Learning Uncertainty Sets in Robust Optimization

Bartolomeo Stellato

Department of Operations Research and Financial Engineering
Department of Electrical and Computer Engineering
Center for Statistics and Machine Learning

PRINCETON UNIVERSITY

stellato.io
bstellato@princeton.edu

Problem family with uncertain constraints

family/context parameter (known)
y

optimization variable

$$\begin{aligned}&\text{minimize} && f(x, y)\\ &\text{subject to} && g(x, u, y) \leq 0\end{aligned}$$

uncertain parameter (unknown)

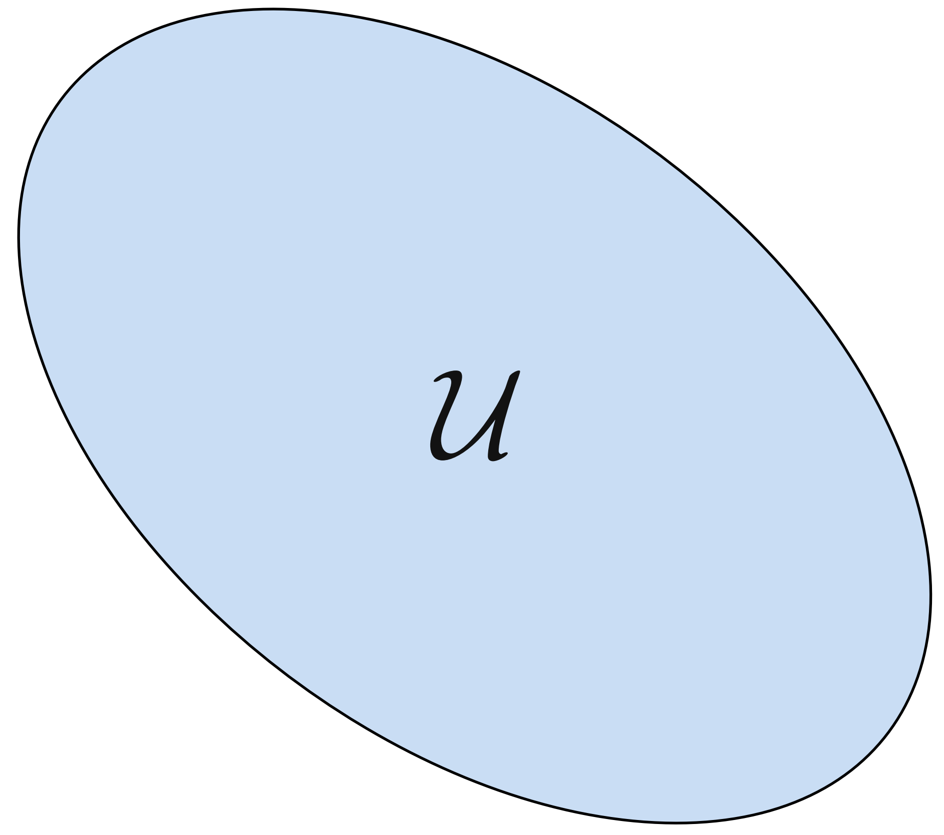
We want to guarantee constraint satisfaction

Example
Inventory Management

- y: inventory levels
- u: uncertain demands
- x: order decisions

Robust optimization recipe

1. Pick uncertainty set $\mathcal{U}$
2. Ensure constraint satisfaction $\forall u \in \mathcal{U}$

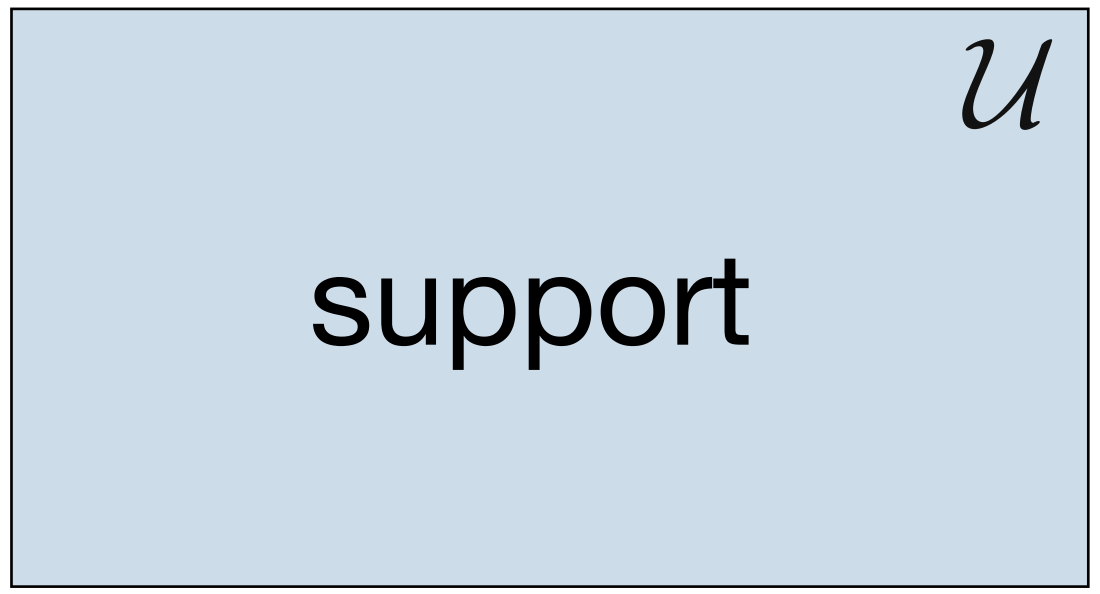


$$\begin{array}{ll} \text{minimize} & f(x, y) \\ \text{subject to} & g(x, u, y) \leq 0 \quad \forall u \in \mathcal{U} \end{array}$$

How do we pick the uncertainty set?

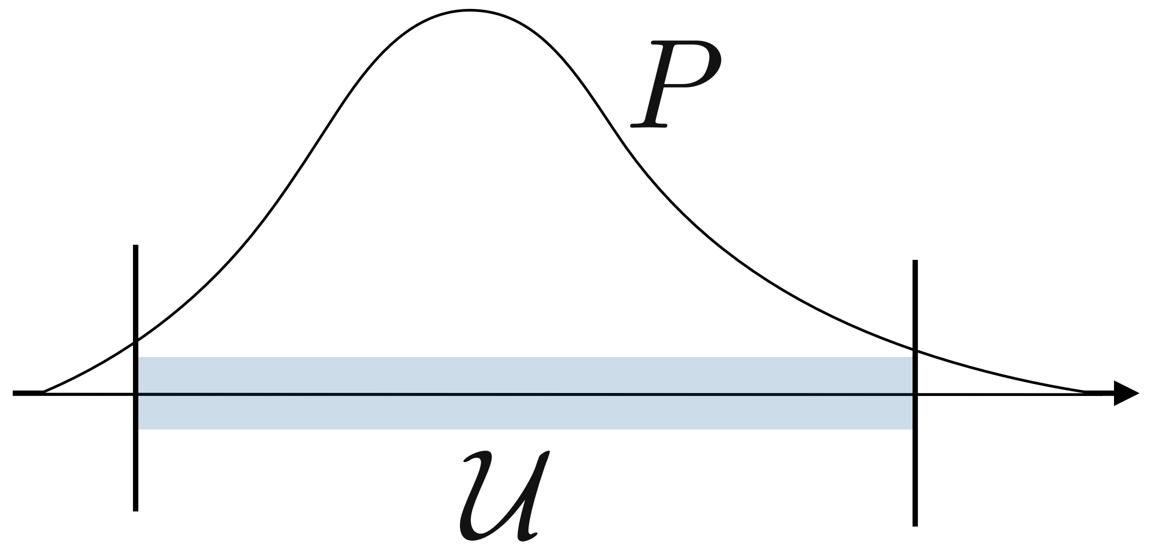
Picking the uncertainty set is difficult

Worst-case approach



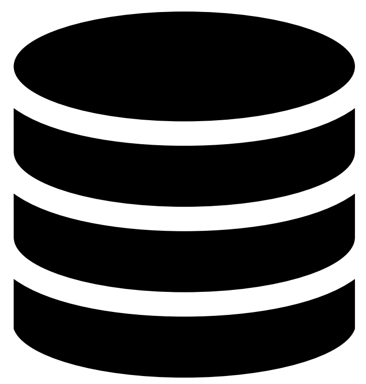
❌ Very conservative

Probabilistic approach



❌ nobody knows P

Data-driven approach



Can we use data to construct uncertainty sets?

“Probabilistic Guarantees in Robust Optimization”, D. Bertsimas, D. den Hertog, and J. Pauphilet (2019)

Data-driven methods for Robust Optimization

Hypothesis testing

D. Bertsimas, V. Gupta, and N. Kallus (2014)

Quantile Estimation

L. Jeff Hong, Z. Huang, and H. Lam (2021)

Wasserstein Distributionally Robust Optimization

P. M. Esfahani and D. Kuhn. (2018).

D. Bertsimas, S. Shtern, B. Sturt (2022)

I. Wang, C. Becker, B. Van Parys, and B. Stellato (2022)

Deep Learning

M. Goerigk, J. Kurtz (2023)

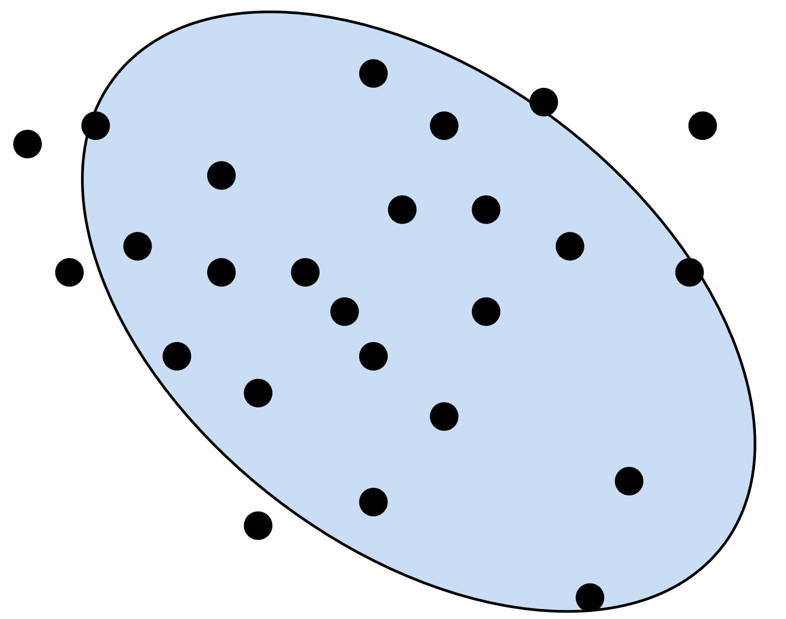
Most approaches decouple

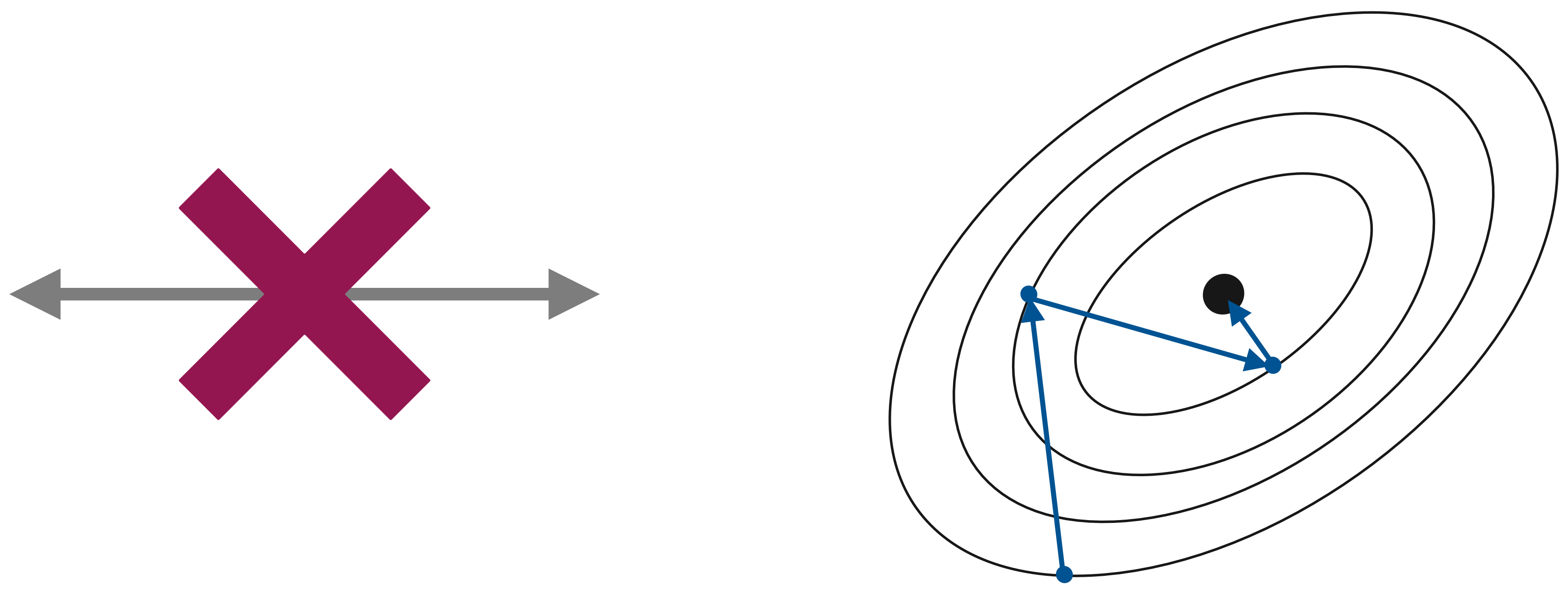
Uncertainty set construction

Downstream optimization task

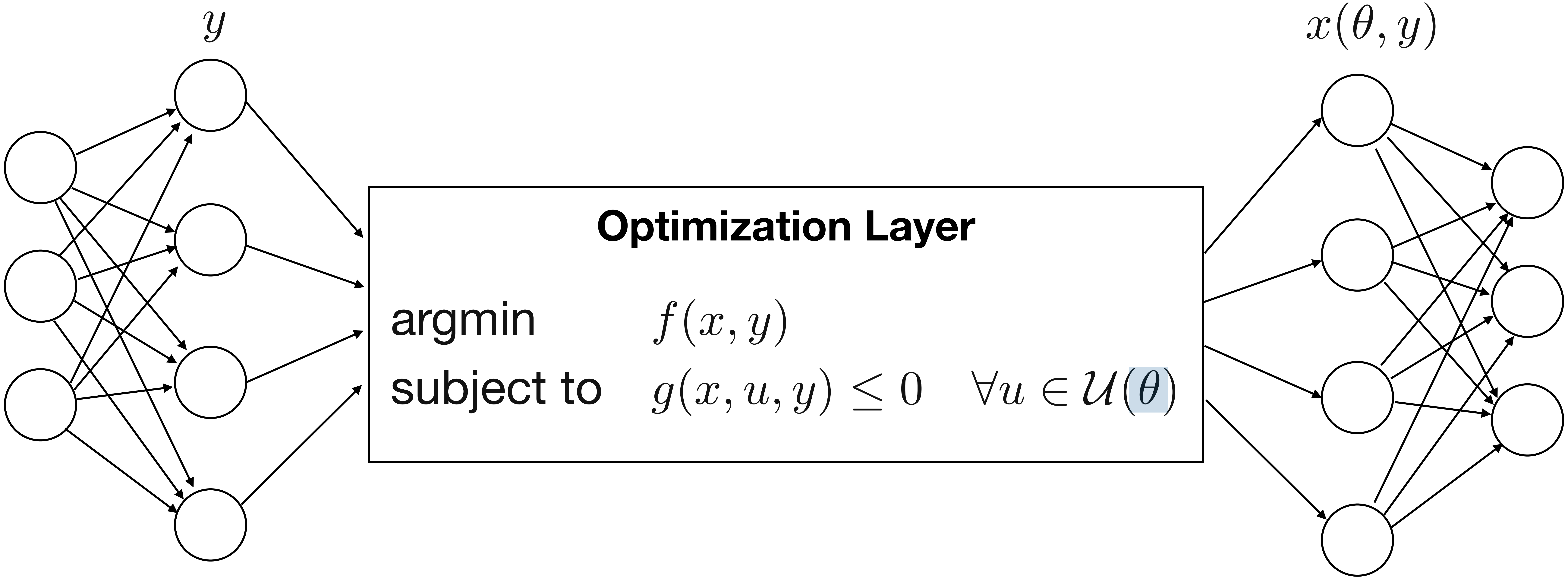
high coverage requirement

$$\mathbf{P}(u \in \mathcal{U}) \geq 1 - \epsilon$$





Leveraging solution structure to choose uncertainty sets



Main idea

Use **differentiable optimization**
to automatically learn
shape and size

Connections with Contextual Optimization

Contextual Optimization

$$x(y) \in \underset{x}{\text{argmin}}\ \mathbf{E}_{\mathbf{P}(u|y)}(f(x,u))$$

[D. Bertsimas and N. Kallus (2020)], [Elmachtoub and Grigas (2022)],
[H. Rahimian, B. Pagnoncelli (2022)]

“A Survey of Contextual Optimization Methods for Decision Making under Uncertainty”,
U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, T. Vidal (2023)

Conditional Robust Optimization

$$x(y) \in \underset{x}{\text{argmin}}\ \max_{u \in \mathcal{U}(y)} f(x,u)$$

[A.R. Chenreddy, N. Bandi, E. Delage (2022)], [S. Ohmori (2021)],
[E. Persak, M. F. Anjos (2023)], [C.Sun, L. Liu, X. Li (2023)]

Differences from our work

- Distribution/uncertainty set depends on y
- High coverage requirements $\mathbf{P}(u \in \mathcal{U}(y)) \geq 1 - \epsilon$
- Limited focus on uncertain constraints

Example: newsvendor problem

“photo of a newsvendor cat” Dall-E

Objective

$$k^T x - p^T \min(x, u)$$

order price → k

sell price → p

orders → x

uncertain demand → u

$\log u \sim \mathcal{N}(\mu, \Sigma)$

Robust Reformulation

$$\begin{array}{ll} \text{minimize} & \tau \\ \text{subject to} & k^T x + \max\{-p^T x, -p^T u\} \le \tau \quad \forall u \in \mathcal{U}(\theta) \\ & x \ge 0 \end{array}$$

family parameter $y = (k, p)$

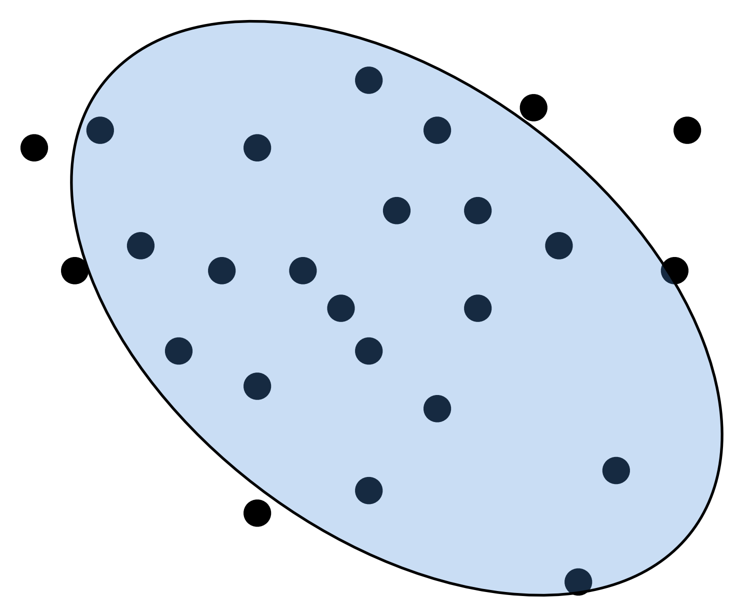
how do we pick the uncertainty set?

Standard vs reshaped uncertainty set for newsvendor

Parameters $\theta = (A, b)$

Standard uncertainty set

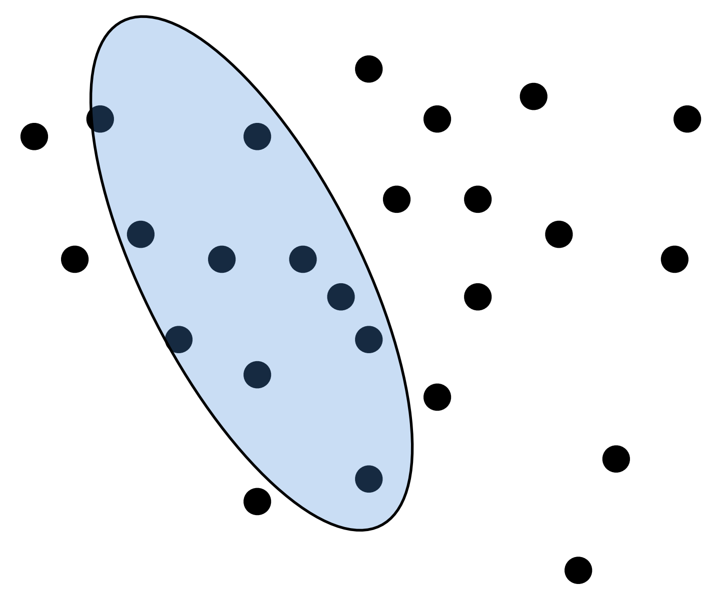
$$\mathcal{U}^{\rm st}(\theta) = \{\hat{\mu} + \hat{\Sigma}^{1/2} u \mid \|u\|_2 \leq \rho\} = \{u \mid \|A^{\rm st} u + b^{\rm st}\|_2 \leq \rho\},$$



empirical
mean and covariance

Reshaped uncertainty set

$$\mathcal{U}^{\rm re}(\theta) = \{\|A^{\rm re} u + b^{\rm re}\|_2 \leq \rho\}$$

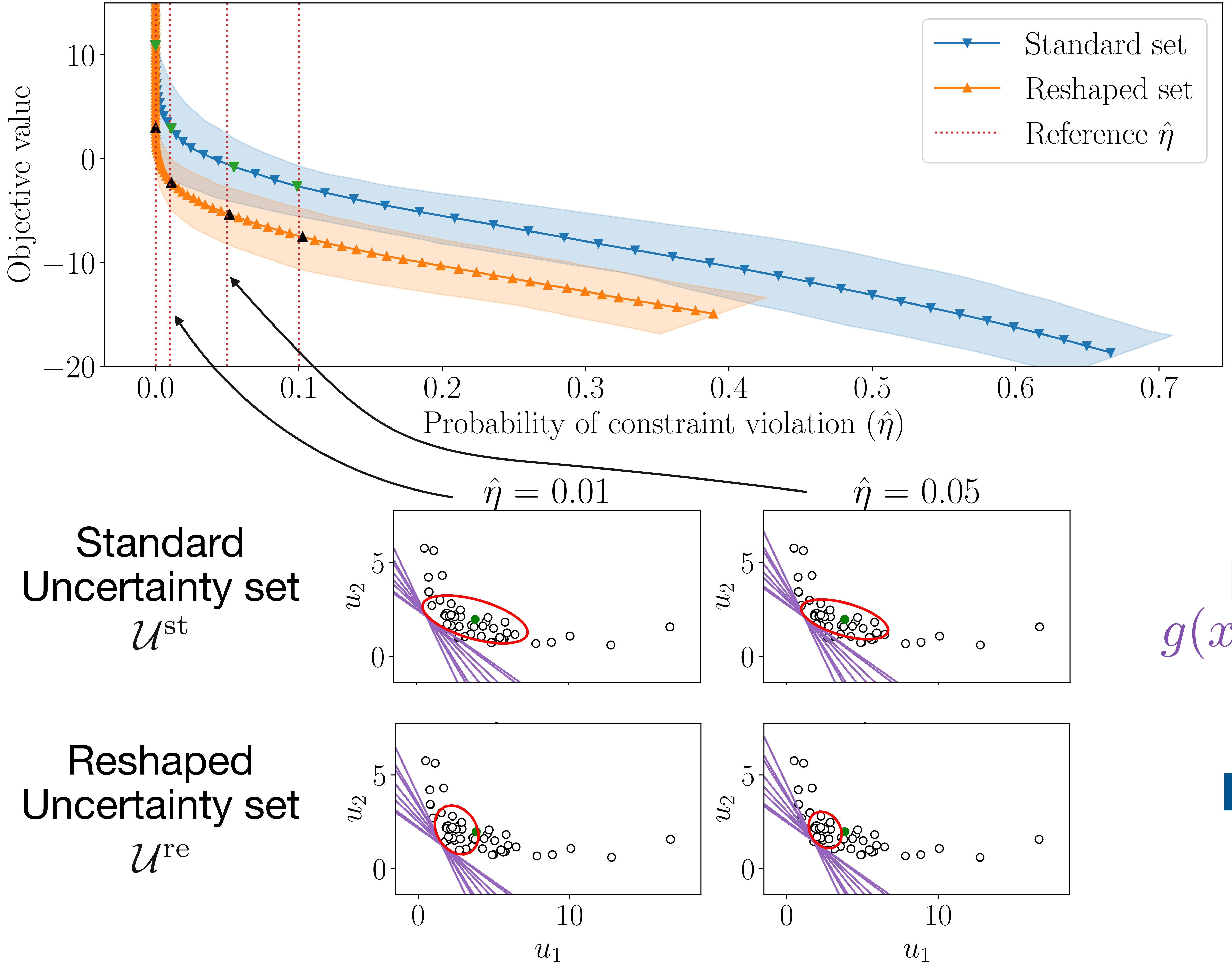


How do we pick the shape and size?

Can the reshaped set be better?

Reshaped set does not have high coverage…
…but performs much better!

Pareto curves for varying size ρ



Standard Uncertainty set $\mathcal{U}^{\text{st}}$

Reshaped Uncertainty set $\mathcal{U}^{\text{re}}$

level curves $g(x(\theta, \bar{y}), u, \bar{y}) = 0$

How can we find this set?

Our parametrization of the uncertainty set shape and size

$$\begin{array}{ll} \text{minimize} & f(x,y) \\ \text{subject to} & g(x,u,y) \leq 0 \quad \forall u \in \mathcal{U}(\theta) \end{array}$$

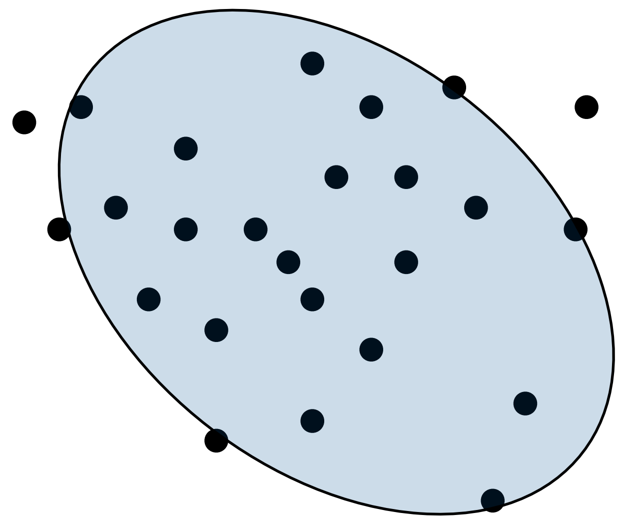
max of concave

key parameters

Shape and size of ellipsoidal set

$$\mathcal{U}_{\text{ellip}}(\theta) = \{\|Au + b\|_2 \leq 1\}$$

$$\theta = (A, b)$$



Linear constraint

$$g(x,u,y) = u^T x \leq 0, \quad \forall u \in \mathcal{U}_{\text{ellip}}(\theta)$$

Robust counterpart

$$-b^T y + \|\lambda\|_2 \leq 0$$
$$A^T \lambda = x$$

Enforcing probabilistic guarantee with CVaR

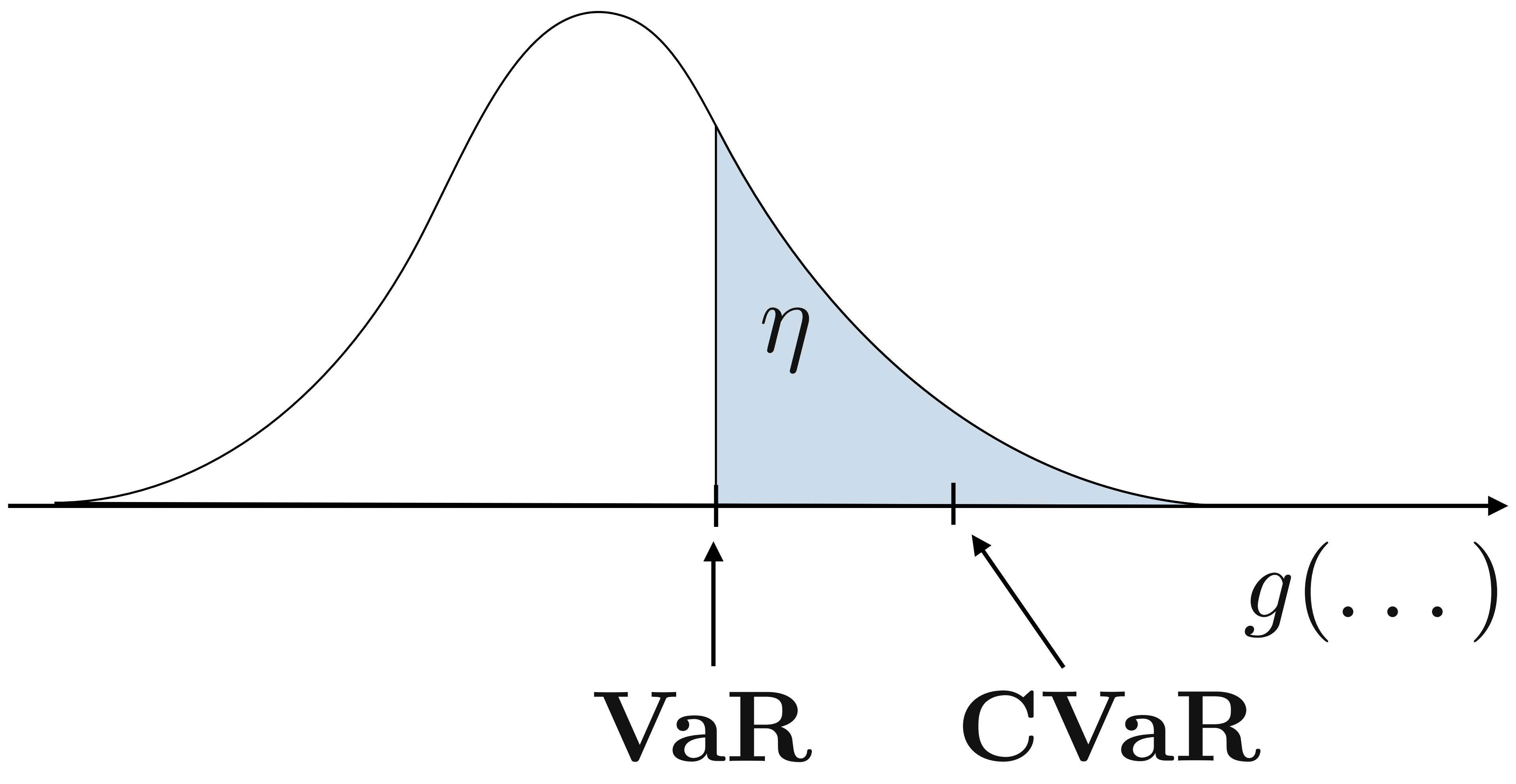
Target guarantee $(\mathbf{VaR}(g(\dots), \eta) \leq 0)$

$$\mathbf{P}_{(u,y)}(g(x(\theta, y), u, y) \leq 0) \geq 1 - \eta$$

over both u and y

Tractable approximation

$$\mathbf{CVaR}(g(x(\theta, y), u, y), \eta) \leq 0$$



Turn into constraint

$$\mathbf{E}_{(u,y)}\left(\frac{g(x(\theta, y), u, y) - \alpha)_+}{\eta} + \alpha\right) \leq 0 \longrightarrow \mathbf{E}_w\left(h(z, w)\right) = \kappa$$

Make it tight

$z = (\theta, \alpha)$

$w = (u, y)$

threshold < 0

Learning the uncertainty sets using stochastic bilevel optimization

training problem

$$\begin{array}{ll} \text{minimize} & \mathbf{E}_w[\ell(z,w)] \\ \text{subject to} & \mathbf{E}_w[h(z,w)] = \kappa \end{array}$$

Loss

$$f(x(\theta, y), y)$$

CVaR constraint

$$\frac{(g(x(\theta, y), u, y) - \alpha)_+}{\eta} + \alpha$$

decision variables

$$z = (\theta, \alpha)$$

random variables

$$w = (u, y)$$

inner robust problem

$$\begin{array}{lll} x(\theta, y) \in \text{argmin} & f(x, y) & \\ \text{subject to} & g(x, u, y) \leq 0 & \forall u \in \mathcal{U}(\theta) \end{array}$$

Learning using Stochastic Augmented Lagrangian algorithm

training problem

$F(z)$

$$\begin{array}{ll} \text{minimize} & \mathbf{E}_w[\ell(z,w)] \\ \text{subject to} & \mathbf{E}_w[h(z,w)] = \kappa \end{array}$$

$H(z)$

Converges to an
ϵ-KKT solution in
$O((1/\epsilon^3)\log(1/\epsilon))$
inner iterations

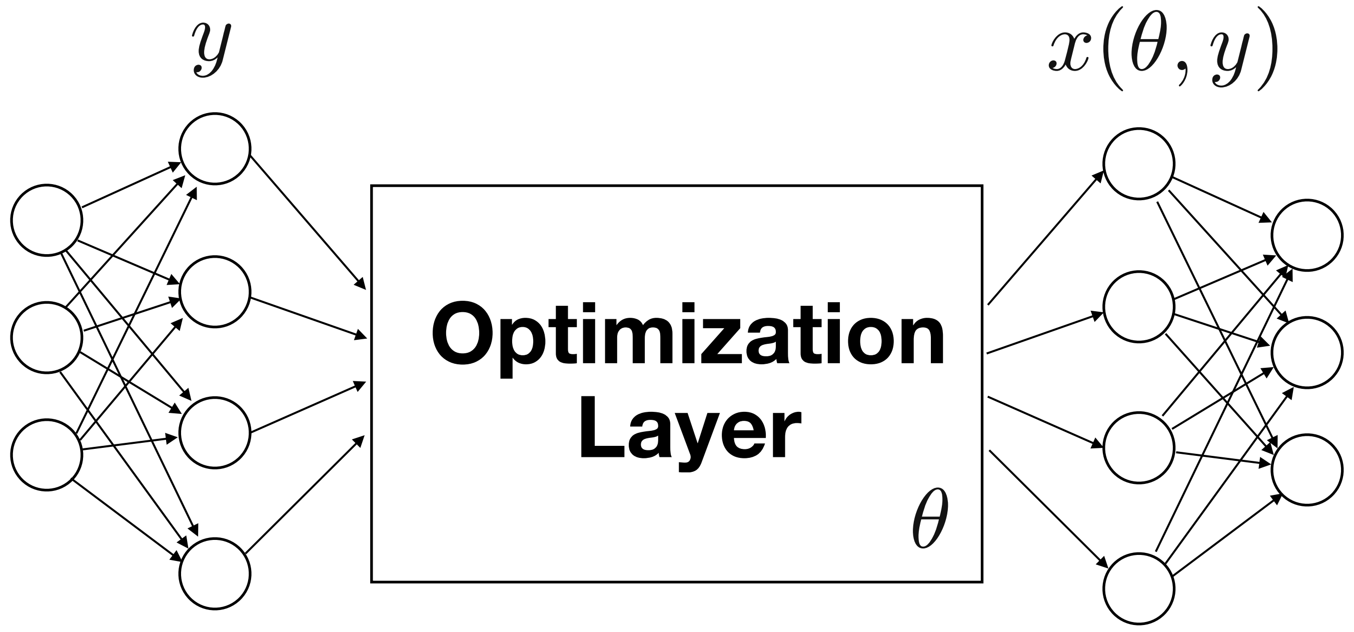
augmented Lagrangian

$$L(z,\lambda,\mu) = F(z) + \lambda(H(z)-\kappa) + \frac{\mu}{2}(H(z)-\kappa)^2$$

stochastic gradient computation

$$\hat{\nabla}_z L(z,\lambda,\mu)$$

Differentiate inner problem optimality conditions



“Differentiable Convex Optimization Layers”, A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Zico Kolter (2019)
“Differentiating Through a Conic Program”, A. Agrawal, S. Barratt, S. Boyd, E. Busseti, W. M. Moursi (2019)

Main idea to get finite-sample probabilistic guarantees

optimality tolerance → target confidence → dimension of x → number of samples of u

Pick $k(\epsilon, \beta, n, N) < 0$ such that

threshold constraint

$$\mathbf{E}_w\left(h(z,w)\right) = \kappa$$

Implies

Finite-sample probabilistic guarantee

$$\mathbf{P}^{N\times J}\left(\mathbf{P}_{(u,y)}(g(x,u,y) \leq 0) \geq 1-\eta \quad \forall x \in \mathcal{X}\right) \geq 1-\beta$$

$\Rightarrow$ it holds also for $x(\theta, y)$

Ingredients

- Tail bounds
- $\mathbf{CVaR} \geq \mathbf{VaR}$

[A. van der Vaart and J.A. Wellner. Weak Convergence and Empirical Processes (1996), Thm. 2.14.9]

LROPT package (coming soon!)

It can be ***hard to dualize*** robust optimization problems

…not to mention ***finding the right uncertainty set***!

LROPT

github.com/stellatogrp/lropt

1. Easily formulate and dualize robust optimization problems
2. Automatically tune uncertainty sets (using cvxpylayers)

$$\begin{array}{ll} \text{minimize} & x^T P x + q^T x \\ \text{subject to} & u^T x \leq b, \qquad \forall u \in \mathcal{U}(\theta) \end{array}$$

$$\mathcal{U} = \{u \mid \|Au + b\|_2 \leq 1\}$$

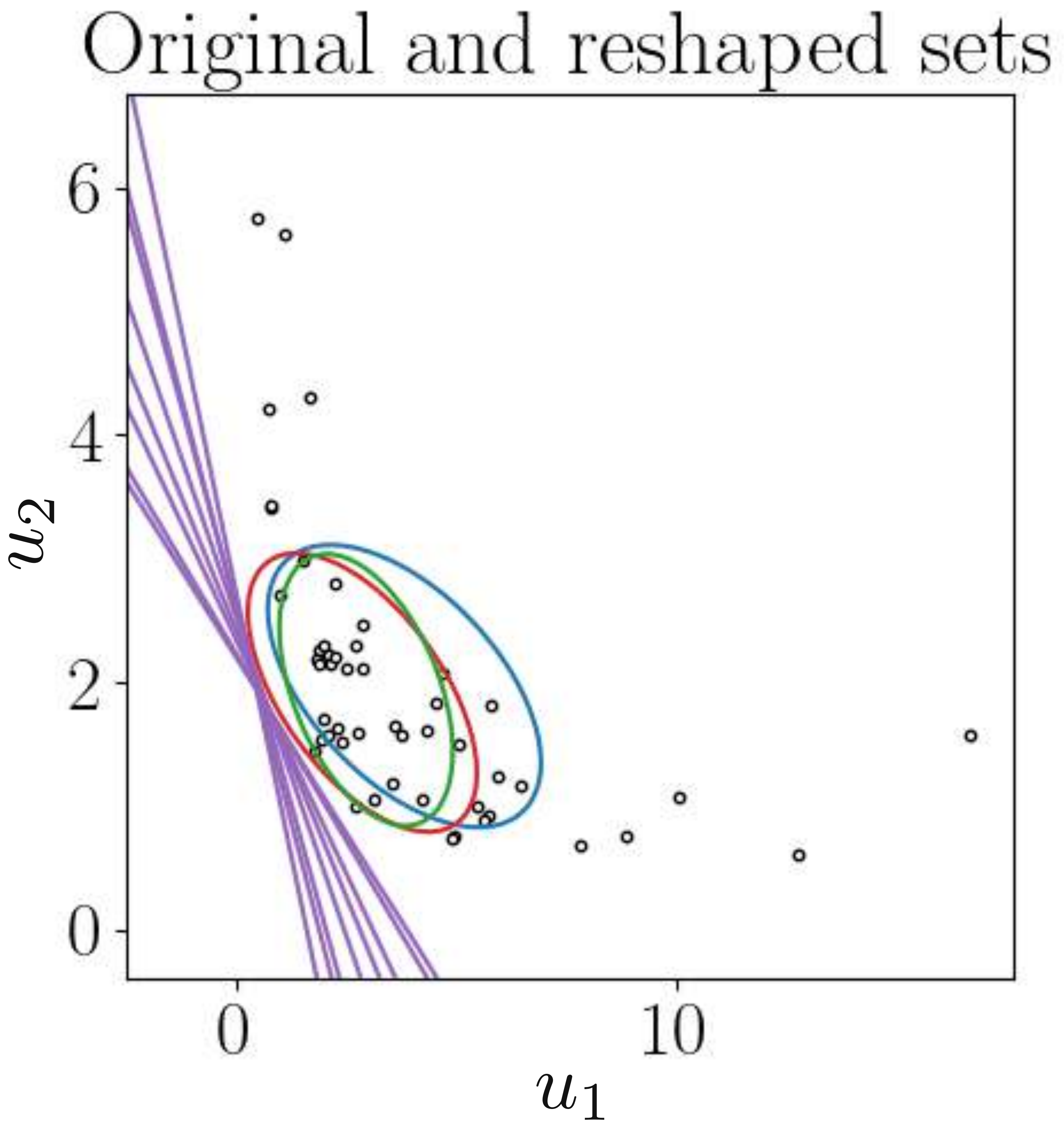
```
x = cp.Variable(n)
u = cp.UncertainParameter(n,
                          uncertainty_set=lr.Ellipsoidal(),
                          data=u_data)

constraints = [u @ x <= b]
objective = cp.Minimize(cp.quad_form(P, x) + q @ x)
problem = lr.RobustProblem(objective, constraints)
problem.train()
```

Training effect on newsvendor example

Robust Reformulation

$$\begin{array}{ll} \text{minimize} & \tau \\ \text{subject to} & k^T x + \max\{-p^T x, -p^T u\} \leq \tau \quad \forall u \in \mathcal{U}(\theta) \\ & x \geq 0 \end{array}$$

family parameter $y = (k, p)$



constraint level curves

(for fixed values $\bar{y}$)

Multi-period inventory management

transportation and holding costs

stocking decisions

sale prices (family parameter)

sales decisions

$$
\begin{array}{ll}
\text{minimize} & \tau \\
\text{subject to} & (t+h)^T s - r^T w(u) \leq \tau, \quad \forall u \in \mathcal{U}(\theta) \\
& w(u) \leq s, \quad \forall u \in \mathcal{U}(\theta) \\
& w(u) \leq \bar{d} + Qu, \quad \forall u \in \mathcal{U}(\theta) \\
& \mathbf{1}^T s = C \\
& 0 \leq s \leq c
\end{array}
$$

demand

Two-stage adjustable optimization

(linear decision rules for $w(u)$)

“Pareto Efficiency in Robust Optimization” D. A. Iancu, N. Trichakis (2014)

Inventory management comparisons with RO and DRO

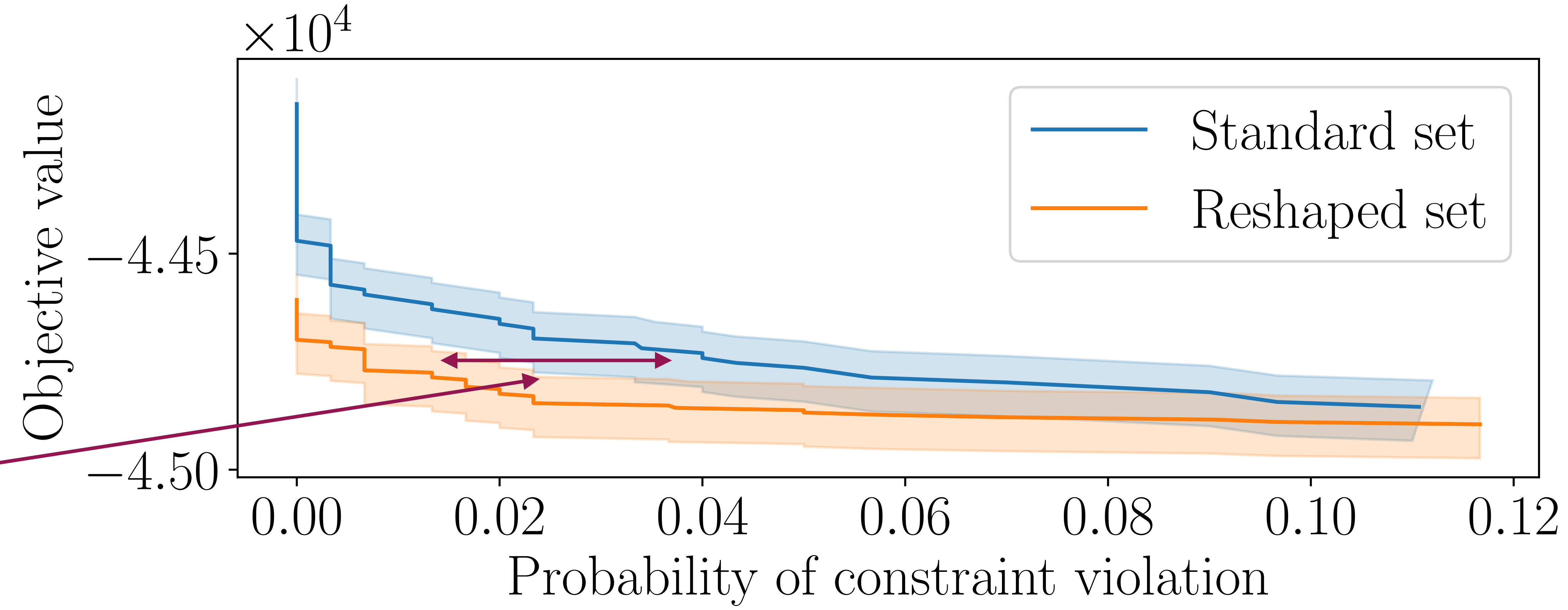
Results on test data

stronger guarantees

Method	LROPT	RO	Wass. DRO
Obj.	-4.477×10^4	-4.471×10^4	-4.465×10^4
$\hat{\eta}$	7.30×10^{-3}	3.51×10^{-2}	2.89×10^{-2}
$\hat{\beta}$	0	1.89×10^{-1}	2.00×10^{-2}
Time	2.73×10^{-3}	2.76×10^{-3}	4.17×10^{-1}

faster than DRO

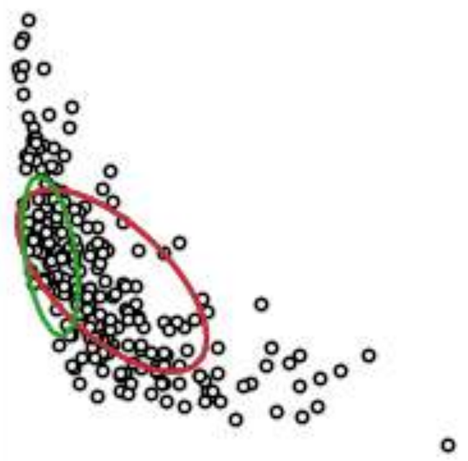
Resizing sets with fixed shape



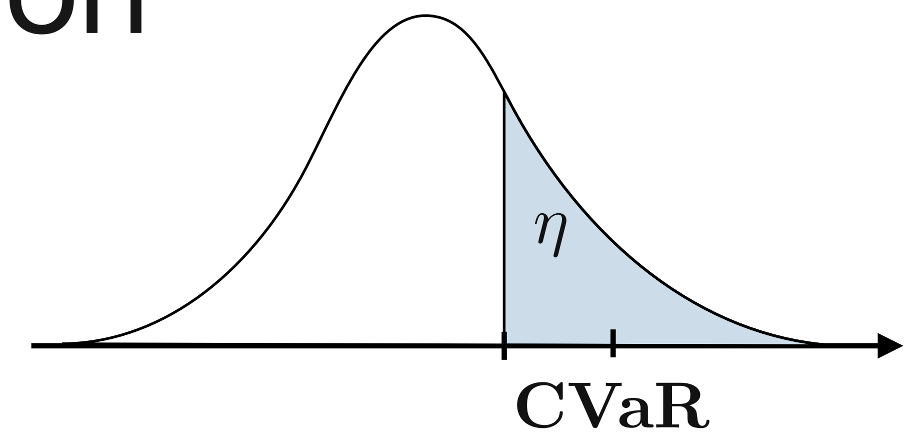
Reduction in constraint violation

Learning Uncertainty sets in Robust Optimization

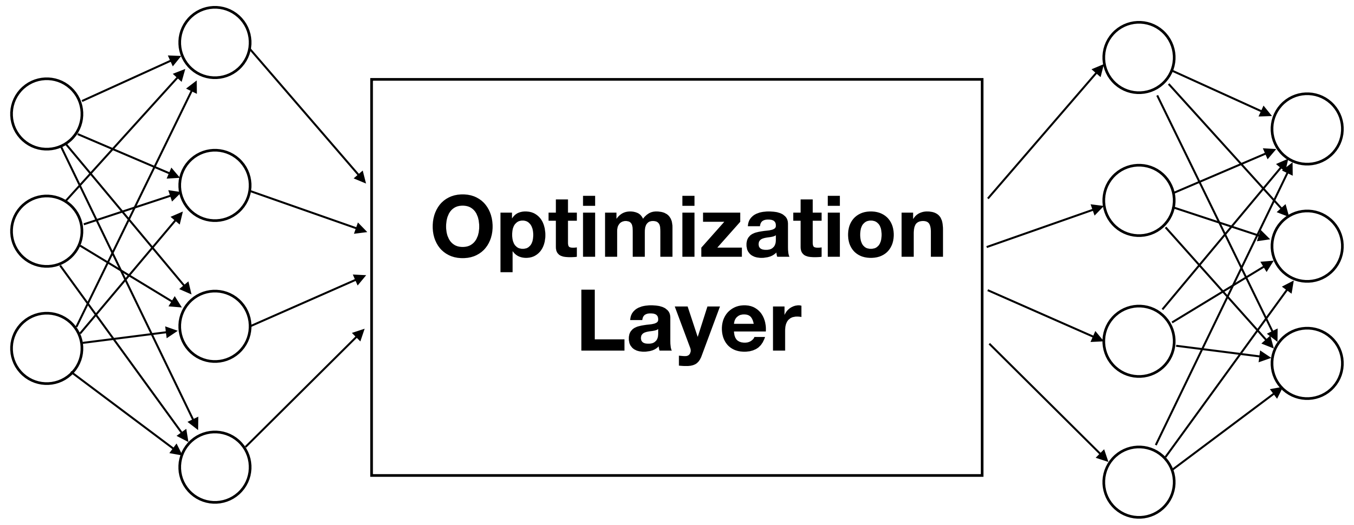
- Optimize **shape and size** of uncertainty sets



- **Bi-level optimization** formulation
    - CVaR as equality constraint



    - Differentiable optimization to compute derivatives



    - Probabilistic guarantees

- **Improvements over RO and DRO formulations**

Acknowledgements

Irina Wang

Princeton

Cole Becker

Princeton

Bart Van Parys

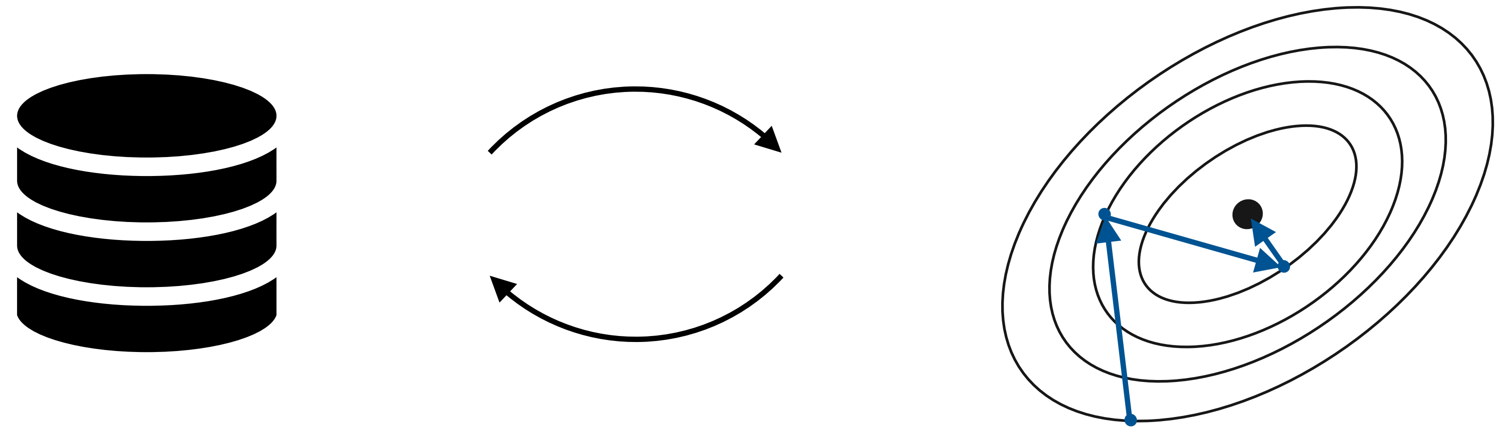
MIT

Conclusion

Machine Learning tools
can help us
formulate optimization problems

We should think
building robust optimization models
as an (automated)
training/validation procedure

Data　　　　Optimization



https://github.com/stellatogrp/lropt

arXiv **Learning for Robust Optimization**
I. Wang, C. Becker, B. Van Parys, and B. Stellato
arxiv.org: 2305.19225, 2023

@b_stellato
bstellato@princeton.edu
stellato.io