

ITEM #220 - Pattern Family Augmentation for Structural Indexing in Time-Series IR Systems – improving Recall While Preserving Structural Correctness

Conversation : Time-Series IR Requirements

20260109

Authors: Sizhe Tan & GPT-Obot

ME :

兄弟，拓展的很好。拉下了一条：Step 5. Structural Indexing -- 应该有一个对每一个 Known Signal Pattern 生成若干个保持同样性质变异 IR，以增加搜索的命中率。

AI :

明白。你补的这一条非常关键：Step 5 不只是“建索引”，还要做“模式族 (pattern family) 扩增”，否则在线匹配会被尺度、速度、相位、噪声形态差异显著拉低命中率。

下面我把它补成一个“可工程落地、DBM 风味”的 Step 5 扩展定义。

Step 5 (增强版) Structural Indexing + Pattern Family Augmentation

5A. 为每个 Known Signal Pattern 生成一组“性质保持”的变异 IR (Pattern Family)

目标不是造假数据，而是把同一模式在现实中出现的合理变体显式建模出来：

- 增加召回 (recall) ，避免“只命中教科书版本”
- 让 Differential Tree / Two-Phases Search 的邻域更连续、更可覆盖
- 为阈值与 RHS 统计提供“同族样本”的稳定估计

关键约束：性质保持 (**property-preserving**)

你要保留的不是“点值”，而是 IR 中的结构性质，例如：

- 方向序列 (Up/Down/Flat 的拓扑结构)
- 关键转折点的相对次序
- 梯次 (Ladder/Tier) 相对关系
- 事件依赖 (Event \rightarrow Event) 的因果链
- Pattern 约束 (而不是 if threshold)

5B. 典型“性质保持变异”操作集 (Mutation Operators)

这些操作应该在 **IR** 层做，而不是 raw series 层做（这就是 DBM 的味道：结构优先）。

1. **Time-warp** (速度变异)
 - 拉伸/压缩某些段的事件间距
 - 保持事件顺序与方向拓扑不变
2. **Amplitude re-quantization** (幅度梯次变异)
 - 让同一方向事件落入邻近 Ladder/Tier
 - 本质是“同结构不同强度”
3. **Phase jitter** (相位抖动)
 - 模式起点/终点前后微移
 - 保持核心事件链不变
4. **Local noise absorption** (噪声吸收)
 - 插入/删除少量“弱事件”但不改变主链
 - 例如：允许出现 1–2 个小反向但被归类为“decorator events”
5. **Substructure optionality** (可选子结构)

- 把某些次要事件标成可选，生成有/无两种版本
- 对应 Two-Phases 的 coarse→refine 很友好

5C. 变异生成策略（避免组合爆炸）

你需要一个 **Mutation Policy** 来控制规模与质量，否则很快爆炸。

推荐一个 DBM 风味的约束：

- 每个 pattern 生成 **K** 个变体（例如 32/64/128），而不是全展开
- 变体采样遵循：
 - 覆盖优先：尽量覆盖距离空间的邻域
 - 结构一致性优先：不破坏核心事件链
 - 成本预算：每个变体有 mutationCost（可直接进入 distance 的解释链）

你甚至可以把它做成：

PatternFamily = { (**IR_variant**, mutationSignature, mutationCost) }

5D. 与索引结构的关系（为什么放在 Step 5）

把 family augmentation 放在 Step 5 的原因是：

- 索引对象变成“模式族”而不是单模式
- Differential Tree 的分割更稳：避免叶子点过稀疏、邻域不连续
- Two-Phases Search 的 phase-1 召回显著提升；phase-2 再用严格约束确认

工程上建议：

- Tree/Index 的点：存 **IR_variant**（或其 embedding/feature key）
- 叶子聚合：挂回到 **patternId**（同族归并）
- 查询命中：先命中 variant，再归并回 patternId，然后输出：
 - 最佳 variant match

- 该 pattern 的 overall score / RHS stats

5E. 在线阈值与统计的直接收益

有了模式族，你的 Step 9/10 会更“可控”：

- 阈值不再是“某个点距离 < thr”
- 而是：
 - minDistance over family
 - top-N family matches distribution
 - family-consistency（同族多个变体同时近 → 置信度更高）

这会显著降低“单点偶然命中”的假阳性。

Step 5 的一句标准化定义（可直接写入 ITEM）

Step 5 builds structural indices (Differential Trees / Two-Phases Search) over **pattern families**, where each Known Signal Pattern is expanded into a bounded set of **property-preserving IR variants**. This turns pattern matching into robust neighborhood search in metric space, increasing recall without sacrificing structural correctness.

如果你接下来要我继续“落地到 runtime 代码骨架”，我建议我直接给：

- `IPatternFamilyGenerator`（输入 pattern IR，输出 variants）
- `IMutationOperator + MutationPolicy`（预算、采样、成本）
- `PatternFamilyIndexBuilder`（把 variants 编入 tree / two-phases）
- 以及 3–5 个 JUnit4 玩具测试：
 - family 命中归并回 patternId
 - noise decorator 不破坏主链
 - 预算控制（不爆炸）

我可以按你现有 `com.dbm.ir.runtime / IrPipelineV2` 的命名风格写成可 drop-in 的最小桥接版。

ME :

很好。请生成 ITEM #220。英文中文各一版。

AI :

DBM-COT ITEM #220

Pattern Family Augmentation for Structural Indexing in Time-Series IR Systems
Improving Recall While Preserving Structural Correctness

Abstract

In large-scale time-series intelligence systems, treating each known signal pattern as a single immutable template leads to fragile matching and poor recall. Real-world occurrences of the same structural pattern vary in speed, amplitude, phase, and noise characteristics. This ITEM introduces **Pattern Family Augmentation** as a mandatory component of **Structural Indexing (Step 5)** in the DBM Time-Series IR stack.

By generating a bounded set of **property-preserving IR variants** for each known pattern, structural indexing is upgraded from isolated template matching to robust **neighborhood search in metric space**. This approach significantly improves recall while maintaining interpretability, metric consistency, and runtime efficiency.

1. Motivation

Traditional time-series pattern matching systems suffer from a fundamental contradiction:

- Tight thresholds preserve precision but miss real signals.
- Loose thresholds increase recall but introduce false positives.

This contradiction arises because patterns are treated as **point objects** rather than **structural families**.

DBM resolves this by recognizing that:

A signal pattern is not a point in metric space, but a **local manifold of structurally equivalent IRs**.

Pattern Family Augmentation makes this manifold explicit and computable.

2. Pattern Family Definition

A **Pattern Family** is defined as a finite set of IR variants derived from a canonical pattern IR, subject to **property-preserving constraints**.

Formally:

```
PatternFamily(P) = {  
    (IR_i, mutationSignature_i, mutationCost_i)  
}
```

Where all IR_i preserve the essential structural properties of pattern P .

Preserved properties typically include:

- Event ordering and causality ($Event \rightarrow Event$)
- Directional topology (Up / Down / Flat relations)
- Relative ladder / tier structure
- Core pattern constraints (pattern logic, not raw thresholds)

3. Property-Preserving Mutation Operators

Pattern Family Augmentation operates **at the IR level**, not at the raw time-series level. This ensures structural correctness and metric consistency.

Typical mutation operators include:

1. **Time-Warp Mutation**
Varies event spacing while preserving order and topology.
2. **Amplitude Re-Quantization**
Shifts events across adjacent ladder tiers without changing direction structure.
3. **Phase Jitter**
Applies small temporal offsets to pattern boundaries.
4. **Noise Absorption / Decoration**
Inserts or removes weak auxiliary events without breaking the main event chain.
5. **Substructure Optionality**
Marks secondary substructures as optional, generating with/without variants.

Each mutation contributes a **mutation cost**, which can be incorporated into metric distance explanations.

4. Controlled Expansion and Budgeting

Unbounded mutation leads to combinatorial explosion. Therefore, Pattern Family generation must be governed by a **Mutation Policy**, enforcing:

- A maximum number of variants per pattern (e.g., $K = 32-128$)
- Coverage-oriented sampling in metric space
- Preservation of a minimal structural core
- Explicit mutation cost accounting

This ensures scalability and interpretability.

5. Integration with Structural Indexing

Pattern Family Augmentation is applied **before** building:

- Metric Differential Trees
- Two-Phases Search indices

Indexing strategy:

- **Index nodes store IR variants**
- **Leaf aggregation maps variants back to canonical pattern IDs**
- Query results are merged at the pattern-family level

This transforms indexing from pattern lookup into **family-aware neighborhood search**.

6. Runtime Benefits

Pattern Family Augmentation enables:

- Higher recall without relaxed thresholds
 - More stable Differential Tree partitions
 - Improved Two-Phases Search pruning
 - Pattern-level confidence scoring based on multi-variant consistency
 - Reduced false positives caused by single accidental matches
-

7. Impact on Online Decision and Learning Loop

With Pattern Families:

- Thresholding decisions are made over **family distance distributions**
- RHS statistics become more stable
- Pattern scores evolve based on family-level evidence
- Newly observed variants can be fed back to Offline pattern evolution

This turns pattern matching into a **living structural evidence system**.

8. Position in the DBM Time-Series Stack

Pattern Family Augmentation is a **mandatory enhancement** of:

Step 5: Structural Indexing

It bridges Offline structure discovery and Online robust recognition, ensuring that DBM Time-Series Intelligence remains both precise and resilient.

9. Summary Statement

Pattern Family Augmentation upgrades time-series pattern matching from fragile template comparison to robust structural neighborhood reasoning in metric space, preserving DBM's core principles of interpretability, consistency, and evolution.

DBM-COT ITEM #220 (中文版)

时间序列 IR 结构索引中的模式族扩增

在保持结构正确性的前提下显著提升命中率

摘要

在大规模时间序列智能系统中，将每一个已知信号模式视为单一、不可变模板，会导致匹配脆弱、召回率低下。现实世界中，同一结构模式在速度、幅度、相位与噪声形态上天然存在变化。

本 ITEM 提出：**模式族扩增（Pattern Family Augmentation）** 应作为 DBM 时间序列 IR 应用栈中 **Step 5（Structural Indexing）** 的必要组成部分。

通过为每一个已知模式生成一组**保持结构性质不变的 IR 变体**，结构索引从“模板匹配”升级为**度量空间中的邻域搜索**，在不牺牲可解释性与一致性的前提下，大幅提升系统召回能力与稳定性。

1. 问题动机

传统时间序列系统面临一个根本矛盾：

- 阈值严格 → 精度高但漏信号
- 阈值放宽 → 召回高但假阳性泛滥

根源在于：

模式被错误地建模为“点”，而不是“结构族”。

DBM 的核心认知是：

一个信号模式不是度量空间中的单点，而是一个由结构等价 IR 构成的局部结构流形。

模式族扩增正是对这一结构事实的工程化表达。

2. 模式族定义

模式族（Pattern Family） 是由一个规范模式 IR 派生出的一组有限 IR 变体集合，所有变体均满足**性质保持约束**。

形式化表示为：

```
PatternFamily(P) = {  
    (IR_i, mutationSignature_i, mutationCost_i)  
}
```

所有 IR_i 必须保持模式 P 的核心结构性质。

3. 性质保持型 IR 变异操作

模式族扩增必须发生在 **IR 层**，而非原始时间序列层。

典型操作包括：

1. **时间扭曲 (Time-Warp)**

改变事件间距，不改变事件顺序与拓扑。

2. **幅度梯次重映射**

在相邻 Ladder/Tier 之间移动事件。

3. **相位抖动**

微调模式起止边界。

4. **噪声吸收 / 装饰事件**

插入或删除弱事件而不破坏主事件链。

5. **子结构可选化**

为次级结构生成有/无两类版本。

每一次变异都会产生**变异成本**，可直接参与度量距离解释。

4. 规模控制与预算机制

为避免组合爆炸，模式族生成必须受 **变异策略 (Mutation Policy)** 控制：

- 每个模式限制生成 K 个变体（如 32–128）
 - 优先覆盖度量空间邻域
 - 保持最小结构核心
 - 明确记录变异成本
-

5. 与结构索引的集成方式

模式族扩增在构建以下索引之前完成：

- 度量差分树 (Differential Tree)
- 两阶段搜索引擎 (Two-Phases Search)

工程原则：

- 索引节点存储 IR 变体
 - 叶子节点回归到 canonical pattern ID
 - 查询结果在模式族层面合并与解释
-

6. 运行期收益

引入模式族后：

- 召回率显著提升而无需放宽阈值
 - 差分树结构更稳定
 - 两阶段搜索剪枝更有效
 - 置信度来自“同族一致性”而非单点命中
 - 假阳性明显减少
-

7. 对在线决策与学习闭环的影响

模式族使得：

- 阈值判断基于族级距离分布
- RHS 统计更稳定
- 模式评分随证据演化
- 新变体可回灌 Offline，推动模式进化

8. 在 DBM 时间序列体系中的定位

模式族扩增是 Step 5 (Structural Indexing) 的必要增强，而非可选优化。

它连接了 Offline 结构发现与 Online 稳健识别，是 DBM 时间序列智能具备工程可行性的关键环节。

9. 总结陈述

模式族扩增使时间序列识别从脆弱的模板匹配，跃迁为度量空间中的结构邻域推理，完整体现了 DBM 的结构智能哲学。

如果你愿意，下一步我可以直接：

- 为 ITEM #220 配一张 **Pattern → Pattern Family → Differential Tree** 的教学级结构图；
- 或直接生成 **Java 8 Skeleton (PatternFamilyGenerator / MutationPolicy / IndexBuilder)**，与 IrPipelineV2 对齐。

这一条补充非常关键，#220 是一个“工程转折点级”的 ITEM。