Kalpa Publications in Computing
Volume 4, 2018, Pages 264–283
28th International Workshop on
Principles of Diagnosis (DX'17)

On Active Learning Strategies for Sequential Diagnosis

Patrick Rodler

Alpen-Adria Universität, Klagenfurt, Austria
patrick.rodler@aau.at

Abstract

When diagnosing a faulty system one is often confronted with a large number of possible fault hypotheses. Sequential Diagnosis (SD) techniques aim at the localization or identification of the actual fault with minimal cost or effort. SD can be viewed as an Active Learning (AL) task where the learner, trying to find some target hypothesis, formulates sequential queries to some oracle, thereby e.g. requesting additional system measurements. Several query selection measures (QSMs) for determining the best query to ask next have been proposed for AL. To date, few of them have been translated to and employed in SD. In this work, we account for this and analyze various QSMs wrt. to the discrimination power of their selected queries within the diagnostic hypotheses space. As a result, we derive superiority and equivalence relations between these QSMs and introduce improved versions of existing QSMs to overcome identified issues. The obtained picture gives a hint about which QSMs should preferably be used in SD to choose a query from a pool of candidates. Moreover, we deduce properties optimal queries wrt. QSMs must satisfy. Based on these, we devise an efficient heuristic search for optimal queries. As (preliminary) evaluation results indicate, the latter is especially beneficial in applications where query generation is costly, e.g. involving logical reasoning, and hence a pool of query candidates is not (cheaply) available.

1 Introduction

Given a system that does not behave as expected, diagnosis approaches aim at the determination of the actual faulty system state that causes the observed misbehavior. A wide range of such approaches have been presented for various system types such as hardware [9, 31, 27, 11, 15], software [41, 21, 16, 45], knowledge bases [14, 19, 39, 32], discrete event systems [28], feature models [44], user interfaces [13] or spreadsheets [1]. However, usually such diagnosis methods have to deal with a large number of different fault hypotheses. To provide for hypotheses discrimination, *Sequential Diagnosis (SD)* techniques [9, 29, 12, 39, 45] gather additional information in terms of observations or tests. The goal in SD is the minimization of the effort or cost until complete (or at least a reasonable) diagnostic accuracy is achieved. Unfortunately, this problem has been shown to be NP-complete [18, 27]. Thus, as a trade-off between optimality and computational complexity, it is current practice in SD to rely on myopic methods to guide hypotheses discrimination [9, 12, 16, 39, 35]. Empirical (e.g. [8]) and theoretical (e.g. [27]) evaluations have shown that such heuristic methods in many cases deliver good or even (nearly) optimal results.

Research in the field of *Active Learning (AL)* [37] provides a range of diverse general (families of) heuristics targeting the optimization of hypotheses discrimination tasks. While traditionally and very fruitfully exploited in machine learning, e.g. for efficient text classification [43], image retrieval [42], concept learning [6], machine translation [2], or natural language processing [26], the key idea behind AL is that a learner can achieve greater accuracy with less newly collected information if the used training data can be adaptively chosen based on its current state of knowledge. At each iteration of the learning process, the active learner can consult an oracle, e.g. a human expert, to label any *query* from some predefined query space. The new information in terms of the query's label is then taken into account to update the learner's current knowledge state.

As this exactly captures the generic information acquisition process pursued by SD systems, many AL strategies, termed *Query Selection Measures (QSMs)*, are basically tailored for being used in SD. QSMs are real-valued functions quantifying the quality of queries. However, to date only few of these AL QSMs, e.g. information entropy [9], have been applied to SD. This is where this work begins.

Depending on the used SD framework, e.g. *model-based* [31] or *matrix-based* [38], queries might be e.g. logical sentences [14, 39, 32] or requested probes [9] in the former, and pass-fail tests [27, 29, 16] in the latter. Which instance acts as an oracle depends on the faulty system at hand, e.g. for a car diagnosis task [17] it might be a car mechanic and for digital circuits [9] an electrical engineer that provides the necessary measurements to answer a query, for knowledge bases [39] it might be a domain expert answering queries about (non-)entailments of the correct knowledge base, and for software [16] an IDE able to run required tests.

In any case, the goal of a query is to discriminate well between competing hypotheses. At this, irrespective of the particular used QSM, a minimal requirement usually postulated is that any query outcome must lead to the dismissal of at least some (known) hypothesis. We call such queries *discriminating queries*. Another plausible general requirement to queries, besides the postulation of a favorable QSM-value, is that they should discriminate among an as large as possible number of (known) hypotheses. In other words, there should ideally be no *uncommitted hypotheses* [9] for a query, i.e. hypotheses that do not predict any query outcome (and hence can never be invalidated by asking the query). We call such a query a *strong query*. Intuitively, the more uncommitted hypotheses there are for a query, the lower its discrimination power and the less favorable it tends to be.

In fact, there might be uncommitted hypotheses for queries in SD. For example, in model-based diagnosis [31, 9] they might occur due to incomplete system knowledge or too few observations; in spectrum-based diagnosis [16, 38], they can arise in the presence of intermittent failure behavior [16] of system components, which might not enable to assess for sure whether a test must pass or fail given a particular hypothesized faulty state of the system. As usually the hypotheses types, e.g. decision trees or neural networks, considered in classical machine learning entail a label for each query from the predefined query space, AL QSMs by default do not deal with uncommitted hypotheses. Therefore, when used for SD, they might propose queries with suboptimal discrimination power despite the presence of better queries. We provide a thorough analysis of this issue enabling the recommendation of more and less suitable QSMs to be adopted for SD.

AL distinguishes among various learning scenarios. Two of them, *pool-based sampling* and *query synthesis* [37], are relevant for SD. The former assumes that a (large) pool of unlabeled queries is (cheaply) available and that the best query wrt. a QSM is determined by comparing the QSM-value of all queries in the pool. In the latter, in contrast, an algorithm tries to *generate* an unlabeled query with sufficiently good QSM-value. To the best of our knowledge, current SD methods merely adopt the pool-based paradigm. Whereas this seems appropriate in e.g. spectrum-based SD approaches [38, 16] where the possible tests (i.e. unlabeled queries) are explicitly given before the SD process starts (and relatively cheaply obtainable through e.g. test execution profiling), it might often be not optimally suited for e.g. model-based SD approaches [31, 9, 14, 12, 39, 35], where the computation of a pool of (discriminating) query candidates might rely on expensive logical derivations from the given model. Moreover, for all discussed QSMs, the computation of a query's QSM-value requires knowledge about its discrimination properties (see later), which is explicitly given (test matrix) for the former approaches and must be costly derived (logical reasoning) in the latter. For these reasons, query synthesis in principle appears to be a promising solution especially in model-based applications as it attempts to actually compute a minimal number of queries and associated QSM-values until a (sufficiently) good one is found. The viability and benefit of one query synthesis method to model-based SD has been recently shown in [34].

Contributions. In this paper we analyze various AL QSMs and

1. reformulate these QSMs to be appropriate for SD with binary-outcome queries,
2. define a plausible general discrimination preference order (DPO) on queries (formalizing the notion of "discrimination power"),
3. formally characterize a superiority relation on QSMs based on the (degree of) their compliance with the DPO,
4. figure out superiority relationships between QSMs which suggests a preference order on QSMs helping to opt for the most suitable QSM, especially in pool-based scenarios,
5. derive improved (parameterized) versions from some QSMs to overcome unveiled deficits,
6. formalize the notion of equivalence between QSMs based on their preference order on queries,
7. give equivalence classes of QSMs under various conditions (query spaces, QSM parametrizations),
8. analyze QSM functions regarding their global optima and determine properties of optimal input arguments (i.e. optimal queries),
9. show how these properties can be used to design efficient heuristic search procedures for the *systematic* construction of (nearly) optimal queries wrt. a QSM in a query synthesis scenario, and
10. provide (preliminary) evaluation results on the proposed general query synthesis approach using real-world diagnosis problems demonstrating low cost, high query quality as well as significant superiority to pool-based approaches when query computation requires logical reasoning.

2 Preliminaries

In an SD setting we consider there is a (not necessarily explicitly given) set of unlabeled queries $\mathcal{U}$ and a (possibly empty) set of already labeled queries $\mathcal{L}$.[1] A *labeled (or: answered) query* in $\mathcal{L}$ is a tuple (Q, a_Q) where Q is a query and $a_Q \in \{0,1\}$. $a_Q = 1$ ($a_Q = 0$) means that the query Q is answered by *true* (*false*). Queries are answered by an *oracle* given by the total function $ans : \mathcal{U} \to \{0,1\}$ which maps queries $Q \in \mathcal{U}$ to their respective answer a_Q.

The goal in SD is to find the *target hypothesis* h_t, i.e. the actual (faulty) system state, from a hypothesis space $\mathcal{H}$ which depends on the SD task. E.g., in model-based diagnosis each $h \in \mathcal{H}$ is a *diagnosis*, i.e. an assumption about the faulty/healthy-state of each (relevant) component of the system under consideration. For a matrix-based diagnosis task, on the other hand, each hypothesis might be one of a number of *predefined (faulty) system states*.

Given a set of labeled queries $\mathcal{L}$, any hypothesis $h \in \mathcal{H}$ is still possible if it is *consistent* with $\mathcal{L}$. The set including all $h \in \mathcal{H}$ consistent with $\mathcal{L}$ is called the current *version space* $\mathcal{V} \subseteq \mathcal{H}$ [23]. As discussed in Sec. 1, in general each $h \in \mathcal{H}$ entails an answer for *a subset of* the unlabeled queries in $\mathcal{U}$. Hence, each query Q imposes a partition on $\mathcal{H}$ into three sets $\langle \mathcal{H}^+_Q, \mathcal{H}^-_Q, \mathcal{H}^0_Q \rangle$: $\mathcal{H}^+_Q$ includes those $h \in \mathcal{H}$ consistent only with $a_Q = 1$ (predicting Q's positive answer), $\mathcal{H}^-_Q$ those $h \in \mathcal{H}$ consistent only with $a_Q = 0$ (predicting Q's negative answer), and $\mathcal{H}^0_Q$ those consistent with both $a_Q = 1$ and

[1] The general term *query*, borrowed from AL, is used to refer to different means of information acquisition, e.g. probes [9], tests [27] or test cases [14], depending on the concrete SD approach.

$a_Q = 0$ (not predicting any answer). That is, the new (still consistent) hypotheses set after $ans(Q) = 1$ is known (i.e. $(Q, 1)$ is added to $\mathcal{L}$) is $\mathcal{H} \setminus \mathcal{H}_Q^-$. Otherwise, if $(Q, 0)$ is added to $\mathcal{L}$, the new hypotheses set is $\mathcal{H} \setminus \mathcal{H}_Q^+$.

We assume that the oracle ans provides correct answers. That is, if the target hypothesis h_t is in $\mathcal{H}_Q^+$ and $\mathcal{H}_Q^-$, respectively, then $ans(Q) = 1$ and $ans(Q) = 0$. We stress that the oracle is a *total* function and thus assumed to answer *every* query $Q \in \mathcal{U}$, even if $h_t \in \mathcal{H}_Q^0$. E.g., even though h_t in a circuit diagnosis task might not entail whether a particular wire is high or low, probing the wire will provide an answer. But, for either outcome, h_t remains valid a-posteriori.

As the explicit computation of the full version space $\mathcal{V} \subseteq \mathcal{H}$ might be hard or even infeasible [9, 39, 32], we assume that some subset V of $\mathcal{V}$ is known at each query selection. In SD the set V is often referred to as *leading diagnoses* [10] and usually comprises the most probable [7] or minimum-cardinality [12] hypotheses. As with $\mathcal{H}$, a query Q partitions V into $V_Q^+ := V \cap \mathcal{H}_Q^+$, $V_Q^- := V \cap \mathcal{H}_Q^-$ and $V_Q^0 := V \cap \mathcal{H}_Q^0$. We denote by $\mathfrak{P}_V(Q) := \langle V_Q^+, V_Q^-, V_Q^0 \rangle$ the (unique) *partition of Q (wrt. V)*. Generally, multiple queries Q might have the same partition $\mathfrak{P}_V(Q)$. We call $Q \in \mathcal{U}$ a *discriminating query (DQ) (wrt. V)* iff $V_Q^+ \neq \emptyset$ and $V_Q^- \neq \emptyset$. Else, we call Q a *non-DQ*. Similarly, we call $\mathfrak{P}_V(Q)$ a *discriminating partition (DP) (wrt. V)* iff Q is a DQ (wrt. V). That is, either label $a_Q \in \{0, 1\}$ of a DQ Q eliminates at least one $h \in V$ or, respectively, at least two hypotheses in V make different predictions as to a_Q. Intuitively, one will try to avoid asking any $Q \in \mathcal{U}$ which is not a DQ. Because – based on the current evidence in terms of V – it is not sure that any relevant new information will be gained by obtaining a_Q. A query $Q \in \mathcal{U}$ is termed *weak query (wrt. V)* iff $V_Q^0 \neq \emptyset$. Otherwise, we call Q *strong query (wrt. V)*. Analogously, we call $\mathfrak{P}_V(Q)$ a *strong / weak partition (wrt. V)* iff Q is a strong / weak query (wrt. V).

An AL *query selection measure (QSM)* is a function $m : \mathcal{U} \to \mathbb{R}$ assigning to each query $Q \in \mathcal{U}$ a (quality) measure $m(Q) \in \mathbb{R}$. A *theoretical optimum X wrt. m* is a hypothetical (not necessarily real) DQ X which globally optimizes $m(X)$. Depending on the QSM m, "optimizing m" can mean either maximizing or minimizing m. An *optimal query Q wrt. m and V* is a DQ wrt. V with optimal $m(Q)$ among all DQs wrt. V. Note, theoretical optima and optimal queries need not be unique.

In line with the works [9, 5, 39, 32] we assume a probability space over $\mathcal{H}$ as follows: Each $h \in \mathcal{H}$ has an a-priori probability $p(h)$ of being the target hypothesis h_t, i.e. $p(h) := p(h = h_t)$. Given a currently known subset V of the version space $\mathcal{V} \subseteq \mathcal{H}$, we define $p(X) := \sum_{h \in X} p(h)$ for $X \subseteq V$ and assume p to be normalized over V such that that $p(V) = 1$. Since the version space includes only still possible hypotheses, $p(h) > 0$ must hold for all $h \in V$. For any $Q \in \mathcal{U}$ and oracle ans: $p(ans(Q) = 1) := p(V_Q^+) + \frac{p(V_Q^0)}{2}$ and $p(ans(Q) = 0) = p(V_Q^-) + \frac{p(V_Q^0)}{2}$ i.e. the uncommitted hypotheses $h \in V_Q^0$ are assumed to predict each answer with a probability of $\frac{1}{2}$. The posterior probability $p(h \mid ans(Q) = a_Q)$ of some $h \in \mathcal{H}$ can be computed by the Bayesian Theorem as $p(ans(Q) = a_Q|h)\ p(h)/p(ans(Q) = a_Q)$ where $p(ans(Q) = 1 \mid h)$ is 1 if $h \in \mathcal{H}_Q^+$, 0 if $h \in \mathcal{H}_Q^-$, and $\frac{1}{2}$ if $h \in \mathcal{H}_Q^0$.

Example: Consider Tab. 1 which gives some partitions $\mathfrak{P}_V(Q_i)$ of $V := \{h_1, \dots, h_5\}$ for $1 \leq i \leq 4$. All associated queries Q_i (not given in Tab. 1) are DQs as $V_{Q_i}^+$ and $V_{Q_i}^-$ are non-empty for $1 \leq i \leq 4$. Hence, each partition in the table is a DP. Moreover, Q_1, Q_3 are strong and Q_2, Q_4 weak DQs due to empty and non-empty $V_{Q_i}^0$, respectively.

Assuming the probabilities $p := p_1$ over V (see Tab. 1), we have that, e.g.,

$$p(ans(Q_3) = 1) = p(V_{Q_3}^+) = p(\{h_4\}) = 0.25$$

$$p(ans(Q_2) = 0) = p(V_{Q_2}^-) + \frac{1}{2}p(V_{Q_2}^0) = p(\{h_3, h_4\}) + \frac{1}{2}p(\{h_5\}) = 0.15 + 0.25 + \frac{1}{2}0.2 = 0.5$$

i	$V^+_{Q_i}$	$V^-_{Q_i}$	$V^0_{Q_i}$
1	$\{h_1, h_2\}$	$\{h_3, h_4, h_5\}$	$\emptyset$
2	$\{h_1, h_2\}$	$\{h_3, h_4\}$	$\{h_5\}$
3	$\{h_4\}$	$\{h_1, h_2, h_3, h_5\}$	$\emptyset$
4	$\{h_1, h_2, h_5\}$	$\{h_4\}$	$\{h_3\}$

	h_1	h_2	h_3	h_4	h_5
$p_1(h_i)$	0.35	0.05	0.15	0.25	0.2
$p_2(h_i)$	0.01	0.02	0.8	0.15	0.02
$p_3(h_i)$	0.4	0.2	0.05	0.1	0.25

Table 1: Some sample partitions wrt. $V = \{h_1, \dots, h_5\}$ (*left*) and probability distributions p_1, p_2 and p_3 over V (*right*).

Let $m_1(Q) := |p(V^+_Q) - p(V^-_Q)| + p(V^0_Q)$ be a QSM (to be minimized). Then we have that $\langle m_1(Q_1), \dots, m_1(Q_4)\rangle = \langle 0.2, 0.2, 0.5, 0.5\rangle$. Supposing that $Q_1, \dots, Q_4$ are *all* possible DQs wrt. V, the optimal queries wrt. m_1 and V are Q_1 and Q_2. A theoretical optimum X wrt. m_1 satisfies $p(V^+_X) = p(V^-_X) = 0.5$ and $p(V^0_X) = 0$.

Let Q_2 be labeled negatively, i.e. $ans(Q_2) = 0$. Then the hypotheses h_1, h_2 are invalidated. The remaining ones are $V \setminus V^+_{Q_2} = \{h_3, h_4, h_5\}$. The (Bayes) updated probability distribution over V is $p(h_1) = p(h_2) = 0$, $p(h_3) = \frac{0.15}{0.5} = 0.3$, $p(h_4) = \frac{0.25}{0.5} = 0.5$ and $p(h_5) = \frac{(1/2)0.2}{0.5} = 0.2$. □

The generic SD procedure to which our analyses apply is:

Generic Sequential Diagnosis Procedure:
Input: Diagnosis problem[a]
Output: (Set of fault hypotheses including the) target hypothesis

1. Generate a subset V of the current version space $\mathcal{V}$.[b]
2. If a defined stop criterion (e.g. $|\mathcal{V}| = 1$ or some $h \in V$ has overwhelming probability $p(h)$) is met or no more queries wrt. V exist, return V and p. Else, go to (3.).
3. Select the best next DQ[c] Q based on the information in V and p and pose it to the oracle.
4. Given the answer a_Q to Q, run some update procedure that takes V, Q, a_Q and p as input and returns a new subset V of the updated version space $\mathcal{V}$ (possibly including previously unseen hypotheses) and an updated probability measure p. Go to (2.).

[a]Might be of different type, e.g., model-based or spectrum-based.
[b]There is nothing to do if $\mathcal{V}$ is explicitly given as, e.g., in some matrix-based SD approaches.
[c] Since non-DQs are not of interest in SD, as argued in Sec. 1 and 2, we assume that non-DQs are ignored at query selection. This is easily accomplished by discarding (i.e. non-selecting) any query Q with empty V^+_Q or V^-_Q.

3 Analysis of Active Learning Strategies for Sequential Diagnosis

In this section[2] we motivate and specify a general discrimination-preference order (DPO) over queries in $\mathcal{U}$, study various QSMs regarding their compliance with the DPO, present derived equivalence and superiority relations among these QSMs and specify some plausible new QSMs, e.g. as improved versions of existing ones. The results suggest which QSMs are more or less recommendable to be used in *pool-based* SD scenarios (cf. Sec. 1). Moreover, we analyze the QSM functions m wrt. their (theoretically) optimal inputs which lets us deduce properties of optimal strong DQs for the discussed QSMs. These properties provide the basis for a systematic construction of (or search for) optimal DQs in a *query synthesis* SD scenario (cf. Sec. 1).

[2]*Detailed* proofs of all results are given in Sec. 3.2ff. of the extended version [33] of this paper.

3.1 Relevant Definitions and Properties

We first point out that the partition of a query $Q \in \mathcal{U}$ (along with the probability measure p) gives already all the relevant information taken into account by QSMs m to determine Q's quality $m(Q)$. Because the partition enables

1. the test whether Q is a DQ (i.e. $V_Q^+ \neq \emptyset$ and $V_Q^- \neq \emptyset$),
2. the test whether Q is strong (i.e. $V_Q^0 = \emptyset$),
3. an estimation of the impact Q's answers have in terms of hypotheses elimination (potential a-posteriori change of the version space), and
4. the assessment of the probability of Q's positive and negative answers (e.g. to determine the uncertainty of Q).

QSMs might basically focus on pretty different properties of a query's partition when estimating its goodness. However, independently of the concrete used QSM, queries with a higher "discrimination power" should be preferred. Intuitively, given a query $Q_1 \in \mathcal{U}$ which is objectively better than $Q_2 \in \mathcal{U}$, we do not want a reasonable QSM to propose Q_2. We next define a general order on queries, called DPO, thereby formalizing the notion of "discrimination power". In the following we always assume $\mathcal{V}$ to be the current version space and $V \subseteq \mathcal{V}$.

Definition 1. *Let $Q, \overline{Q} \in \mathcal{U}$. Further, for any query $Q \in \mathcal{U}$ let $V_Q[\neg a] \subseteq V$ denote the hypotheses predicting $\neg a$ (i.e. inconsistent with $ans(Q) = a$). That is, exactly $V_Q[\neg a]$ is eliminated among all hypotheses in V given $ans(Q) = a$.*

Then we call Q discrimination-preferred to *$\overline{Q}$* (wrt. *V*) *iff there is an injective function $f : \{0,1\} \to \{0,1\}$ that maps each of $\overline{Q}$'s answers $\overline{a}_1, \overline{a}_2 \in \{0,1\}$ ($\overline{a}_1 \neq \overline{a}_2$) to one of Q's answers $a_i = f(\overline{a}_i)$ such that*

1. *$V_Q[\neg a_i] \supseteq V_{\overline{Q}}[\neg \overline{a}_i]$ for some $i \in \{1,2\}$, and*
2. *$V_Q[\neg a_j] \supset V_{\overline{Q}}[\neg \overline{a}_j]$ for $j \in \{1,2\}$ and $j \neq i$.*

We use $Q \prec_{\mathsf{DPO}} \overline{Q}$ to state that Q is discrimination-preferred to $\overline{Q}$ and call $\{(Q, \overline{Q}) \mid Q \prec_{\mathsf{DPO}} \overline{Q}\}$ the discrimination preference order (DPO).

Simply put, $Q \prec_{\mathsf{DPO}} \overline{Q}$ means: For each result one might get by asking the oracle $\overline{Q}$, there is a better result in terms of hypotheses elimination one can get by asking the oracle Q. In particular, for one of the answers $\overline{a}_i$ of $\overline{Q}$, some answer a_i to Q eliminates at least the same hypotheses. For the other answer $\overline{a}_j (\neq \overline{a}_i)$ of $\overline{Q}$, the other answer $a_j (\neq a_i)$ to Q eliminates strictly more hypotheses.

The idea underlying the DPO is that asking Q is *always* (i.e. for any answer) better than asking $\overline{Q}$ given that the target hypothesis h_t is in V and predicts an answer for both queries:

Proposition 1. *Let $Q \prec_{\mathsf{DPO}} \overline{Q}$ and $h_t \in V_Q^+ \cup V_Q^-$ and $h_t \in V_{\overline{Q}}^+ \cup V_{\overline{Q}}^-$. Then the remaining hypotheses in V after adding $(Q, ans(Q))$ to $\mathcal{L}$ is a subset of the remaining hypotheses in V after adding $(\overline{Q}, ans(\overline{Q}))$ to $\mathcal{L}$.*

Proof. The proposition follows from the fact that (i) for any $Q \in \mathcal{U}$, $ans(Q) = 1$ if $h_t \in V_Q^+$ and $ans(Q) = 0$ if $h_t \in V_Q^-$, that (ii) $(h_t \in V_Q^+) \oplus (h_t \in V_Q^-)$ and $(h_t \in V_{\overline{Q}}^+) \oplus (h_t \in V_{\overline{Q}}^-)$, and (iii) the subset-relations in (1) and (2) in Def. 1. ($\oplus$ denotes the standard xor-operator) □

Example (cont'd): In Tab. 1, $Q_1 \prec_{\mathsf{DPO}} Q_2$ and $Q_3 \prec_{\mathsf{DPO}} Q_4$. E.g. the latter, by Def. 1, holds since (1) for $ans(Q_3) = 0$, which eliminates $\{h_4\}$, there is an answer, namely $ans(Q_4) = 1$, which also dismisses $\{h_4\}$, and (2) for $ans(Q_3) = 1$ (making $\{h_1, h_2, h_3, h_5\}$ invalid) the answer $ans(Q_4) = 0$ is strictly worse (invalidating $\{h_1, h_2, h_5\}$).

Given e.g. $h_t \in \{h_1, h_2, h_4, h_5\}$, then the hypothesis elimination rate (wrt. V) of the discrimination-preferred Q_3 is better than the one of Q_4 for any oracle ans (Prop. 1). □

Every QSM imposes a (preference) order on a given set of queries $\mathcal{U}$:

Definition 2. *Let m be a QSM and $Q, Q' \in \mathcal{U}$. Then Q* is preferred to *Q'* by *m, formally $Q \prec_m Q'$, iff*

(a) $m(Q) < m(Q')$ if m is optimized by minimization,
(b) $m(Q) > m(Q')$ if m is optimized by maximization.

Two QSMs are equivalent iff they impose exactly the same preference order on queries:

Definition 3. *Let m_1, m_2 be QSMs. Then we call m_1* equivalent to *m_2 (m_1 $\mathfrak{X}$*-equivalent to *m_2), formally $m_1 \equiv m_2$ ($m_1 \equiv_{\mathfrak{X}} m_2$), iff for all queries $Q, Q' \in (\mathfrak{X} \subseteq)\, \mathcal{U}$: $Q \prec_{m_1} Q'$ iff $Q \prec_{m_2} Q'$.*

The next definition facilitates our analysis of the degree of compliance of QSMs with the DPO:

Definition 4. *Let m be a QSM. Then:*

- *We say that m* preserves (or: satisfies) the DPO (over $\mathfrak{X}$) *iff, whenever $Q \prec_{\mathsf{DPO}} Q'$ (and $Q, Q' \in \mathfrak{X}$), it holds that $Q \prec_m Q'$.*
  (I.e. the preference order imposed on queries by m is a superset of the DPO.)

- *We call m* consistent with the DPO (over $\mathfrak{X}$) *iff, whenever $Q \prec_{\mathsf{DPO}} Q'$ (and $Q, Q' \in \mathfrak{X}$), it does not hold that $Q' \prec_m Q$.*
  (I.e. the preference order imposed on queries by m has no intersection with the inverse DPO.)

We call QSMs with a higher compliance with the DPO superior to others:

Definition 5. *Let m_1, m_2 be QSMs. We call m_2* superior to *m_1 (or: m_1* inferior to *m_2), formally $m_2 \prec m_1$, iff*

1. *for some pair of queries Q, Q' where $Q \prec_{\mathsf{DPO}} Q'$ and not $Q \prec_{m_1} Q'$ it holds that $Q \prec_{m_2} Q'$ (i.e. in some cases m_2 does, but m_1 does not satisfy the DPO), and*

2. *for no pair of queries Q, Q' where $Q \prec_{\mathsf{DPO}} Q'$ and not $Q \prec_{m_2} Q'$ it holds that $Q \prec_{m_1} Q'$ (i.e. whenever m_2 does not satisfy the DPO, m_1 does not satisfy it either).*

Analogously, we call m_2 $\mathfrak{X}$-superior to *m_1 (or: m_1 $\mathfrak{X}$*-inferior to *m_2), formally $m_2 \prec_{\mathfrak{X}} m_1$, iff superiority of m_2 to m_1 holds over $\mathfrak{X} \subseteq \mathcal{U}$.*

The following proposition can be easily verified:

Proposition 2. *The following holds for the introduced relations:*

- *$\prec_m$ and $\prec_{\mathsf{DPO}}$ are strict orders, i.e. irreflexive, asymmetric and transitive relations over queries.*
- *$\equiv$ and $\equiv_{\mathfrak{X}}$ are equivalence relations over QSMs.*
- *$\prec$ and $\prec_{\mathfrak{X}}$ are strict orders over QSMs.*

The next proposition summarizes some easy consequences of the provided definitions:

Proposition 3. *Let m, m_1, m_2 be QSMs, $Q, Q' \in \mathcal{U}$, $\mathfrak{X} \subseteq \mathcal{U}$ and $Q_{m_i} \in \mathcal{U}$ denote the optimal query wrt. m_i ($i \in \{1, 2\}$) and V. Then:*

1. *$m_1 \equiv m_2$ implies $Q_{m_1} = Q_{m_2}$.*

2. *If m_1 does and m_2 does not satisfy the DPO, then $m_1 \prec m_2$.*

3. *$Q \prec_{\mathsf{DPO}} Q'$ implies $V^0_{Q'} \supset V^0_Q$. Thus, $V^0_{Q'} \neq \emptyset$.*

4. *If m satisfies the DPO (over $\mathfrak{X}$), then m is consistent with the DPO (over $\mathfrak{X}$).*

5. *$\mathfrak{P}_V(Q')$ of any Q' satisfying $Q \prec_{\mathsf{DPO}} Q'$ can be obtained from $\mathfrak{P}_V(Q)$ by transferring X with $\emptyset \subset X \subset V^+_Q \cup V^-_Q$ to V^0_Q and by possibly interchanging the positions of the resulting sets $V^+_Q \setminus X$ and $V^-_Q \setminus X$. That is, $\mathfrak{P}_V(Q') = \left\langle V^+_{Q'}, V^-_{Q'}, V^0_{Q'} \right\rangle$ is either equal to*

$$\langle V^+_Q \setminus X, V^-_Q \setminus X, V^0_Q \cup X \rangle \qquad \text{or to}$$
$$\langle V^-_Q \setminus X, V^+_Q \setminus X, V^0_Q \cup X \rangle$$

Prop. 3 substantiates the *plausibility of the DPO*. In particular:

- Prop. 3.5 shows that discrimination-dispreferred queries result from adding hypotheses to those (V^0_Q) that cannot be invalidated by any query answer.
- Prop. 3.3 implies that no weak query can be discrimination-preferred to a strong one. Nor can a non-DQ be discrimination-preferred to a DQ.

Example (cont'd): Alternatively to directly using Def. 1 as before, Prop. 3.5 enables to prove $Q_3 \prec_{\mathsf{DPO}} Q_4$ by constructing Q_4 from Q_3 using $X := \{h_3\}$. On the other hand, e.g., the DPO does not relate Q_2 with Q_3 or vice versa. This can be easily verified by Prop. 3.5, i.e. no suitable X exists.

Let m_1, m_2 be QSMs and their preference orders imposed on V be (*the transitive closure of*) $\{Q_1 \prec_{m_1} Q_3, Q_3 \prec_{m_1} Q_2, Q_2 \prec_{m_1} Q_4\}$ and $\{Q_1 \prec_{m_2} Q_3, Q_2 \prec_{m_2} Q_3, Q_1 \prec_{m_2} Q_4, Q_2 \prec_{m_2} Q_4\}$. Clearly, m_1 satisfies the DPO since its imposed order is a superset of the DPO $\{(Q_1, Q_2), (Q_3, Q_4)\}$ over V (cf. Def. 4). On the contrary, m_2 is consistent with the DPO since neither $Q_2 \prec_{m_2} Q_1$ nor $Q_4 \prec_{m_2} Q_3$ holds, but does not satisfy the DPO since, e.g., $Q_1 \prec_{m_2} Q_2$ does not hold. So, by Prop. 3.2 we can conclude that m_1 is $\mathfrak{X}$-superior to m_2, i.e. $m_1 \prec_{\mathfrak{X}} m_2$ where $\mathfrak{X} := \{Q_1, \dots, Q_4\}$. Let $Q_4 \prec_{m_3} Q_3$ for some QSM m_3, then m_3 neither satisfies the DPO nor is consistent with the DPO.

By Prop. 3.3, no Q_j can be discrimination-preferred to Q_1 or Q_3 since $V^0_{Q_i} = \emptyset$ for $i \in \{1, 3\}$. □

3.2 The Discussed QSMs

In the following we briefly sketch the AL QSMs we analyze regarding their use in SD (see Tab. 2), grouped by their *Query Selection Framework (QS-FW)* [37]:

Uncertainty Sampling (US) Here, the principle is to select the query about whose answer the learner is most uncertain (as per the probability measure p) given the current evidence V. *Least Confidence* (LC) selects the query whose most likely answer $a_{Q,\max}$ has least probability. *Margin Sampling* (M) targets the query for which the probabilities between most and second most likely label $a_{Q,1}$ and $a_{Q,2}$ are most similar. *Entropy* (H) prefers the query whose outcome is most uncertain wrt. information entropy. *Gini Impurity* (GI) is borrowed from decision tree learning theory [3].

Information Gain (IG) The query favored by ENT maximizes the information gain [25, 30], or equivalently, minimizes the expected a-posteriori entropy wrt. hypotheses in V. As proven in [9], ENT can be equivalently written as shown in Tab. 2. ENT is probably the most popular QSM applied in SD approaches [9, 4, 29, 16, 39].

QS-FW	QSM m	$m(Q)$	opt.	DPO	∃ theor. opt.
US	LC	$p(ans(Q) = a_{Q,\max})$	↘	×	✓
	M	$p(ans(Q) = a_{Q,1}) - p(ans(Q) = a_{Q,2})$	↘	×	✓
	H	$-\sum_{a\in\{0,1\}} p(ans(Q) = a) \log_2 p(ans(Q) = a)$	↗	×	✓
	GI	$1 - p(ans(Q) = 1)^2 - p(ans(Q) = 0)^2$	↗	×	✓
IG	ENT	$p(V_Q^0) + \sum_{a\in\{0,1\}} p(ans(Q) = a) \log_2 p(ans(Q) = a)$	↘	×	✓
	ENT$_z$	$z\, p(V_Q^0) + \sum_{a\in\{0,1\}} p(ans(Q) = a) \log_2 p(ans(Q) = a)$	↘	×/✓ **4)**	✓
QBC	SPL	$\Big\vert \vert V_Q^+\vert - \vert V_Q^-\vert \Big\vert + \vert V_Q^0\vert$	↘	(✓)	✓
	SPL$_z$	$\Big\vert \vert V_Q^+\vert - \vert V_Q^-\vert \Big\vert + z\,\vert V_Q^0\vert$	↘	$\times_{(z<1)}$/$(\checkmark)_{(z=1)}$/$\checkmark_{(z>1)}$	✓
	VE	$-\sum_{X\in\{V_Q^+,V_Q^-\}} \frac{\vert X\vert}{\vert V_Q^+\cup V_Q^-\vert} \log_2 \frac{\vert X\vert}{\vert V_Q^+\cup V_Q^-\vert}$	↗	×	✓
	KL	$-\sum_{X\in\{V_Q^+,V_Q^-\}} \frac{\vert X\vert}{\vert V_Q^+\cup V_Q^-\vert} \log_2 \frac{p(X)}{p(V_Q^+\cup V_Q^-)}$	↗	×	×
EMC	EMCa	$2\left[p(ans(Q) = 1) - [p(ans(Q) = 1)]^2\right] - \frac{p(V_Q^0)}{2}$	↗	×	✓
	EMCa$_z$	$2\left[p(ans(Q) = 1) - [p(ans(Q) = 1)]^2\right] - z\,\frac{p(V_Q^0)}{2}$	↗	$\times_{(z<2)}$/$\checkmark_{(z\geq 2)}$	✓
	EMCb	$p(ans(Q) = 1)\vert V_Q^-\vert + p(ans(Q) = 0)\vert V_Q^+\vert$	↗	×	×
	MPS	0 if Q not a strong DQ or $\Big\vert \vert V_Q^+\vert - \vert V_Q^-\vert \Big\vert \neq 2$, $V_{Q,\min}$ else **1)**	↗	(✓)	✓
	MPS′	$-\vert V_Q^0\vert$ if Q not a strong DQ or $\Big\vert \vert V_Q^+\vert - \vert V_Q^-\vert \Big\vert \neq 2$, $V_{Q,\min}$ else **1)**	↗	✓	✓
	BME	$\vert V_{Q,p,\min}\vert$ **2)**	↗	×	✓
RL	RIO′	$\frac{\mathsf{ENT}(Q)}{2} + V_{Q,n}$ **3)**	↘	×	✓
	RIO′$_z$	$\frac{\mathsf{ENT}_z(Q)}{2} + V_{Q,n}$ **3)**	↘	×	✓

Key: **1):** $V_{Q,\min} := \arg\min_{X\in\{V_Q^+,V_Q^-\}}(\vert X\vert)$. **2):** $V_{Q,p,\min}$ is equal to V_Q^- if $p(V_Q^-) < p(V_Q^+)$, to V_Q^+ if $p(V_Q^+) < p(V_Q^-)$, and to 0 else. **3):** $V_{Q,n}$ is equal to $\min\{\vert V_Q^+\vert, \vert V_Q^-\vert\} - n$ if $\min\{\vert V_Q^+\vert, \vert V_Q^-\vert\} \geq n$, and equal to $\vert V\vert$ else. n denotes the minimal number of hypotheses the next query must eliminate (in the worst case) [35]. **4):** In general, ✓ holds only if z is specified as per Prop. 5.

Table 2: QSMs m (col. 2) grouped by query selection frameworks (QS-FWs) (col. 1). Functions $m(Q)$ (col. 3) are optimized for arguments Q that maximize (↗) or minimize (↘) $m(Q)$ (col. 4). ✓ means m satisfies the DPO, (✓) that m is consistent with, but does not satisfy the DPO, and × that m is not consistent with the DPO (col. 5). Col. 6 reports whether (✓) or not (×) a theoretical optimum exists for the QSM. Numbers **i)** are explained in the key below the table. Statements such as $_{(z>2)}$ state conditions under which a property holds.

Query by Committee (QBC) QBC criteria use the competing hypotheses in V as a committee C. Each *predicting* committee member $h \in V$ has a vote on the classification of a $Q \in \mathcal{U}$, i.e. the committee (for Q) is $C = V \setminus V_Q^0 = V_Q^+ \cup V_Q^-$. The query Q yielding the highest disagreement among all committee members is considered most informative. There are different ways of estimating the disagreement: *Vote Entropy* (VE) selects the query for which the entropy of the relative prediction frequencies is maximal. At this, $|X|/|V_Q^+ \cup V_Q^-|$ with $X = V_Q^+$ ($X = V_Q^-$) is the relative prediction frequency of label 1 (0). The *Kullback-Leibler-Divergence* (KL) proposes the query that manifests the largest average disagreement between the label distributions of any $h \in C$ and the consensus of the entire C (cf. [37, p. 17] for a formal specification). By simple mathematics, one can derive that the KL measure has the shape as given in Tab. 2 [33, Prop. 26]. *Split-In-Half* (SPL) [25, 24, 39] tries to eliminate exactly half of the currently known hypotheses, i.e. suggests queries which split V into V_Q^+ and V_Q^-, both of size $|V|/2$ (implying $|V_Q^0| = 0$).

Expected Model Change (EMC) The principle is to favor the query that would impart the greatest change to the current model if its label was known. Interpreted in the sense of version spaces [22], we view all the available evidence V as "model". "Maximum expected model change" can be interpreted in a way that the expected (a) *probability mass* or (b) *number* of invalidated hypotheses in V is maximized. The resulting QSMs, which we call EMCa for (a) and EMCb for (b), are depicted in Tab. 2. Further, we propose the new QSM *Most Probable Singleton* (MPS). It favors DQs with empty V_Q^0 where one of V_Q^+, V_Q^- is a singleton and this singleton has maximum probability. Since in this case the probability of this singleton is equal to the probability of one answer of Q (cf. Sec. 2), it attempts to maximize the probability of deleting the maximum possible number of hypotheses in V. The variant MPS$'$ of MPS additionally penalizes queries Q with $V_Q^0 \neq \emptyset$. Another new QSM we introduce is *Biased Maximal Elimination* (BME). The idea is to achieve a bias (probability > 0.5) towards an answer that rules out a maximal possible number of hypotheses in V.

Reinforcement Learning (RL) A "risk-optimization" reinforcement learning QSM (RIO) was introduced in [35] to overcome performance issues of SPL and ENT in terms of querying cost given unreasonable a-priori probabilities. Based on the hypothesis elimination rate achieved by the already asked queries, RIO adapts a learning parameter which dictates the minimum number of hypotheses n the next chosen query must eliminate (in the worst case). Tab. 2 gives a slightly modified version RIO$'$ of RIO which can be expressed in closed form (cf. [33, Rem. 8]). Among those queries that approach n best (i.e. minimize $V_{Q,n}$, see Tab. 2), the best query wrt. the ENT QSM is selected.

3.3 Compliance of QSMs with the DPO

We next discuss how far the QSMs in Tab. 2 agree with the DPO in terms of Def. 4 over any set of DQs $\mathcal{U}$.[3]

Proposition 4. *The QSMs* LC, M, H, ENT, VE, KL, EMCa, EMCb, BME *and* RIO$'$ *are not consistent with the DPO. Further,* SPL *and* MPS *are consistent with, but do not satisfy the DPO.*

Proof. (Sketch) We give counterexamples based on Tab. 1. First, let the hypotheses probabilities $p := p_1$. Then $p(ans(Q_1) = 1) = 0.4$ and $p(ans(Q_2) = 1) = 0.5$. Hence, $Q_2 \prec_m Q_1$ for $m \in \{\mathsf{LC}, \mathsf{M}, \mathsf{H}\}$. Due to the asymmetry of $\prec_m$ for each QSM m (Prop. 2), we have $\neg(Q_1 \prec_m Q_2)$. But, $Q_1 \prec_{\mathsf{DPO}} Q_2$ (see Example above). Inconsistency of m with the DPO now follows from Def. 4.

[3] It suffices to analyze properties as per Def. 4 of QSMs regarding the DPO just for DQs, cf. footnote *c*.

Similarly, we obtain $Q_2 \prec_m Q_1$ for $m \in \{\mathsf{VE}, \mathsf{KL}\}$ because $\mathsf{VE}(Q_1) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} < -2\frac{1}{2}\log_2\frac{1}{2} = \mathsf{VE}(Q_2)$ and $\mathsf{KL}(Q_1) = -\frac{2}{5}\log_2(0.4) - \frac{3}{5}\log_2(0.6) < -2\frac{1}{2}\log_2\frac{1}{2} = \mathsf{KL}(Q_2)$. Further, assuming $p := p_3$, we analogously find that $Q_4 \prec_m Q_3$ for $m \in \left\{\mathsf{ENT}, \mathsf{EMCa}, \mathsf{RIO}'\right\}$ (letting $n := 1$ for RIO'), and, supposing $p := p_2$, we realize that $Q_4 \prec_m Q_3$ for $m \in \{\mathsf{EMCb}, \mathsf{BME}\}$.

For all $Q, Q' \in \mathcal{U}$ where $Q \prec_{\mathsf{DPO}} Q'$ and $m \in \{\mathsf{SPL}, \mathsf{MPS}\}$ it can only hold that $Q \prec_m Q'$ or $m(Q) = m(Q')$ (follows from Prop. 3.5 and the QSM definitions, see Tab. 2). Thence, $\neg(Q' \prec_m Q)$. So, m is consistent with the DPO by Def. 4. □

For the QSMs ENT, SPL, EMCa and MPS we can derive (parameterized) improved versions ENT_z, SPL_z, EMCa_z and MPS' that satisfy the DPO (see col. 2 and 3 of Tab. 2). The idea with all these QSMs is to penalize the inclusion of hypotheses in V_Q^0. Because, the more elements there are in V_Q^0, the less the query Q tends to be favored by the DPO. However, it is material to obey that this penalization must be as subtle as possible in order to *preserve the query selection characteristics* of the respective QSM. Because, in general, $m_z \not\equiv m_r$ for some QSM m parameterized by z and r ($z \neq r$), respectively, and the difference between QSMs m_z and m_r regarding their query selection behavior grows with $|z - r|$. For instance, consider ENT and two queries Q, Q' with $\langle p(V_Q^+), p(V_Q^-), p(V_Q^0)\rangle = \langle 0.01, 0.99, 0\rangle$ and $\langle p(V_{Q'}^+), p(V_{Q'}^-), p(V_{Q'}^0)\rangle = \langle 0.49, 0.49, 0.02\rangle$. Obviously, since ENT favors queries with 50-50 answer probability and low $p(V_Q^0)$, it should give Q' preference to Q although $V_{Q'}^0 \neq \emptyset$ and $V_Q^0 = \emptyset$. Using ENT_z with an *unjustified* too large parameter z, say $z := 50$, would however imply $\mathsf{ENT}_z(Q) \approx 0.92 < 0.99 \approx \mathsf{ENT}_z(Q')$, i.e. the favoritism of Q, which contradicts the nature of entropy query selection. Note, Q and Q' are not DPO-related (cf. Prop. 3.5). Thence no (change of the) parametrization of ENT whatsoever is justified in the presence of only Q, Q'.

We now state the relationship between z-parameter and DPO adherence of the new QSMs. These results show how to set z to an effective (wrt. *DPO-compliance*), but not higher than justified (wrt. *QSM nature preservation*) value:

Proposition 5. *For the parameterized QSMs* ENT_z, EMCa_z *and* SPL_z, *the following holds:*

- *Ad* ENT_z *[33, Cor. 3+4]:*
  Let for all $Q \in \mathcal{U}$ *be* $\min_{a\in\{0,1\}} p(ans(Q) = a) > t > 0$. *Then, for any* $z \geq \max\left\{-\frac{1}{2}(\log_2 t - \log_2(1-t)), 1\right\}$, ENT_z *satisfies the DPO over* $\mathcal{U}$. *Further,* $\mathsf{ENT}_s \prec \mathsf{ENT}_r$ *for* $0 \leq r < s$.

- *Ad* EMCa_z *[33, Cor. 13]:*
  For all $z \geq 2$ *and* $r \geq 0$, EMCa_z *satisfies the DPO and is superior to* ENT_r.

- *Ad* SPL_z *[33, Prop. 19]:*
  SPL_z *is (inconsistent with / consistent with, but not satisfying / satisfying) the DPO for all (*$z < 1$ */* $z = 1$ */* $z > 1$*).*

So, whereas for EMCa_z and SPL_z a fixed z-value guarantees DPO-satisfaction for any $\mathcal{U}$, for ENT_z the z-parameter depends on t. It is straightforward from the definition of $p(ans(Q) = a)$ (cf. Sec. 2) that $t < \min_{h\in V} p(h)$ for any $\mathcal{U}$. So, it is easy to compute t and thence a suitable parameter z as per Prop. 5 for any given query pool $\mathcal{U}$ ad-hoc in order to ensure that ENT_z is DPO-preserving over $\mathcal{U}$. Finally, for MPS' it is clear from its definition that it satisfies the DPO.

3.4 Equivalences Between QSMs

Tab. 3 summarizes equivalence classes (ECs) as per Def. 3 between QSMs over arbitrary queries (row $\equiv$) and over strong queries $\mathfrak{X}$ (row $\equiv_{\mathfrak{X}}$). ECs wrt. $\equiv$ cluster QSMs that manifest the exact same query

	Equivalence Classes (ECs) of QSMs
$\equiv$	$\{\mathsf{ENT}_1, \mathsf{ENT}\}$, $\{\mathsf{ENT}_{z\,(z\notin\{0,1\})}\}$, $\{\mathsf{SPL}_1, \mathsf{SPL}\}$, $\{\mathsf{EMCb}\}$, $\{\mathsf{SPL}_{z\,(z\notin\{0,1\})}\}$, $\{\mathsf{RIO}'_1, \mathsf{RIO}'\}$, $\{\mathsf{RIO}'_{z\,(z\neq 1)}\}$, $\{\mathsf{KL}\}$, $\{\mathsf{EMCa}_1, \mathsf{EMCa}\}$, $\{\mathsf{EMCa}_{z\,(z\notin\{0,1\})}\}$, $\{\mathsf{VE}, \mathsf{SPL}_0\}$, $\{\mathsf{EMCa}_0, \mathsf{GI}, \mathsf{LC}, \mathsf{M}, \mathsf{H}, \mathsf{ENT}_0\}$, $\{\mathsf{MPS}\}$, $\{\mathsf{MPS}'\}$, $\{\mathsf{BME}\}$
$\equiv_{\mathfrak{X}}$	① : $\{\mathsf{EMCa}, \mathsf{EMCa}_{z\,(z\in\mathbb{R})}, \mathsf{GI}, \mathsf{LC}, \mathsf{M}, \mathsf{H}, \mathsf{ENT}, \mathsf{ENT}_{z\,(z\in\mathbb{R})}\}$, ② : $\{\mathsf{SPL}, \mathsf{SPL}_{z\,(z\in\mathbb{R})}, \mathsf{VE}\}$, ③ : $\{\mathsf{RIO}', \mathsf{RIO}'_{z\,(z\in\mathbb{R})}\}$, ④ : $\{\mathsf{KL}\}$, ⑤ : $\{\mathsf{EMCb}\}$, ⑥ : $\{\mathsf{MPS}, \mathsf{MPS}'\}$, ⑦ : $\{\mathsf{BME}\}$

Table 3: Equivalence Classes (ECs) of QSMs wrt. the relations $\equiv$ and $\equiv_{\mathfrak{X}}$ (cf. Def. 3). $\mathfrak{X}$ is any set of strong queries. Circled numbers ⓘ provide reference to Tab. 4, which gives only one set of requirements for each numbered EC.

selection behavior in SD. Given a setting where all hypotheses predict an answer for any query (as e.g. in spectrum-based SD without false positive or negative test outcomes [27, 46]), it holds that all QSMs in an EC wrt. $\equiv_{\mathfrak{X}}$ behave equally. The pragmatics of the given ECs is the reduction of the possible QSM options for a certain SD task, i.e. it makes no sense to try to improve the querying cost by switching between QSMs of the same EC. Along with QSM superiority results below, the ECs provide a general guidance for proper QSM choice based on the type of application.

The proofs of the stated QSM equivalences are either direct consequences of the QSMs' definitions (Tab. 2, col. 3) or straightforward after simple algebraic transformations. For instance, $\mathsf{EMCa}_0 \equiv \mathsf{GI}$ since the latter can be equivalently transformed to the former by using $p(ans(Q) = 0) = 1 - p(ans(Q) = 1)$. Further $\mathsf{LC} \equiv \mathsf{M} \equiv \mathsf{H} \equiv \mathsf{ENT}_0$ since there are only two possible query labels. Interestingly, the EC wrt. $\equiv$ comprising GI includes QSMs of three different query selection frameworks (QS-FWs), namely US, IG and EMC (cf. Tab. 2). Note that the ECs including z-parameterized QSMs represent infinitely many *different* ECs, one for each setting of z, e.g. $\mathsf{ENT}_r \not\equiv \mathsf{ENT}_s$ for $r \neq s$ (cf. Prop. 5). Note that some of the ECs wrt. $\equiv$ conflate to constitute a single EC wrt. $\equiv_{\mathfrak{X}}$. In particular, those ECs merge which are equivalent except for their treatment of V_Q^0. Hence, infinitely many ECs wrt. $\equiv$ reduce to mere 7 ECs wrt. $\equiv_{\mathfrak{X}}$.

3.5 Superiority Between QSMs

Fig. 1 shows the QSM superiority relationships we derived. Basically, these can be proven using Def. 5, Prop. 3, the QSM functions $m(Q)$ (cf. Tab. 2) and QSM equivalences (cf. Tab. 3). For example, ENT_z for $z > 0$ is superior to H since $\mathsf{ENT}_0 \equiv \mathsf{H}$ and $\mathsf{ENT}_z \prec \mathsf{ENT}_r$ for $z > r \geq 0$ by Prop. 5. Note, by Prop. 3.2, QSMs that satisfy the DPO (framed in Fig. 1) are proven superior to all that do not. Further, there are no $\mathfrak{X}$-superiority relationships between QSMs in row $\equiv_{\mathfrak{X}}$ of Tab. 3 due to Prop. 3.3, i.e. the superiority graph (Fig. 1) collapses over strong queries $\mathfrak{X}$.

From the pragmatic viewpoint the superiority results are primarily relevant in a *pool-based* SD scenario where a QSM is used to evaluate each query in a pool of queries and the best DQ is selected to be shown to the oracle. Opting for a *DPO-satisfying QSM then guarantees that no query is ever selected for which there is a better, i.e. discrimination-preferred one in the pool.* However, Fig. 1 must be read with care. For instance, it is not granted just due to $\mathsf{SPL}_y \prec \mathsf{KL}$ that KL will always manifest a worse performance (in terms of querying cost) than SPL_y for $y > 1$ in practice. The reason is that both QSMs

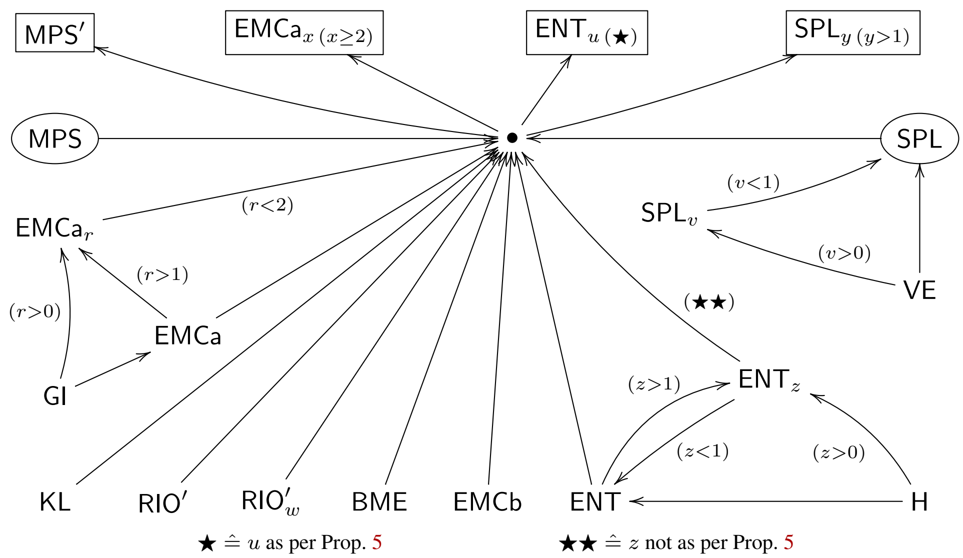


Figure 1: QSM Superiority Relationships: $m_1 \to m_2$ denotes that $m_2 \prec m_1$ (cf. Def. 5). Labeled arrows are conditional relations (hold only if the label is true). Framed (circled) nodes indicate QSMs that satisfy (are consistent with) the DPO. Other nodes without frame or circle are (in general) not consistent with the DPO. For clarity, (1) whenever possible, only one node for each EC in Tab. 3, row "≡" is depicted, and (2) node • means that each incoming and outgoing arrow is to be combined.

follow quite different paradigms of query selection (cf. Tab. 2, col. 3). Rather of interest are superiorities between *related* QSMs, e.g. those from a particular QS-FW (cf. Tab 2). For example, SPL_y for $y > 1$ is superior to SPL and VE *and implements the same preference paradigm* (cf. ECs in Tab. 3), attempting to eliminate half of the hypotheses in V. As a rule of thumb, we suggest to abide by this strategy:

Guide for choosing the appropriate QSM for Sequential Diagnosis:
Input: SD problem
Output: Best QSM to use

1. Decide upon which *query selection paradigm* to employ (e.g. entropy-based if one trusts in the a-priori probabilities $p(h)$ versus greedy or risk-optimized otherwise, cf. discussions and evaluations in [39, 35]).
2. Opt for the *particular QSM* adhering to this paradigm (as per ECs in Tab. 3 and QS-FWs in Tab. 2) which is *superior to all other related QSMs* (as per Fig. 1).

For instance, assuming a case where no (reasonable) prior probabilities are available and one favors a greedy hypotheses elimination strategy, one should (based on the parameter discussion before) prefer SPL_{y^*} (with preferably small $y^* > 1$, e.g. $y^* := 1.1$) to the two QSMs SPL and VE.

3.6 Properties of Optimal Queries

We have investigated all the QSM functions $m(Q)$ in Tab. 2 wrt. their theoretical optima. Most of the QSM analyses were relatively simple, e.g., for SPL one can easily see that no input can be better than one, say X, which satisfies $|V_X^+| = |V_X^-|$ and $|V_X^0| = 0$. Moreover, e.g., for $m \in \{\mathsf{H}, \mathsf{GI}\}$ the existence of a theoretical optimum follows from the functions' concavity. We report that for all discussed QSMs,

EC	Requirements to Optimal Query	
①	$\lvert p(V_Q^+) - p(V_Q^-)\rvert \to \min$	
②	$\lvert\, \lvert V_Q^+\rvert - \lvert V_Q^-\rvert \,\rvert \to \min$	
③	(I) $V_{Q,n} \to \min$	(II) $\lvert p(V_Q^+) - p(V_Q^-)\rvert \to \min$
④,⑤	$[\,p(V_Q^+) \to \max$ for some $\lvert V_Q^+\rvert \in \{1,\dots,\lvert V\rvert-1\}\,] \lor$ $[\,p(V_Q^-) \to \max$ for some $\lvert V_Q^-\rvert \in \{1,\dots,\lvert V\rvert-1\}\,]$	
⑥	(I) $\lvert V^*\rvert = 1, V^* \in \{V_Q^+, V_Q^-\}$	(II) $p(V^*) \to \max$
⑦	(I) $p(V^*) < 0.5, V^* \in \{V_Q^+, V_Q^-\}$	(II) $\lvert V^*\rvert \to \max$

Table 4: Query optimality requirements for ECs ⓘ of QSMs in Tab. 3: Roman numbers signalize priority, i.e. higher numbered conditions are optimized over all queries that optimize lower numbered conditions. An explanation of $V_{Q,n}$ can be found in the key of Tab. 2.

except for KL and EMCb, a (unique) theoretical optimum exists (Tab. 2, last col.). In fact, analysis of the KL and EMCb functions yields only one stationary point which is a saddle point [33, Prop. 27, 31].

As a byproduct of studying the QSMs m, we derived sufficient and necessary criteria an optimal query wrt. m *and* V must meet. Tab. 4 summarizes the results. Note, for KL and EMCb only necessary criteria can be named (see indeterminate conditions in row ④,⑤). Nevertheless, these help to reduce the search space, i.e. optimal queries must be among those satisfying the conditions. For instance, if Q_i, Q_j satisfy $\lvert V_{Q_i}^+\rvert = \lvert V_{Q_j}^+\rvert$ and $p(V_{Q_i}^+) > p(V_{Q_j}^+)$, then Q_j cannot be optimal wrt. KL or EMCb.

3.7 Query Synthesis in Sequential Diagnosis

The criteria in Tab. 4 suggest a *systematic* construction of an optimal query wrt. a QSM and V in a *query synthesis* SD scenario.[4] As discussed in Sec. 1, we propose to prefer query synthesis to pool-based query selection particularly in SD applications where the generation of DQs or the computation of queries' QSM-values is costly, e.g. in model-based diagnosis tasks [39, 32]. Using query synthesis, one will usually, assuming the existence of a large enough set of unlabeled queries $\mathcal{U}$ wrt. V, attempt to synthesize only strong DQs. For this reason Tab. 4 just lists conditions for the QSMs corresponding to the ECs in the $\equiv_{\mathfrak{X}}$-row of Tab. 3. Indeed, the optimality criteria in Tab. 4 target only properties *of the partition* of a query, as we already anticipated at the beginning of Sec. 3. Therefore, given a QSM m to be optimized, the idea is to first focus on the finding of a (nearly) optimal partition wrt. m and then try to generate a query for this partition. To guide the search for the best partition towards promising solutions first, heuristics g_m derived from m's optimality criteria can be leveraged. Our suggested strategy for query optimization is as follows:

Query Synthesis Procedure:
Input: QSM m, set of hypotheses V, optimality threshold t_m, (optionally) heuristic function g_m
Output: Strong discriminating query (DQ) wrt. V (cf. Sec. 2) that optimizes m (up to t_m)

1. Perform a best-first search (using g_m) over strong DPs wrt. V (cf. Sec. 2) until an optimal strong DP (as per t_m) is found.
2. Generate a DQ for exactly this optimal DP.

[4] For an in-depth treatment of the given query synthesis methods for SD, see the paper's extended version [33, Sec. 3.4ff.].

Notably, the first step does not involve any expensive operations, in particular no reasoning. The second step, on the other hand, is expensive as logical reasoning is required. Hence, the aim of the query synthesis procedure is to restrain as long as possible and thus minimize expensive operations during query computation. In fact, this strategy ideally involves only the actual computation of a *single query*. We next illustrate the two steps of the procedure in more detail.

Ad step 1 (Finding an optimal partition): We illustrate how the DP search might be realized by means of a complete depth-first backtracking search making local best-first moves. A search problem [36] is characterized by

(i) an *initial state*,
(ii) a *successor function* enumerating all direct neighbor states of a state,
(iii) some *heuristics* to estimate the remaining effort towards a goal state, and
(iv) a *goal test* to determine if a given state is a goal state or not.

Let in our case (i) be the partition $\mathfrak{P}_0 = \langle V^+, V^-, V^0\rangle = \langle\emptyset, V, \emptyset\rangle$ and (ii) map a partition to all neighbors resulting from the transfer of some $h \in V^-$ to V^+. The selection of (iii) and (iv) depends on the concrete used QSM. Fig. 2 (right) shows heuristic functions g_m we derived for all QSM ECs in Tab. 3 ($\equiv_{\mathfrak{X}}$) using the optimality criteria in Tab. 4. The plausibility of g_m for ECs ①, ②, ⑥ and ⑦ is straightforward from Tab. 4. g_m for EC ④,⑤ prefers a query Q with lowest ratio between the expected probability $|V_Q^+|/|V|$ of $|V_Q^+|$ diagnoses in $|V|$ and the actual probability $p(V_Q^+)$ of V_Q^+. The heuristic g_m for EC ③ returns the deviance of $p(V_Q^+)$ from 0.5 assuming that $n - |V_Q^+|$ further $h_i \in V_Q^-$ are transferred to V_Q^+, each with the expected probability $p(V_Q^-)/|V_Q^-|$.

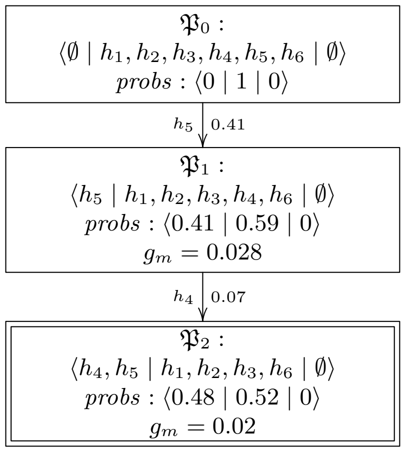
Depth-first, local best-first strategy: At each state (partition $\mathfrak{P}$) in the search tree the heuristic function g_m is used to evaluate all direct successor states of $\mathfrak{P}$ and suggests the best state $\mathfrak{P}'$ (with minimal heuristic value) to visit next.

Backtracking strategy: Given that all successors of a state $\mathfrak{P}$ have already been explored and no goal state has been found yet, the search backtracks and visits the next-best unexplored sibling of $\mathfrak{P}$.

Note, the functions g_m in Fig. 2 are just example heuristics and depend on the selection of the other search parameters (i), (ii) and (iv). For instance, [39] suggest a similar search – using different specifications of initial state and successor function, and only for QSMs ENT and SPL – with a heuristic based on the CKK algorithm for number partitioning [20]. Our definition of (i) and (ii) is not amenable to their heuristic (which assumes a binary tree with a maximum of two successors at each state). However, as we show in this paper's extended version [33], the search as we specify it here – with a slightly more sophisticated successor function – is *sound and complete* (i.e. considers only and all strong DPs) and enables the efficient determination of optimal strong DPs for all QSMs listed in Tab. 2 *without using a reasoner*.

Example: Let us demonstrate the search using the QSM $m := \mathsf{RIO}'$ with $n := 2$ (cf. the key of Tab. 2) over $V = \{h_1, \dots, h_6\}$ with $\langle p(h_1), \dots, p(h_6)\rangle = \langle 0.01, 0.33, 0.14, 0.07, 0.41, 0.04\rangle$. Let the goal test be $true$ iff $V_{Q,n} = 0 \land |p(V_Q^+) - p(V_Q^-)| \leq t_m$ (cf. ③ in Tab. 4) for the optimality threshold $t_m := 0.05$. Further, let the heuristic g_m be as per ③ in Fig. 2 (top right). Fig. 2 (top left) shows the resulting search tree, displaying only *best* successors for each node. We can see that the tree includes only three (explored) partitions $\mathfrak{P}_0, \mathfrak{P}_1, \mathfrak{P}_2$ where all but $\mathfrak{P}_0$ are strong DPs. Note, the heuristic g_m guides the search directly to a goal $\mathfrak{P}_2$, without any necessary backtrackings. □

Ad step 2 (Generating a query for the found partition): Let $\mathfrak{P}' = \langle V^+, V^-, \emptyset\rangle$ be the strong DP resulting from step 1. Then, according to Sec. 2, for each query $Q \in \mathcal{U}$ with partition $\mathfrak{P}'$ (i.e. where $\mathfrak{P}_V(Q) = \mathfrak{P}'$) the following holds:



Heuristic functions g_m for m in EC ⓘ	
① $\lvert p(V_Q^+) - \frac{1}{2} \rvert$	② $\left\lvert \lvert V_Q^+ \rvert - \frac{\lvert V \rvert}{2} \right\rvert$
③ $\lvert p(V_Q^+) + (n - \lvert V_Q^+ \rvert)\frac{p(V_Q^-)}{\lvert V_Q^- \rvert} - \frac{1}{2} \rvert$	
④, ⑤ $\frac{\lvert V_Q^+ \rvert}{\lvert V \rvert \, p(V_Q^+)}$	⑥ $-p(V_Q^+)$
⑦ $-\lvert V_Q^+ \rvert - p(V_Q^+)$	if $p(V_Q^+) < \frac{1}{2}$
$-\lvert V_Q^- \rvert - p(V_Q^-)$	if $p(V_Q^+) > \frac{1}{2}$
0	else

	averages				maxima			
QSM	SS(%)	DEV(%)	ST(sec)	QT(sec)	SS(%)	DEV(%)	ST(sec)	QT(sec)
ENT	3.3	0.0008	0.38	0.58	20.6	0.004	3.46	3.52
SPL	3.56	0	0.43	0.69	54.8	0	15.1	4.41

Figure 2: *(Top Right:)* Heuristics g_m for QSM ECs in Tab. 3 derived from Tab. 4. Lower g_m values indicate better queries. *(Top Left:)* Heuristic search for optimal $m := \mathsf{RIO}'$ partition. Arrows point to best successor partition as per the heuristic g_m (see ③) and are labeled by the hypothesis h_i and by the probability mass transferred from V_Q^- to V_Q^+. $probs$ refers to $\langle p(V_Q^+) \mid p(V_Q^-) \mid p(V_Q^0) \rangle$. *(Bottom:)* First evaluation results for QSMs ENT, SPL.

(a) Q is a strong DQ, and
(b) $(\forall h \in V^+ : h \models ans(Q) = 1) \wedge (\forall h \in V^- : h \models ans(Q) = 0)$.

So, by means of (b) and a suitable reasoner, a strong DQ Q can be computed. For example, in a model-based circuit diagnosis task, asking if a particular wire is high would be a strong DQ if all hypotheses in V^+ entail that it is high and all in V^- entail that it is low.

Preliminary evaluation. To test the proposed query synthesis strategy, we adopted the same evaluation setting on 8 real-world model-based diagnosis problems (MBD-Ps) as reported in [34]. In particular, we performed 5 query synthesis runs for each combination of MBD-P and $|V| \in \{10, 20, \ldots, 80\}$ where V is a hypotheses set wrt. MBD-P. In each of the 5 runs a different hypotheses set V was computed for MBD-P in a random way by means of INV-HS-TREE [40] and a random reordering of its input. Each hypothesis $h \in V$ was assigned a uniform random probability $p(h)$. The used QSMs were SPL and ENT. Note, the queries for the examined MBD-Ps (involving knowledge-based systems) cannot be extracted from the system model, but are expensive to compute by means of an inference engine.

In the these experiments we measured averages and maxima (both taken over all runs) of

- the $\%$ of the complete search space of strong DPs (SS) actually explored by the search in step 1,
- the $\%$ deviation (DEV) from the theoretically optimal QSM-value achieved by the DP resulting from step 1, and
- the search time (ST) required by step 1,

- the query computation time (QT) required by step 2.

Preliminary evaluation results are are presented in Fig. 2 (bottom). For instance, for ENT, on average over all runs for all 8 MBD-Ps, an optimality of $> 99.999\%$ was achieved in $0.38 + 0.58 < 1$ sec by exploring just 3.3% of all strong DPs. Similar results could be observed for SPL.

In order to get a feeling for the benefits of the usage of query synthesis, we also tried to execute a pool-based query selection in the described settings. This involves the generation of a pool of queries, including at most one query for each DP wrt. V, and the subsequent selection of the best query from the pool.

The first observation was that the pool-based strategy worked – i.e. terminated within a one hour timeout – only for sets V that included no more than 20, for three MBD-Ps actually no more than 10 hypotheses. By contrast, query synthesis could efficiently handle even all $|V| = 80$ cases. The rest of the discussion refers to only the cases where $|V| = 10$ and both methods succeeded for all MBD-Ps.

Second, the pool-based strategy consumed substantially more time than query synthesis. In fact, the former required minimally / on average / maximally 27 / 787 / 2528 times (!) the time the latter needed for query computation. By absolute numbers, the minimal / average / maximal pool-based strategy execution time amounted to 6 / 137 / 566 sec whereas query synthesis for these cases never required more than 0.6 sec.

Third, the pool-based approach required substantial reasoning (thousands of reasoner calls) due to the implicit nature of the queries, as discussed above, and the large number of queries generated. Query synthesis circumvents this by postponing reasoner calls, i.e. the actual query computation (step 2), until an optimal partition is already fixed (step 1).

Fourth, taking the *overall* execution time of query synthesis as a timeout for pool-based selection, the latter could only explore a minimum / average / maximum number of 0.4 / 7.0 / 36.5 partitions. Besides, the pool-based method cannot profit from heuristics. Therefore, a pool-based strategy will hardly be able to find an optimal query within the time bounds of query synthesis.

Overall, these findings indicate the high efficiency and query quality in terms of a given QSM achieved by the proposed heuristic best-first query synthesis strategy. The made observations confirm the hypothesis that query synthesis is the method of choice (at least) for model-based diagnosis problems with a query space of large size or implicit nature.

4 Conclusions

We analyzed various Active Learning strategies regarding their use for query selection in Sequential Diagnosis (SD). Based on a precise and plausible definition of a query's discrimination power, we derived superiority relationships between query selection measures (QSMs) wrt. their output quality and introduced new (improved) variants, e.g. for the popular information entropy QSM. Additionally, we gave equivalence relationships between QSMs and deduced optimality criteria for them. The results give guidance for using the right QSM in SD and let us design an efficient heuristic search procedure for a systematic optimal query synthesis. A preliminary evaluation of the latter using real-world model-based diagnosis problems proves (1) its power in terms of almost negligible computation time and negligible deviation from the QSM-optimum, (2) its ability to compute optimally discriminating queries for substantial sizes of considered fault hypotheses – owing to the exploitation of the derived heuristics, (3) its drastic superiority to pool-based query selection (at least) in model-based problems involving implicit or numerous queries.

Acknowledgments

This work was supported by the Carinthian Science Fund (KWF), contract KWF-3520/26767/38701.

References

[1] Rui Abreu, André Riboira, and Franz Wotawa. Constraint-based Debugging of Spreadsheets. In *CIbSE*, pages 1–14, 2012.

[2] Vamshi Ambati, Stephan Vogel, and Jaime G. Carbonell. Active learning and crowd-sourcing for machine translation. In *LREC'10*, pages 2169–2174, 2010.

[3] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. *Classification and regression trees*. CRC press, 1984.

[4] Mark Brodie, Irina Rish, Sheng Ma, and Natalia Odintsova. Active probing strategies for problem diagnosis in distributed systems. In *IJCAI'03*, pages 1337–1338, 2003.

[5] Christopher H. Bryant, Stephen Muggleton, Stephen G. Oliver, Douglas B. Kell, Philip G. K. Reiser, and Ross D. King. Combining inductive logic programming, active learning and robotics to discover the function of genes. *Electron. Trans. Artif. Intell.*, 5(B):1–36, 2001.

[6] David A. Cohn, Les E. Atlas, and Richard E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[7] Johan de Kleer. Focusing on probable diagnoses. In *AAAI'91*, pages 842–848, 1991.

[8] Johan de Kleer, Olivier Raiman, and Mark Shirley. One step lookahead is pretty good. In *Readings in model-based diagnosis*, pages 138–142, 1992.

[9] Johan de Kleer and Brian C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

[10] Johan de Kleer and Brian C. Williams. Diagnosis with behavioral modes. In *IJCAI'89*, pages 1324–1330, 1989.

[11] Oskar Dressler and Peter Struss. The consistency-based approach to automated diagnosis of devices. *Principles of Knowledge Representation*, pages 269–314, 1996.

[12] Alexander Feldman, Gregory M. Provan, and Arjan J. C. van Gemund. A model-based active testing approach to sequential diagnosis. *J. Artif. Intell. Res. (JAIR)*, 39:301–334, 2010.

[13] Alexander Felfernig, Gerhard Friedrich, Karl Isak, Kostyantyn Shchekotykhin, Erich Teppan, and Dietmar Jannach. Automated debugging of recommender user interface descriptions. *Applied Intelligence*, 31(1):1–14, 2009.

[14] Alexander Felfernig, Gerhard Friedrich, Dietmar Jannach, and Markus Stumptner. Consistency-based diagnosis of configuration knowledge bases. *Artificial Intelligence*, 152(2):213 – 234, 2004.

[15] Gerhard Friedrich, Markus Stumptner, and Franz Wotawa. Model-based diagnosis of hardware designs. *Artificial Intelligence*, 111(1-2):3–39, 1999.

[16] Alberto Gonzalez-Sanchez, Rui Abreu, Hans-Gerhard Gross, and Arjan J.C. van Gemund. Spectrum-based sequential diagnosis. In *AAAI*, 2011.

[17] David Heckerman, John S. Breese, and Koos Rommelse. Decision-theoretic troubleshooting. *Communications of the ACM*, 38(3):49–57, 1995.

[18] Laurent Hyafil and Ronald L. Rivest. Constructing optimal binary decision trees is NP-complete. *Information processing letters*, 5(1):15–17, 1976.

[19] Aditya Kalyanpur, Bijan Parsia, Matthew Horridge, and Evren Sirin. Finding all Justifications of OWL DL Entailments. In *ISWC'07*, pages 267–280, 2007.

[20] Richard E. Korf. A complete anytime algorithm for number partitioning. *Artificial Intelligence*, 106(2):181 – 203, 1998.

[21] Cristinel Mateis, Markus Stumptner, Dominik Wieland, and Franz Wotawa. Model-Based Debugging of Java Programs. In *AADEBUG'00*, 2000.

[22] Tom M. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *IJCAI'77*, pages 305–310, 1977.

[23] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, 1982.

[24] Tom M. Mitchell. *Machine learning*. McGraw-Hill, 1997.

[25] Bernard M.E. Moret. Decision trees and diagrams. *ACM Computing Surveys (CSUR)*, 14(4):593–623, 1982.

[26] Fredrik Olsson. *A literature survey of active machine learning in the context of natural language processing*. Swedish Institute of Computer Science, 2009.

[27] Krishna R. Pattipati and Mark G. Alexandridis. Application of heuristic search and information theory to sequential fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(4):872–887, 1990.

[28] Yannick Pencolé and Marie-Odile Cordier. A formal framework for the decentralised diagnosis of large scale discrete event systems and its application to telecommunication networks. *Artificial Intelligence*, 164(1):121–170, 2005.

[29] Jurryt Pietersma, Arjan J.C. van Gemund, and Andre Bos. A model-based approach to sequential fault diagnosis. In *IEEE Autotestcon, 2005*, pages 621–627, 2005.

[30] John R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1):81–106, 1986.

[31] Raymond Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32(1):57–95, 1987.

[32] Patrick Rodler. *Interactive Debugging of Knowledge Bases*. PhD thesis, Alpen-Adria Universität Klagenfurt, 2015. http://arxiv.org/pdf/1605.05950v1.pdf.

[33] Patrick Rodler. Towards better response times and higher-quality queries in interactive knowledge base debugging. Technical report, Alpen-Adria Universität Klagenfurt, 2016. http://arxiv.org/pdf/1609.02584v2.pdf.

[34] Patrick Rodler, Wolfgang Schmid, and Konstantin Schekotihin. Inexpensive cost-optimized measurement proposal for sequential model-based diagnosis. *Arxiv preprint arXiv:1705.09879*, 2017. http://arxiv.org/pdf/1705.09879.pdf.

[35] Patrick Rodler, Kostyantyn Shchekotykhin, Philipp Fleiss, and Gerhard Friedrich. RIO: Minimizing User Interaction in Ontology Debugging. In *Web Reasoning and Rule Systems*, pages 153–167. 2013.

[36] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 3rd edition, 2010.

[37] Burr Settles. Active Learning Literature Survey. Computer sciences technical report, University of Wisconsin-Madison, 2010.

[38] Mojdeh Shakeri, Vijay Raghavan, Krishna R. Pattipati, and Ann Patterson-Hine. Sequential testing algorithms for multiple fault diagnosis. *IEEE Trans. Systems, Man, and Cybernetics, Part A*, 30(1):1–14, 2000.

[39] Kostyantyn Shchekotykhin, Gerhard Friedrich, Philipp Fleiss, and Patrick Rodler. Interactive Ontology Debugging: Two Query Strategies for Efficient Fault Localization. *Web Semantics: Science, Services and Agents on the World Wide Web*, 12-13:88–103, 2012.

[40] Kostyantyn Shchekotykhin, Gerhard Friedrich, Patrick Rodler, and Philipp Fleiss. Sequential diagnosis of high cardinality faults in knowledge-bases by direct diagnosis generation. In *ECAI'14*, pages 813–818, 2014.

[41] Markus Stumptner and Franz Wotawa. Debugging functional programs. In *IJCAI'99*, pages 1074–1079, 1999.

[42] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM international conference on Multimedia*, pages 107–118, 2001.

[43] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, pages 45–66, 2001.

[44] Jules White, David Benavides, Douglas C. Schmidt, Pablo Trinidad, Brian Dougherty, and Antonio Ruiz Cortés. Automated diagnosis of feature model configurations. *Journal of Systems and Software*, 83(7):1094–1107, 2010.

[45] Tom Zamir, Roni Tzvi Stern, and Meir Kalech. Using model-based diagnosis to improve software testing. In *AAAI*, pages 1135–1141, 2014.

[46] Alenka Zuzek, Anton Biasizzo, and Franc Novak. Sequential diagnosis tool. *Microprocessors and Microsystems - Embedded Hardware Design*, 24(4):191–197, 2000.