Advanced Mixed Integer Programming Formulations for Non-Convex Optimization Problems in Statistical Learning

Juan Pablo Vielma

Massachusetts Institute of Technology

2016 IISA International Conference on Statistics.
Corvallis, Oregon, August, 2016.

(Custom) Product Recommendations via CBCA

Feature	SX530	RX100
Zoom	50x	3.6x
Prize	$249.99	$399.99
Weight	15.68 ounces	7.5 ounces
Prefer	☑	☐

Feature	TG-4	G9
Waterproof	Yes	No
Prize	$249.99	$399.99
Weight	7.36 lb	7.5 lb
Prefer	☐	☑

Feature	TG-4	Galaxy 2
Waterproof	Yes	No
Prize	$249.99	$399.99
Viewfinder	Electronic	Optical
Prefer	☑	☐

We recommend:

Towards Optimal Product Recommendation

- Find enough information about preferences to recommend

Feature	SX530	RX100
Zoom	50x	3.6x
Prize	$249.99	$399.99
Weight	15.68 ounces	7.5 ounces
Prefer	✓	☐

Feature	TG-4	G9
Waterproof	Yes	No
Prize	$249.99	$399.99
Weight	7.36 lb	7.5 lb
Prefer	☐	✓

Feature	TG-4	Galaxy 2
Waterproof	Yes	No
Prize	$249.99	$399.99
Viewfinder	Electronic	Optical
Prefer	✓	☐

We recommend:

- How do I pick the **next (1st) question** to obtain the largest reduction of uncertainty or “variance” on preferences

Choice-based Conjoint Analysis

Feature	Chewbacca	BB-8
Wookiee	Yes	No
Droid	No	Yes
Blaster	Yes	No
I would buy toy	☑	☐
Product Profile	x^1	x^2

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = x^2$$

MNL Preference Model

- Utilities for 2 products, n features (e.g. n = 12)

$$U_1 = \beta \cdot x^1 + \epsilon_1 = \sum_{i=1}^{n} \beta_i x_i^1 + \epsilon_1$$

$$U_2 = \beta \cdot x^2 + \epsilon_2 = \sum_{i=1}^{n} \beta_i x_i^2 + \epsilon_2$$

part-worths

product profile

noise (gumbel)

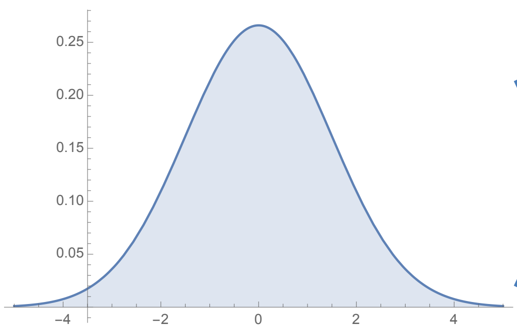
- Utility maximizing customer: $x^1 \succeq x^2 \Leftrightarrow U_1 \text{“}\geq\text{”} U_2$

- Noise can result in response error:

$$L\left(\beta \,|x^1 \succeq x^2\right) = \mathbb{P}\left(x^1 \succeq x^2 \mid \beta\right) = \frac{e^{\beta \cdot x^1}}{e^{\beta \cdot x^1} + e^{\beta \cdot x^2}}$$

Next Question To Reduce "Variance": Bayesian

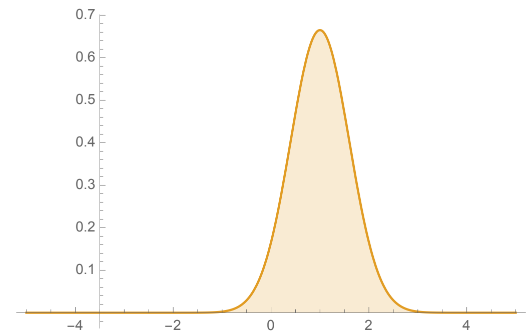
Prior Distribution of β



Feature	SX530	RX100
Zoom	50x	3.6x
Prize	$249.99	$399.99
Weight	15.68 ounces	7.5 ounces
Prefer	✓	☐

Bayesian Update

Posterior Distribution

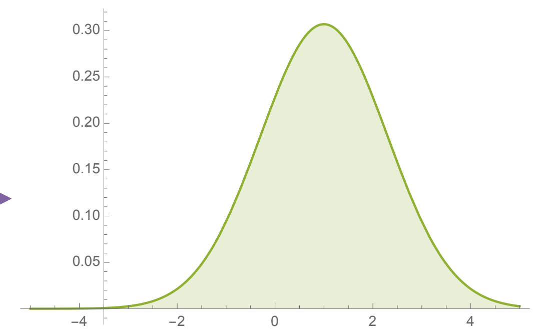


MCMC

Feature	TG-4	Galaxy 2
Waterproof	Yes	No
Prize	$249.99	$399.99
Viewfinder	Electronic	Optical
Prefer	✓	☐

Bayesian Update

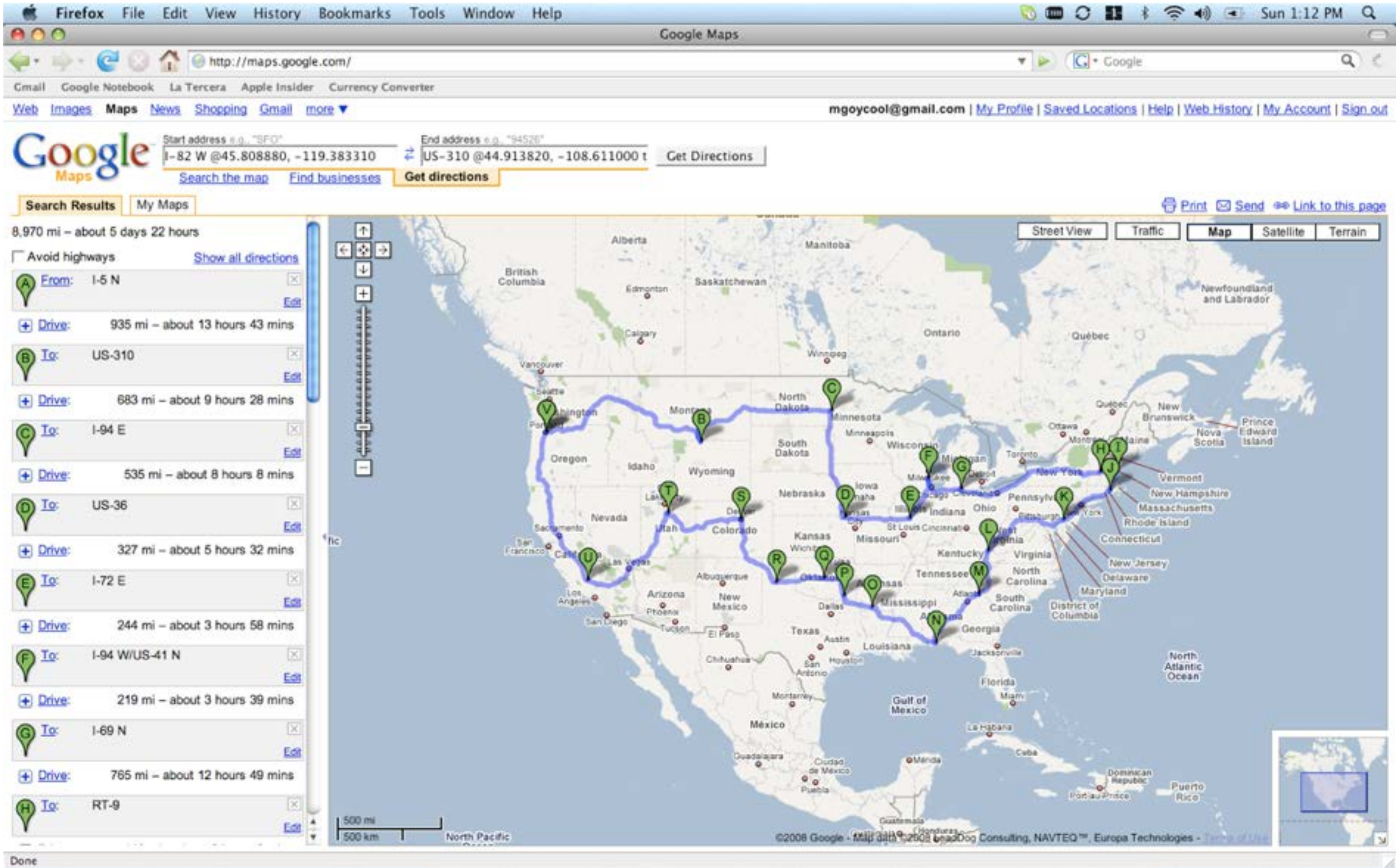
Posterior Distribution



- Black-box objective: Question Selection = Enumeration ☹
- Question selection by Mixed Integer Programming (MIP)

Avoiding Enumeration with MIP

Traveling Salesman Problem (TSP): Visit Cities Fast



How about 49 cities?

- Number of tours $= 48!/2 \approx 10^{60}$
- Fastest supercomputer $\approx 10^{17}$flops
- Assuming one floating point operation per tour:
  $> 10^{35}$ years $\approx 10^{25}$ times the age of the universe!
- How long does it take on an iphone?
  - Less than a second!
  - 4 iterations of **cutting plane** method!
  - Dantzig, Fulkerson and Johnson 1954 did it by hand!
  - For more info see tutorial in ConcordeTSP app
  - **Cutting planes** are the key for effectively solving (even NP-hard) MIP problems in practice.

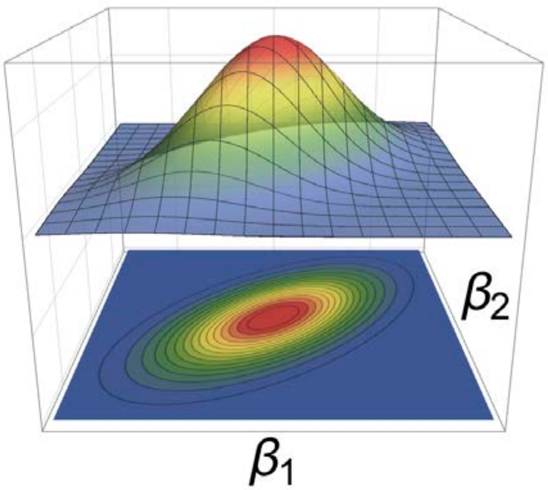
50+ Years of MIP = Significant Solver Speedups

- Algorithmic Improvements (Machine Independent):
  - CPLEX v1.2 (1991) – v11 (2007): 29,000x speedup
  - Gurobi v1 (2009) – v6.5 (2015): 48.7x speedup
  - Commercial, but free for academic use
- (Reasonably) effective free / open source solvers:
  - GLPK, CBC and SCIP (free only for non-commercial)
- Easy to use, fast and versatile modeling languages
  - Julia based JuMP modelling language
- Linear MIP solvers very mature and effective:
  - Convex nonlinear MIP getting there (quadratic nearly there)

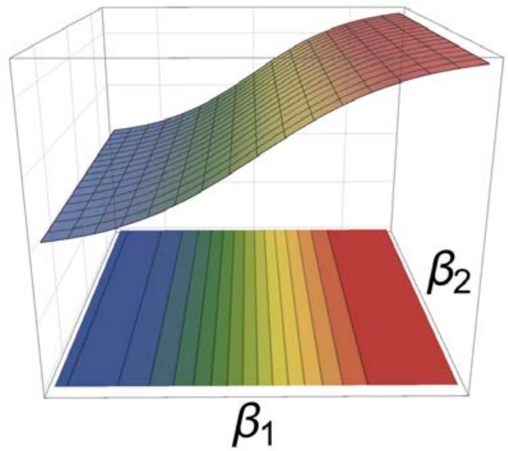
Question Selection with MIP

Bayesian Update and Geometric Updates

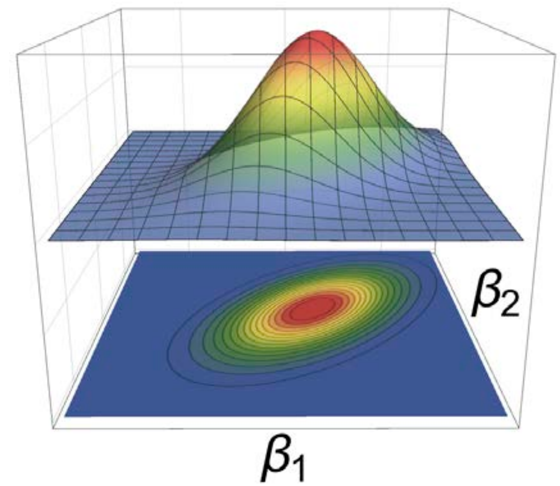
Prior distribution



Answer likelihood



Posterior distribution



$$\beta \sim N(\mu, \Sigma)$$

$$x^1 \succeq x^2$$

$$f\left(\beta \mid x^1 \succeq x^2\right)$$

$$\phi(\beta; \mu, \Sigma)$$

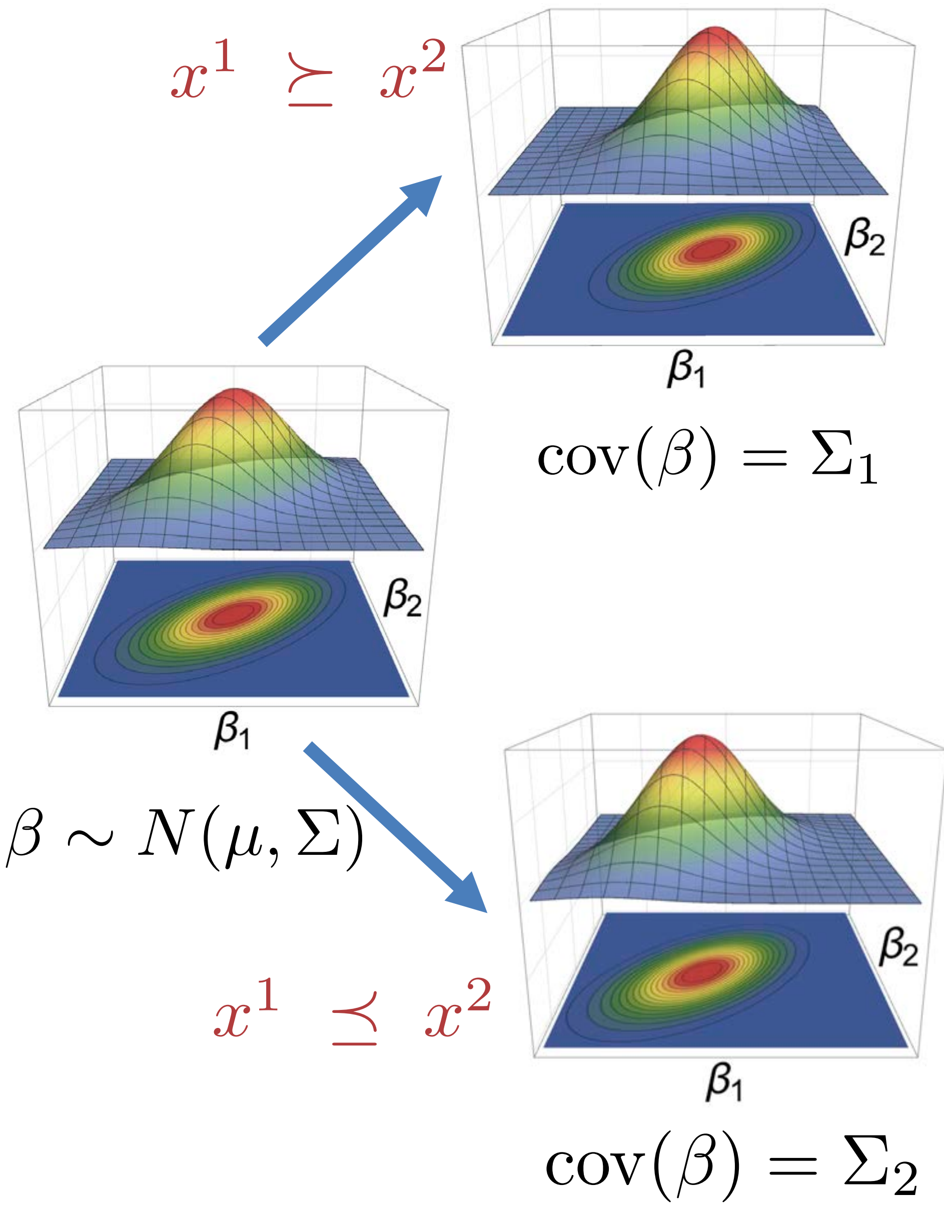
$$L\left(\beta \mid x^1 \succeq x^2\right)$$

Multidimensional Integration?

$$f\left(\beta \mid x^1 \succeq x^2\right) = \frac{\boxed{\phi(\beta\,;\,\mu, \Sigma)\, L\left(\beta \mid x^1 \succeq x^2\right)}}{\int_{\mathbb{R}} \phi(\beta\,;\,\mu, \Sigma)\, L\left(\beta \mid x^1 \succeq x^2\right) d\beta}$$

non-convex on $x^1, x^2 \in \{0,1\}^n$

D-Efficiency and Posterior Covariance Matrix



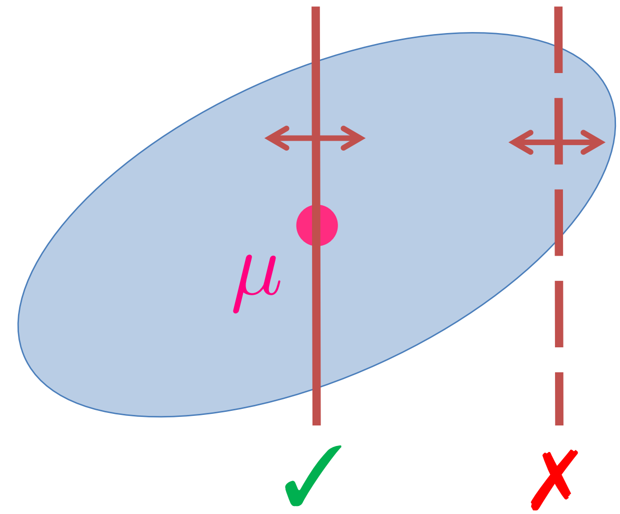
- "Variance" = D-Efficiency:
- $f\left(x^1, x^2\right) := \mathbb{E}_{\beta, x^1 \preceq/\succeq x^2}\left(\det(\Sigma_i)^{1/p}\right)$
- Non-convex function
- Even evaluating expected D-Efficiency for a question requires multidimensional integration

Standard Question Selection Criteria

$$(\beta - \mu)' \cdot \Sigma^{-1} \cdot (\beta - \mu) \leq r$$

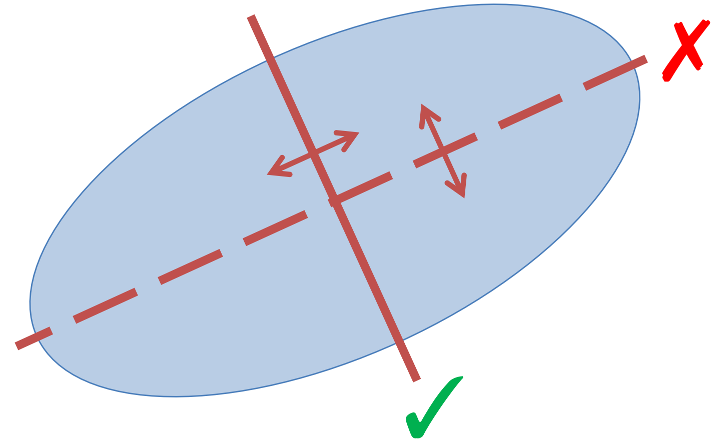
- Choice balance:
  - Minimize distance to center

$$\mu \cdot (x^1 - x^2)$$



- Postchoice symmetry:
  - Maximize variance of question

$$(x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2)$$

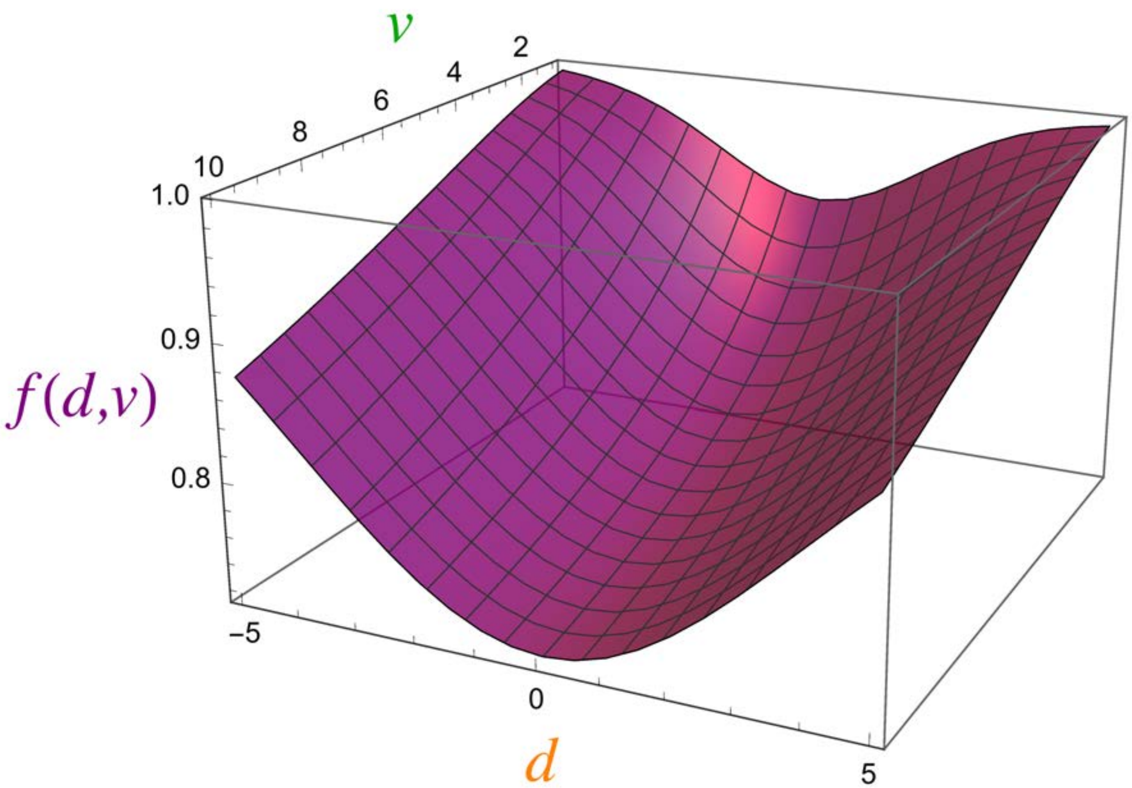


D-efficiency: Balance Question Trade-off

- D-efficiency = Non-convex function $f(d, v)$ of

  distance: $d := \mu \cdot (x^1 - x^2)$

  variance: $v := (x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2)$



Can evaluate $f(d, v)$ with 1-dim integral 🙂

Optimization Model

$$\min \quad f(d, v) \quad ✗$$

$$s.t.$$

$$\mu \cdot (x^1 - x^2) = d \quad ✓$$

$$(x^1 - x^2)' \cdot \Sigma \cdot (x^1 - x^2) = v \quad ✗$$

$$A^1 x^1 + A^2 x^2 \leq b \quad ✓$$

$$x^1 \neq x^2 \quad ✗$$

$$x^1, x^2 \in \{0, 1\}^n$$

Technique 1: Binary Quadratic $x^1, x^2 \in \{0,1\}^n$

$$(x^1 - x^2)' \cdot \Sigma \cdot (x^1 - x^2) = v$$

$$X^l_{i,j} = x^l_i \cdot x^l_j \quad (l \in \{1,2\}, \quad i,j \in \{1,\ldots,n\}):$$

$$X^l_{i,j} \leq x^l_i, \quad X^l_{i,j} \leq x^l_j, \quad X^l_{i,j} \geq x^l_i + x^l_j - 1, \quad X^l_{i,j} \geq 0$$

$$W_{i,j} = x^1_i \cdot x^2_j :$$

$$W_{i,j} \leq x^1_i, \quad W_{i,j} \leq x^2_j, \quad W_{i,j} \geq x^1_i + x^2_j - 1, \quad W_{i,j} \geq 0$$

$$\sum_{i,j=1}^{n} \left(X^1_{i,j} + X^2_{i,j} - W_{i,j} - W_{j,i}\right) \Sigma_{i,j} = v$$

Technique 1: Binary Quadratic $x^1, x^2 \in \{0,1\}^n$

$$x^1 \neq x^2 \quad \Leftrightarrow \quad \|x^1 - x^2\|_2^2 \geq 1$$

$$X^l_{i,j} = x^l_i \cdot x^l_j \quad (l \in \{1,2\}, \quad i,j \in \{1,\ldots,n\}):$$

$$X^l_{i,j} \leq x^l_i, \quad X^l_{i,j} \leq x^l_j, \quad X^l_{i,j} \geq x^l_i + x^l_j - 1, \quad X^l_{i,j} \geq 0$$

$$W_{i,j} = x^1_i \cdot x^2_j :$$

$$W_{i,j} \leq x^1_i, \quad W_{i,j} \leq x^2_j, \quad W_{i,j} \geq x^1_i + x^2_j - 1, \quad W_{i,j} \geq 0$$

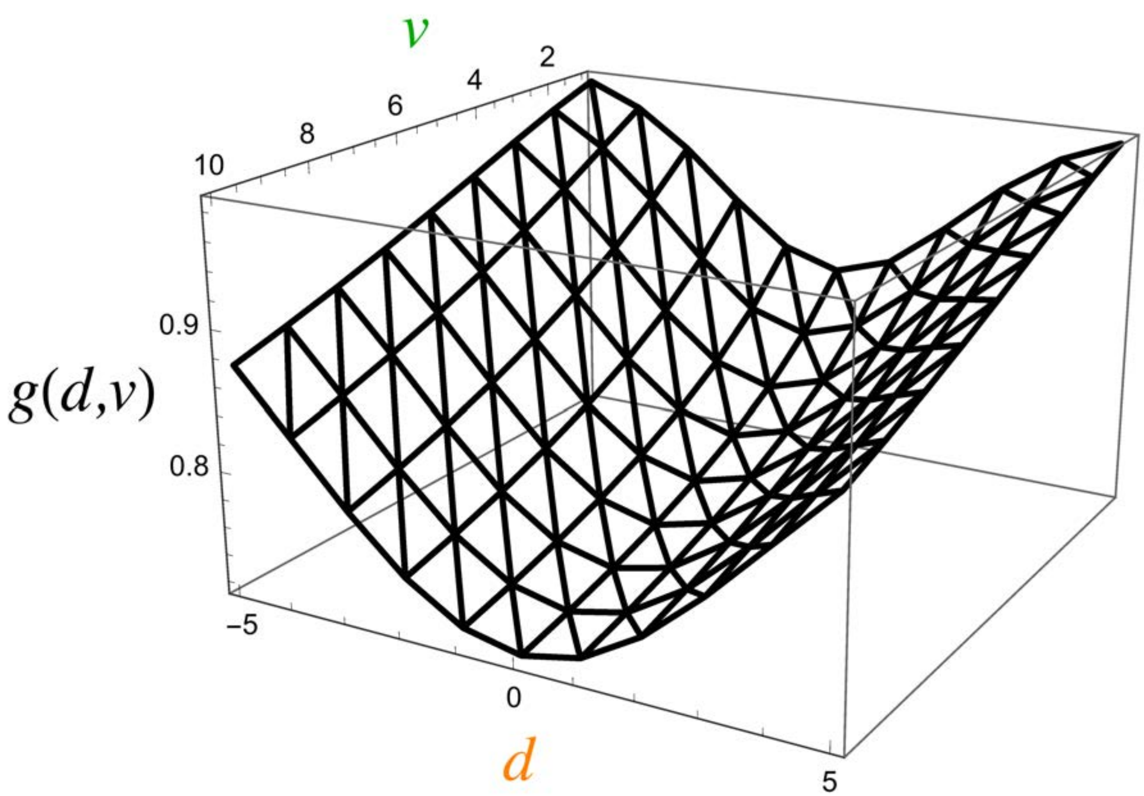
$$\sum_{i,j=1}^{n} \left(X^1_{i,j} + X^2_{i,j} - W_{i,j} - W_{j,i}\right) \geq 1$$

Technique 2: Piecewise Linear Functions

- D-efficiency = Non-convex function $f(d, v)$ of

  distance: $d := \mu \cdot (x^1 - x^2)$

  variance: $v := (x^1 - x^2)' \cdot \sum \cdot (x^1 - x^2)$



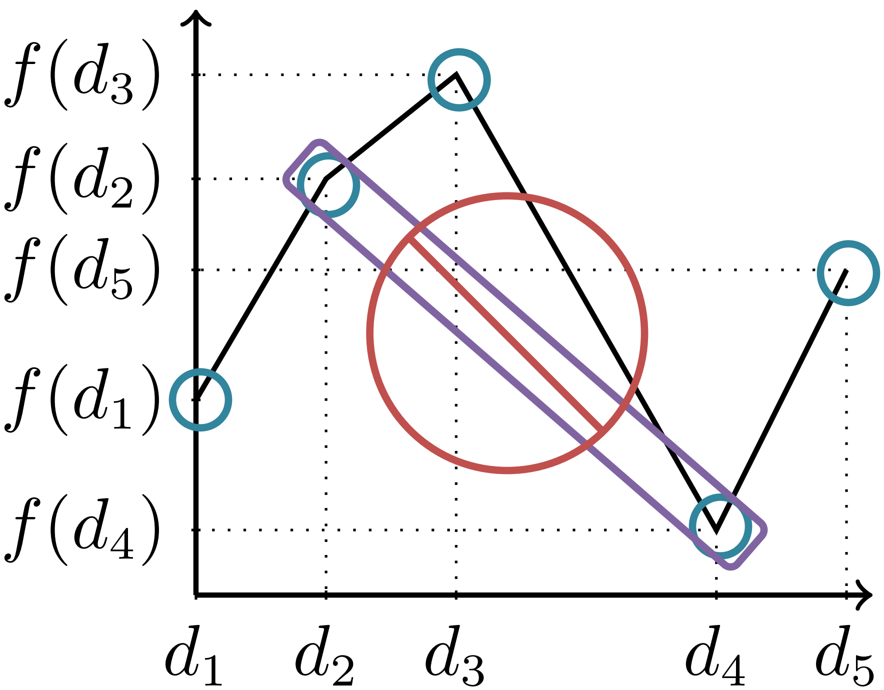
Can evaluate $f(d, v)$ with 1-dim integral 🙂

Piecewise Linear Interpolation

MIP formulation

Simple Formulation for Univariate Functions

$$z = f(x)$$



$$\text{Size} = O(\#\text{ of segments})$$

$$\begin{pmatrix} x \\ z \end{pmatrix} = \sum_{j=1}^{5} \begin{pmatrix} d_j \\ f(d_j) \end{pmatrix} \lambda_j$$

$$1 = \sum_{j=1}^{5} \lambda_j, \quad \lambda_j \geq 0$$

$$y \in \{0,1\}^4, \quad \sum_{i=1}^{4} y_i = 1$$

$$0 \leq \lambda_1 \leq y_1$$

$$0 \leq \lambda_2 \leq y_1 + y_2$$

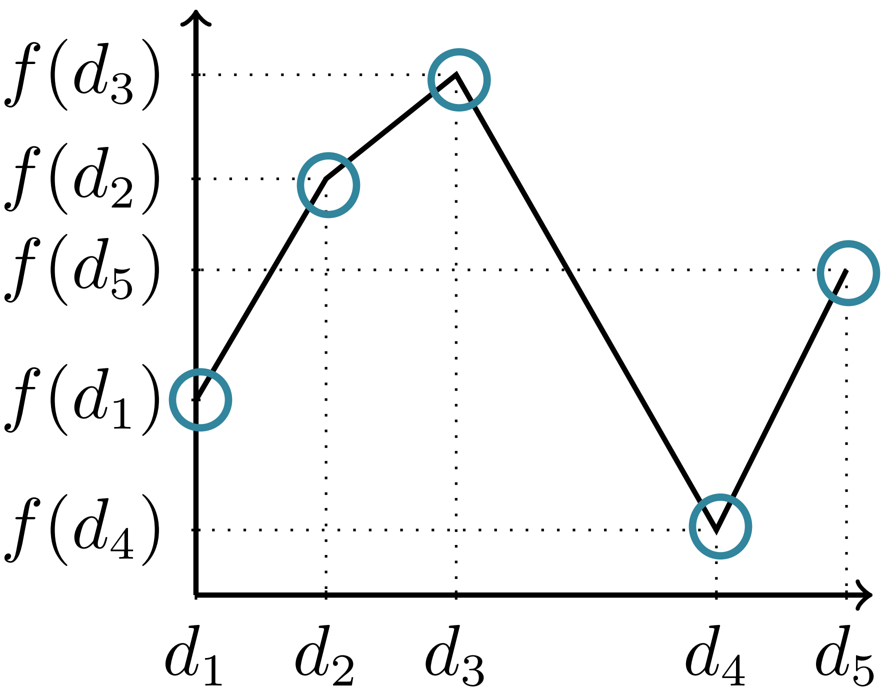
$$0 \leq \lambda_3 \leq y_2 + y_3$$

$$0 \leq \lambda_4 \leq y_3 + y_4$$

$$0 \leq \lambda_5 \leq y_4$$

Advanced Formulation for Univariate Functions

$$z = f(x)$$



$$\text{Size} = O\left(\log_2 \# \text{ of segments}\right)$$

$$\begin{pmatrix} x \\ z \end{pmatrix} = \sum_{j=1}^{5} \begin{pmatrix} d_j \\ f(d_j) \end{pmatrix} \lambda_j$$

$$1 = \sum_{j=1}^{5} \lambda_j, \quad \lambda_j \geq 0$$

$$y \in \{0,1\}^2$$

$$0 \leq \lambda_1 + \lambda_5 \leq 1 - y_1$$

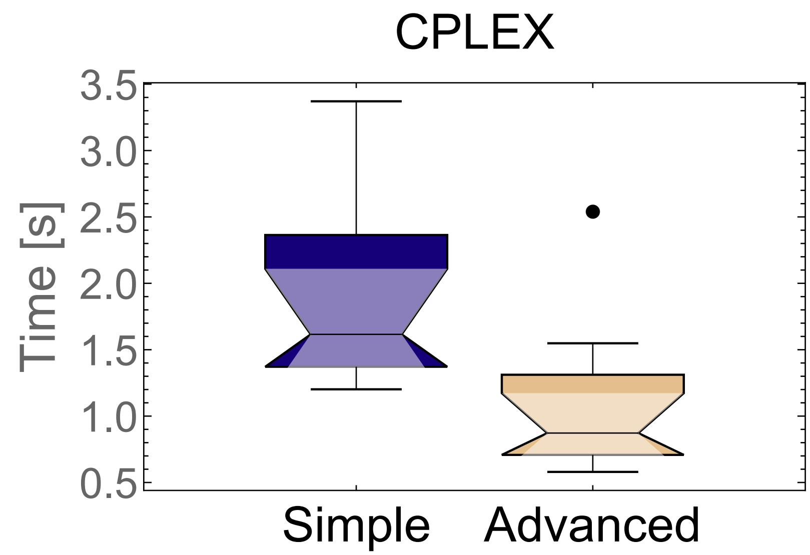
$$0 \leq \lambda_3 \leq y_1$$

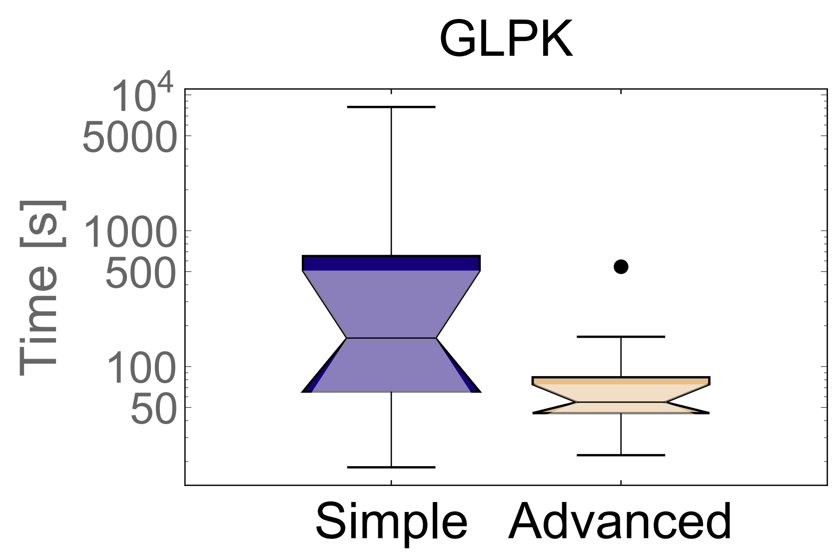
$$0 \leq \lambda_4 + \lambda_5 \leq 1 - y_2$$

$$0 \leq \lambda_1 + \lambda_2 \leq y_2$$

Computational Performance

- Advanced formulations provide an computational advantage
- Advantage is significantly more important for free solvers
- State of the art commercial solvers can be significantly better that free solvers
- Still, free is free!





Summary and Main Messages

- Always choose Chewbacca!

- MIP can solve very challenging problems in practice
- Commercial solvers best, but free solvers reasonable
  - Easily accessible and integrated into complex systems through the JuMP modeling language github.com/JuliaOpt/JuMP.jl
- Formulations = speed-ups and are (relatively) easy to learn
  - Mixed integer linear programming formulation techniques. J. P. Vielma. SIAM Review 57, 2015. pp. 3-57.
- CBC application: http://ssrn.com/abstract=2798984