペア (i, j) に対し、上記の系統関係のうち該当するものを総合し、最も高い類似度スコアをそのペアの系統関係ベース類似度とする本研究における実験では、$w_{ft} = 0.8$, $w_{\text{same-base}} = 0.6$, $w_{\text{merge-source}} = 0.7$, $w_{\text{same-merge-source}} = 0.5$ などの重みを割り当て、最大値を採用した。このようにしてモデル間の系統関係類似度行列 S_{lineage} を構築する。

A.3　特徴ベースと系統関係ベースの統合

特徴ベースと系統関係ベースの類似度を統合するため、次式のような線形結合を行う。

$$S_{\text{combined}} = \alpha \, S_{\text{feature}} + (1 - \alpha) \, S_{\text{lineage}},$$

ここで、α は特徴ベースの類似度に対する重みを表す。α の値を大きくすると、各 LLM の内的特徴が類似度に強く反映され、$1 - \alpha$ の値を大きくすると系統関係がより重視される。なお、本研究における実験では $\alpha = 0.5$ を採用した。