知識多様体解析仕様書（2次元・均等配置制約付き）改訂版 v2.0

改訂目的

本改訂版は、従来仕様で測地線最適化が実行時フォールバックにより直線初期経路のまま出力され得た問題を防ぎ、測地線が最初から確実に「最適化済み経路」として出力されるようにするための仕様である。

本仕様では、2次元知識マップにおいて、cosine distance に基づく意味的距離関係の保存を主目的としつつ、可視化上できるだけ -1 < x < 1, -1 < y < 1 の領域内に均等にばらけることを副目的とする。

本仕様に書かれていない方法を任意に変更してはならない。実行環境やライブラリ差で結果が変わる可能性がある箇所は、必ず「差が出る可能性のある箇所」として manifest.json と analysis_report.md に記録する。

1. 共通条件

1. PDF本文は、ページ順にテキスト抽出する。図表キャプション・参考文献も含める。OCRは使わない。抽出不能ページがある場合のみ、そのページ番号を記録する。
2. テキスト前処理は、Unicode
   NFKC正規化、英字小文字化、連続空白の1空白化のみとする。ストップワード除去、語幹化、翻訳、要約は行わない。
3. TF-IDFは scikit-learn の TfidfVectorizer 相当を用い、以下の設定に固定する。

- analyzer="char_wb"
- ngram_range=(4,7)
- sublinear_tf=True
- norm="l2"
- max_features=250000
- min_df=1

4. 特徴数が250,000を超える場合は、TF-IDF語彙を頻度順ではなく、ライブラリ標準の語彙順・上限処理に従って固定する。
5. 乱数を使う全処理では random_state=0 とする。
6. 各文書ベクトルはL2正規化する。以後の類似度は cosine similarity とする。
7. 文書間距離は以下で定義する。

```
cosine distance = 1 - cosine similarity
```

2. 2次元知識マップ

2.1 代表4本の選定

1. 代表4本は、20本から4本を全探索し、4本間のペアワイズ cosine distance 総和が最大となる組を選ぶ。
2. 同点の場合は、ファイル名の辞書順で最初の組を採用する。
3. 選ばれた代表4本を、ファイル名の辞書順に並べ、順に以下の四隅に固定する。

- 1本目: (-1,-1)
- 2本目: (1,-1)
- 3本目: (1,1)
- 4本目: (-1,1)

2.2 残り16本の配置

残り16本は、代表4本を固定したまま、cosine distance に基づく意味的距離関係を保ちつつ、2次元領域内にできるだけ均等にばらけるように配置する。均等化は可視化のための副目的であり、意味的距離保存を主目的とする。

残り16本の座標は、以下の目的関数 J を最小化して求める。

```
J = E_stress + lambda_rep E_rep + lambda_var E_var + lambda_center E_center
```

距離保存項

```
E_stress = sum_{i<j} w_ij ( ||y_i-y_j|| - alpha d_ij )^2 / sum_{i<j} w_ij (alpha d_ij)^2
```

ここで、d_ij は文書 i,j 間の cosine distance、y_i は文書 i の2次元座標、w_ij=1 とする。alpha
は、代表4本間の平均2次元距離が、代表4本間の平均 cosine distance に対応するように決める固定スケール係数とする。

反発項

```
E_rep = 2/(N(N-1)) sum_{i<j} exp( -||y_i-y_j||^2 / (2 sigma^2) )
```

ここで N=20, sigma=0.35 とする。

分散項

```
E_var = ((var_x - 1/3)^2 + (var_y - 1/3)^2) / (1/3)^2
```

中心化項

```
E_center = mean_x^2 + mean_y^2
```

係数

```
lambda_rep = 0.10
lambda_var = 0.05
lambda_center = 0.01
```

2.3 最適化条件

1. 最適化は L-BFGS-B を用いる。
2. 代表4本以外の16本について以下の範囲制約を課す。

- -0.98 <= x <= 0.98

- -0.98 <= y <= 0.98

3. 代表4本のみ、四隅 (-1,-1), (1,-1), (1,1), (-1,1) に固定する。
4. 最適化の初期値は、random_state=0 の SMACOF/MDS 結果を代表4本にアフィン整合させたものとする。
5. 最終座標に対して、全点を一括で線形スケーリングしてはならない。
6. 最終座標CSVを locked_knowledge_map_coordinates.csv
   として保存し、以後のSPH、GPR、勾配、測地線解析では2次元マップを再計算せず、このロック済み座標CSVを読み込む。

2.4 2次元知識マップの出力

以下を必ず出力する。

1. 座標CSV
2. locked_knowledge_map_coordinates.csv
3. cosine similarity 行列CSV
4. cosine distance 行列CSV
5. 配置目的関数サマリーJSON
6. 2次元知識マップPNG

7. 2次元知識マップPDF
8. 模式図PNG
9. 模式図PDF

目的関数サマリーJSONには以下を含める。

- E_stress
- E_rep
- E_var
- E_center
- J
- 最近傍距離の平均
- 最近傍距離の最小値
- 最近傍距離の標準偏差
- var_x
- var_y
- mean_x
- mean_y

3. SPH補間

3.1 評価点

```
p = (0.25, 0.75)
```

3.2 SPHカーネル

SPHカーネルは2次元 cubic spline kernel とする。q=r/h に対し、以下で定義する。

```
0 <= q < 1: W = 1 - 1.5 q^2 + 0.75 q^3
1 <= q < 2: W = 0.25 (2-q)^3
q >= 2: W = 0
```

20本すべてを使うため、平滑化長 h は以下で固定する。

```
h = max_i distance(p,p_i) / 1.98
```

SPH重みは以下で正規化する。

```
w_i = W_i / sum_j W_j
```

補間知識ベクトルは以下で定義する。

```
v(p) = normalize(sum_i w_i x_i)
```

ここで x_i は文書 i のL2正規化済みTF-IDFベクトルである。

3.3 SPH補間の出力

以下を必ず出力する。

1. 文書別SPH重みCSV
2. 文書別正規化寄与率CSV

3. 上位TF-IDF特徴CSV
4. 意味カテゴリ集計CSV
5. 言語化Markdown
6. SPH重み模式図PNG
7. SPH重み模式図PDF

上位TF-IDF特徴を抽出し、文字 n-gram
断片を根拠として日本語で言語化する。言語化では、TF-IDFに存在しない概念を追加しない。

4. SPH知識勾配

4.1 定義

x方向・y方向の知識勾配は、正規化SPH補間ベクトル v(x,y) の偏微分とする。

```
x方向: partial v / partial x
y方向: partial v / partial y
```

解析微分が実装困難な場合は、中心差分 delta=1e-3 を用いる。

4.2 方向間類似度

方向間類似度は、以下の両方で評価する。

1. 内積 <partial v/partial x, partial v/partial y>
2. cosine similarity

4.3 方向定義

```
0度方向: partial v / partial x
45度方向: (partial v / partial x + partial v / partial y) / sqrt(2)
90度方向: partial v / partial y
```

各方向について、正方向で増えるTF-IDF特徴、逆方向で増えるTF-IDF特徴を分けて日本語で言語化する。

4.4 SPH知識勾配の出力

以下を必ず出力する。

1. 勾配サマリーCSV
2. 方向別TF-IDF特徴CSV
3. 方向間類似度CSV
4. 意味カテゴリ別勾配CSV
5. 言語化Markdown
6. 方向ベクトル模式図PNG
7. 方向ベクトル模式図PDF

5. GPR

5.1 モデル定義

GPRは、2次元座標を入力、TF-IDF知識ベクトルを出力とする。高次元TF-IDFに直接GPRを適用せず、TruncatedSVD
で10次元潜在表現に圧縮してカーネルハイパーパラメータを推定する。random_state=0 とする。

カーネルは以下で固定する。

```
ConstantKernel * RBF(length_scale=[lx,ly]) + WhiteKernel
```

GPRの設定は以下とする。

```
n_restarts_optimizer=10
random_state=0
normalize_y=False
```

予測平均は、学習済みGPRカーネルから得られる後方線形係数により、20本文書ベクトルの線形結合として求める。

5.2 不確実性

不確実性は、以下を出力する。

1. GPR後方標準偏差
2. GPR事前標準偏差
3. 相対不確実性 = 後方標準偏差 / 事前標準偏差

5.3 寄与率

各論文の寄与率は以下で求める。

```
abs(後方線形係数) * max(cosine(文書ベクトル, 予測平均), 0)
```

その後、20本の合計が1になるように正規化する。後方線形係数が負の場合は、その論文が不要であるという意味ではなく、局所平均場を調整する補正的寄与であると説明する。

5.4 GPRの出力

以下を必ず出力する。

1. GPRサマリーJSON
2. ベイズ寄与率CSV
3. GPR予測平均の上位TF-IDF特徴CSV
4. 意味カテゴリCSV
5. 言語化Markdown
6. GPR模式図PNG
7. GPR模式図PDF

6. 予想論文の生成

評価点 (0.25,0.75)
で推定される知識ベクトルに基づき、この位置に対応する仮想的・予想的な論文概要を日本語500字程度で生成する。

1. 生成文は、TF-IDF特徴、SPH補間結果、GPR予測平均の内容に基づく。
2. 根拠のない新概念、実在しない実験結果、存在しない数値を追加しない。
3. 文体は学術論文の要旨に近いものとし、対象、方法、結果の方向性、意義を含める。

7. 測地線（改訂・確実化版）

7.1 基本定義

SPH補間写像 v(x,y) から誘導計量を以下で定義する。

```
g_ij = < partial v / partial p_i, partial v / partial p_j >
```

ここで p_1=x, p_2=y とする。

測地線は、以下の端点固定経路として求める。

```
始点: (0.25, 0.75)
終点: (-0.3, 0.35)
離散点数: 31
```

離散エネルギーは以下で定義する。

```
E = sum_k Delta p_k^T g((p_k+p_{k+1})/2) Delta p_k
```

経路点の座標範囲は以下に制限する。

```
-1 <= x <= 1
-1 <= y <= 1
```

7.2 計量評価の固定条件

1. 計量 g は、ロック済み座標 locked_knowledge_map_coordinates.csv
   とL2正規化済みTF-IDF文書ベクトルから構成したSPH補間写像に基づいて評価する。
2. 2次元知識マップは測地線解析中に再計算してはならない。
3. partial v / partial x と partial v / partial y は中心差分で求める。
4. 中心差分幅は delta=1e-3 とする。
5. v(x,y) は常にL2正規化済みSPH補間ベクトルとする。
6. 計量行列が数値的に特異または非正定値になる場合のみ、次の正則化を適用する。

```
g_regularized = g + epsilon I
epsilon = 1e-10
```

7. 正則化を適用した点数、最大正則化量、正則化前の最小固有値を geodesic_optimization_status.json に記録する。

7.3 最適化対象

1. 31点のうち、始点と終点を除く29点のみを最適化変数とする。
2. 各内部点の範囲制約は -1 <= x <= 1, -1 <= y <= 1 とする。
3. 始点と終点は固定し、最適化中に変更してはならない。

7.4 初期経路と再試行条件

初期経路は、まずユークリッド直線とする。ただし、測地線が直線初期経路のまま未最適化で出力されることを防ぐため、以下の決定論的再試行を必ず実装する。

初期経路候補

a を直線に垂直な単位ベクトル方向の正弦波摂動振幅とする。

```
candidate_0: a = 0.00 # 直線
candidate_1: a = 0.01
candidate_2: a = -0.01
candidate_3: a = 0.03
candidate_4: a = -0.03
candidate_5: a = 0.05
candidate_6: a = -0.05
```

内部点 k=1,...,29 に対し、摂動は以下で与える。

```
p_k = straight_p_k + a * sin(pi k / 30) * n_perp
```

ここで n_perp は始点から終点への直線方向に垂直な単位ベクトルである。摂動後に範囲外となる点は、各座標を [-1,1] にクリップする。

7.5 L-BFGS-B設定

各候補について、以下の設定で L-BFGS-B を実行する。

```
method = "L-BFGS-B"
maxiter = 2000
maxfun = 200000
ftol = 1e-12
gtol = 1e-8
maxls = 50
```

可能であれば目的関数勾配は数値差分ではなく最適化ライブラリの標準近似に任せてよい。ただし、その場合も上記設定を記録する。

7.6 成功判定と採用規則

各候補について、以下を記録する。

- candidate_id
- initial_amplitude
- optimizer_success
- optimizer_message
- nit
- nfev
- initial_energy
- final_energy
- straight_energy
- energy_drop_from_straight
- relative_energy_drop_from_straight
- max_lateral_deviation_from_straight
- mean_lateral_deviation_from_straight

採用経路は以下の規則で決める。

1. optimizer_success=True の候補が1つ以上ある場合、その中で final_energy が最小の候補を採用する。
2. optimizer_success=True の候補がないが、final_energy < straight_energy - 1e-12 を満たす候補がある場合、その中で final_energy が最小の候補を暫定採用し、geodesic_status="energy_improved_but_optimizer_not_success" と記録する。
3. いずれの候補も上記を満たさない場合、直線経路を「測地線」として出力してはならない。この場合は geodesic_status="failed" とし、出力ファイル名に straight_baseline を付ける。analysis_report.md には「測地線最適化は失敗したため、測地線としては採用していない」と明記する。

7.7 フォールバック禁止事項

以下を禁止する。

1. 実行時間や最適化失敗を理由に、直線初期経路を無条件に測地線として報告すること。
2. optimizer_success=False のまま、geodesic_path.csv という名前で直線経路のみを保存すること。
3. 測地線図に直線初期経路だけを描き、最適化済み測地線であるかのように表示すること。
4. 直線経路と測地線のエネルギー差、最大横ずれ、成功判定を省略すること。

7.8 直線経路との比較

必ず直線経路と採用測地線を比較し、以下をJSONとMarkdownに記録する。

- 直線経路エネルギー
- 採用測地線エネルギー
- エネルギー差 = 直線経路エネルギー - 採用測地線エネルギー
- 相対エネルギー低下率
- 直線リーマン長
- 測地線リーマン長
- リーマン長差
- 最大横ずれ
- 平均横ずれ
- 端点線分からの最大横ずれ
- 経路最大折れ角
- 経路平均折れ角
- optimizer success
- optimizer message
- 採用した初期経路候補ID

7.9 計量の「曲率っぽい」変化の出力

厳密なリーマン曲率テンソルを要求しない。ただし、計量が経路上でどの程度変化しているかを示すため、以下の曲率類似指標を必ず出力する。

各経路点または各区間中点で以下を計算する。

- g_xx
- g_xy
- g_yy
- trace_g = g_xx + g_yy
- det_g
- lambda_min
- lambda_max
- anisotropy_ratio = lambda_max / lambda_min
- coupling = abs(g_xy) / sqrt(g_xx g_yy)
- frobenius_norm_g

さらに隣接点差分から以下を計算する。

- ||dg||/ds
- d_trace_g/ds
- d_det_g/ds
- d_anisotropy_ratio/ds
- d_coupling/ds

サマリーとして以下を記録する。

- metric_variation_mean_norm_dg_ds
- metric_variation_max_norm_dg_ds
- anisotropy_ratio_min
- anisotropy_ratio_max
- coupling_min
- coupling_max
- trace_g_min
- trace_g_max
- det_g_min
- det_g_max

7.10 測地線の出力

以下を必ず出力する。

1. geodesic_reoptimized_path.csv
2. geodesic_straight_baseline_path.csv
3. geodesic_reoptimized_metric.csv
4. geodesic_reoptimized_metric_variation.csv
5. geodesic_reoptimized_comparison.json
6. geodesic_optimization_status.json
7. geodesic_reoptimized_report.md
8. geodesic_reoptimized_schematic.png
9. geodesic_reoptimized_schematic.pdf
10. geodesic_reoptimized_metric_variation.png
11. geodesic_reoptimized_metric_variation.pdf

図には必ず以下を含める。

- 20本の文書点
- 始点と終点
- 直線経路
- 採用測地線
- 最大横ずれ位置
- 採用測地線が直線とほぼ一致する場合でも、エネルギー差と最大横ずれを図注に表示する。

8. 出力・再現性確認

8.1 必須出力

すべての数値結果はCSVまたはJSONで保存する。図はPNGとPDFの両方で保存する。

最終回答には、以下を明記する。

1. 主要パラメータ
2. 乱数seed

3. TF-IDF次元数
4. 代表4本
5. 評価点
6. SPH平滑化長
7. GPRカーネル
8. GPR不確実性
9. 方向間類似度
10. 測地線最適化ステータス
11. 採用した測地線候補ID
12. 直線経路エネルギー
13. 測地線エネルギー
14. エネルギー差
15. 相対エネルギー低下率
16. 最大横ずれ
17. 測地線長
18. 直線経路との比較
19. 計量変化サマリー
20. 主要な出力ファイル
21. 差が出る可能性のある箇所

8.2 再現性ファイル

再現性確認のため、以下を必ず出力する。

1. manifest.json
2. pdf_text_extraction_summary.csv
3. analysis_report.md
4. 使用したPythonバージョン
5. 使用した主要ライブラリのバージョン
6. 入力PDFファイル名一覧
7. 入力PDFごとの抽出文字数
8. geodesic_optimization_status.json
9. 測地線最適化の全候補ログCSVまたはJSON

8.3 差が出る可能性のある箇所

以下は必ず列挙する。

- PDFテキスト抽出ライブラリの差
- scikit-learn の TF-IDF 語彙順・上限処理の差
- SMACOF/MDS の実装差
- L-BFGS-B の実装差および停止判定差
- GPR ハイパーパラメータ最適化の局所解差
- 浮動小数点演算およびBLAS/LAPACK差
- matplotlib のフォント・描画差

8.4 測地線に関する最終回答テンプレート

測地線解析を含む最終回答では、最低限以下の形式で報告する。

```
測地線最適化: success / failed / energy_improved_but_optimizer_not_success
採用候補ID: ...
直線経路エネルギー: ...
測地線エネルギー: ...
エネルギー差: ...
相対エネルギー低下率: ...
最大横ずれ: ...
計量変化 ||dg||/ds 平均: ...
計量変化 ||dg||/ds 最大: ...
異方性固有値比: min ... / max ...
結合係数 |g_xy|/sqrt(g_xx g_yy): min ... / max ...
```

9. 改訂版での重要な変更点

1. 測地線最適化の直線フォールバックを禁止した。
2. L-BFGS-B の最大反復数、許容誤差、ラインサーチ回数を明記した。
3. 直線、正負の正弦波摂動を含む複数初期経路候補を固定し、決定論的に再試行するようにした。
4. 成功判定と採用規則を明文化した。
5. 測地線が直線に近い場合でも、エネルギー差、相対低下率、最大横ずれを必ず出力するようにした。
6. 計量の「曲率っぽい」変化を、厳密曲率ではなく $\|dg\|/ds$、異方性、結合係数、trace、det の経路上変化として必ず出力するようにした。
7. 測地線失敗時は、直線経路を測地線として表示・命名することを禁止した。