Data-Driven Decision-Making
In Enterprise Applications

Robust Optimization Concepts

Rainer Schlosser

Hasso Plattner Institute (EPIC)

June 24, 2019

The Worlds is Full of . . .

- Problems & Goals
- Decisions
- Uncertainty

The Worlds is Full of . . .

- Problems & Goals
- Decisions
- Uncertainty

- We often hear: "Risk-Sensitive Decision-Making", "Robustness", "Minimizing Risk", "Eliminate Risk", "Balance Risks", . . .

  But: What exactly does it mean? What is your definition?

Database Configuration

Examples

- Index Selection, Partial Replication
- Data Layout / Compression, etc.

Database Configuration

Examples

- Index Selection, Partial Replication
- Data Layout / Compression, etc.

Goals & Objectives

- Optimize "**performance**": min runtime, min data, max throughout
- Constraints: budgets, bounds, computation time, reconfiguration costs
- Robustness: variance? worst case? uncertainty? risk aversion?

Data-Driven Workload Anticipation

Optimize “performance” based on historical data via . . .

- **observed** workloads (static + deterministic)
- **forecasted** workloads (dynamic + deterministic)
- **multiple potential** future workload scenarios (dynamic + stochastic)

Data-Driven Workload Anticipation

Optimize "performance" based on historical data via . . .

- **observed** workloads (static + deterministic)
- **forecasted** workloads (dynamic + deterministic)
- **multiple potential** future workload scenarios (dynamic + stochastic)

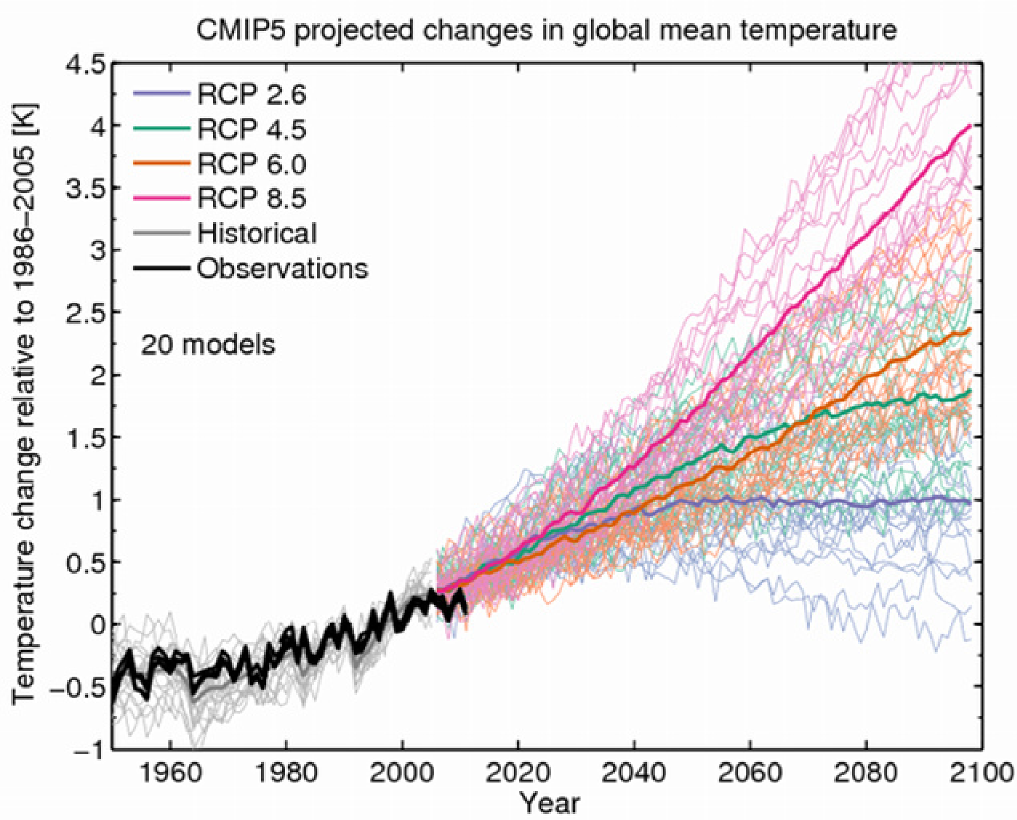
Let "workload" be characterized by number & costs of queries over time

(a) types of queries vary

(b) number of queries vary

(c) costs of queries vary (cf. skewness)

Observed, Forecasted, and Potential Workloads



Solutions for Deterministic Approaches

Existing solutions for deterministic workloads (observed & forecast)

- numerical algorithms (e.g., IBM approach for Index Selection (IS))
- linear programming (e.g., CoPhy for IS or Stefan for Replication)

Solutions for Deterministic Approaches

Existing solutions for deterministic workloads (observed & forecast)

- numerical algorithms (e.g., IBM approach for Index Selection (IS))
- linear programming (e.g., CoPhy for IS or Stefan for Replication)

In general, solution approaches have in common: $\max_{\vec{x}} \; f(\vec{x})$

- decisions $\vec{x}$ satisfy certain constraints (budget, etc.)
- decisions are chosen based on "performance" comparisons, cf. $f(\vec{x})$

Approaches for Stochastic Problems & Robustness

Performance F is random! Assume K potential scenarios with probability p_k :

- expected performance $$\max_{\vec{x}} \; E\left[F(\vec{x})\right] := \sum_{k=1,\ldots,K} p_k \cdot f_k(\vec{x})$$

Approaches for Stochastic Problems & Robustness

Performance F is random! Assume K potential scenarios with probability p_k :

- expected performance $\max_{\vec{x}} \; E\left[F(\vec{x})\right] := \sum_{k=1,\ldots,K} p_k \cdot f_k(\vec{x})$

- mean-variance opt. $\max_{\vec{x}} \; E\left[F(\vec{x})\right] - \alpha \cdot \underbrace{Var\left[F(\vec{x})\right]}_{:= \sum_{k=1,\ldots,K} P_k \cdot \left(f_k(\vec{x}) - E\left[F(\vec{x})\right]\right)^2}$

- utility functions

- worst case

Approaches for Stochastic Problems & Robustness

Performance F is random! Assume K potential scenarios with probability p_k:

- expected performance $$\max_{\vec{x}} \; E\left[F(\vec{x})\right] := \sum_{k=1,\ldots,K} p_k \cdot f_k(\vec{x})$$

- mean-variance opt. $$\max_{\vec{x}} \; E\left[F(\vec{x})\right] - \alpha \cdot \underbrace{Var\left[F(\vec{x})\right]}_{:= \sum_{k=1,\ldots,K} P_k \cdot \left(f_k(\vec{x}) - E\left[F(\vec{x})\right]\right)^2}$$

- utility functions $$\max_{\vec{x}} \; E\left[u\left(F(\vec{x})\right)\right] := \sum_{k=1,\ldots,K} p_k \cdot u\left(f_k(\vec{x})\right)$$

- worst case

Approaches for Stochastic Problems & Robustness

Performance F is random! Assume K potential scenarios with probability p_k :

- expected performance $\max_{\vec{x}} \; E\left[F(\vec{x})\right] := \sum_{k=1,\dots,K} p_k \cdot f_k(\vec{x})$

- mean-variance opt. $\max_{\vec{x}} \; E\left[F(\vec{x})\right] - \alpha \cdot \underbrace{Var\left[F(\vec{x})\right]}_{:= \sum_{k=1,\dots,K} P_k \cdot \left(f_k(\vec{x}) - E\left[F(\vec{x})\right]\right)^2}$

- utility functions $\max_{\vec{x}} \; E\left[u\left(F(\vec{x})\right)\right] := \sum_{k=1,\dots,K} p_k \cdot u\left(f_k(\vec{x})\right)$

- worst case $\max_{\vec{x}} \min_{k=1,\dots,K} f_k(\vec{x}) \;\cong \max_{\vec{x},L} L \quad s.t. \quad f_k(\vec{x}) \geq L \quad \forall k$

Robustinisierung of Existing Approaches

Idea: Adapt existing deterministic solutions $\max_{\vec{x}} \; f(\vec{x})$

(1) use multiple workloads & (2) adapt the performance criteria

Robustinisierung of Existing Approaches

Idea: Adapt existing deterministic solutions $\max_{\vec{x}} \; f(\vec{x})$

(1) use multiple workloads & (2) adapt the performance criteria

Derive stochastic solutions as follows:

(i) expected performance: let $f(\vec{x}) := E\left[F(\vec{x})\right] = \sum_{k=1,\ldots,K} p_k \cdot f_k(\vec{x})$

Robustinisierung of Existing Approaches

Idea: Adapt existing deterministic solutions $\max_{\vec{x}} \ f(\vec{x})$

(1) use multiple workloads & (2) adapt the performance criteria

Derive stochastic solutions as follows:

(i) expected performance: let $f(\vec{x}) := E\left[F(\vec{x})\right] = \sum_{k=1,\dots,K} p_k \cdot f_k(\vec{x})$

(ii) mean-variance optimization: let $f(\vec{x}) := E\left[F(\vec{x})\right] - \alpha \cdot Var\left[F(\vec{x})\right]$

(iii) utility functions: let $f(\vec{x}) := E\left[u\left(F(\vec{x})\right)\right]$

(iv) worst case: let $f(\vec{x}) := \max_{\vec{x}} \min_{k=1,\dots,K} f_k(\vec{x})$

Discussion of Increased Problem Complexity

- Definition of workload scenarios:

- Number of scenarios limited:

- More decisions/variables:

- Adaption of constraints:

Discussion of Increased Problem Complexity

- Definition of workload scenarios: based on historical data
  use Conf.Intvls of Forecasts (TSA)
  let practitioners decide
- Number of scenarios limited:
- More decisions/variables:
- Adaption of constraints:

Discussion of Increased Problem Complexity

• Definition of workload scenarios:	based on historical data
	use Conf.Intvls of Forecasts (TSA)
	let practitioners decide
• Number of scenarios limited:	no
• More decisions/variables:	no
• Adaption of constraints:	no

Discussion of Increased Problem Complexity

• Definition of workload scenarios:	based on historical data
	use Conf.Intvls of Forecasts (TSA)
	let practitioners decide
• Number of scenarios limited:	no
• More decisions/variables:	no
• Adaption of constraints:	no
• New **Nonlinearity**:	no, BQP, yes, yes
Is this a problem?	no, hardly, yes (iif a solver is used), no

Examples (Index Selection)

- Numerical heuristic “IBM”

  (i) define index candidates

  (ii) pick indexes greedily following the criteria: saved runtime/space

  (iii) shuffle to account for index interaction

  To do: Adjust step (ii), i.e., use some adapted “robust” criteria (easy)

Examples (Index Selection)

- Numerical heuristic "IBM"
  - (i) define index candidates
  - (ii) pick indexes greedily following the criteria: saved runtime/space
  - (iii) shuffle to account for index interaction

  To do: Adjust step (ii), i.e., use some adapted "robust" criteria (easy)

- Solver-based heuristic "CoPhy"
  - (i) define index candidates
  - (ii) solve integer LP, i.e., minimize total runtime

  To do: Adjust step (ii) for Mean-Variance optimization (interested?)

Robustinisierung of CoPhy

Old:

$$\underset{\substack{z_{ji}, x_m \in \{0,1\}, m \in I, \\ j=1,\ldots,Q, i \in I_j \cup 0}}{minimize} \sum_{k=1,\ldots,1} p_k \cdot \sum_{j=1,\ldots,Q, i \in I_j \cup 0} b_{j,k} \cdot f_j(i) \cdot z_{ji}$$

$$\sum_{i \in I_j \cup 0} z_{ji} = 1 \qquad \forall j$$

Index decision for query *j*

$$z_{ji} \le x_i \qquad \forall j, i$$

Index *i* used at all?

$$\sum_{m \in I} s_m \cdot x_m \le A$$

Budget constraint

Robustinisierung of CoPhy

Old:

$$\underset{\substack{z_{ji}, x_m \in \{0,1\}, m \in I, \\ j=1,\ldots,Q, i \in I_j \cup 0}}{minimize} \sum_{k=1,\ldots,1} p_k \cdot \sum_{j=1,\ldots,Q, i \in I_j \cup 0} b_{j,k} \cdot f_j(i) \cdot z_{ji}$$

$$\sum_{i \in I_j \cup 0} z_{ji} = 1 \qquad \forall j$$ Index decision for query *j*

$$z_{ji} \leq x_i \qquad \forall j, i$$ Index *i* used at all?

$$\sum_{m \in I} s_m \cdot x_m \leq A$$ Budget constraint

New:

$$\underset{\substack{z_{ji}, x_m \in \{0,1\}, m \in I, \\ j=1,\ldots,Q, i \in I_j \cup 0}}{minimize} \underbrace{\sum_{k=1,\ldots,K} p_k \cdot \sum_{j=1,\ldots,Q, i \in I_j \cup 0} b_{j,k} \cdot f_j(i) \cdot z_{ji}}_{=:EW} \qquad \text{(quadratic)}$$

$$+\alpha \cdot \sum_{k=1,\ldots,K} p_k \cdot \left(\sum_{j=1,\ldots,Q, i \in I_j \cup 0} b_{j,k} \cdot f_j(i) \cdot z_{ji} - EW \right)^2$$

Summary

- Use multiple stochastic workloads instead of forecasted ones
- Different techniques to include robustness can be used
- Existing approaches just slightly have to be extended
- Additional complexity is tolerable
- Concepts are general applicable
- Approach is suitable if (i) randomness is high & (ii) robustness is needed

Summary

- Use multiple stochastic workloads instead of forecasted ones
- Different techniques to include robustness can be used
- Existing approaches just slightly have to be extended
- Additional complexity is tolerable
- Concepts are general applicable
- Approach is suitable if (i) randomness is high & (ii) robustness is needed

Overview

2	April 25	Linear Programming I
3	April 29	Linear Programming II
4	May 2	Linear/Logistic Regression + Homework (3 weeks time)
5	May 16	Exercise Implementations
6	May 20	Dynamic Programming
7	May 23	Pricing in Competitive Markets
8	May 27	Project Assignments + Homework 2 (until June 13)
9	June 3	Workshop / Group Meetings
10	June 13	Workshop / Group Meetings (hand in Homework II)
11	June 20	Presentations (First Results)
12	June 24	Robust Optimization Concepts
13	**June 27**	**Workshop / Group Meetings**
14/15	July 1/4	Workshop / Group Meetings
16	July 11	Presentations (Final Results), Feedback, Documentation (Aug 31)