Project A3
Methods for Efficient Resource Utilization in Machine Learning Algorithms
Prof. Dr. Peter Marwedel, Prof. Dr. Jörg Rahnenführer

Problem

Methodology

Model-Based Optimisation (MBO)
- Regression model as a surrogate to approximate relationship between \(x\) and \(f(x)\)
- Predictions of model help to move quickly to regions with promising prediction quality
- Infill criterion balances exploitation and exploration:
  \[ LCR(x, \lambda) = \frac{1}{L(x)} \cdot L(x) \]

Infill criterion with different weighting of uncertainty (different values for \(\lambda\)) leads to multiple independent proposals

RAMBO Uses Resources Efficiently
- Asynchronous strategy
  - asyn.ei.bel: Worker determines next evaluation immediately. Estimated results of running jobs are fed into the surrogate model
- Synchronous strategy with multipoint proposals
- RAMBO: Central process creates set of scheduled jobs for all workers
- qLCB: One proposal for each worker after evaluations are finished on all workers

Result:
- RAMBO reduces idle time in comparison to qLCB
- RAMBO has similar resource utilization compared to asyn.ei.bel

RAMBO Approaches Optimum Faster
- RAMBO: beneficial due to controlled exploration
- asyn.ei: Overhead of calculating the EEl via MC Simulation decreases performance
- asyn.ei.bel: Estimated results deteriorate performance
- qLCB + asyn.ei: Less evaluations due to idling

Software:
- self-contained, well-documented, open-source packages, ensuring reproducibility:
  - mlrMBO, mlrMBO + RAMBO, mlrHyperopt, traceR

Resource-Aware Model-Based Optimisation (RAMBO)
- RAMBO extends MBO to parallel systems with optimized resource utilization for heterogeneous runtimes
- Resource Demands Estimator: Additional regression model to predict runtime
- Job Selection: runtime estimates + execution priority → interaction between scheduling and infill
- Scheduling: Controlled job selection to converge faster to optimum

Complexity of Hyperparameter Space: Prediction Quality versus Resource Utilization
- Misclassification Rate Depending on Hyperparameters
- Complex search space
- SVM: good prediction quality only for specific hyperparameters
- RF: stable prediction quality

Scheduling Strategies for Efficient Resource Utilization
- Inputs for scheduling in each MBO iteration:
  - \(f_i\): Priority of jobs determined by infill criterion
  - \(f_j\): Estimated runtime of jobs
- Resource-aware strategy
- Determine job with highest priority and set MBO iteration time bound to its runtime
- Map most promising job to one CPU exclusively
- Solve knapsack problem to maximize sum of priorities and map remaining jobs on CPUs
- Job priority refinement: based on hierarchical clustering to prefer jobs scattered across the search space

MBO Improves Prediction of Survival Times
Heterogeneous patient cohort (e.g. different clinical centers):
- Prediction of survival curves for subgroup

Problem:
- Subgroup model: small sample size
- Pooled model: high heterogeneity between subgroups

Solution:
- Group-specific weights to select subgroups that improve prediction quality
- Weighted version of partial log-likelihood
- Optimize C-index by setting weights using MBO

Results:
- Subgroup 1: Few similar subgroups with large weight
- Subgroup 2+3: Most subgroups with medium weight
- MBO improves prediction quality by selecting suitable weights as hyperparameters

Project A3
Methods for Efficient Resource Utilization in Machine Learning Algorithms
Prof. Dr. Peter Marwedel, Prof. Dr. Jörg Rahnenführer