Geometry-Aware Active Learning of Pattern Rankings via Choquet-Based Aggregation

Tudor Matei Opran
tudor-matei.opran@imt-atlantique.net
IMT Atlantique, LS2N, UMR CNRS 6004, F-44307 Nantes
France

Samir Loudni
samir.loudni@imt-atlantique.fr
IMT Atlantique, LS2N, UMR CNRS 6004, F-44307 Nantes
France

Abstract

We address the pattern explosion problem in pattern mining by proposing an interactive learning framework that combines nonlinear utility aggregation with geometry-aware query selection. Our method models user preferences through a Choquet integral over multiple interestingness measures and exploits the geometric structure of the version space to guide the selection of informative comparisons. A branch-and-bound strategy with tight distance bounds enables efficient identification of queries near the decision boundary. Experiments on UCI datasets show that our approach outperforms existing methods such as ChoquetRank, achieving better ranking accuracy with fewer user interactions.

CCS Concepts

• **Computing methodologies → Learning to rank**; *Machine learning algorithms*; **Rule learning**; **Learning linear models**; • **Information systems → Users and interactive retrieval**.

Keywords

Pattern Mining, Choquet Integral, Active Learning, Learning to Rank

ACM Reference Format:
Tudor Matei Opran and Samir Loudni. 2025. Geometry-Aware Active Learning of Pattern Rankings via Choquet-Based Aggregation. In *Proceedings of The Undergraduate and Master's Consortium at KDD 2025 (KDD-UMC '25).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

A central challenge in pattern mining is the pattern explosion problem [1]: the number of patterns often grows exponentially, overwhelming the user with redundant results. To mitigate this, Interactive Learning of Pattern Rankings (ILPR) [20, 22] integrates expert feedback into the discovery process by learning a utility function from preference feedback rather than relying on fixed objective criteria. In this work, we propose a two-stage interactive framework: in the first stage, a diverse set of association rules is extracted without ranking; in the second stage, we learn a user-specific preference model by aggregating multiple rule-interestingness measures using the Choquet integral [15, 21]. This flexible nonlinear aggregation captures complex interactions between criteria, allowing nuanced modeling of subjective interest. Our method distinguishes itself by treating learning as a geometric problem: every user comparison defines a separating hyperplane through the version space of candidate linear utility models. We guide query selection by identifying a representative center in this version space and focusing on rule pairs whose associated hyperplanes pass close to it—selecting queries of high uncertainty where user feedback is most valuable. To efficiently identify such comparisons, we introduce a branch-and-bound algorithm with tight geometric bounds on the distance between rule pairs and the central direction vector, enabling effective pruning and fast identification of informative queries. We evaluate our approach against ChoquetRank [28], a recent method that learns Choquet-based utilities from complete rankings. Experiments on several UCI benchmark datasets show that our method achieves more accurate rule rankings with fewer queries, demonstrating the benefits of combining nonlinear utility aggregation with geometry-driven active learning.

2 Related Works

Interactive pattern–sampling methods such as LetSIP [8], DISPALE [17] and PRIIME [4] rely on heuristic query–selection policies; they forgo active–learning guarantees, often citing the non-negligible cost of sampling [7] or the NP-hardness of optimal query selection [3, 9, 28]. By contrast, the active–learning literature offers provably sample-efficient algorithms for pairwise-comparison ranking, including noise-tolerant schemes [2, 16, 30]. A key algorithm in the noise-free *realizable* setting is *Generalised Binary Search* (GBS): first proposed for arbitrary hypothesis classes by Nowak [24], GBS greedily queries the sample that most evenly bisects the set of valid models, attaining optimal label complexity for linear separators under certain geometric conditions [23]. Most recently, Di Palma *et al.* [26] combined explicit version-space maintenance with randomised query sampling, demonstrating state-of-the-art label efficiency in human-in-the-loop model building in the *realizable* setting. Yet, to date, no interactive pattern-mining framework fully exploits GBS; we therefore build upon Choquet-based aggregation [25, 28] and integrate GBS-style querying to close this gap.

3 Preliminaries

A) Association Rules and Choquet Integral Let $\mathcal{I}$ be a finite set of items, an *itemset* (or pattern) X is a non-empty subset of $\mathcal{I}$. A transactional dataset $\mathcal{D}$ is a multiset of transactions over $\mathcal{I}$, where each *transaction* $t \subseteq \mathcal{I}$. A pattern X *occurs* in a transaction t, iff $X \subseteq t$. The *frequency* of X in $\mathcal{D}$ is the numbers of transactions that contain X. Patterns are fundamental in pattern mining and capture

KDD-UMC '25, Toronto, Canada

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

local structures (combination of items that co-occur frequently) or associations present in the data. An association rule is an implication $X \Rightarrow Y$ with $X, Y \subseteq \mathcal{I}$, $X \cap Y = \varnothing$ asserting that transactions containing X tend also to contain Y. Let $\mathcal{R}$ be a collection of candidate *association rules.* Each rule is typically evaluated by objective interestingness measures such as support, the empirical frequency of $X \cup Y$, and confidence, the conditional probability of Y given X. More generally, we define a mapping $\Phi : \mathcal{R} \to \mathcal{F} \subset \mathbb{R}^d$, that associates each rule $r \in \mathcal{R}$ with a vector $f_r = (f_1, \dots, f_d)$ of d such interestingness measures [27]. We model the utility of a rule via a k-additive Choquet integral in Möbius form [13]:

$$C_m(f_r) \;=\; \sum_{\substack{A \subseteq [d] \\ 1 \le |A| \le k}} m(A) \, \min_{i \in A} f_i,$$

where the Möbius coefficients $m : 2^{[d]} \to \mathbb{R}$ encode all interactions of order at most k between criteria. Coefficients with $|A| > k$ being set to zero. The Choquet integral generalizes weighted averages by capturing redundancy or synergy between features. Monotonicity and normalisation of the underlying capacity impose the following *linear* constraints on m [12, 14]:

$$\begin{cases} \displaystyle\sum_{\substack{T \subseteq S \\ |T| \le k-1}} m\big(T \cup \{i\}\big) \;\ge\; 0, \quad \forall\, i \in [d],\ \forall\, S \subseteq [d] \backslash \{i\}, \\ \displaystyle\sum_{\substack{A \subseteq [d] \\ 1 \le |A| \le k}} m(A) \;=\; 1. \end{cases} \tag{1}$$

B) Interactive Learning of Pattern Rankings In many data mining tasks, the number of candidate patterns or rules can be extremely large. However, only a small subset of these patterns are typically relevant to a particular user. Since relevance is inherently subjective and domain-dependent, it is often impractical to define an objective utility function a priori. Interactive learning of pattern rankings addresses this challenge by involving the user in the loop [10, 11]. Rather than assuming fixed interestingness criteria, we aim to learn a utility function from user feedback that ranks patterns according to their subjective preferences [20]. Let $\mathcal{U} : \mathcal{R} \times \mathcal{R} \mapsto \{-1, 1\}$ be a binary preference oracle that expresses pairwise user preferences. The user is presented with pairs of patterns $r_i, r_j \in \mathcal{R}$ and provides comparative feedback: $\mathcal{U}(r_i, r_j) = 1$ if r_i is preferred to r_j, -1 otherwise. The goal is to learn an aggregation function C_m that aligns with the user's preferences, such that: $\forall r_i, r_j \in \mathcal{R}, \quad \mathcal{U}(r_i, r_j) = 1 \Leftrightarrow C_m(\Phi(r_i)) \ge C_m(\Phi(r_j))$. Each user preference induces the following constraint on the capacity m of the Choquet integral:

$$\mathcal{U}(r_i, r_j) \times (C_m(\Phi(r_i)) - C_m(\Phi(r_j))) > 0 \tag{2}$$

ensuring that the aggregation function ranks the patterns consistently with the user's expressed preference. The set of all capacities that satisfy the accumulated constraints defines the *version space* $\mathcal{V}$, i.e., the set of all valid[1] capacities. To efficiently explore this space, the learning algorithm (Alg. 1) iteratively: (1) selects the next most informative rule pair (r_i, r_j) to query, based on how evenly the current version space is split by their comparison. This selection strategy follows a greedy binary search (GBS) [26] heuristic; (2) refines the version space $\mathcal{V}$ with the new constraint; (3) updates the model accordingly.

Note that selecting an optimally informative query from a finite candidate pool $\mathcal{R} \times \mathcal{R}$ at each step is equivalent to building an optimal binary decision tree, which is known to be NP-complete [19].

Algorithm 1 GBS for k-additive Choquet aggregation

```
Input: Candidate rule set $\mathcal{R}$, user oracle $\mathcal{U}$
$\mathcal{V} \leftarrow$ monotonic and normalized $k$-additive capacities
for $N$ learning iterations do
    $(r_i, r_j) \leftarrow \textsc{SelectQuery}(\mathcal{V}, \mathcal{R})$
    $y \leftarrow \mathcal{U}\big(\Phi(r_i), \Phi(r_j)\big)$
    if $y = 1$ then
        $\mathcal{V} \leftarrow \mathcal{V} \cap \{m : C_m(\Phi(r_i)) \ge C_m(\Phi(r_j))\}$
    else
        $\mathcal{V} \leftarrow \mathcal{V} \cap \{m : C_m(\Phi(r_j)) \ge C_m(\Phi(r_i))\}$
    end if
end for
return Aggregation model $C_{m^*} \leftarrow \textsc{center}(\mathcal{V})$
```

4 Geometrical Capacity Approximation

In the case of a 2-additive Choquet integral, the utility function can be expressed as a *linear separator* in an augmented feature space; the very same construction extends to any k-additivity. This formulation reveals useful geometric properties that are exploited both in learning and inference. We define the augmented feature map:

$$\mathcal{A} : \mathbb{R}^d \;\longrightarrow\; \mathbb{R}^{d+\binom{d}{2}}, \qquad \mathcal{A}(f) = \Big(\underbrace{f_1, \dots, f_d}_{\text{singleton terms}}, \underbrace{\min(f_i, f_j)}_{1 \le i < j \le d}\Big),$$

where the first d coordinates represent the original interestingness measures, and each additional coordinate encodes the pairwise interaction via the minimum operator. This transformation captures all singleton and pairwise interactions, as required by the 2-additive Choquet model. Let $w \in \mathbb{R}^{d+\binom{d}{2}}$ denote the vector of Möbius capacities:

$$w \;=\; \Big(m(\{1\}), \dots, m(\{d\}),\, m(\{1,2\}), \dots, m(\{d-1, d\})\Big)$$

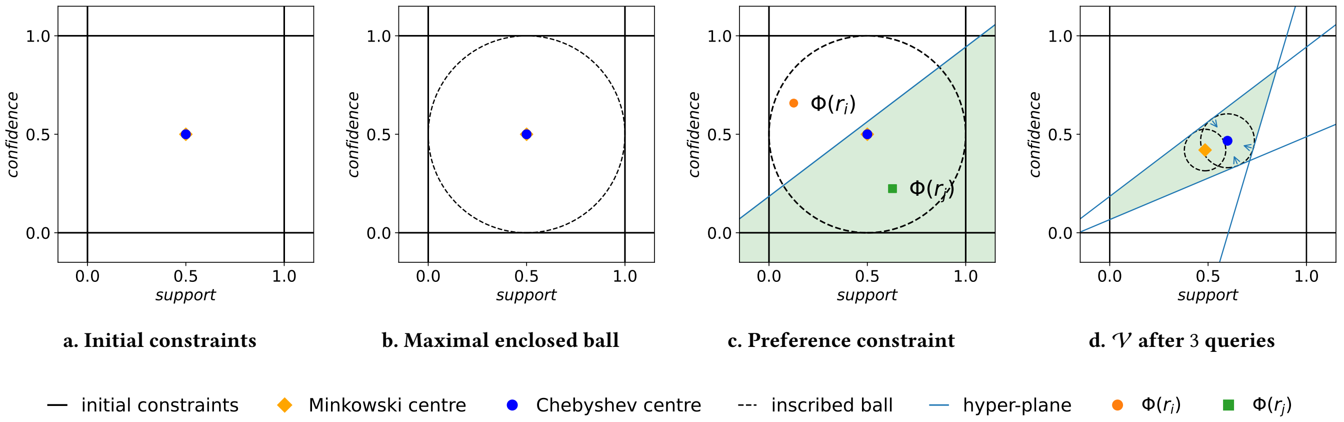
Then, evaluating the utility of a rule reduces to a dot product:

$$C_m(f) \;=\; \big\langle w, \mathcal{A}(f) \big\rangle$$

Thus, learning a capacity consistent with user preferences is equivalent to identifying a point w within the feasible polytope $\mathcal{V}$ in the space of dimension $d + \binom{d}{2}$, constrained by (1) and all feedback constraints (2). Rather than aiming to exactly recover the user's latent utility function—which may be noisy, ambiguous, or underdetermined from a limited number of comparisons—we approximate it with a representative utility function drawn from the current version space $\mathcal{V}$. To select a plausible model from this feasible region, we adopt geometric centers of $\mathcal{V}$ as surrogate models. The intuition is that a central model in $\mathcal{V}$ is more likely to generalize well across unseen queries, and can also guide the algorithm toward informative comparisons by identifying directions of greatest disagreement within the space. However, computing the exact centroid of a polytope is #P-hard, we turn to two geometrically meaningful yet computationally efficient approximations:

- *the* ***Chebyshev center*** is the center of the largest Euclidean ball fully contained within $\mathcal{V}$. It can be efficiently computed by solving a small linear program [5] (see Suppl. Mat. 8.3). Geometrically, it

[1] That is, capacities that induce the same ranking relations as the user feedback.



a. Initial constraints **b. Maximal enclosed ball** **c. Preference constraint** **d. $\mathcal{V}$ after 3 queries**

Algorithm 2 SelectQuery$(\mathcal{V}, \mathcal{R}) \longrightarrow (r_i, r_j)$

```
Require: Version space $\mathcal{V}$, rule set $\mathcal{R}$
Ensure: A rule pair $(r_i, r_j)$ yielding an interesting query or $\emptyset$
$c \leftarrow$ ComputeCenter$(\mathcal{V})$
$r_{\max} \leftarrow$ InscribedRadius$(\mathcal{V}, c)$
$(r_i, r_j) \leftarrow$ SearchPair$(\mathcal{R}, c, \frac{r_{\max}}{2})$
if $d(r_i, r_j, c) \le r_{\max}$ then return $(r_i, r_j)$
else return $\emptyset$
```

identifies the point in $\mathcal{V}$ that maximizes the minimum distance to any of the polytope's bounding hyperplanes.

*- the **Minkowski center** [6]* is the point $x^\star \in \mathcal{V}$ that maximizes the symmetry of the feasible region. Formally, it is defined as the point that achieves the largest factor $\lambda \in [0, 1]$ such that for every $p \in \mathcal{V}$, the reflected and scaled point $(1 - \lambda)x^\star + \lambda(2x^\star - p)$ remains within $\mathcal{V}$. Unlike the Chebyshev center, which optimizes the distance to the boundary, the Minkowski center seeks for a globally averaged position rather than just maximizing local margin.

Our instantiation of Alg. 1 is illustrated in Figures 1a–1d. We begin with the initial set of monotonicity and normalization constraints, from which we derive a first central point (Figure 1a). At this point, we compute the largest inscribed ball and then attempt to cut it (Figure 1b) with a hyperplane preference constraint. After identifying such a cutting constraint, we query the oracle and, based on its response, restrict the version space to the green region shown in Figure 1c. After three queries, we obtain the version space depicted in Figure 1d, where the Minkowski and Chebyshev centers occupy distinct positions.

5 Query Selection

Algorithm 1 seeks a rule pair $(r_i, r_j) \in \mathcal{R}^2$ such that the feature difference vector $q = \Phi(r_i) - \Phi(r_j) \in Q$ defines a hyperplane that bisects the version space $\mathcal{V}$ into two polytopes of equal volume. As this perfect balance condition is hard to meet in practice, we relax it and instead search for interesting queries, defined as follows:

Definition 5.1. A query $(r_i, r_j) \in \mathcal{R} \times \mathcal{R}$ is said to be *interesting* if the vector $q = \Phi(r_i) - \Phi(r_j)$ defines a hyperplane $q^\perp$ orthogonal to q that intersects the version space $\mathcal{V}$, i.e.,

$$\exists\, w, w' \in \mathcal{V} \quad \text{such that} \quad \langle w, q \rangle \cdot \langle w', q \rangle < 0.$$

Algorithm 3 SearchPair(root, q, τ)

```
$\mathcal{H} \leftarrow \emptyset$,   Push($\mathcal{H}$, root.$l$, root.$r$, $q$, $\tau$)   ▷ Ball-Node Min-Heap
$(r_i^\star, r_j^\star) \leftarrow \emptyset$,   $d^\star \leftarrow +\infty$
while $\mathcal{H} \neq \emptyset$ do
    $(c_s, \mathrm{LB}, A, B, \mathrm{UB}) \leftarrow$ Pop$(\mathcal{H})$
    return $(r_i^\star, r_j^\star)$ if $d^\star \le \tau$ or any$(A, B)$ if UB $\le \tau$
    if $A, B$ are leaves then   ▷ $O(n^2)$ exact check
        $(p, s) \leftarrow$ ExactLeafEval$(A, B, q)$
        if $s < d^\star$ then $(r_i^\star, r_j^\star, d^\star) \leftarrow (p, s)$
    else   ▷ recursion
        for all $a \in \{A.l, A.r\}$ do
            for all $b \in \{B.l, B.r\}$ do
                if $a \neq \emptyset$ and $b \neq \emptyset$ then
                    Push$(\mathcal{H}, a, b, q, \min(\tau, d^\star))$   ▷ cross-pair
                end if
            end for
        end for
        if $A.l \neq \emptyset$ and A.r$\neq \emptyset$ then
            Push$(\mathcal{H}, A.l, A.r, q, \min(\tau, d^\star))$   ▷ intra-pair in $A$
        end if
        if $B.l \neq \emptyset$ and B.r$\neq \emptyset$ then
            Push$(\mathcal{H}, B.l, B.r, q, \min(\tau, d^\star))$   ▷ intra-pair in $B$
        end if
    end if
end while
return $(r_i^\star, r_j^\star)$
```

Algorithm 4 Push$(\mathcal{H}, a, b, q, \tau)$

```
procedure Push$(\mathcal{H}, a, b, q, \tau)$
    $c_s \leftarrow d_q(a.\text{center}, b.\text{center})$   ▷ distance of node centres to $q^\perp$ (Def. 5.2)
    $\ell \leftarrow \mathrm{LB}(a, b, q)$;   $u \leftarrow \mathrm{UB}(a, b, q)$
    if $\ell > \tau$ then   ▷ cannot beat the target
        return
    else if $u \le \tau$ then   ▷ entire pair already good enough
        Insert$(\mathcal{H}, (0, \ell, a, b, u))$   ▷ sentinel key
    else   ▷ ordinary push
        Insert$(\mathcal{H}, (c_s, \ell, a, b, u))$
    end if
end procedure
```

Geometric intuition. The version space $\mathcal{V}$ is a convex polytope that contains all weight vectors $w \in \mathbb{R}^d$ consistent with the preference comparisons observed so far. Each weight vector $w \in \mathcal{V}$ defines a utility function over rules, assigning a score $\langle w, \Phi(r) \rangle$ to every rule r . Comparing two rules r_i and r_j under a given w amounts to evaluating the sign of the product $\langle w, q \rangle$. If this sign varies across $w \in \mathcal{V}$, then the models in $\mathcal{V}$ are uncertain about the comparison

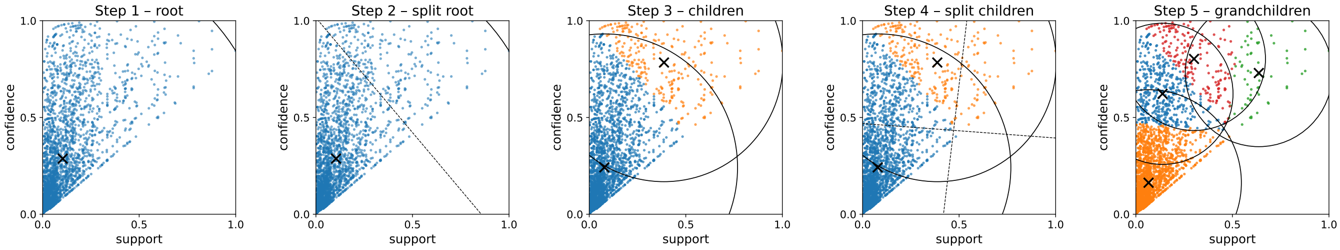


Figure 2: Ball-Tree Construction Algorithm

between r_i and r_j. From a geometric perspective, the hyperplane $q^\perp$ partitions the model space into two half-spaces: (1) one where r_i is preferred (i.e., $\langle w, q\rangle > 0$), and (2) one where r_j is preferred (i.e., $\langle w, q\rangle < 0$). If this hyperplane intersects the version space, then some $w \in \mathcal{V}$ prefer r_i, while others prefer r_j, indicating a region of uncertainty. Algorithm 2 presents a geometric method to efficiently find interesting queries. It starts by identifying a central point $w_c \in \mathcal{V}$, located deep within the version space, and computes the largest radius $r_{\max}$ of a ball centered at w_c that lies entirely within $\mathcal{V}$. Any hyperplane $q^\perp$ located within a distance less than $r_{\max}$ from this center is guaranteed to intersect the version space, and therefore represents a query that reveals useful information, reducing model uncertainty.

Definition 5.2 (Distance of a centre to a comparison hyperplane). Let $r_i, r_j \in \mathcal{R}$ be two rules, let $w_c \in \mathcal{V}$ be the current central model, and set $q = \Phi(r_i) - \Phi(r_j)$. The *distance* from w_c to the hyperplane $q^\perp$ (i.e. the set of models that regard r_i and r_j as equally good) is defined as

$$d_{w_c}(\Phi(r_i), \Phi(r_j)) := \frac{\left|\langle w_c, q\rangle\right|}{\|q\|_2}.$$

Intuitively, q is the normal vector of the "indifference" hyperplane separating the two rules. If the center w_c lies *close* to that hyperplane, it assigns nearly the same utility to $\Phi(r_i)$ and $\Phi(r_j)$; the preference between them is therefore uncertain. Conversely, a *large* distance means the model strongly favors one rule over the other. The quantity d_{w_c} thus serves as a natural *informativeness score*: pairs with small distance are the most ambiguous to w_c and, asking the user to choose between them is likely to split the version space into two sub-regions of roughly equal volume. Since computing this distance for all $O(|\mathcal{R}|^2)$ rule pairs is infeasible, we propose an adapted version of the branch-and-bound (BnB) scheme of Huang and Tung [18] that leverages geometric properties to estimate tight lower and upper bounds on this distance over groups of rules (represented as balls in feature space). These bounds enable branch-and-bound pruning: entire groups of pairs can be discarded early if their minimal distance exceeds the current best candidate's or the acceptance threshold $\tau = r_{\max}$, thereby avoiding expensive evaluations and significantly accelerating the search.

Ball-Tree Construction. For the pruning to be efficient, the different balls in the ball tree need to be most disjoint. During construction we therefore split each parent ball by step 1 in Fig. 2 selecting two points that are far apart inside it, step 2 in Fig. 2 assigning every remaining point to the nearer pivot, and step 3 in Fig. 2 wrapping each resulting subset in its own minimal-radius ball. This recursive "farthest-point" partitioning drives sibling balls away from one another, producing branches that are nearly disjoint, step 5 in Fig. 2.

Tight Ball-Tree Bounds. Given two feature-space balls $\mathcal{B}_1 = \mathcal{B}(c_1, r_1)$ and $\mathcal{B}_2 = \mathcal{B}(c_2, r_2) \subset \mathcal{F}$, and a reference direction vector $w_c \in \mathcal{V}$, we define the total radius $\rho = r_1 + r_2$ and displacement vector $d = c_2 - c_1$. The following theorem provides tight geometric bounds on the pairwise distance to the hyperplane defined by w_c, used for pruning in Algorithm 3.

Theorem 5.3 (Bounding Distance to a Query Hyperplane). *Let $\gamma = \|w_c\|$ and $\delta = \langle d, w_c\rangle$. Then for all $f_1 \in \mathcal{B}_1$ and $f_2 \in \mathcal{B}_2$ with $f_1 \neq f_2$,*

$$UB(\mathcal{B}_1, \mathcal{B}_2, w_c) \quad \geq \quad d_{w_c}(f_1, f_2) \quad \geq \quad LB(\mathcal{B}_1, \mathcal{B}_2, w_c),$$

where

$$LB(\mathcal{B}_1, \mathcal{B}_2, w_c) := \begin{cases} 0, & |\delta| \leq \rho\gamma \\ \gamma \cdot \frac{|\delta - \rho\gamma|}{\|d\gamma - \rho w_c\|}, & \textit{otherwise} \end{cases}$$

$$UB(\mathcal{B}_1, \mathcal{B}_2, w_c) := \begin{cases} \gamma \cdot \frac{|\delta - \rho\gamma|}{\|d\gamma - \rho w_c\|}, & \delta \leq 0 \\ \gamma \cdot \frac{|\delta + \rho\gamma|}{\|d\gamma + \rho w_c\|}, & \delta \geq 0 \end{cases}$$

Proof. We begin by taking the Minkowsky difference of $\mathcal{B}(c_1, r_1)$ with $\mathcal{B}(c_1, r_1)$ which results in $\mathcal{B}(d, \rho) = \{c_2 - c_1 + z \mid \|z\| \leq \rho\}$. Regarding the *lower bound* we can adapt the bound from [18] to the un-normalized case of Def.5.2. Let us distinguish between two scenarios, one in which the hyperplane defined by $w_c^\perp$ intersects $\mathcal{B}(d, \rho)$ and one in which it does not. Intersection necessitates that the distance from the center d to the hyper-plane $w_c^\perp$ be less than ρ. If the intersection condition is met, there exists a point in $\mathcal{B}(d, \rho) \cap w_c^\perp$ rendering the distance null. When the hyper-plane and the ball are disjoint, we seek for the nearest point to the hyper-plane achievable in the ball. Such a point $v = d + z, \|z\| \leq \rho$ minimizes the parallel component of d w.r.t. w_c. This point is attained at $z^\star = -\rho\frac{w_c}{\gamma}$ with $\gamma = \|w_c\|$. Evaluating the distance from w_c to $v^\perp$, we get:

$$\frac{|\langle d - \rho\frac{w_c}{\gamma}, w_c\rangle|}{\|d - \rho\frac{w_c}{\gamma}\|} = \gamma\frac{|\langle d, w_c\rangle - \rho\gamma|}{\|d\gamma - \rho w_c\|}$$

We derive $|\langle d, w_c\rangle| \leq \rho\gamma$ from the intersection condition. When searching for an *upper bound* we seek the furthest achievable point from the hyper-plane $w_c^\perp$. Yet again we distinguish between two scenarios: one where the center d lies in the positive half-space

$\{\langle x, w_c \rangle \geq 0, x \in \mathcal{F}\}$ and one where it lies in the negative half-space. Contrary to the lower bound, we seek a point $v = d + z, \|z\| \leq \rho$ that maximizes the parallel component of d w.r.t. w_c. This point is attained at $z^\star_+ = \rho \frac{w_c}{\gamma}$ with $\gamma = \|w_c\|$ in the positive half-space scenario and at $z^\star = -\rho \frac{w_c}{\gamma}$ in the negative half-space scenario. We obtain the desired upper bound by evaluating the distance from w_c to $v^\perp$. □

Algorithm 3. The procedure follows a best-first search strategy guided by c_s: the minimum distance from two node centroids to the query hyperplane $q^\perp$. Algorithm 4 maintains the visitation order. Given the target threshold $\tau = \min\{d^\star, r_{\max}\}$, it (i) discards every node pair whose lower bound already exceeds τ and (ii) immediately reinserts any pair whose *upper* bound is below τ, using a sentinel key so it will be popped next. The search stops when the heap is empty or when the current best distance $d^\star \leq \tau$. When the most promising entry (A, B) is popped at iteration i, two cases arise. If both nodes are leaves, the algorithm evaluates the pair-to-hyperplane distance for every rule combination in $A \times B$ (Def. 5.2) and updates $(r^\star_i, r^\star_j)$ whenever a smaller distance is found. Otherwise it expands the branch by pushing to the heap the four cross-child pairs $(A_l, B_l), (A_l, B_r), (A_r, B_l), (A_r, B_r)$ as well as the two intra-child pairs (A_l, A_r) and (B_l, B_r).

6 Experiments

For all experiments we employ the following statistical features: Yule's Q, Cosine, Kruskal's τ, Added Value, and the Certainty Factor (see Suppl. Mat. 8.1 for more experimental configuration details).

Statistical-rule–measure oracle (ϕ). Tan et al. [27] showed that many statistical interestingness measures fall into a few *equivalence classes*: measures in the same class rank rules in exactly the same order, whereas measures from *different* classes do not. The measure ϕ belongs to one such class. For our feature set we deliberately picked five measures, each drawn from a class *different* from ϕ's. Hence every single feature is, by construction, *agnostic* to ϕ. Their *aggregate*, however, need not be: a non-linear aggregator such as the Choquet integral could exploit interactions between the features and recover information that none of them carries alone. By evaluating Alg. 1 against the ϕ oracle we therefore test whether the Choquet integral can "bridge the gap" from individually ϕ-agnostic features to a collective representation that aligns with the ϕ ranking.

Surprise against a maximum–entropy oracle. If all you know about a *transactional database* are the single–item frequencies p_i, the most neutral assumption is that items occur *independently*. The probability that every item in a rule $X \Rightarrow Y$ appears together is $\prod_{i \in X \cup Y} p_i$, so in n transactions we *expect* $f_{\text{exp}} = n \prod_{i \in X \cup Y} p_i$ such co-occurrences. The oracle compares this to the *observed* count f_{obs} and assigns the score $S = \log_2 \frac{f_{\text{obs}}}{f_{\text{exp}}} = \log_2 f_{\text{obs}} - \log_2 f_{\text{exp}}$. By evaluating against this surprise oracle we test the adaptivity of Alg. 1 to a highly agnostic setting.

Consequences of increasing additivity. Figures 3a–3c reveal a clear trade-off between model expressiveness and convergence speed. In the 1-additive setting, the Minkowski and Chebyshev centres (orange and blue solid lines) converge fastest and attain the highest scores on every metric for the ϕ oracle, showing that even a linear aggregation of ϕ-agnostic features can closely approximate ϕ's ranking. Moving to the 2- and 3-additive settings (Fig. 3b and 3c) slows convergence but raises performance on the highly agnostic surprise oracle (dashed lines). Hence, greater additivity yields more expressive—and ultimately more accurate—models, but at the cost of slower convergence.

Query selection and iteration time. Figure 3d reports the mean wall-clock time per iteration (averaged over folds and datasets) for the 2-additive experiments. As ChoquetRank chooses queries by random sampling, its running time is essentially constant, which explains the near-flat curve. The branch-and-bound strategy behaves differently: as learning proceeds the radius of the largest admissible ball shrinks exponentially, and finding a rule pair whose separating hyperplane still intersects that ball becomes increasingly costly, thus the search time grows. Inexplicably, the Minkowski center shows a rise–fall pattern which may be due to its evolving position in the version space, starting near a boundary and gradually settling inward. To verify that the algorithm does not keep cutting the version space with nearly parallel hyperplanes, we also plot the cumulative distribution of pairwise angles between successive preference constraints. The almost-uniform CDF in Figure 3d indicates that the cutting directions remain well diversified throughout the run (Cf. Supp. 8.1).

Top-*k* diversity and overall performance (Cf. Supp. 8.1) Figures 3e–3g plot the Jaccard similarity (lower is better) between the transaction covers of the top-15 rules. All methods follow the same general pattern, yet a clear trend towards *greater diversity* appears as we move from 1- to 3-additive models. Thus higher additivity not only improves accuracy but also yields rule lists with more diversified covers—another manifestation of the expressiveness-versus-convergence trade-off.

Implementation details. The entire pipeline is implemented in Python and released under an open-source license here[2]. All raw datasets together with their evaluations can be downloaded here[3].

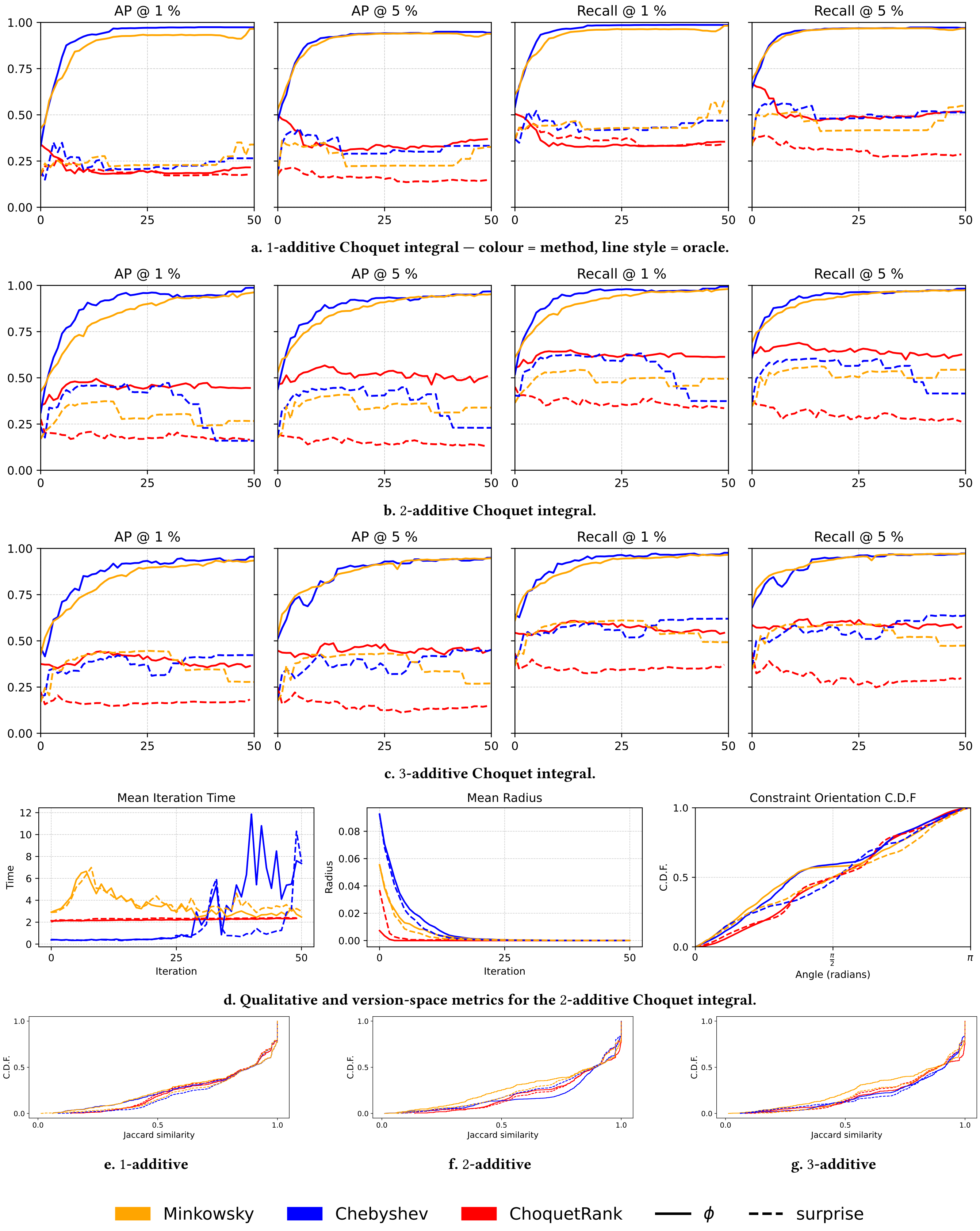
7 Conclusion

Across more than 1 000 query iterations per oracle–method pair, our geometry-driven strategy consistently outperforms the heuristic baseline of ChoquetRank [28] on precision, recall, and convergence rate at low cut-offs. The branch-and-bound strategy entails higher per-iteration cost, but it deterministically finds a query hyperplane that intersects the current version space, whereas the random sampling in ChoquetRank offers no such guarantee. The additional computation therefore yields consistently better rankings and more informative queries—a favorable trade-off in most scenarios.

Acknowledgments We gratefully acknowledge Laurent Truffet for the insightful discussions on polytopal geometry. This research project was financially supported by the French national research project FIDD under grant agreement ANR-24-CE23-0711.

[2] https://github.com/Tudor1415/KDD-UMC-Geometric-AL

[3] https://drive.google.com/drive/folders/132GJjjRn1ypJYsFCil8GY51zZHWUU8Ji



a. 1**-additive Choquet integral — colour = method, line style = oracle.**

b. 2**-additive Choquet integral.**

c. 3**-additive Choquet integral.**

d. Qualitative and version-space metrics for the 2**-additive Choquet integral.**

e. 1**-additive**

f. 2**-additive**

g. 3**-additive**

Figure 3: Choquet-integral performance and diversity analyses across additive orders.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th VLDB*. San Francisco, CA, USA, 487–499.

[2] Nir Ailon. 2012. An Active Learning Algorithm for Ranking from Pairwise Preferences with an Almost Optimal Query Complexity. *J. Mach. Learn. Res.* 13 (2012), 137–164.

[3] Noga Alon. 2006. Ranking Tournaments. *SIAM J. Discret. Math.* 20 (2006), 137–142. https://api.semanticscholar.org/CorpusID:3074207

[4] Mansurul A Bhuiyan and Mohammad Al Hasan. 2016. PRIIME: A generic framework for interactive personalized interesting pattern discovery. In *2016 IEEE International Conference on Big Data (Big Data)*. 606–615. doi:10.1109/BigData.2016.7840653

[5] Stephen Boyd and Lieven Vandenberghe. 2004. *Analytic Center of Linear Inequalities*. Cambridge University Press, Cambridge, UK.

[6] Dick den Hertog, Jean Pauphilet, and Mohamed Yahya Soali. 2024. Minkowski Centers via Robust Optimization: Computation and Applications. *Operations Research* 72, 5 (2024), 2135–2152.

[7] Kyriaki Dimitriadou, Olga Papaemmanouil, and Yanlei Diao. 2016. AIDE: An Active Learning-Based Approach for Interactive Data Exploration. *IEEE Transactions on Knowledge and Data Engineering* 28, 11 (2016), 2842–2856. doi:10.1109/TKDE.2016.2599168

[8] Vladimir Dzyuba and Matthijs Leeuwen. 2017. Learning What Matters – Sampling Interesting Patterns. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. doi:10.1007/978-3-319-57454-7_42

[9] Vladimir Dzyuba, Matthijs Van Leeuwen, Siegfried Nijssen, and Luc De Raedt. 2013. Active Preference Learning for Ranking Patterns. In *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*. 532–539. doi:10.1109/ICTAI.2013.85

[10] Vladimir Dzyuba and Matthijs van Leeuwen. 2017. Learning What Matters - Sampling Interesting Patterns. In *PAKDD 2017*. 534–546.

[11] Vladimir Dzyuba, Matthijs van Leeuwen, Siegfried Nijssen, and Luc De Raedt. 2014. Interactive Learning of Pattern Rankings. *International Journal on Artificial Intelligence Tools* 23, 6 (2014).

[12] Michel Grabisch. 1997. k-order additive discrete fuzzy measures and their representation. *Fuzzy Sets Syst.* 92, 2 (1997), 167–189.

[13] Michel Grabisch. 2004. The Möbius transform on symmetric ordered structures and its application to capacities on finite sets. *Discret. Math.* 287, 1-3 (2004), 17–34.

[14] Michel Grabisch and Christophe Labreuche. 2016. *Fuzzy Measures and Integrals in MCDA*. Springer New York, New York, NY, 553–603. doi:10.1007/978-1-4939-3094-4_14

[15] M. Grabisch and M. Roubens. 2000. Application of the Choquet integral in multicriteria decision making. In *Fuzzy Measures and Integrals - Theory and Applications*. 348–374.

[16] Margot Herin, Patrice Perny, and Nataliya Sokolovska. 2024. Noise-Tolerant Active Preference Learning for Multicriteria Choice Problems. In *Algorithmic Decision Theory: 8th International Conference, ADT 2024, New Brunswick, NJ, USA, October 14–16, 2024, Proceedings* (New Brunswick, NJ, USA). Springer-Verlag, Berlin, Heidelberg, 191–206. doi:10.1007/978-3-031-73903-3_13

[17] A. Hien, S. Loudni, N. Aribi, A. Ouali, and A. Zimmermann. 2023. Interactive Pattern Mining Using Discriminant Sub-patterns as Dynamic Features. In *PAKDD 2023 Proceedings (Lecture Notes in Computer Science, Vol. 13935)*. Springer, 252–263.

[18] Qiang Huang and Anthony K. H. Tung. 2023. Lightweight-Yet-Efficient: Revitalizing Ball-Tree for Point-to-Hyperplane Nearest Neighbor Search . In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE Computer Society, Los Alamitos, CA, USA, 436–449. doi:10.1109/ICDE55515.2023.00040

[19] Laurent Hyafil and Ronald L. Rivest. 1976. Constructing optimal binary decision trees is NP-complete. *Inform. Process. Lett.* 5, 1 (1976), 15–17. doi:10.1016/0020-0190(76)90095-8

[20] Hang Li. 2011. Learning to Rank for Information Retrieval and Natural Language Processing. In *Synthesis Lectures on Human Language Technologie*. Morgan & Claypool.

[21] J.-L. Marichal. 2000. An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria. *IEEE Transactions on Fuzzy Systems* 8, 6 (2000), 800–807. doi:10.1109/91.890347

[22] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2023. Human-in-the-loop machine learning: a state of the art. *Artif. Intell. Rev.* 56, 4 (2023), 3005–3054.

[23] Stephen Mussmann and Percy Liang. 2018. Generalized binary search for split-neighborly problems. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1561–1569.

[24] Robert Nowak. 2008. Generalized binary search. *2008 46th Annual Allerton Conference on Communication, Control, and Computing* (2008), 568–574. https://api.semanticscholar.org/CorpusID:8787894

[25] Tudor Matei Opran and Samir Loudni. 2025. Échantillonnage Actif pour la Découverte de Règles Classification via des Comparaisons par Paires. In *Extraction et Gestion des Connaissances, EGC'2025*. 231–238.

[26] Luciano Di Palma, Yanlei Diao, and Anna Liu. 2024. Efficient Version Space Algorithms for Human-in-the-loop Model Development. *ACM Transactions on Knowledge Discovery from Data* 18, 3 (2024), 1–49.

[27] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. 2004. Selecting the right objective measure for association analysis. *Information Systems* 29, 4 (June 2004), 293–313.

[28] C. Vernerey, N. Aribi, S. Loudni, Y. Lebbah, and N. Belmecheri. 2024. Learning to Rank Based on Choquet Integral: Application to Association Rules. In *PAKDD 2024*. 313–326.

[29] Charles Vernerey and Samir Loudni. 2023. A Java Library for Itemset Mining with Choco-solver. *Journal of Open Source Software* 8, 88 (2023), 5654. doi:10.21105/joss.05654

[30] Yichong Xu, Hongyang Zhang, Kyle Miller, Aarti Singh, and Artur Dubrawski. 2017. Noise-Tolerant Interactive Learning Using Pairwise Comparisons. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/e11943a6031a0e6114ae69c257617980-Paper.pdf

8 Supplementary Material

8.1 Experimental Details

Experimental Protocol. We carried out our experiments on the five UCI datasets listed in Table 1. For every dataset we mined, with *Choco–Mining* [29], up to ten million *minimal, non-redundant* association rules that satisfy a confidence of at least 99% and a support of at least 10. Each rule was evaluated on five statistical interestingness measures: **Yule's Q**, **Cosine similarity**, **Goodman–Kruskal's τ**, **Added Value**, and the **Certainty Factor**. Rules with identical five-dimensional feature vectors were deduplicated, but dominated rules were *not* removed so as not to bias the study against oracles that might legitimately prefer a dominated rule over a Pareto-optimal one. After de-duplication we capped the rule set of every dataset at 100 000 and evaluated all methods under *three-fold cross-validation.* For both ChoquetRank and our branch-and-bound learner, the loop stopped as soon as the oracle replied 'indifferent' or 0, since neither algorithm currently supports indifference feedback.

Jaccard Top-k similarity plots. To measure how diverse the highest–ranked rules are, we proceeded as follows. First, every rule was ranked with the final iteration Choquet capacity for its oracle–center–fold setting. Next, we kept the *top fifteen* rules and turned each one into a binary cover vector that flags the transactions it captures. For every pair of covers $(\mathbf{v}_i, \mathbf{v}_j)$ we then computed the Jaccard similarity

$$J(\mathbf{v}_i, \mathbf{v}_j) = \frac{\|\mathbf{v}_i \wedge \mathbf{v}_j\|_1}{\|\mathbf{v}_i \vee \mathbf{v}_j\|_1},$$

where $\wedge$ and $\vee$ are element-wise logical AND / OR. The empirical distribution of these J values—aggregated over folds—underlies the similarity curves shown in the paper.

Constraint orientation plot. For every query we record the *augmented* difference vector $\mathbf{h} = \mathcal{A}(\Phi(r_i)) - \mathcal{A}(\Phi(r_j))$ To derive a linear constraint in the $d-1$-dimensional where the version space polytope is fully dimensional (does not contain any equality constraints), we substitute the weight sum equality, yielding

$$\mathbf{a} \;=\; (h_1, \ldots, h_d) - h_{d+1}\mathbf{1}, \qquad b \;=\; -h_{d+1},$$

such that the preference reads $\mathbf{a}^\top \mathbf{w} \le b$. Normalizing $\mathbf{a}$ gives the unit normal $\mathbf{n} = \mathbf{a}/\|\mathbf{a}\|_2$. Collecting all such normals for a given *(oracle, centre)* pair, forming every unordered pair $(\mathbf{n}_i, \mathbf{n}_j)$, and computing $\theta_{ij} = \arccos(\mathbf{n}_i^\top \mathbf{n}_j)$ produces a set of angles whose empirical C.D.F. is the constraint–orientation curve in Figure 3d.

Table 1: Statistics of the datasets used in the experiments.

Dataset	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{D}\rvert$	$\lvert\mathcal{R}\rvert$
mushroom	95	8 124	100 000
tictactoe	28	958	23 473
magic	80	19 020	100 000
credit	73	30 000	100 000
twitter	214	49 999	100 000

8.2 The Minkowski Center

It maximizes a measure λ of central symmetry and can be computed using the following LPs from [6].

Minkowski–centre LP

$$\max_{w,\lambda} \lambda \tag{3}$$

$$\text{s.t.} \quad Aw = (1+\lambda)\, b \tag{4}$$

$$Cw - \lambda\, \delta \le d \tag{5}$$

$$\lambda \ge 0 \tag{6}$$

Facet–minimum LPs for δ

$$\delta_i = \min_y c_i^\top y \tag{7}$$

$$\text{s.t.} \quad Ay = b \tag{8}$$

$$Cy \le d \tag{9}$$

Figure 4: LPs used to compute the Minkowski center and the facet offsets δ.

The program in Fig. 4 maximizes the symmetry factor λ of the polytope $C = \{x \mid Ax = b,\ Cx \le d\}$. Constraint $Aw = (1+\lambda)b$ ensures that the scaled point $w/(1+\lambda)$ still satisfies all equalities. The inequality $Cw - \lambda\delta \le d$ tightens every facet by the amount $\lambda\delta$, guaranteeing that every λ-scaled reflection of C about the candidate center remains inside C. The bound $\lambda \ge 0$ rules out negative symmetry radii. At optimality, $\lambda^\star$ equals the Minkowski symmetry and $x^\star = w^\star/(1+\lambda^\star)$ is a Minkowski center.

8.3 The Chebyshev Center

It maximizes the radius r of the largest inscribed ball in a polytope. In Fig. 5 $x \in \mathbb{R}^n$ is the candidate center of the largest inscribed Euclidean ball and $r \in \mathbb{R}_+$ its radius. The equalities $Ax = b$ keep the center on the affine subspace common to every feasible point. For each facet $c_i^\top x \le d_i$, the term $\kappa_i = \|c_i\|_2$ measures how far that facet shifts when the ball grows; the constraint $Cx + r\kappa \le d$ therefore pulls every facet inward by exactly r so that the entire ball stays inside the polytope. Maximizing r then yields the Chebyshev radius $r^\star$ and its associated center $x^\star$.

Chebyshev–centre LP

$$\max_{x,r} r \tag{10}$$

$$\text{s.t.} \quad Ax = b \tag{11}$$

$$Cx + r\,\kappa \;\le\; d \tag{12}$$

$$r \ge 0 \tag{13}$$

Figure 5: LP used to compute the Chebyshev center.