### Problem Formulation

**Switched System Identification**

**Inputs:**
- Full-state observations of a system (with noise):
  \((x(t), x(t+1)), t = 0, \ldots, N - 1,\) number of modes \(m,\)
  error tolerances \(\epsilon, \tau > 0,\)

**Output:**
- Find \(m \times d\) matrices \(A_1, \ldots, A_m\) s.t.
  \(\|x(t+1) - A_m x(t)\|_2 \leq \epsilon\) \(\forall t\)

### Overall Algorithm

Organize constraints using a tree data structure.

Each node carries the following information:
- Data points that have been assigned to modes.
- Unassigned data points.
- Polyhedra \(P_1, \ldots, P_m\) representing constraints for \(A_1, \ldots, A_m\) resp.

**Key Steps of the Algorithm:**
1. Choose a previously unexplored leaf.
2. Expand the leaf (see below):
   - Discover matrices \(A_1, \ldots, A_m\) or
   - Add more children.

**Expanding a Tree Leaf**

**Leaf with unassigned data points:**
- Polyhedra \(P_1, \ldots, P_m\)
- \(P_j\) child forces matrix \(A_j\) to fit unexplained data point.

**Time Complexity**

**Idea #2:** Choose maximum volume ellipsoid (MVE) center of polyhedra.
- Volume shrinks by at least \(a < 1\) \([1, \S 4.3]\).

**Observations:**
- Combination of simple ideas.
- Easy to implement and works well in practice.

### Reformulation with a Gap

**Original Problem**

There are two possible outcomes:
- Yes: Successfully found \(m\) matrices satisfying error tolerances \(\epsilon, \tau\),
- No: No such matrices can fit the given data.

**Idea #1:** Reformulate problem with a gap.
- Input two relative error tolerances \(\epsilon_1 < \epsilon_2\)
- Yes: Successfully found \(m\) matrices satisfying error tolerances \(\epsilon_2, \tau\)
- No: No such matrices can fit the data for error tolerances \(\epsilon_1, \tau\).

**Main Result**

Algorithm with time complexity

**Implementation**

Implemented in the Python programming language.
- **Gurobi LP solver** (free academic license).
- Use Chebyshev center instead of MVE center.

**Comparison against two methods:**
- **MILP Solver:** Comparison with MILP.
  - Implemented using Gurobi: state-of-the-art solver \([2]\).
  - Worst-case exponential in the number of data points
- **Clustering-Based:** Fast method but inexact \([3]\).

### References