Documentação Técnica – Algoritmo rank_similarity: Cálculos e Premissas

Objetivo

Este documento descreve, em nível matemático e procedimental, o algoritmo rank_similarity para ranquear candidatos (anos ou ano-mês) pela similaridade em relação a um alvo. Cobre notação, pré-processamentos, agregação temporal, remoção de outliers, métricas, normalizações, cálculo de scores e critérios de cobertura.

Notação

Seja X_v(t) a série da variável v após agregação/normalização na malha temporal T. Seja Y_v^c(t) a curva do candidato c para a variável v, alinhada à mesma malha. O conjunto de variáveis é V; os candidatos formam o conjunto C.

Construção do alvo (target)

Quando o alvo não é fornecido, é construído por estatística + método de agregação:

• compare_by='year', agg_freq='monthly': target_v(m) = estatística(média/min/máx) por mês m=1..12, agregando todos os anos.

• compare_by='year', agg_freq='daily': target_v(doy) = estatística por dia-do-ano.

• compare_by='year', agg_freq='hourly': target_v(k) com k=(dia-do-ano, hora).

• compare_by='month': restringe ao mês escolhido e agrega por dia-do-mês (e eventualmente hora).

Agregação temporal

Dados originais s_v(t_raw) são reamostrados conforme agg_freq ∈ {monthly, daily, hourly} e agregados por agg_stat ∈ {mean, min, max}. Pré e/ou pós-agregação podem aplicar filtros de outliers.

Remoção de outliers

Os métodos marcam pontos extremos como NaN (não removem linhas fisicamente):

• Z-score: $z_i = (x_i - \mu) / \sigma$; marca $|z_i| > z_{thresh}$.

• IQR: Q1, Q3, IQR = Q3−Q1; marca fora de [Q1−k·IQR, Q3+k·IQR].

• MAD: mediana m; MAD = median(|x_i−m|); z_rob = 0.6745·(x_i−m)/MAD; marca |z_rob| > mad_thresh.

Escopos: pre_agg (dentro de cada janela antes da estatística), post_agg (na curva agregada), both.

Normalização (opcional)

Após agregação, cada série pode ser normalizada:

• zscore: x' = (x − mean(x)) / std(x)

• minmax: x' = (x − min(x)) / (max(x) − min(x))

• none: mantém nível original.

Alinhamento e cobertura

Alinhamos alvo X_v(t) e candidato Y_v^c(t) via:

• inner: T = interseção dos índices.

• left_target: T = índices do alvo; candidato é reindexado.

Removem-se pares com NaN. Define-se coverage_v^c = (# pares válidos) / (# pontos válidos do alvo). O candidato c só é considerado se coverage_v^c ≥ min_coverage para TODAS as variáveis v ∈ V.

Métricas (modo curve)

Para sequências alinhadas x = X_v(t), y = Y_v^c(t):

• Correlação de Pearson (corr, ↑ melhor): corr = Cov(x,y)/(σ_x·σ_y).

• RMSE (↓ melhor): RMSE = sqrt((1/n)·Σ (x_i − y_i)^2).

• MAE (↓ melhor): MAE = (1/n)·Σ |x_i − y_i|.

• DTW (↓ melhor): distância de Dynamic Time Warping com janela Sakoe–Chiba (raio r).

Métricas (modo value)

Quando compare_mode='value', comparamos escalares por grupo: abs_diff = |μ_y − μ_x|; pct_diff = abs_diff / (|μ_x| + ε).

Normalização das métricas (min–max por variável)

Para cada variável v e métrica m, normaliza-se para [0,1]:

• Se ↑ melhor (ex.: corr): score_m = (m − m_min)/(m_max − m_min).

• Se ↓ melhor (ex.: rmse, mae, dtw): score_m = (m_max − m)/(m_max − m_min).

Se m_max == m_min ou não finito, define-se score_m = 1.0.

Score por variável

Com pesos de métrica w_m (distribuição que soma 1):

score_var(v,c) = Σ_m w_m · score_m(v,c). No modo value, utiliza-se 0.5 para abs_diff e 0.5 para pct_diff.

Score final do candidato

Com pesos por variável W_v (normalizados):

score_final(c) = Σ_{v∈V} W_v · score_var(v,c). O ranking ordena score_final em ordem decrescente.

Tratamento de NaNs

A remoção de outliers insere NaNs; o alinhamento remove pares com NaN. Coverage é calculada antes do scoring; candidatos abaixo do limiar são descartados.

Limitações e extensões

Exemplo numérico (ilustrativo)

Suponha, para uma variável v e três candidatos A,B,C: corr={0.90,0.60,0.30}, rmse={2,4,1}, mae={1.2,2.2,1.5}.

Normalização (min–max por métrica):

• corr↑: A=1.00, B=0.50, C=0.00; rmse↓: A≈0.667, B=0.00, C=1.00; mae↓: A=1.00, B=0.00, C=0.70.

Com metric_weights {corr:0.5, rmse:0.3, mae:0.2}: score_var(A)=0.90; B=0.25; C=0.44. Com múltiplas variáveis, o score_final é a média ponderada por var_weights.

Gerado automaticamente – rank_similarity (documentação de cálculos e premissas).