Foundations and Recent Developments in Scenario Optimization

Giuseppe Carlo Calafiore

Dipartimento di Automatica e Informatica
Politecnico di Torino – ITALY

PGMO Days – Paris, Nov. 8, 2016

Summary of contents

RCP theory
Introduction

- Random convex programs (RCPs) are convex optimization problems subject to a finite number of constraints (scenarios) that are extracted according to some probability distribution.
- The optimal objective value of an RCP and its associated optimal solution (when it exists), are random variables: RCP theory is mainly concerned with providing probabilistic assessments on the objective and on the probability of constraint violation for RCPs.
- We give a synthetic overview of RCP theory.
- Discuss impact and some applicative examples.

RCP theory

Preliminaries

- A finite-dimensional convex optimization problem

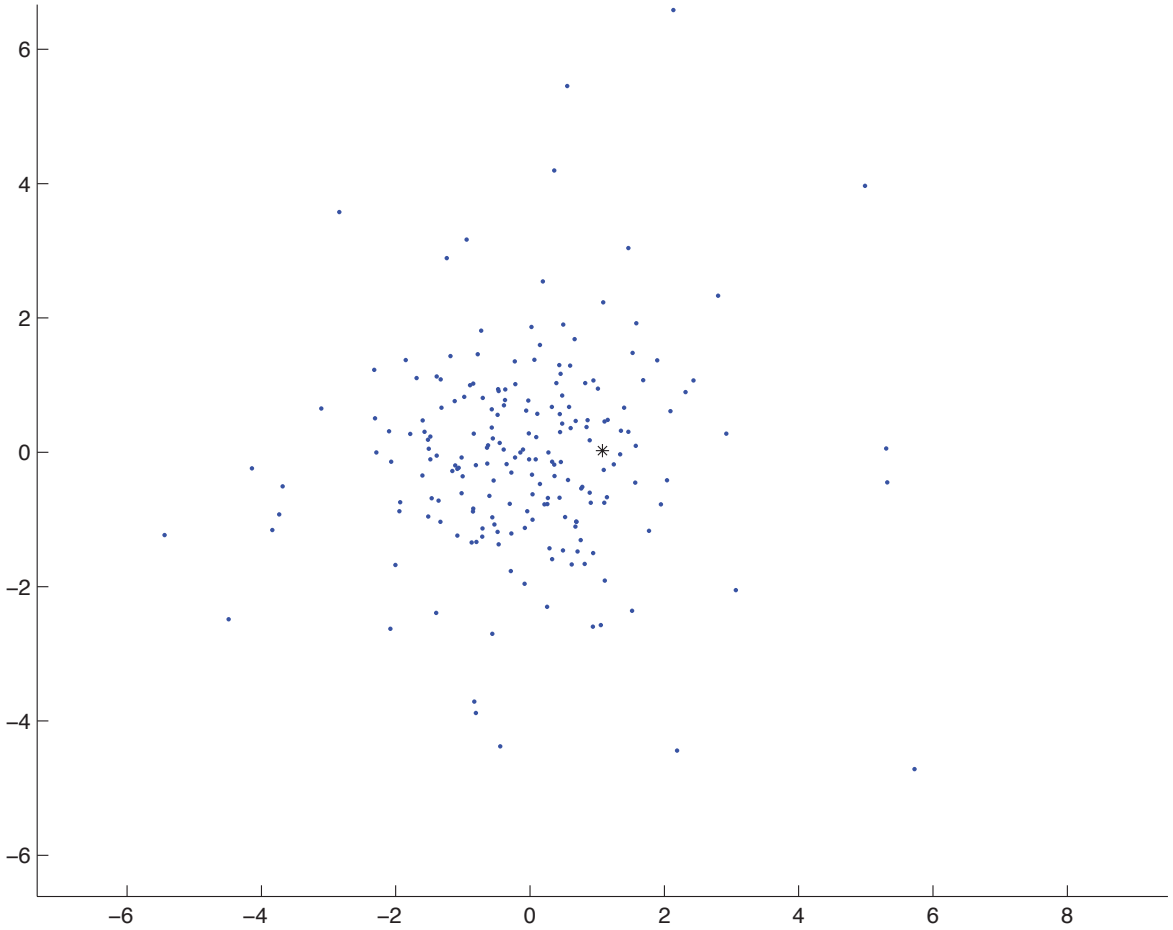
$$P[K]: \quad \min_{x \in \mathcal{X}} \; c^\top x \quad \text{subject to:} \qquad (1)$$
$$f_j(x) \leq 0, \qquad \forall j \in K,$$

  $x \in \mathcal{X}$ is the optimization variable, $\mathcal{X} \subset \mathbb{R}^d$ is a compact and convex domain, $c \neq 0$ is the objective direction, K is a *finite* set of indices, and $f_j(x) : \mathbb{R}^d \to \mathbb{R}$ are convex in x for each $j \in K$.

- Each constraint thus defines a convex set $\{x : f_j(x) \leq 0\}$.
- We denote with $\mathrm{Opx}[K]$ an optimal solution of problem $P[K]$, whenever it exists, and with $\mathrm{Obj}[K]$ the optimal objective.
- If $P[K]$ is unfeasible, we assume by convention that $\mathrm{Obj}[K] = +\infty$;

A model paradigm

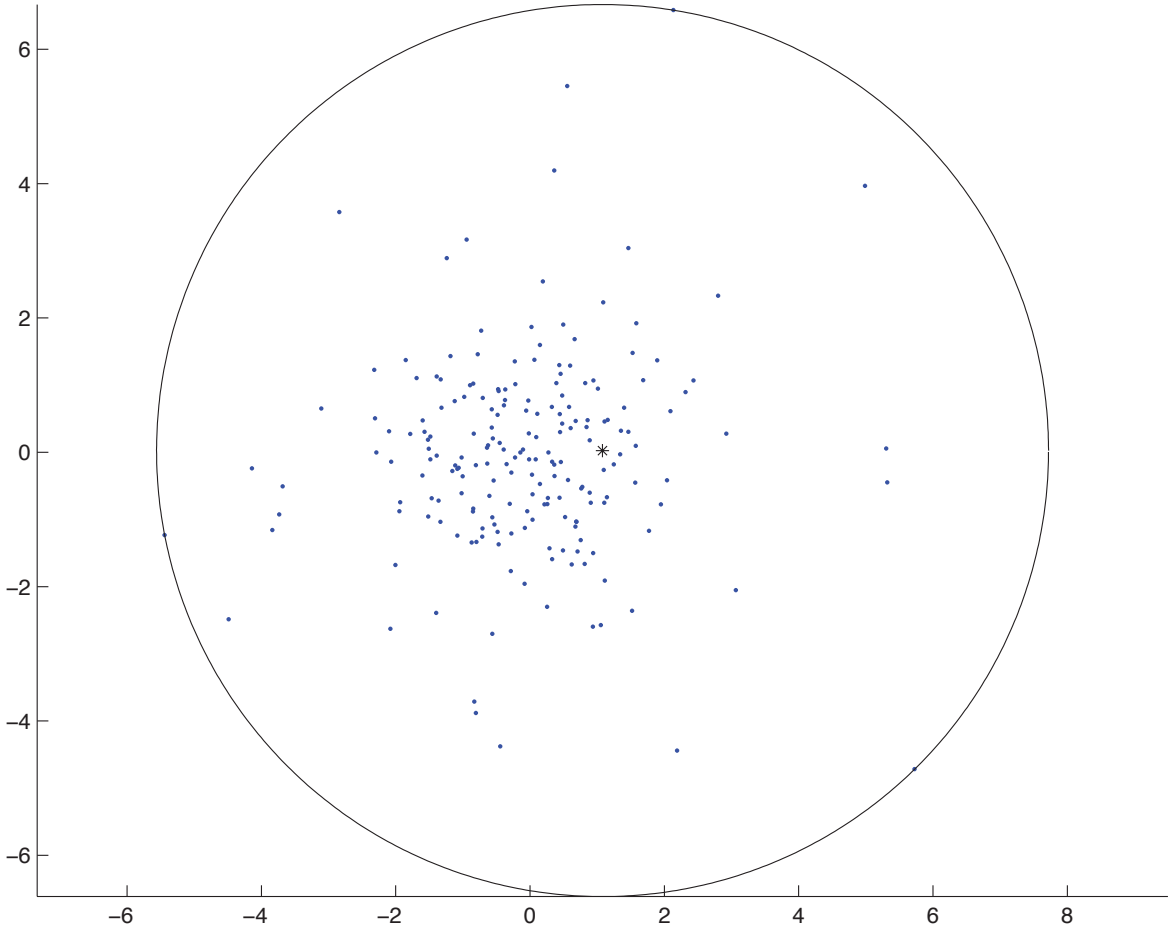
See N points



A model paradigm

See N points

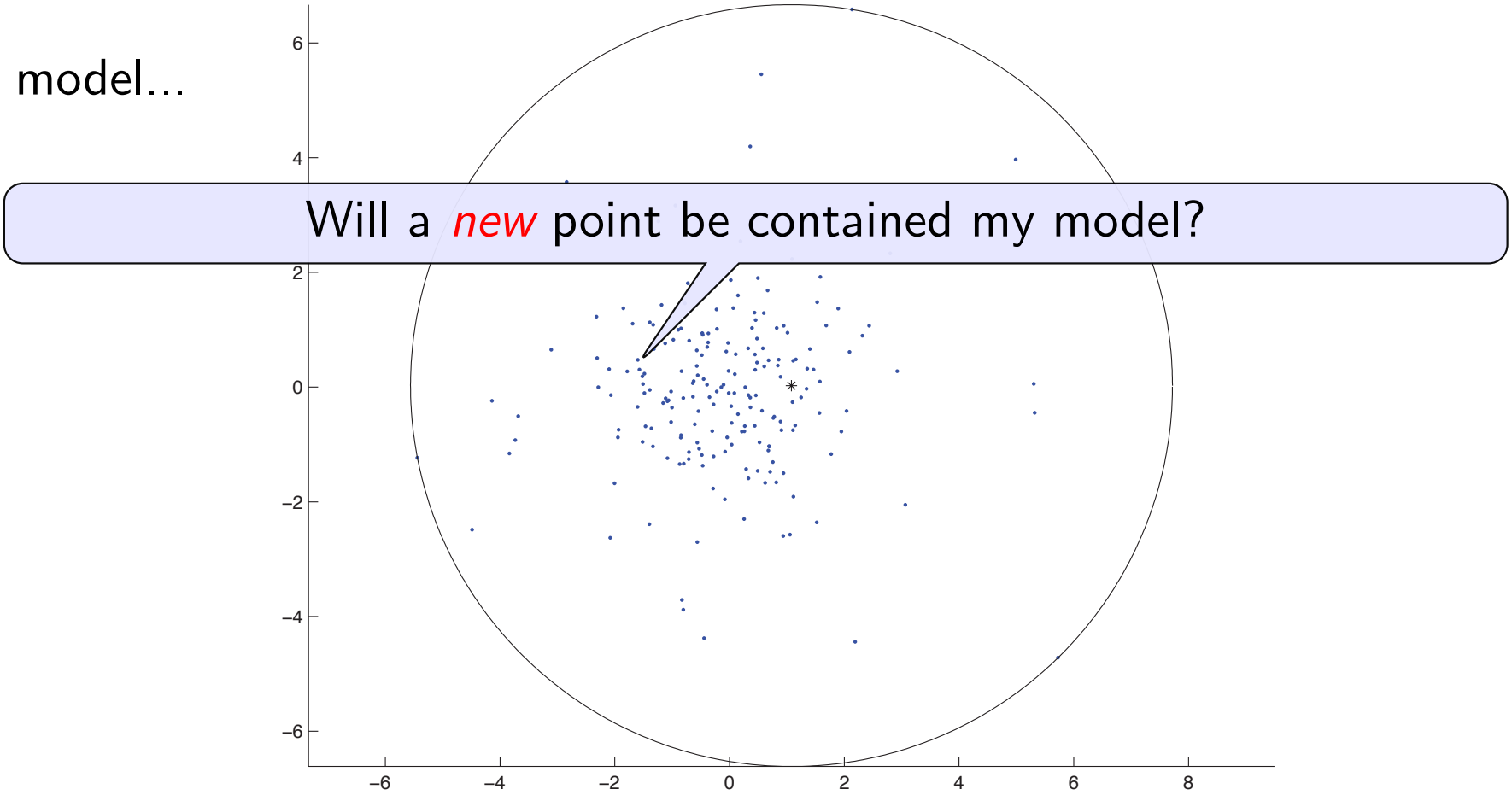
Fit model...



A model paradigm

See N points

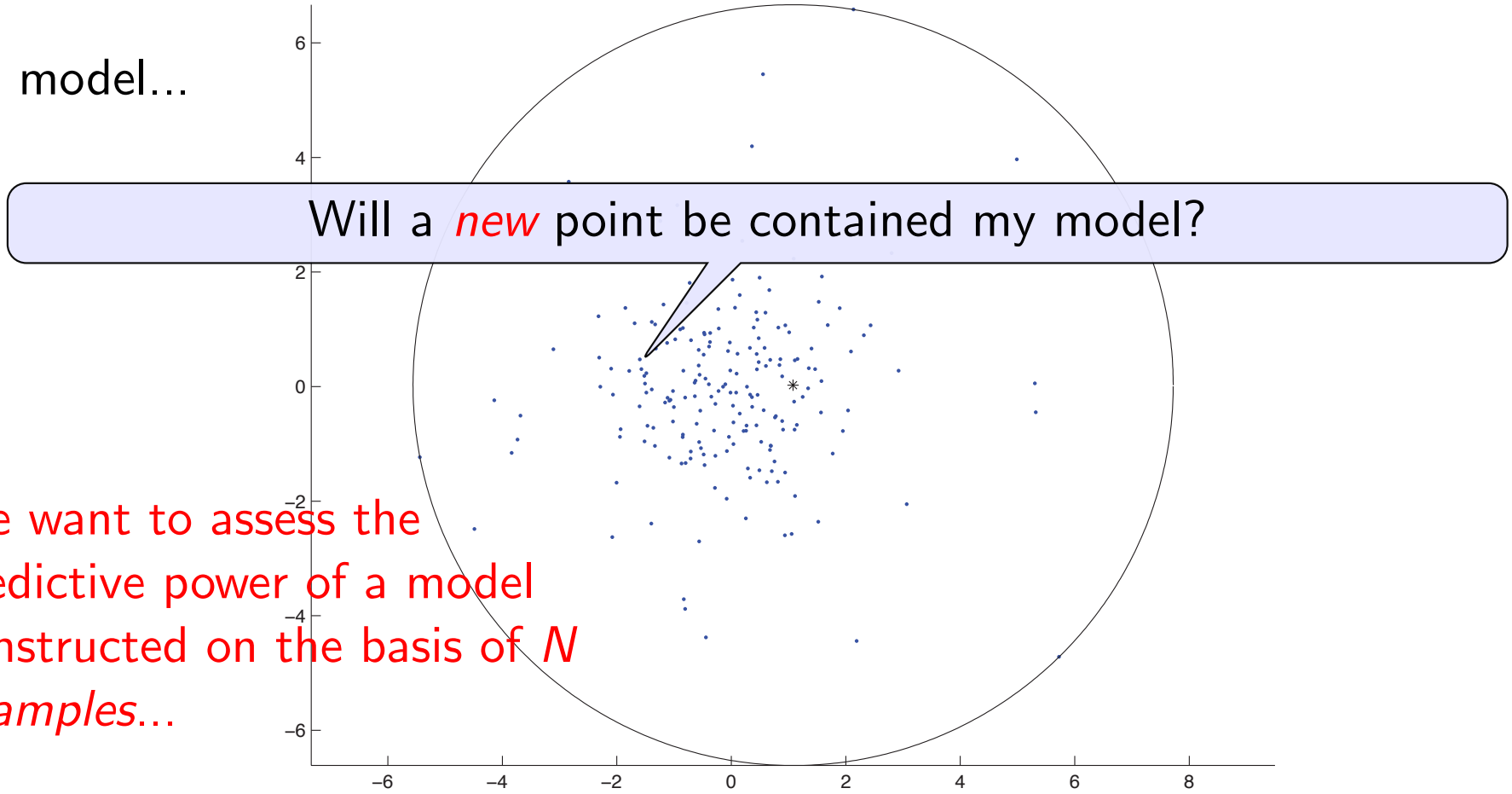
Fit model...



A model paradigm

See N points

Fit model...



We want to assess the predictive power of a model constructed on the basis of N *examples*...

RCP theory
Formalization

- Let $\delta \in \Delta$ denote a vector of random parameters, with $\Delta \subseteq \mathbb{R}^{\ell}$, and let $\mathbb{P}$ be a probability measure on Δ.
- Let $x \in \mathbb{R}^d$ be a design variable, and consider a family of functions $f(x, \delta) : (\mathbb{R}^d \times \Delta) \to \mathbb{R}$ defining the design constraints and parameterized by δ.

  Specifically, for a given design vector x and realization δ of the uncertainty, the design constraint are satisfied if $f(x, \delta) \leq 0$.

Assumption (convexity)

The function $f(x, \delta) : (\mathbb{R}^d \times \Delta) \to \mathbb{R}$ *is* convex *in* x*, for each fixed* $\delta \in \Delta$*.*

RCP theory
Formalization

- Define
$$\omega \doteq (\delta^{(1)}, \ldots, \delta^{(N)}) \in \Delta^N,$$
where $\delta^{(i)} \in \Delta$, $i = 1, \ldots, N$, are independent random variables, identically distributed (iid) according to $\mathbb{P}$, and where $\Delta^N = \Delta \times \Delta \cdots \Delta$ (N times).
- Let $\mathbb{P}^N$ denote the product probability measure on Δ^N.
- To each $\delta^{(j)}$ we associate a constraint function
$$f_j(x) \doteq f(x, \delta^{(j)}), \quad j = 1, \ldots, N.$$
Therefore, to each randomly extracted ω there correspond N random constraints $f_j(x)$, $j = 1, \ldots, N$.

RCP theory
Formalization

- Given $\omega = (\delta^{(1)}, \ldots, \delta^{(N)}) \in \Delta^N$ we define the following convex optimization problem:

$$P[\omega]: \quad \min_{x \in \mathcal{X}} \; c^\top x \quad \text{subject to:} \qquad (2)$$
$$f_j(x) \leq 0, \quad j = 1, \ldots, N,$$

  where $f_j(x) = f(x, \delta^{(j)})$.

- For each random extraction of ω, problem (2) has the structure of a generic convex optimization problem $P[\omega]$, as defined in (1).
- We denote with $J^* = J^*(\omega) = \mathrm{Obj}(\omega)$ the optimal objective value of $P[\omega]$, and with $x^* = x^*(\omega) = \mathrm{Opx}[\omega]$ the optimal solution of problem (2), when it exists.
- Problem (2) is named a *random convex program* (RCP), and the corresponding optimal solution x^* is named a *scenario solution*.

RCP theory

Remarks on the generality of the model

Model (2) encloses a quite general family of uncertain convex programs.

- Problems with multiple uncertain (convex) constraints of the form

$$\begin{aligned} \min_{x\in\mathcal{X}} \; & c^\top x \quad \text{subject to:} \\ & f^{(1)}(x,\delta^{(j)}) \leq 0, \ldots, f^{(m)}(x,\delta^{(j)}) \leq 0; \\ & j = 1,\ldots,N, \end{aligned}$$

  can be readily cast in the form of (2) by condensing the multiple constraints in a single one:

$$f(x,\delta) \doteq \max_{i=1,\ldots,m} f^{(i)}(x,\delta).$$

- The case when the problem has an uncertain and nonlinear (but convex) objective function $g(x,\delta)$ can also be fit in the model by adding one slack decision variable t and reformulating the problem with linear objective t and an additional constraint $g(x,\delta) - t \leq 0$.

Violation probability

Definition (Violation probability)

The violation probability of problem $P[\omega]$ is defined as

$$V^*(\omega) \doteq \mathbb{P}\{\delta \in \Delta : \ J^*(\omega, \delta) > J^*(\omega)\}.$$

- To each random extraction of $\omega \in \Delta^N$ it corresponds a value of V^*, which is therefore itself a random variable with values in $[0, 1]$.
- For given $\epsilon \in (0, 1)$, let us define the "bad" event of having a violation larger than ϵ:
$$\mathcal{B} \doteq \{\omega \in \Delta^N : \ V^* > \epsilon\}$$
- We prove that it holds that $\mathbb{P}^N\{\mathcal{B}\} \leq \beta(\epsilon)$, for some explicitly given function $\beta(\epsilon)$ that goes to zero as N grows.
- In other words, if N is large enough, the scenario objective is a-priori guaranteed with probability at least $1 - \beta(\epsilon)$ to have violation probability smaller than ϵ.

RCP theory
Main result

Theorem

- *Consider problem (2), with* $N \geq d+1$. *Let*

$$V^*(\omega) \doteq \mathbb{P}\{\delta \in \Delta : J^*(\omega, \delta) > J^*(\omega)\}.$$

- *Then,*

$$\mathbb{P}^N\{\omega \in \Delta^N : V^*(\omega) > \epsilon\} \leq \Phi(\epsilon; d, N)$$

where

$$\Phi(\epsilon; d, N) \doteq \sum_{j=0}^{d} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}$$

is a beta cumulative distribution.

RCP theory
Main result

Theorem

- *Consider problem (2), with* $N \geq d+1$. *Let*

$$V^*(\omega) \doteq \mathbb{P}\{\delta \in \Delta : J^*(\omega, \delta) > J^*(\omega)\}.$$

- *Then,*

$$\mathbb{P}^N\{\omega \in \Delta^N : V^*(\omega) > \epsilon\} \leq \Phi(\epsilon; d, N)$$

where

$$\Phi(\epsilon; d, N) \doteq \sum_{j=0}^{d} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}$$

is a beta cumulative distribution.

The proof of this result is far from obvious...

Remark

$$\mathbb{P}^N\{\omega \in \Delta^N : V^*(\omega) > \epsilon\} \leq \Phi(\epsilon; d, N)$$

$$\Phi(\epsilon; d, N) \doteq \sum_{j=0}^{d} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}$$

This bound is UNIVERSAL:

- Does not depend on problem type (LP, QP, SDP, generic convex);
- Does not depend on the *distribution* law $\mathbb{P}$ of the uncertain parameters;
- Depends on the problem structure only via the *dimension*, d;
- Provides an explicit assessment on the violation probability tail, for *finite* N.

Learning-theoretic flavor: "training" on a finite batch of samples N provides a solution which is still optimal, with high probability, on a new, unseen, scenario...

Remark

$$\mathbb{P}^N\{\omega \in \Delta^N : V^*(\omega) > \epsilon\} \leq \Phi(\epsilon; d, N)$$

$$\Phi(\epsilon; d, N) \doteq \sum_{j=0}^{d} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}$$

This bound is UNIVERSAL:

- Does not depend on problem type (LP, QP, SDP, generic convex);
- Does not depend on the *distribution* law $\mathbb{P}$ of the uncertain parameters;
- Depends on the problem structure only via the *dimension*, d;
- Provides an explicit assessment on the violation probability tail, for *finite* N.

Learning-theoretic flavor: "training" on a finite batch of samples N provides a solution which is still optimal, with high probability, on a new, unseen, scenario...

Reversing the bound

Corollary

Given $\epsilon \in (0,1)$, $\beta \in (0,1)$. *If* N *is an integer such that*

$$N \geq \frac{2}{\epsilon}\left(\ln \beta^{-1} + d\right).$$

then it holds that

$$\mathbb{P}^N\{V^* > \epsilon\} \leq \beta.$$

Observe that β^{-1} is under a log: achieving small β is "cheap" in terms of the required number of samples N.

Reversing the bound

Corollary

Given $\epsilon \in (0,1)$, $\beta \in (0,1)$. *If* N *is an integer such that*

$$N \geq \frac{2}{\epsilon}\left(\ln \beta^{-1} + d\right).$$

then it holds that

$$\mathbb{P}^N\{V^* > \epsilon\} \leq \beta.$$

Observe that β^{-1} is under a log: achieving small β is "cheap" in terms of the required number of samples N.

A Practitioner's viewpoint

- Set β to a very small level, say $\beta = 10^{-10}$
- Bound becomes

$$N \geq \frac{2}{\epsilon}\left(21 + d\right).$$

- The event $\{V^* > \epsilon\}$ has vanishing probability $\leq 10^{-10}$, that is, the complementary event $\{V^* \leq \epsilon\}$ holds with *practical certainty*.
- Scenario optimization guarantees, with practical certainty, that

$$V^* \leq \epsilon.$$

- These statements are more easily understandable by engineers. The neglected event is so remote that before worrying about it the designer should better check many other simplifying assumptions and uncertainties on her model...

A Practitioner's viewpoint

- Set β to a very small level, say $\beta = 10^{-10}$
- Bound becomes

$$N \geq \frac{2}{\epsilon}\left(21 + d\right).$$

- The event $\{V^* > \epsilon\}$ has vanishing probability $\leq 10^{-10}$, that is, the complementary event $\{V^* \leq \epsilon\}$ holds with *practical certainty*.
- Scenario optimization guarantees, with practical certainty, that

$$V^* \leq \epsilon.$$

- These statements are more easily understandable by engineers. The neglected event is so remote that before worrying about it the designer should better check many other simplifying assumptions and uncertainties on her model...

A Practitioner's viewpoint

- Set β to a very small level, say $\beta = 10^{-10}$
- Bound becomes

$$N \geq \frac{2}{\epsilon}\left(21 + d\right).$$

- The event $\{V^* > \epsilon\}$ has vanishing probability $\leq 10^{-10}$, that is, the complementary event $\{V^* \leq \epsilon\}$ holds with *practical certainty*.
- Scenario optimization guarantees, with practical certainty, that

$$V^* \leq \epsilon.$$

- These statements are more easily understandable by engineers. The neglected event is so remote that before worrying about it the designer should better check many other simplifying assumptions and uncertainties on her model...

Let's see an example...

Example
Linear Support Vector Machine

- $z^{(i)} \in \mathbb{R}^n$, $i = 1, \ldots, N$, are n-dimensional training features (representing, e.g., patients, images, etc.).
- $y_i \in \{-1, +1\}$, $i = 1, \ldots, N$, are given labels for the observed features.
- The linear SVM is a supervised linear classifier: it seeks to separate the points with positive labels from the negative ones via a slab of maximum thickness.

Linear SVM

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \quad \|w\|_2^2$$
$$\text{s.t.:} \quad 1 - y_i(w^\top z^{(i)} + b) \leq 0 \quad i = 1, \ldots, N$$

Example
Linear Support Vector Machine

Linear SVM

$$\min_{w \in \mathbb{R}^n, b \in \mathbb{R}} \quad \|w\|_2^2$$
$$\text{s.t.:} \quad 1 - y_i(w^\top z^{(i)} + b) \leq 0 \quad i = 1, \ldots, N$$

- If the data set is *linearly separable*, all points with $y_i = +1$ lie in the halfspace

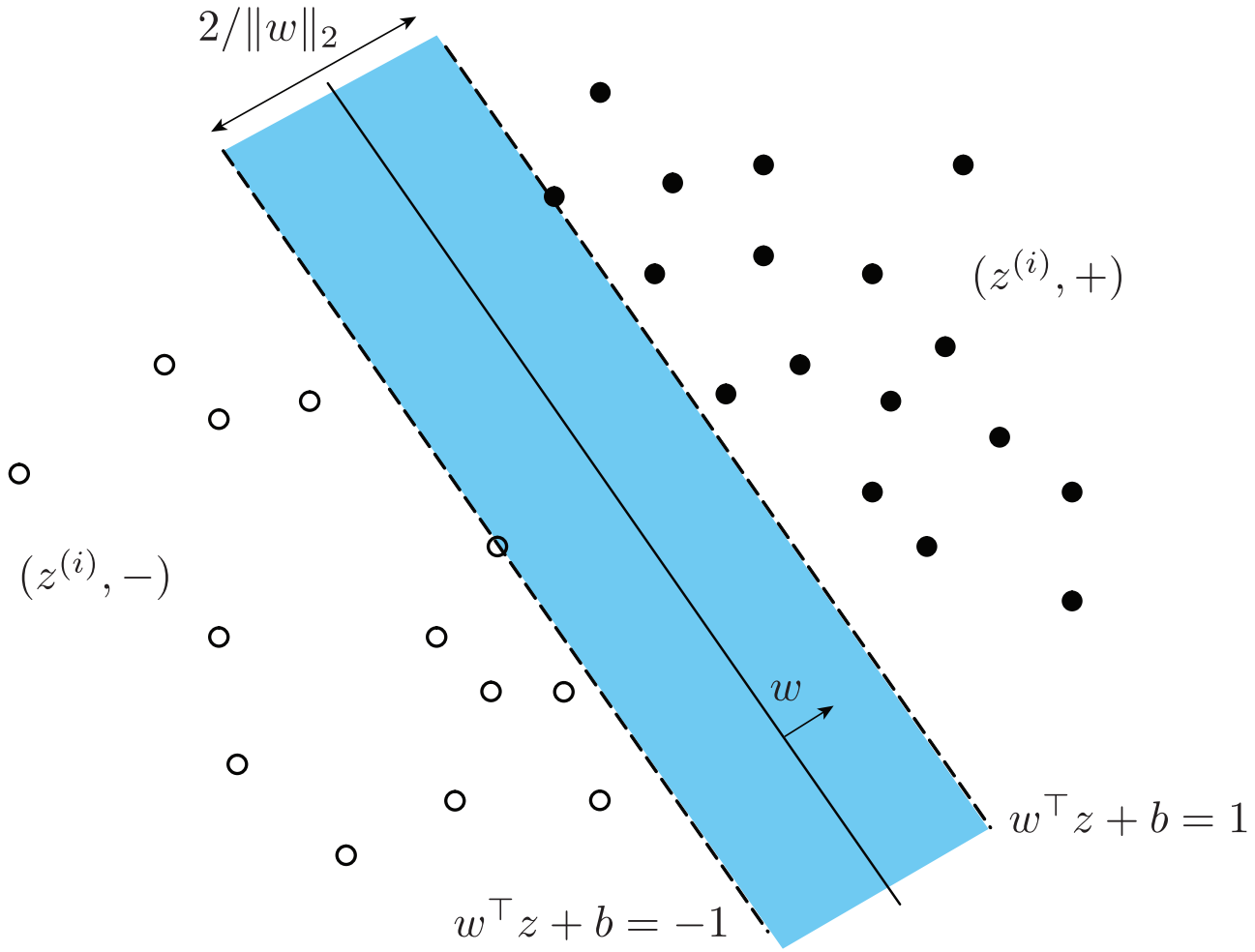
$$\mathcal{H}_+ = \{z \in \mathbb{R}^n : \ w^\top z + b \geq 1\}$$

while points with $y_i = -1$ lie in the halfspace

$$\mathcal{H}_- = \{z \in \mathbb{R}^n : \ w^\top z + b \leq -1\}$$

- The two halfspaces are separated by a *margin* of thickness $2/\|w\|_2$.

Example
Linear Support Vector Machine



Example
Linear Support Vector Machine

Linear SVM

$$\min_{w\in\mathbb{R}^n, b\in\mathbb{R}} \quad \|w\|_2^2$$
$$\text{s.t.:} \quad 1 - y_i(w^\top z^{(i)} + b) \leq 0 \quad i = 1, \ldots, N$$

- Decision variable is $x = (w, b)$, of dimension $d = n + 1$;
- Uncertainty parameters are $\delta^{(i)} = (z^{(i)}, y_i)$, $i = 1, \ldots, N$;
- Constraint function is $f(x, \delta) = 1 - y_i(w^\top z + b)$;
- Reliability is $R^* = 1 - V^* = \mathbb{P}\{(z, y) \text{ is correctly classified with margin}\}$.

Assuming the data is separable:
If $N \geq \frac{2}{\epsilon}(22 + n)$, then $R^* \geq 1 - \epsilon$, with practical certainty!

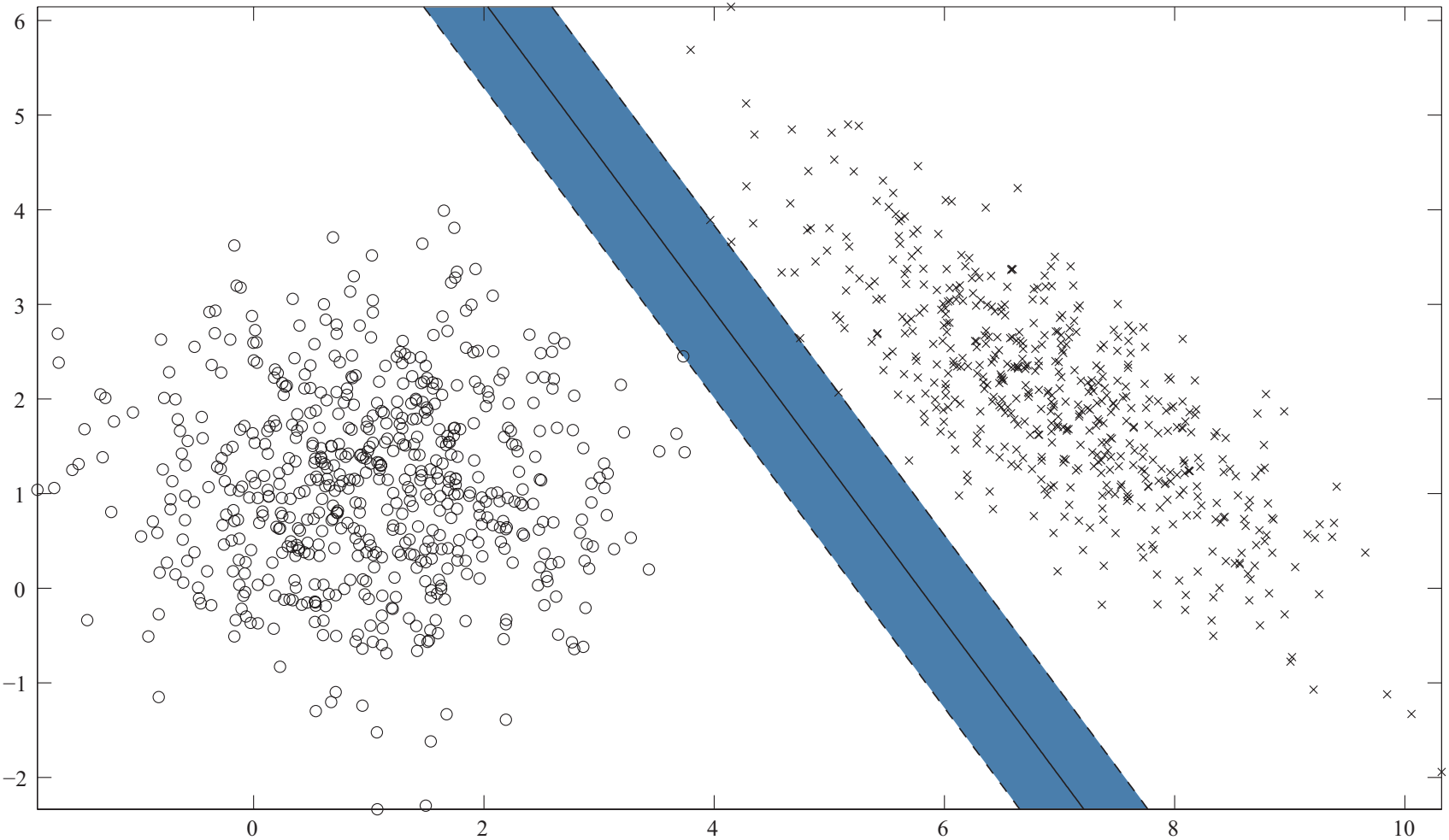
Example
Linear Support Vector Machine

- For a numerical example in $\mathbb{R}^2$, taking $\epsilon = 0.05$ we have that $N = 1000$ training samples suffice for guaranteeing an out-of-sample reliability $R^* \geq 95\%$, with practical certainty.
- Data:

$$\begin{cases} z = \begin{bmatrix} 7 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \xi, \ y = 1, & \text{with probability 0.5;} \\ z = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \xi, \ y = -1, & \text{with probability 0.5,} \end{cases}$$

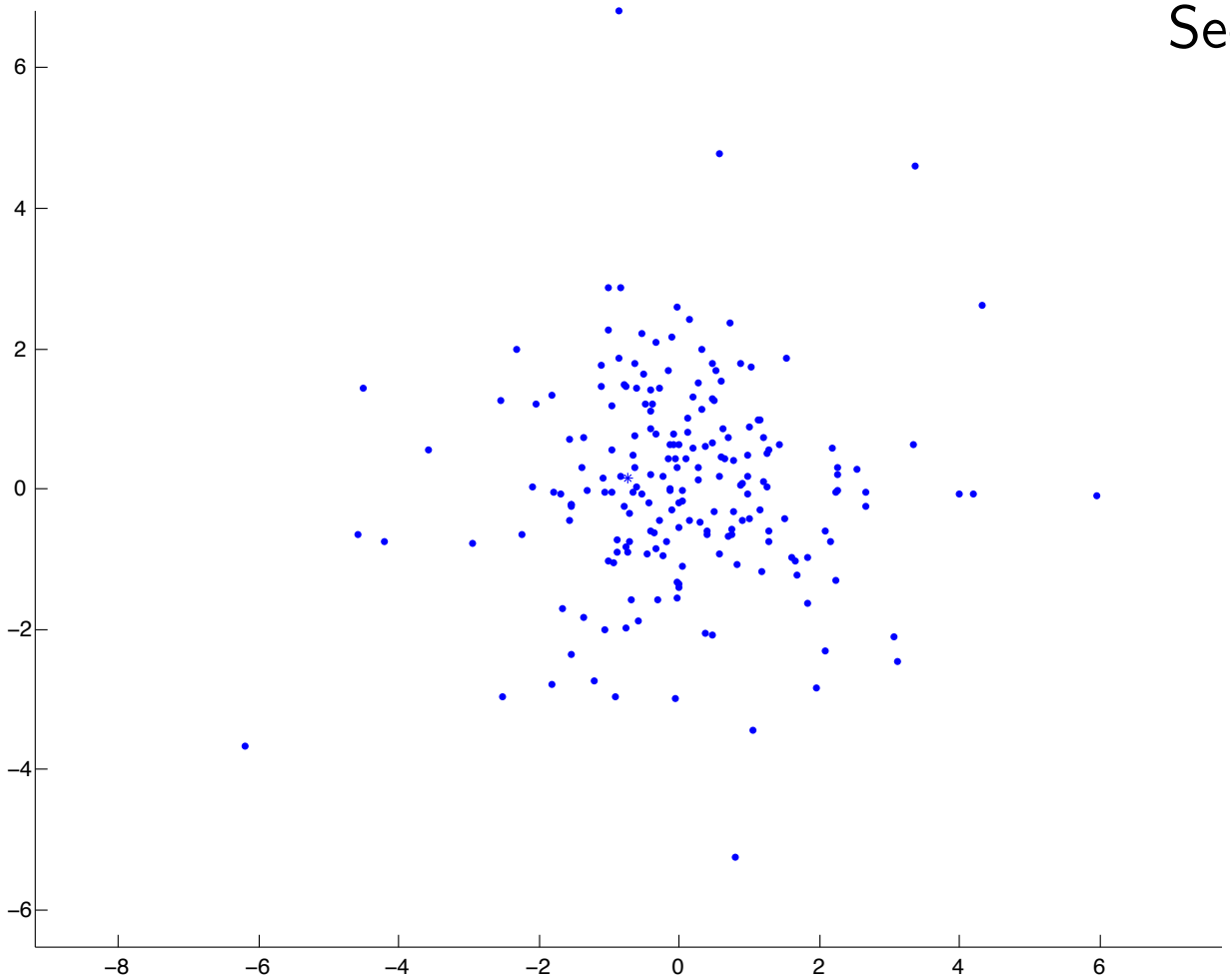
  where ξ is a standard Normal vector.
- We achieve separation $1/\|w^*\|_2 = 0.4753$.

Example
Linear Support Vector Machine



A step further

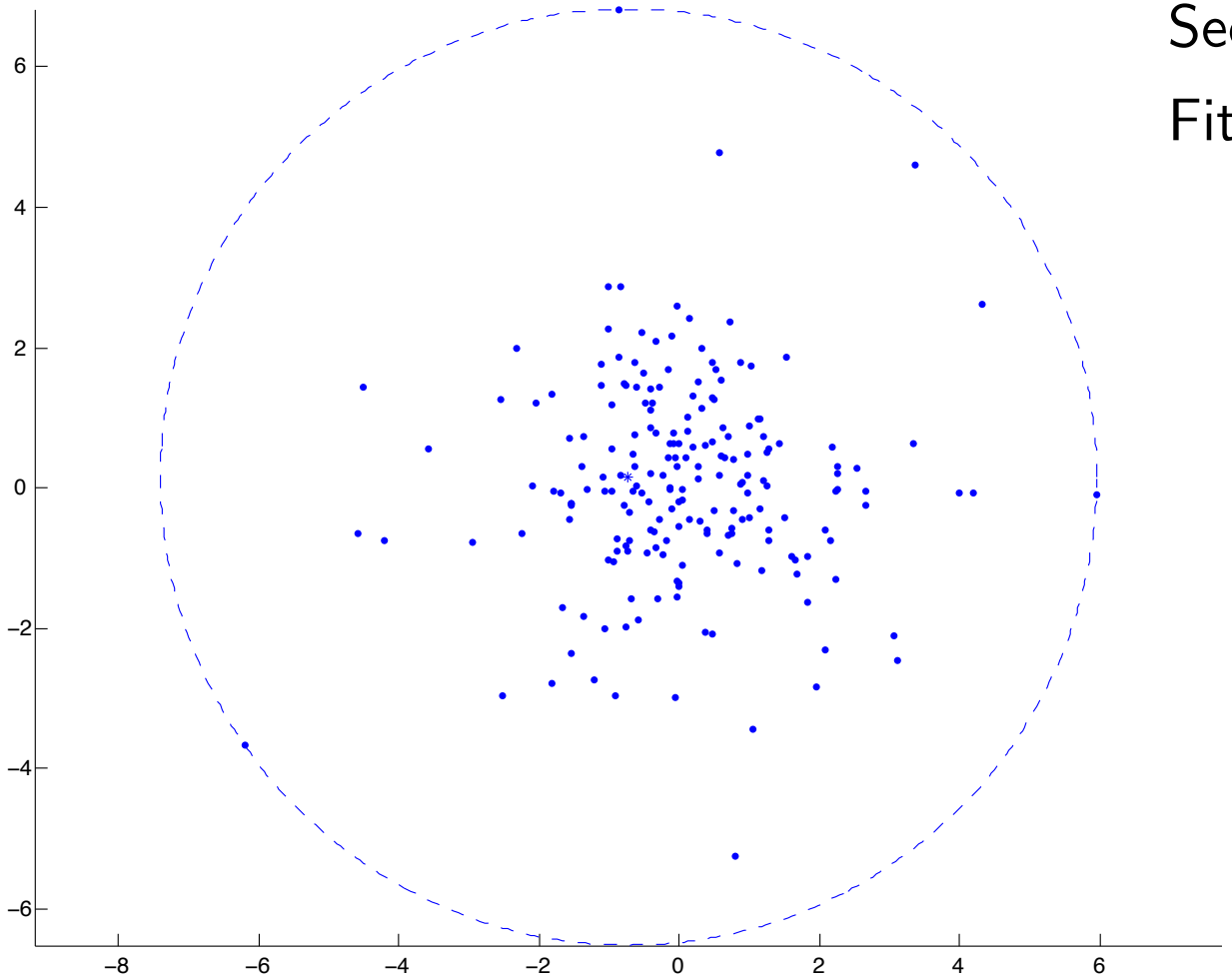
RCPs with violated constraints (RCPV)



See N points

A step further

RCPs with violated constraints (RCPV)

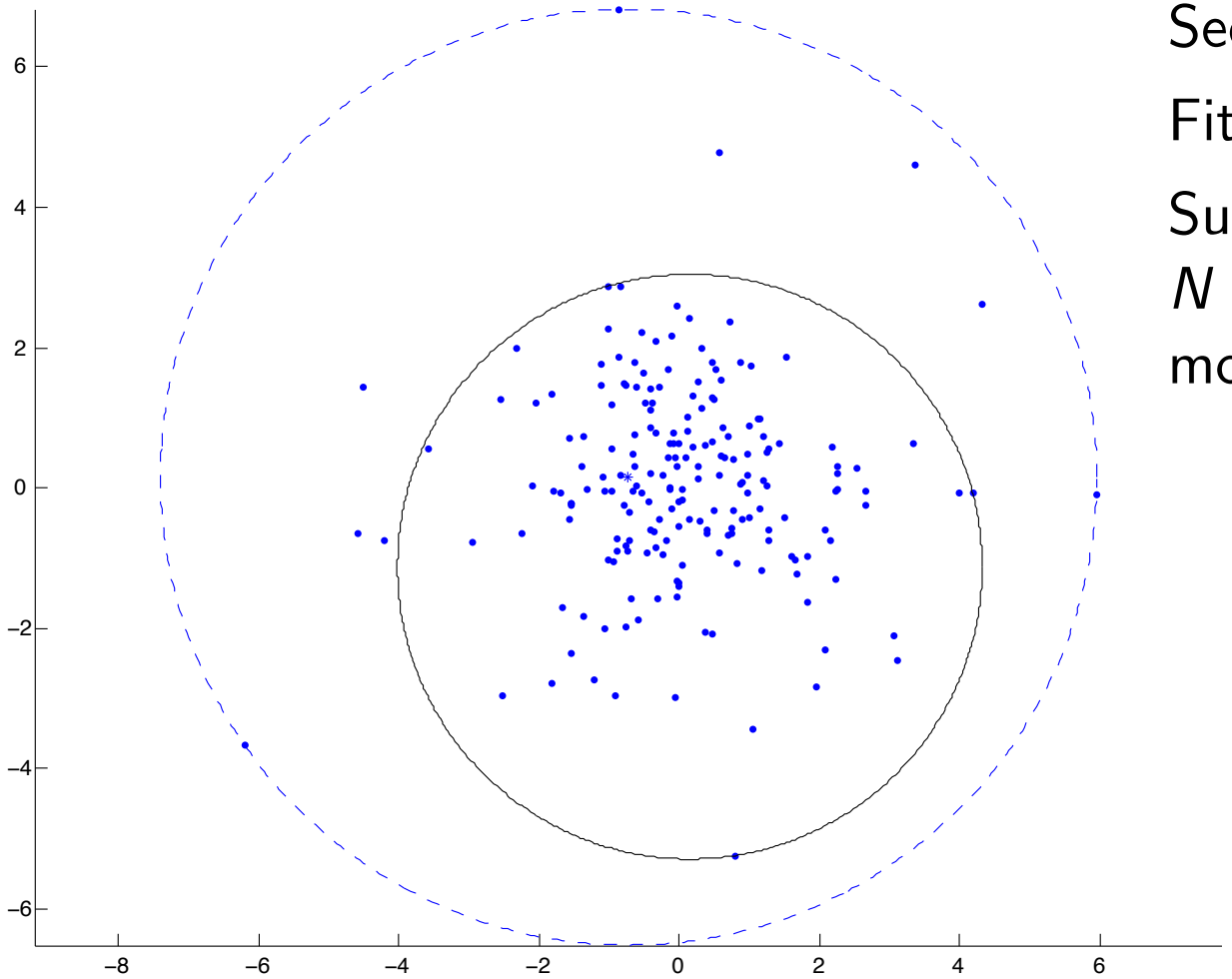


See N points

Fit model...

A step further

RCPs with violated constraints (RCPV)



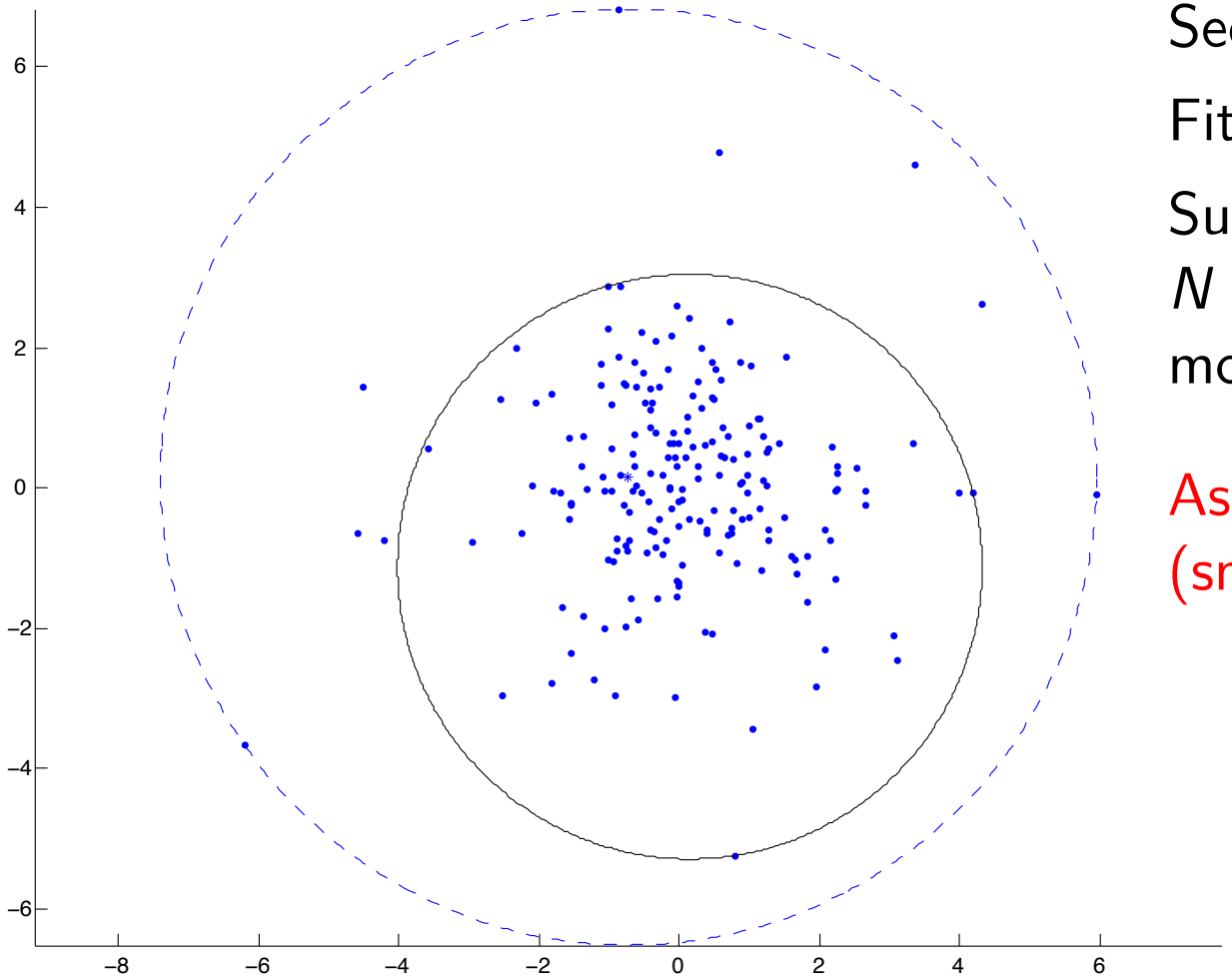
See N points

Fit model...

Suitably discard r of the N points, and refit model...

A step further

RCPs with violated constraints (RCPV)



See N points

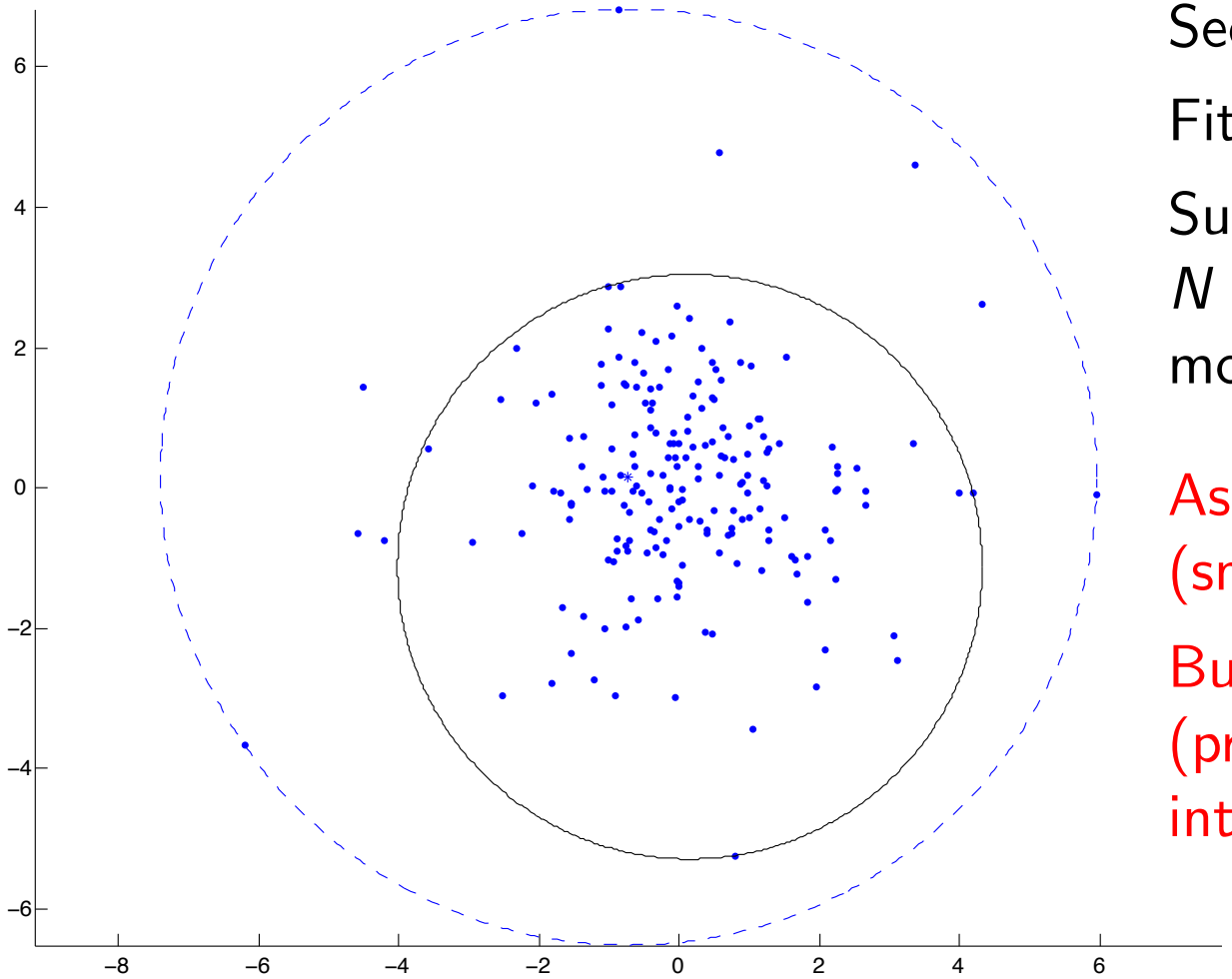
Fit model...

Suitably discard r of the N points, and refit model...

As $r \nearrow$ we get better (smaller) models...

A step further

RCPs with violated constraints (RCPV)



See N points

Fit model...

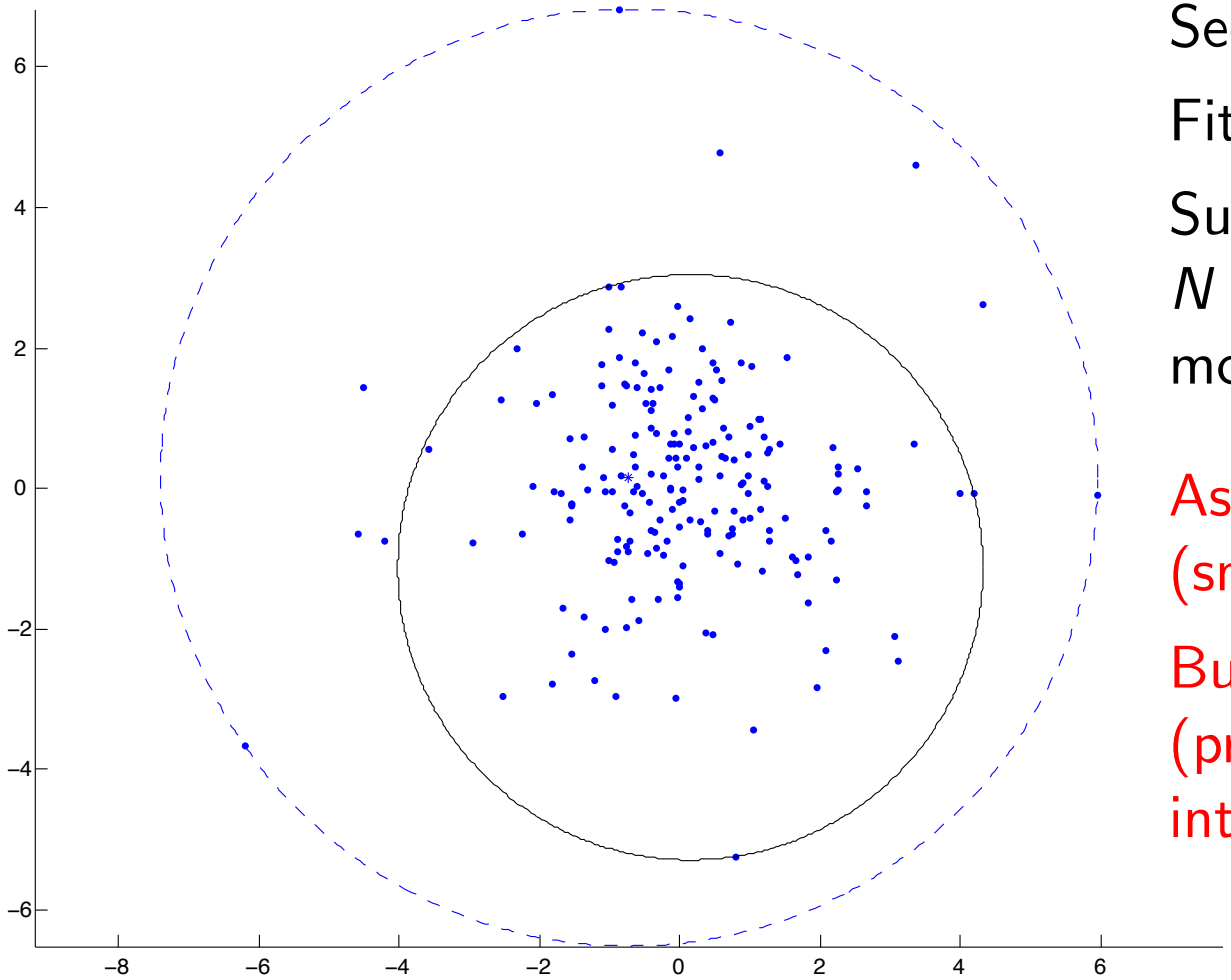
Suitably discard r of the N points, and refit model...

As $r \nearrow$ we get better (smaller) models...

But model reliability (prob. *new* point is in) intuitively gets worse...

A step further

RCPs with violated constraints (RCPV)



See N points

Fit model...

Suitably discard r of the N points, and refit model...

As $r \nearrow$ we get better (smaller) models...

But model reliability (prob. *new* point is in) intuitively gets worse...

...FUNDAMENTAL TRADEOFF!

RCPs with violated constraints (RCPV)

- Extend the basic RCP setup to the case when $r \geq 0$ of the N extracted constraints are purposely a-posteriori violated, with the aim of improving the optimal objective value.

- We select r and apply some optimal or sub-optimal strategy in order to find a subset of the N sampled constraints of cardinality $N - r$ so that the objective of the optimization problem with this subset of constraints is significantly reduced. We call this modified class of problems *RCPs with violated constraints*, or RCPVs for short.

- RCPVs provide an effective tool for modulating the robustness of the solution (i.e. the violation probability) and the achievable optimality level.

- Call ω_r the subset of $N - r$ elements from $\omega = (\delta^{(1)}, \ldots, \delta^{(N)})$ that are a-posteriori selected by the rule.

RCPs with violated constraints (RCPV)

- Extend the basic RCP setup to the case when $r \geq 0$ of the N extracted constraints are purposely a-posteriori violated, with the aim of improving the optimal objective value.

- We select r and apply some optimal or sub-optimal strategy in order to find a subset of the N sampled constraints of cardinality $N - r$ so that the objective of the optimization problem with this subset of constraints is significantly reduced. We call this modified class of problems *RCPs with violated constraints*, or RCPVs for short.

- RCPVs provide an effective tool for modulating the robustness of the solution (i.e. the violation probability) and the achievable optimality level.

- Call ω_r the subset of $N - r$ elements from $\omega = (\delta^{(1)}, \ldots, \delta^{(N)})$ that are a-posteriori selected by the rule.

RCPs with violated constraints (RCPV)

- Extend the basic RCP setup to the case when $r \geq 0$ of the N extracted constraints are purposely a-posteriori violated, with the aim of improving the optimal objective value.
- We select r and apply some optimal or sub-optimal strategy in order to find a subset of the N sampled constraints of cardinality $N - r$ so that the objective of the optimization problem with this subset of constraints is significantly reduced. We call this modified class of problems *RCPs with violated constraints*, or RCPVs for short.
- RCPVs provide an effective tool for modulating the robustness of the solution (i.e. the violation probability) and the achievable optimality level.
- Call ω_r the subset of $N - r$ elements from $\omega = (\delta^{(1)}, \ldots, \delta^{(N)})$ that are a-posteriori selected by the rule.

RCPs with violated constraints (RCPV)

- Extend the basic RCP setup to the case when $r \geq 0$ of the N extracted constraints are purposely a-posteriori violated, with the aim of improving the optimal objective value.

- We select r and apply some optimal or sub-optimal strategy in order to find a subset of the N sampled constraints of cardinality $N - r$ so that the objective of the optimization problem with this subset of constraints is significantly reduced. We call this modified class of problems *RCPs with violated constraints*, or RCPVs for short.

- RCPVs provide an effective tool for modulating the robustness of the solution (i.e. the violation probability) and the achievable optimality level.

- Call ω_r the subset of $N - r$ elements from $\omega = (\delta^{(1)}, \ldots, \delta^{(N)})$ that are a-posteriori selected by the rule.

RCPs with violated constraints

A key result

Theorem

Define

$$V^*(\omega) \doteq \mathbb{P}\{\delta \in \Delta : J(\omega_r, \delta) > J(\omega_r)\}.$$

and $\mathcal{B} = \{\omega \in \Delta^N : V^*(\omega) > \epsilon\}$. *Then,*

$$\mathbb{P}^N\{\mathcal{B}\} \leq \beta_{r,d}(\epsilon),$$

where

$$\beta_{r,d}(\epsilon) \doteq \binom{r+d}{r} \Phi(\epsilon; r+d, N),$$

being Φ *the cumulative beta distribution.*

RCPs with violated constraints

Reversing the bound

Corollary

Given $\epsilon \in (0,1)$, $\beta \in (0,1)$ *and integer* $r \leq N-d-1$. *If* N *is an integer such that*

$$N \geq \frac{2}{\epsilon} \ln \beta^{-1} + \frac{4}{\epsilon}(r+d), \quad r > 0,$$

then it holds that

$$\mathbb{P}^N\{V^* > \epsilon\} \leq \beta.$$

For $r = 0$, *the lower bound simplifies to previous one:*

$$N \geq \frac{2}{\epsilon}\left(\ln \beta^{-1} + d\right), \quad \textit{for } r = 0.$$

Some numerical values

For various values of N and ϵ, we can find explicitly the maximum allowed number r of constraints that can be a-posteriori removed while still guaranteeing that $\mathbb{P}^N\{\mathcal{B}\} \leq \beta$, with $\beta = 10^{-9}$.

	$\epsilon = 0.5$	$\epsilon = 0.4$	$\epsilon = 0.3$	$\epsilon = 0.2$	$\epsilon = 0.1$	$\epsilon = 0.01$	$\epsilon = 0.001$
$N = 60$	2	-	-	-	-	-	-
$N = 100$	11	5	0	-	-	-	-
$N = 200$	42	26	13	3	-	-	-
$N = 500$	153	109	69	34	6	-	-
$N = 1000$	358	265	179	100	32	-	-
$N = 2000$	793	602	421	250	96	-	-
$N = 5000$	2160	1673	1201	748	322	5	-
$N = 10000$	4506	3523	2563	1629	735	30	-
$N = 40000$	18959	14992	11071	7206	3426	237	2

Table: Maximum number r of removable constraints guaranteeing $\mathbb{P}^N\{\mathcal{B}\} \leq 10^{-9}$, for problems of dimension $d = 5$.

RCPV
Application in finance

- A collection of assets $a_1, \ldots, a_n$
- $p_i(k)$ is the market price of a_i at time kT, where T is a fixed period of time
- Simple return of an investment in asset i over the k-th period:

$$r_i(k) = \frac{p_i(k) - p_i(k-1)}{p_i(k-1)}, \; i = 1, \ldots, n; \; k = 1, 2, \ldots$$

$r(k) \doteq [r_1(k) \; \cdots \; r_n(k)]^\top$ is the vector of returns over the k-th period.

Assumption

The returns $\{r(k)\}_{k=1,2,\ldots}$ form an iid (independent, identically distributed) random sequence. In particular, each $r(k)$ is distributed according to the same and possibly unknown probability distribution $\mathbb{P}$ having support $\Delta \subseteq \mathbb{R}^n$.

Preliminaries

- A *portfolio* of assets $a_1, \ldots, a_n$ is defined by a vector $x \in \mathbb{R}^n$ whose entry x_i, $i = 1, \ldots, n$, describes the (signed) fraction of an investor's wealth invested in asset a_i, where $x_i \geq 0$ denotes a "long" position, and $x_i < 0$ denotes a "short" position
- Constraints on the portfolio vector (e.g., no short-selling: $x \geq 0$) are taken into account in generality, by considering that x is constrained in a given polytope $\mathcal{X}$

Assumption

Portfolio composition constraints are expressed by the condition $x \in \mathcal{X}$, where $\mathcal{X}$ is a given nonempty polytope. $\star$

- E.g., in the classical (Markovitz) case $\mathcal{X}$ is the standard simplex:

$$x_i \geq 0, \quad \sum_{i=1}^{n} x_i = 1.$$

Preliminaries

- A *portfolio* of assets $a_1, \ldots, a_n$ is defined by a vector $x \in \mathbb{R}^n$ whose entry x_i, $i = 1, \ldots, n$, describes the (signed) fraction of an investor's wealth invested in asset a_i, where $x_i \geq 0$ denotes a "long" position, and $x_i < 0$ denotes a "short" position
- Constraints on the portfolio vector (e.g., no short-selling: $x \geq 0$) are taken into account in generality, by considering that x is constrained in a given polytope $\mathcal{X}$

Assumption

Portfolio composition constraints are expressed by the condition $x \in \mathcal{X}$, where $\mathcal{X}$ is a given nonempty polytope. $\star$

- E.g., in the classical (Markovitz) case $\mathcal{X}$ is the standard simplex:

$$x_i \geq 0, \quad \sum_{i=1}^{n} x_i = 1.$$

Preliminaries

- A *portfolio* of assets $a_1, \ldots, a_n$ is defined by a vector $x \in \mathbb{R}^n$ whose entry x_i, $i = 1, \ldots, n$, describes the (signed) fraction of an investor's wealth invested in asset a_i, where $x_i \geq 0$ denotes a "long" position, and $x_i < 0$ denotes a "short" position
- Constraints on the portfolio vector (e.g., no short-selling: $x \geq 0$) are taken into account in generality, by considering that x is constrained in a given polytope $\mathcal{X}$

Assumption

Portfolio composition constraints are expressed by the condition $x \in \mathcal{X}$, where $\mathcal{X}$ is a given nonempty polytope. $\star$

- E.g., in the classical (Markovitz) case $\mathcal{X}$ is the standard simplex:

$$x_i \geq 0, \quad \sum_{i=1}^{n} x_i = 1.$$

Preliminaries

- A *portfolio* of assets $a_1, \ldots, a_n$ is defined by a vector $x \in \mathbb{R}^n$ whose entry x_i, $i = 1, \ldots, n$, describes the (signed) fraction of an investor's wealth invested in asset a_i, where $x_i \geq 0$ denotes a "long" position, and $x_i < 0$ denotes a "short" position
- Constraints on the portfolio vector (e.g., no short-selling: $x \geq 0$) are taken into account in generality, by considering that x is constrained in a given polytope $\mathcal{X}$

Assumption

Portfolio composition constraints are expressed by the condition $x \in \mathcal{X}$, where $\mathcal{X}$ is a given nonempty polytope. $\star$

- E.g., in the classical (Markovitz) case $\mathcal{X}$ is the standard simplex:

$$x_i \geq 0, \quad \sum_{i=1}^{n} x_i = 1.$$

Preliminaries

- The portfolio return over any period of duration Δ is described by a random variable $\varrho(x) = r^\top x$, where r is distributed according to $\mathbb{P}$
- Consider a sequence of returns of finite length N: $r(1), r(2), \ldots, r(N)$, collected by rows in matrix R_N:

$$R_N^\top = [r(1)\ r(2)\ \cdots\ r(N)] \in \mathbb{R}^{n,N}$$

- R_N is a random matrix, with each column independently distributed according to the unknown distribution $\mathbb{P}$; events related to R_N are measured by the product probability measure $\mathbb{P}^N$
- If $x \in \mathcal{X}$ is a given portfolio vector, then such a portfolio produces a random sequence of returns for $k = 1, \ldots, N$:

$$\rho_N(x) = R_N x = [\varrho_1(x)\ \varrho_2(x)\ \cdots\ \varrho_N(x)]^\top \in \mathbb{R}^N, \quad \varrho_i(x) \doteq r^\top(i)x$$

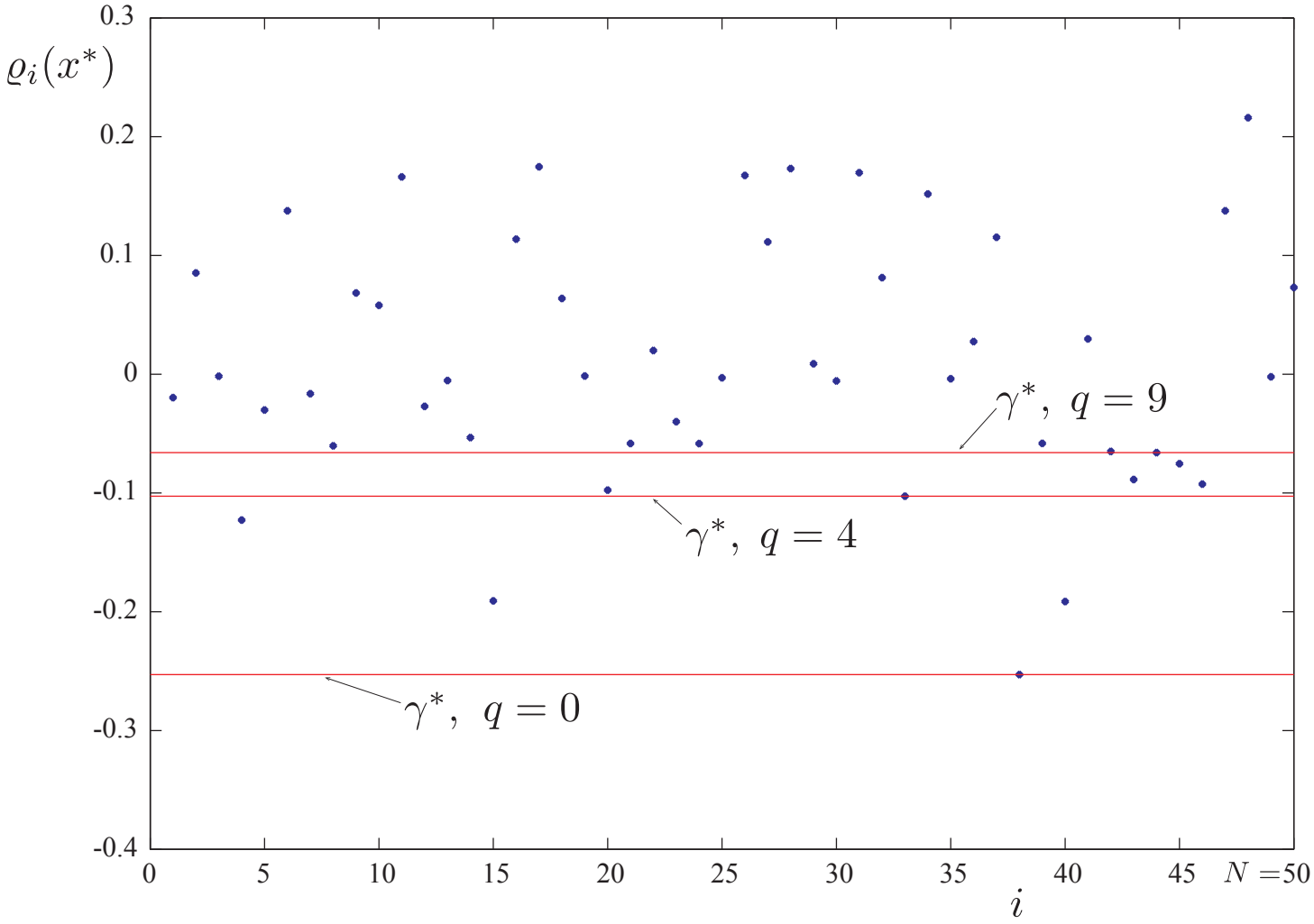
Return selection rule

- Let $q \leq N - n - 1$ be a given nonnegative integer. We introduce a *rule* $\mathcal{S}_q$ for selecting a subset of cardinality $N - q$ of returns in R_N.
- The rule selects q returns in the sequence $\{\varrho_i(x)\}$ such that γ^* is the largest lower bound over a (suitably selected) subset of $N - q$ returns, while q of the returns fall below γ^*.
- The advantage of doing so is to obtain a return level γ^* which is generally larger than the level obtained for $q = 0$.
- The portfolio allocation strategy x^* that we propose is a solution of the following LP

$$\begin{aligned} \gamma^* = \max_{x \in \mathcal{X}, \gamma} \quad & \gamma \\ \text{subject to:} \quad & r^\top(i)x \geq \gamma, \quad i \in \mathcal{I}_{N-q}, \end{aligned} \tag{3}$$

where $\mathcal{I}_{N-q}$ are the return indices selected by the rule.

Intuition behind selection rule



Intuition behind selection rule

- We are in the presence of a fundamental tradeoff here:

  while level γ^* increases by increasing q, intuitively this level also becomes less and less *reliable* that is, informally, the probability of the actual portfolio return $\varrho(x^*)$ being larger than γ^* decreases

- This fact should not come too much as a surprise, since level γ^* can also be interpreted as the empirical (q/N)-quantile of the return sequence $\{\varrho_i(x^*)\}_{i=1,\ldots,N}$

- We shall make this tradeoff rigorously precise.

Intuition behind selection rule

- We are in the presence of a fundamental tradeoff here:

  while level γ^* increases by increasing q, intuitively this level also becomes less and less *reliable* that is, informally, the probability of the actual portfolio return $\varrho(x^*)$ being larger than γ^* decreases

- This fact should not come too much as a surprise, since level γ^* can also be interpreted as the empirical (q/N)-quantile of the return sequence $\{\varrho_i(x^*)\}_{i=1,\ldots,N}$

- We shall make this tradeoff rigorously precise.

Intuition behind selection rule

- We are in the presence of a fundamental tradeoff here:

  while level γ^* increases by increasing q, intuitively this level also becomes less and less *reliable* that is, informally, the probability of the actual portfolio return $\varrho(x^*)$ being larger than γ^* decreases

- This fact should not come too much as a surprise, since level γ^* can also be interpreted as the empirical (q/N)-quantile of the return sequence $\{\varrho_i(x^*)\}_{i=1,\ldots,N}$

- We shall make this tradeoff rigorously precise.

The short-fall probability of optimal portfolio

- For a *fixed* portfolio $x \in \mathcal{X}$ and return level $\gamma \in \mathbb{R}$, we define the short-fall probability as

$$V(x, \gamma) = \mathbb{P}\{r : r^\top x < \gamma\};$$

such a probability is a *number* in $[0, 1]$.

- However, if we now ask about the short-fall probability relative to the optimal solution of (3), we have

$$V^* \doteq V(x^*, \gamma^*) = \mathbb{P}\{r : r^\top x^* < \gamma^*\},$$

and this is, a priori, a random variable, since x^*, γ^* are so.

- Therefore, V^* is a random variable with support $[0, 1]$, and events related to V^* are measured by the product probability $\mathbb{P}^N$.

The short-fall probability of optimal portfolio

- For a *fixed* portfolio $x \in \mathcal{X}$ and return level $\gamma \in \mathbb{R}$, we define the short-fall probability as
$$V(x, \gamma) = \mathbb{P}\{r : r^\top x < \gamma\};$$
such a probability is a *number* in $[0, 1]$.
- However, if we now ask about the short-fall probability relative to the optimal solution of (3), we have
$$V^* \doteq V(x^*, \gamma^*) = \mathbb{P}\{r : r^\top x^* < \gamma^*\},$$
and this is, a priori, a random variable, since x^*, γ^* are so.
- Therefore, V^* is a random variable with support $[0, 1]$, and events related to V^* are measured by the product probability $\mathbb{P}^N$.

The short-fall probability of optimal portfolio

- We consider as a measure of "riskiness" of the optimal portfolio the expected value (w.r.t. $\mathbb{P}^N$) of the short-fall probability V^*

Definition (Expected short-fall probability)

The expected short-fall probability of the optimal portfolio resulting from (3) is defined as

$$\mathbb{E}_{\mathbb{P}^N}\{V^*\} = \mathbb{E}_{\mathbb{P}^N}\{\mathbb{P}\{r : r^\top x^* < \gamma^*\}\}$$

- Our key result concerns a quantification of an upper bound on the expected short-fall probability of the optimal portfolio.

The short-fall probability of optimal portfolio

- We consider as a measure of "riskiness" of the optimal portfolio the expected value (w.r.t. $\mathbb{P}^N$) of the short-fall probability V^*

Definition (Expected short-fall probability)

The expected short-fall probability of the optimal portfolio resulting from (3) is defined as

$$\mathbb{E}_{\mathbb{P}^N}\{V^*\} = \mathbb{E}_{\mathbb{P}^N}\{\mathbb{P}\{r : r^\top x^* < \gamma^*\}\}$$

- Our key result concerns a quantification of an upper bound on the expected short-fall probability of the optimal portfolio.

The short-fall probability of optimal portfolio

- We consider as a measure of "riskiness" of the optimal portfolio the expected value (w.r.t. $\mathbb{P}^N$) of the short-fall probability V^*

Definition (Expected short-fall probability)

The expected short-fall probability of the optimal portfolio resulting from (3) is defined as

$$\mathbb{E}_{\mathbb{P}^N}\{V^*\} = \mathbb{E}_{\mathbb{P}^N}\{\mathbb{P}\{r : r^\top x^* < \gamma^*\}\}$$

- Our key result concerns a quantification of an upper bound on the expected short-fall probability of the optimal portfolio.

On the expected short-fall probability

Lemma (Upper bound on the expected short-fall probability)

Let x^*, γ^* be the optimal solution of problem (3), under any given selection rule. Then it holds that

$$\mathbb{E}_{\mathbb{P}^N}\{V^*\} \leq \frac{q}{N} + \left(\frac{n}{N} + \frac{\omega(n,q)}{2\sqrt{N}}\right),$$

where $\omega(n,q) = \mathcal{O}(\sqrt{2n\ln(q+n)})$ and, more precisely,

$$\omega(n,q) = \frac{2n(1+\ln(q+n)-\ln n)-2\ln 2+1}{\sqrt{2n(1+\ln(q+n)-\ln n)-2\ln 2}}.$$

$\star$

Explicit conditions on N and q

Corollary (Explicit conditions on N and q)

Let $\beta \in (0,1)$ be a very small probability level chosen by the user (e.g., set $\beta = 10^{-6}$, or lower, for "practical certainty"). Let $z_{\exp} \in (0,1)$ be a desired expected short-fall probability level, and let $q \leq N - n - 1$.
If

$$N \geq 4\frac{q+n}{z_{\exp}} + \frac{(c+1/c)^2}{4z_{\exp}^2}, \tag{4}$$

with $c \doteq \sqrt{2n + 2n \ln \frac{n+q}{q} - 2\ln 2}$, then it holds that $\mathbb{E}_{\mathbb{P}^N}\{V^*\} \leq z_{\exp}$.
For "large" q, bound (4) simplifies approximately to

$$N \geq 4\frac{q+n}{z_{\exp}} + \frac{2n + 2n \ln \frac{n+q}{q} - 2\ln 2}{4z_{\exp}^2}.$$

$\star$

Some remarks

(i) For fixed N and for a given desired level of expected short-fall probability z_{exp}, it is the investor's interest to make γ^* as large as possible.

On a given realization of the returns, level γ^* increases if we increase the number q of suppressed returns, hence we want to make q as large as possible.

However, if one increases q too much, then the resulting portfolio will fail to satisfy the expected short-fall probability requirement.

The right-hand-side of (4) is increasing in q, hence this term tells us precisely how large q can be made, while satisfying the requirement $\mathbb{E}_{\mathbb{P}^N}\{V^*\} \leq z_{\text{exp}}$: we choose q such that the right-hand-side of (4) is the largest integer that does not exceed N.

Some remarks

(i) For fixed N and for a given desired level of expected short-fall probability z_{exp}, it is the investor's interest to make γ^* as large as possible.

On a given realization of the returns, level γ^* increases if we increase the number q of suppressed returns, hence we want to make q as large as possible.

However, if one increases q too much, then the resulting portfolio will fail to satisfy the expected short-fall probability requirement.

The right-hand-side of (4) is increasing in q, hence this term tells us precisely how large q can be made, while satisfying the requirement $\mathbb{E}_{\mathbb{P}^N}\{V^*\} \leq z_{\text{exp}}$:
we choose q such that the right-hand-side of (4) is the largest integer that does not exceed N.

Some remarks

(ii) All the other parameters being the same, the expected short-fall probability bound increases as n (the number of securities) increases.

There is a fundamental tradeoff between the complexity of the random optimization model (here, the number of variables, $n+1$) and the out-of-sample *reliability* of the model: the more complex the model is (i.e., the larger n is), the more training data we need for achieving a given reliability level (i.e., the larger N needs to be, see eq. (4)).

A financial interpretation of this phenomenon is that high diversification of a portfolio (large n) needs a large number N of data in order to provide meaningful portfolios.

This fact is often overlooked in standard portfolio optimization, where one may be lead to think that more diversification leads to less risk: not necessarily so, since the larger n the more accurate the parameter estimation must be, for otherwise the actual decrease in risk level may just be illusory.

Some remarks

(ii) All the other parameters being the same, the expected short-fall probability bound increases as n (the number of securities) increases.

There is a fundamental tradeoff between the complexity of the random optimization model (here, the number of variables, $n+1$) and the out-of-sample *reliability* of the model: the more complex the model is (i.e., the larger n is), the more training data we need for achieving a given reliability level (i.e., the larger N needs to be, see eq. (4)).

A financial interpretation of this phenomenon is that high diversification of a portfolio (large n) needs a large number N of data in order to provide meaningful portfolios.

This fact is often overlooked in standard portfolio optimization, where one may be lead to think that more diversification leads to less risk: not necessarily so, since the larger n the more accurate the parameter estimation must be, for otherwise the actual decrease in risk level may just be illusory.

Some remarks

(ii) All the other parameters being the same, the expected short-fall probability bound increases as n (the number of securities) increases.

There is a fundamental tradeoff between the complexity of the random optimization model (here, the number of variables, $n+1$) and the out-of-sample *reliability* of the model: the more complex the model is (i.e., the larger n is), the more training data we need for achieving a given reliability level (i.e., the larger N needs to be, see eq. (4)).

A financial interpretation of this phenomenon is that high diversification of a portfolio (large n) needs a large number N of data in order to provide meaningful portfolios.

This fact is often overlooked in standard portfolio optimization, where one may be lead to think that more diversification leads to less risk: not necessarily so, since the larger n the more accurate the parameter estimation must be, for otherwise the actual decrease in risk level may just be illusory.

Repetitive scenario design (RSD)
Introduction

Repetitive Scenario Design (RSD)

Repetitive scenario design (RSD)

- Repetitive Scenario Design (RSD) is a randomized approach to robust design based on iterating two phases: a standard scenario design phase that uses N scenarios (design samples), followed by randomized feasibility phase that uses N_o test samples on the scenario solution.
- This novel approach broadens the applicability of the scenario technology, since the user is now presented with a clear tradeoff between the number N of design samples and the ensuing expected number of repetitions required by the RSD algorithm.
- The plain (one-shot) scenario design becomes just one of the possibilities, sitting at one extreme of the tradeoff curve, in which one insists in finding a solution in a single repetition: this comes at the cost of possibly high N. Other possibilities along the tradeoff curve use lower N values, but possibly require more than one repetition.

Repetitive scenario design (RSD)

Idea

- Each time we solve a scenario problem with N scenarios, we "toss a coin." The toss is successful if $V(\theta^*) \leq \epsilon$, while it fails if $V(\theta^*) > \epsilon$.
- The a-priori probability of success in a coin toss is $\geq 1 - \beta_\epsilon(N)$, where

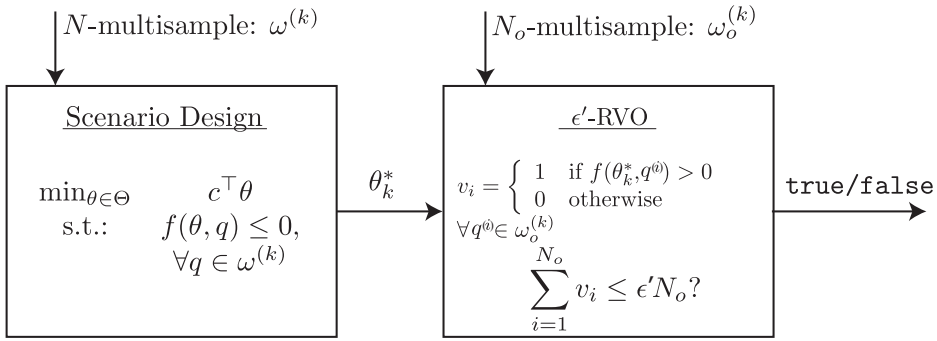
$$\beta_\epsilon(N) = \Phi(\epsilon; d, N) \doteq \sum_{j=0}^{d} \binom{N}{j} \epsilon^j (1-\epsilon)^{N-j}$$

- Plain scenario technology works by selecting N such that $\beta_\epsilon(N)$ is very small (say, $\leq 10^{-9}$).
- This means biasing the coin so to be successful with practical certainty (i.e., w.p. $\geq 1 - 10^{-9}$) in one single coin toss!
- Success in one toss comes at the price of possibly high N...

Repetitive scenario design (RSD)

Idea

- What if we use a lower N (i.e., we bias the coin with higher $\beta_\epsilon(N)$) and then check the resulting solution?
- Idea: while the probability of being successful in one shot is low, if we toss the coin repeatedly, the probability of being successful at some toss becomes arbitrarily high...
- We thus set up an iterative approach in two stages: a scenario optimization stage, and a feasibility check phase.



Repetitive scenario design (RSD)

Ideal oracle

- We first assume we have an ideal feasibility oracle, called a Deterministic Violation Oracle (DVO), that returns `true` if $V(\theta^*) \leq \epsilon$ and `false` otherwise.
- We apply the following algorithm:

Algorithm (RSD with ϵ-DVO)
Input data: integer $N \geq n$, level $\epsilon \in [0, 1]$.
Output data: solution θ^*. Initialization: set iteration counter to $k = 1$.

1. (Scenario step) Generate N i.i.d. samples $\omega^{(k)} \doteq \{q_k^{(1)}, \ldots, q_k^{(N)}\}$ according to $\mathbb{P}$, and solve scenario problem. Let θ_k^* be the resulting optimal solution.
2. (ϵ-DVO step) If $V(\theta_k^*) \leq \epsilon$, then set flag to `true`, else set it to `false`.
3. (Exit condition) If flag is `true`, then exit and return current solution $\theta^* \leftarrow \theta_k^*$; else set $k \leftarrow k + 1$ and goto 1.

Repetitive scenario design (RSD)

Ideal oracle

Theorem

Given $\epsilon \in [0,1]$ and $N \geq n$, define the running time K of the algorithm with DVO as the value of the iteration counter k when the algorithm exits. Then:

1. *The solution θ^* returned by the algorithm is an ϵ-probabilistic robust design, i.e., $V(\theta^*) \leq \epsilon$.*
2. *The expected running time of the algorithm is $\leq (1 - \beta_\epsilon(N))^{-1}$, and equality holds if the scenario problem is f.s. w.p. 1.*
3. *The running time of the algorithm is $\leq k$ with probability $\geq 1 - \beta_\epsilon(N)^k$, and equality holds if the scenario problem is f.s. w.p. 1.*

Repetitive scenario design (RSD)

Randomized oracle

- Since the ideal oracle is hardly realizable in practice, we next introduce a Randomized Violation Oracle (RVO):

ϵ'-**RVO** (*Randomized ϵ'-violation oracle*)
Input data: integer N_o, level $\epsilon' \in [0,1]$, and $\theta \in \mathbb{R}^n$. Output data: a logic flag, `true` or `false`.

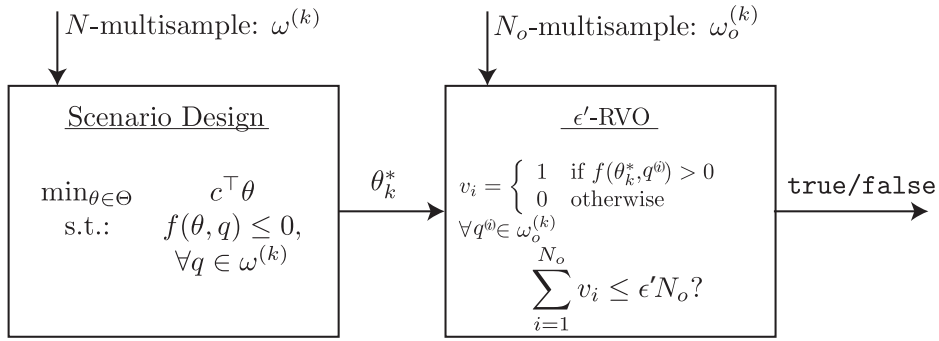
1. Generate N_o i.i.d. samples $\omega_o \doteq \{q^{(1)}, \ldots, q^{(N_o)}\}$, according to $\mathbb{P}$.
2. For $i = 1, \ldots, N_o$, let $v_i = 1$ if $f(\theta, q^{(i)}) > 0$ and $v_i = 0$ otherwise.
3. If $\sum_i v_i \leq \epsilon' N_o$, return `true`, else return `false`.

Repetitive scenario design (RSD)

Randomized oracle

Algorithm (RSD with ϵ'-RVO)
Input data: integers N, N_o, level $\epsilon' \in [0,1]$. Output data: solution θ^*.
Initialization: set iteration counter to $k = 1$.

1. (Scenario step) Generate N i.i.d. samples $\omega^{(k)} \doteq \{q_k^{(1)}, \ldots, q_k^{(N)}\}$ according to $\mathbb{P}$, and solve scenario problem. Let θ_k^* be the resulting optimal solution.
2. (ϵ'-RVO step) Call the ϵ'-RVO with current θ_k^* as input, and set flag to `true` or `false` according to the output of the ϵ'-RVO.
3. (Exit cond.) If flag is `true`, then exit and return current solution $\theta^* \leftarrow \theta_k^*$; else set $k \leftarrow k+1$ and goto 1.



Repetitive scenario design (RSD)

Randomized oracle

Theorem (RSD with ϵ'-RVO)
Let $\epsilon, \epsilon' \in [0,1]$, $\epsilon' \leq \epsilon$, and $N \geq n$ be given. Define the event $\mathrm{BadExit}$ in which the algorithm exits returning a "bad" solution θ^*:

$$\mathrm{BadExit} \doteq \{\text{algorithm returns } \theta^*\text{: } V(\theta^*) > \epsilon\}.$$

The following statements hold.

1. $$\mathbb{P}\{\mathrm{BadExit}\} \leq \frac{\mathrm{Fbeta}((1-\epsilon')N_o, \epsilon' N_o + 1; 1-\epsilon)}{1 - H_{1,\epsilon'}(N, N_o)} \beta_\epsilon(N).$$
2. The expected running time of the algorithm is $\leq (1 - H_{1,\epsilon'}(N, N_o))^{-1}$, and equality holds if the scenario problem is f.s. w.p. 1.
3. The running time of the algorithm is $\leq k$ with probability $\geq 1 - H_{1,\epsilon'}(N, N_o)^k$, and equality holds if the scenario problem is f.s. w.p. 1.

Here, Fbeta denotes the cumulative distribution of a beta density, and $H_{1,\epsilon'}(N, N_o)$ has an explicit expression in terms of beta-Binomial distributions.

Repetitive scenario design (RSD)

Randomized oracle

- A key quantity related to the expected running time of the algorithm is $H_{1,\epsilon'}(N, N_o)$, which is the upper tail of a beta-Binomial distribution.
- This quantity is related to the hypergeometric function ${}_3F_2$, and to ratios of Gamma functions, which may be delicate to evaluate numerically for large values of the arguments. It is therefore useful to have a more "manageable," albeit approximate, expression for $H_{1,\epsilon'}(N, N_o)$.

Corollary

For $N_o \to \infty$ it holds that $H_{1,\epsilon'}(N, N_o) \to \beta_{\epsilon'}(N)$.

- For large N_o, and $\epsilon' \leq \epsilon$, we have $H_{1,\epsilon'}(N, N_o) \simeq \beta_{\epsilon'}(N) \geq \beta_{\epsilon}(N)$, whence

$$\hat{K} \doteq \frac{1}{1 - H_{1,\epsilon'}(N, N_o)} \simeq \frac{1}{1 - \beta_{\epsilon'}(N)} \geq \frac{1}{1 - \beta_{\epsilon}(N)}.$$

This last equation gives us an approximate, asymptotic, expression for the upper bound $\hat{K}$ on the expected running time of the algorithm.

Example

Robust finite-horizon input design

- We consider a system of the form

$$x(t+1) = A(q)x(t) + Bu(t), \quad t = 0, 1, \ldots; \; x(0) = 0,$$

where $u(t)$ is a scalar input signal, and $A(q) \in \mathbb{R}^{n_a, n_a}$ is an interval uncertain matrix of the form

$$A(q) = A_0 + \sum_{i,j=1}^{n_a} q_{ij} e_i e_j^\top, \quad |q_{ij}| \leq \rho, \; \rho > 0,$$

where e_i is a vector of all zeros, except for a one in the i-th entry.

- Given a final time $T \geq 1$ and a target state $\bar{x}$, the problem is to determine an input sequence $\{u(0), \ldots, u(T-1)\}$ such that *(i)* the state $x(T)$ is robustly contained in a small ball around the target state $\bar{x}$, and *(ii)* the input energy $\sum_k u(k)^2$ is not too large.

Example

Robust finite-horizon input design

- We write $x(T) = x(T; q) = \mathcal{R}(q)u$, where $\mathcal{R}(q)$ is the T-reachability matrix of the system (for a given q), and $u \doteq (u(0), \ldots, u(T-1))$.
- We formally express our design goals in the form of minimization of a level γ such that

$$\|x(T; q) - \bar{x}\|_2^2 + \lambda \sum_{t=0}^{T-1} u(t)^2 \leq \gamma,$$

where $\lambda \geq 0$ is a tradeoff parameter. Letting $\theta = (u, \gamma)$, the problem is formally stated in our framework by setting

$$f(\theta, q) \leq 0, \quad \text{where } f(\theta, q) \doteq \|\mathcal{R}(q)u - \bar{x}\|_2^2 + \lambda \|u\|_2^2 - \gamma.$$

Example

Robust finite-horizon input design

- Assuming that the uncertain parameter q is random and uniformly distributed in the hypercube $\mathbb{Q} = [-\rho, \rho]^{n_a \times n_a}$, our scenario design problem takes the following form:

$$\begin{aligned} \min_{\theta=(u,\gamma)} \quad & \gamma \\ \text{s.t.:} \quad & f(\theta, q^{(i)}) \leq 0, \quad i = 1, \ldots, N. \end{aligned}$$

- We set $T = 10$, thus the size of the decision variable $\theta = (u, \gamma)$ of the scenario problem is $n = 11$.
- We set the desired level of probabilistic robustness to $1 - \epsilon = 0.995$, i.e., $\epsilon = 0.005$, and require a level of failure of the randomized method below $\beta = 10^{-12}$, that is, we require the randomized method to return a good solution with "practical certainty."

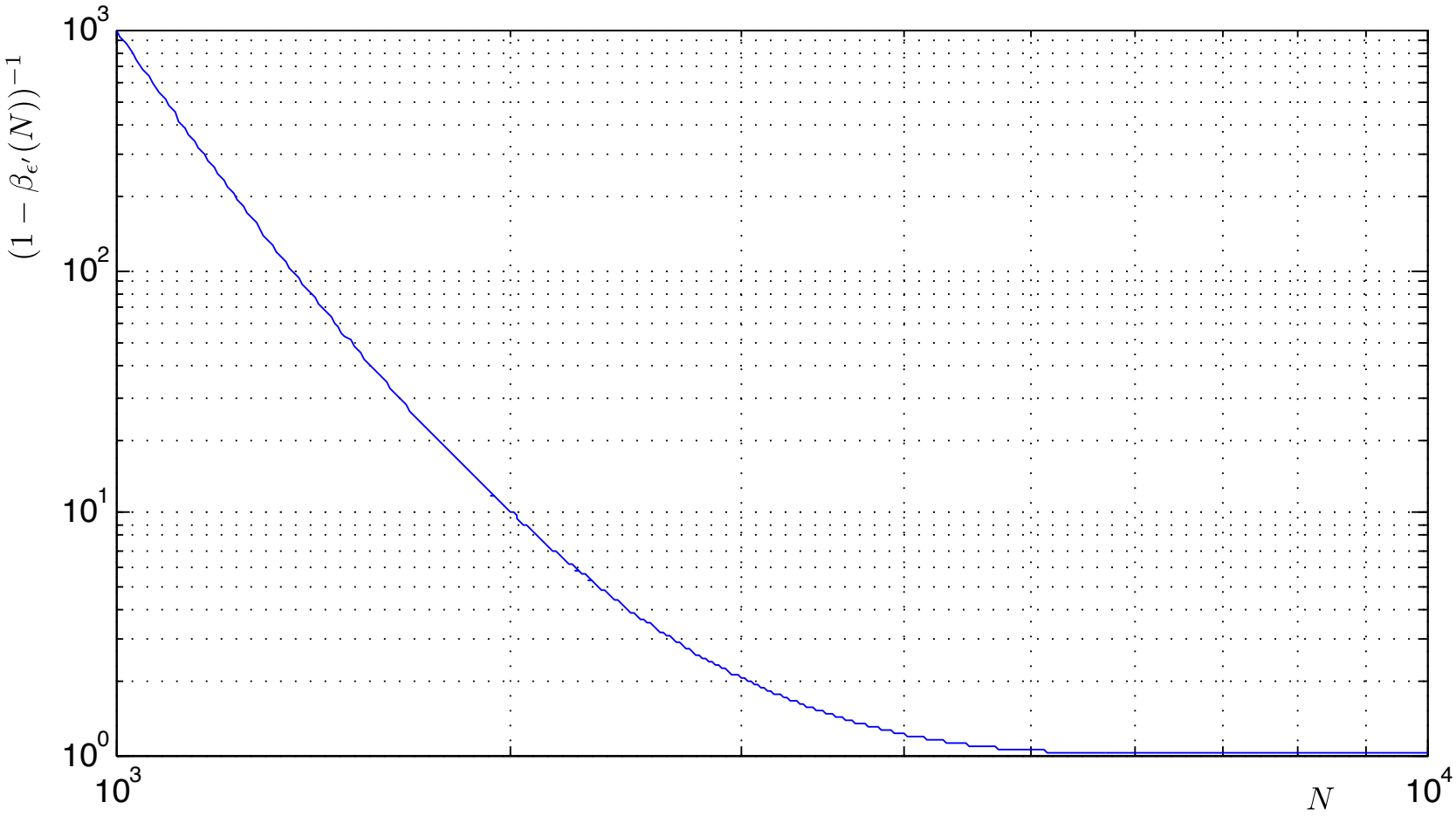
Example

Robust finite-horizon input design

- Using a plain (one-shot) scenario approach, imposing $\beta_\epsilon(N) \leq \beta$ would require $N \geq 10440$ scenarios.
- Let us now see how we can reduce this N figure by resorting to a repetitive scenario design approach.
- Let us fix $\epsilon' = 0.7\epsilon = 0.0035$, thus $\delta = \epsilon - \epsilon' = 0.0015$.
- A plot of the (asymptotic) bound on expected number of iterations, $(1 - \beta_{\epsilon'}(N))^{-1}$, as a function of N is shown in the next figure. We see from this plot, for instance, that the choice $N = 2000$ corresponds to a value of about 10 for the upper bound on the expected number of iterations.
- Let us choose this value of $N = 2000$ for the scenario block.

Example

Robust finite-horizon input design



Log-log plot of $(1-\beta_{\epsilon'}(N))^{-1}$ vs. N.

Example

Robust finite-horizon input design

- For $N = 2000$ a reliability level $\beta = 10^{-12}$ is achieved for $N_o \geq 62403$. Let us then choose $N_o = 63000$ samples to be used in the oracle.
- With the above choices we have $H_{1,\epsilon'}(N, N_o) = 0.8963$, thus the algorithm's upper bound on average running time is

$$\hat{K} = (1 - H_{1,\epsilon'}(N, N_o))^{-1} = 9.64.$$

- Notice that this upper bound is conservative (worst case) in general. Thus, we may expect a performance which is in practice better than the one predicted by the bound.

Example

Robust finite-horizon input design

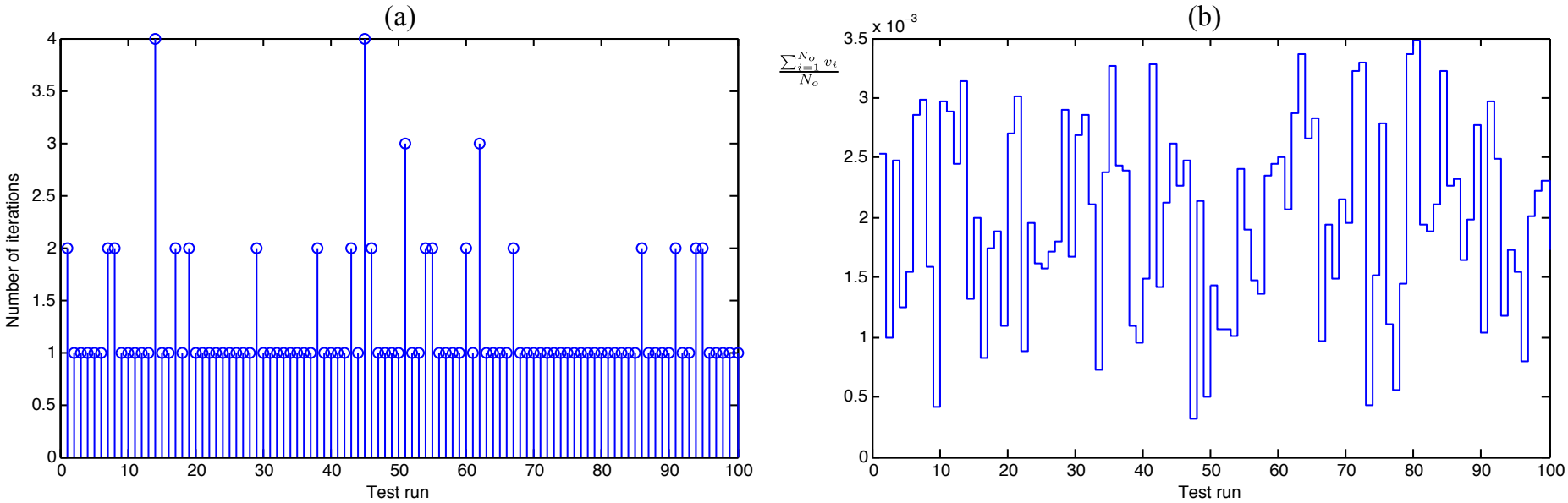
- We considered the following data:

$$A_0 = \begin{bmatrix} -0.7214 & -0.0578 & 0.2757 & 0.7255 & 0.2171 & 0.3901 \\ 0.5704 & 0.1762 & 0.3684 & -0.0971 & 0.6822 & -0.5604 \\ -1.3983 & -0.1795 & 0.1511 & 1.0531 & -0.1601 & 0.9031 \\ -0.6308 & -0.0058 & 0.4422 & 0.8169 & 0.512 & 0.2105 \\ 0.7539 & 0.1423 & 0.2039 & -0.3757 & 0.5088 & -0.6081 \\ -1.3571 & -0.1769 & 0.1076 & 1.0032 & -0.1781 & 0.9151 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

- We set target state $\bar{x} = [1, -1/2, 2, 1, -1, 2]^\top$, $\rho = 0.001$, and $\lambda = 0.005$.
- We run the RSD algorithm for 100 times, and on each test run we recorded the number of iterations and the solution returned upon exit. The algorithm exited most of the times in a single repetition, with a maximum of 4 repetitions.

Example

Robust finite-horizon input design



(a) Repetitions of RSD algorithm in the 100 test runs.
(b) Levels of empirical violation probability evaluated by the oracle upon exit, in the 100 test runs.

Example

Robust finite-horizon input design

Computational improvements

- Substantial reduction of the number of design samples (from the 10440 to just 2000), at the price of a very moderate number of repetitions (the average number of repetitions in the 100 test runs was 1.27).
- On average over the 100 test experiments, the RSD method (with $N = 2000$, $N_o = 63000$) required 224 s to return a solution.
- A plain, one-shot, scenario optimization with $N = 10440$ scenarios required 2790 s. Using the RSD approach instead of a plain one-shot scenario design thus yielded a reduction in computing time of about one order of magnitude.
- The reason for this improvement is due to the fact that the scenario optimization problem in the RSD approach (which uses $N = 2000$ scenarios) took about 173 s to be solved on a typical run, and the subsequent randomized oracle test (with $N_o = 63000$) is computationally cheap, taking only about 3.16 s.

Conclusions

- Scenario design is a flexible technology that permits attacking a class of robust design problems that are hard to deal with via conventional deterministic methods.
- Widely used in control design. Recently became particularly popular in Model Predictive Control.
- Interesting data-driven approaches in computational finance. Potential in many engineering domains.
- The repetitive approach further broadens the applicability of scenario design to problems in which dealing with "large N" may be a problem in practice (e.g., robust SDP problems).

THANK YOU!