Stochastic programming

- stochastic programming
- 'certainty equivalent' problem
- violation/shortfall constraints and penalties
- Monte Carlo sampling methods
- validation

sources: Nemirovsky & Shapiro

Stochastic programming

- objective and constraint functions $f_i(x, \omega)$ depend on optimization variable x *and* a random variable ω

- ω models
  - parameter variation and uncertainty
  - random variation in implementation, manufacture, operation

- value of ω is not known, but its distribution is

- goal: choose x so that
  - constraints are satisfied on average, or with high probability
  - objective is small on average, or with high probability

Stochastic programming

- basic stochastic programming problem:

$$\begin{array}{ll} \text{minimize} & F_0(x) = \mathbf{E}\, f_0(x,\omega) \\ \text{subject to} & F_i(x) = \mathbf{E}\, f_i(x,\omega) \leq 0, \quad i = 1,\ldots,m \end{array}$$

  - variable is x
  - problem data are f_i, distribution of ω

- if $f_i(x,\omega)$ are convex in x for each ω
  - F_i are convex
  - hence stochastic programming problem is convex

- F_i have analytical expressions in only a few cases;
  in other cases we will solve the problem approximately

Example with analytic form for F_i

- $f(x) = \|Ax - b\|_2^2$, with A, b random

- $F(x) = \mathbf{E}\, f(x) = x^T P x - 2q^T x + r$, where

$$P = \mathbf{E}(A^T A), \quad q = \mathbf{E}(A^T b), \quad r = \mathbf{E}(\|b\|_2^2)$$

- only need second moments of (A, b)

- stochastic constraint $\mathbf{E}\, f(x) \leq 0$ can be expressed as standard quadratic inequality

‘Certainty-equivalent’ problem

- ‘certainty-equivalent’ (a.k.a. ‘mean field’) problem:

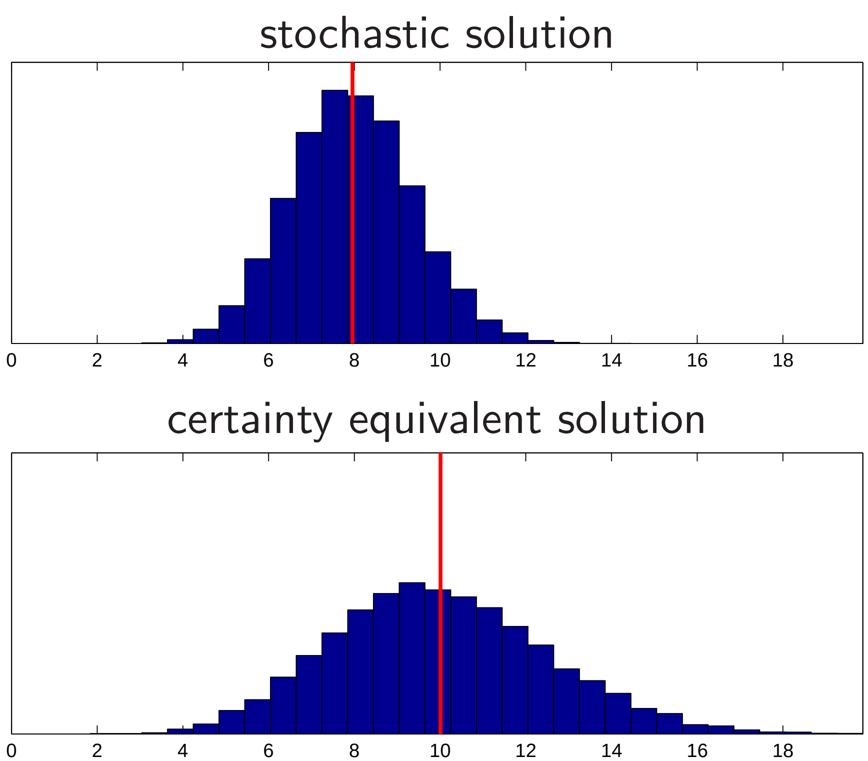
$$\begin{array}{ll} \text{minimize} & f_0(x, \mathbf{E}\,\omega) \\ \text{subject to} & f_i(x, \mathbf{E}\,\omega) \leq 0, \quad i = 1, \ldots, m \end{array}$$

- roughly speaking: ignore parameter variation

- if f_i convex in ω for each x, then
  - $f_i(x, \mathbf{E}\,\omega) \leq \mathbf{E}\, f_i(x, \omega)$
  - so optimal value of certainty-equivalent problem is lower bound on optimal value of stochastic problem

Stochastic programming example

- minimize $\mathbf{E}\,\|Ax - b\|_1$; A_{ij} uniform on $\bar{A}_{ij} \pm \gamma_{ij}$; b_i uniform on $\bar{b}_i \pm \delta_i$
- objective PDFs for stochastic optimal and certainty-equivalent solutions
- lower bound from CE problem: 5.96



Expected violation/shortfall constraints/penalties

- replace $\mathbf{E} f_i(x,\omega) \le 0$ with
  - $\mathbf{E} f_i(x,\omega)_+ \le \epsilon$ (LHS is expected violation)
  - $\mathbf{E}\left(\max_i f_i(x,\omega)_+\right) \le \epsilon$ (LHS is expected worst violation)
- variation: add violation/shortfall penalty to objective

$$\text{minimize} \quad \mathbf{E}\left(f_0(x,\omega) + \textstyle\sum_{i=1}^m c_i f_i(x,\omega)_+\right)$$

  where $c_i > 0$ are penalty rates for violating constraints

- these are convex problems if f_i are convex in x

Chance constraints and percentile optimization

- 'chance constraints' (η is 'confidence level'):

$$\mathbf{Prob}(f_i(x,\omega) \leq 0) \geq \eta$$

  - convex in some cases
  - generally interested in $\eta = 0.9,\ 0.95,\ 0.99$
  - $\eta = 0.999$ meaningless (unless you're sure about the distribution tails)

- percentile optimization (γ is 'η-percentile'):

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \mathbf{Prob}(f_0(x,\omega) \leq \gamma) \geq \eta \end{array}$$

  - convex or quasi-convex in some cases

- these topics covered next lecture

Solving stochastic programming problems

- analytical solution in special cases, $e.g.$, when expectations can be found analytically
  - ω enters quadratically in f_i
  - ω takes on finitely many values
- general case: approximate solution via (Monte Carlo) sampling

Finite event set

- suppose $\omega \in \{\omega_1, \ldots, \omega_N\}$, with $\pi_j = \mathbf{Prob}(\omega = \omega_j)$
- sometime called 'scenarios'; often we have $\pi_j = 1/N$
- stochastic programming problem becomes

$$\begin{array}{ll} \text{minimize} & F_0(x) = \sum_{j=1}^N \pi_j f_0(x, \omega_j) \\ \text{subject to} & F_i(x) = \sum_{j=1}^N \pi_j f_i(x, \omega_j) \leq 0, \quad i = 1, \ldots, m \end{array}$$

- a (standard) convex problem if f_i convex in x
- computational complexity grows *linearly* in the number of scenarios N

Monte Carlo sampling method

- a general method for (approximately) solving stochastic programming problem
- generate N samples (realizations) $\omega_1, \ldots, \omega_N$, with associated probabilities $\pi_1, \ldots, \pi_N$ (usually $\pi_j = 1/N$)
- form sample average approximations

$$\hat{F}_i(x) = \sum_{j=1}^{N} \pi_j f_i(x, \omega_j), \quad i = 0, \ldots, m$$

- these are RVs (via $\omega_1, \ldots, \omega_N$) with mean $\mathbf{E}\, f_i(x, \omega) = F_i(x)$

- now solve finite event problem

$$\begin{array}{ll} \text{minimize} & \hat{F}_0(x) \\ \text{subject to} & \hat{F}_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

- solution $x^\star_{\mathrm{mcs}}$ and optimal value $\hat{F}_0(x^\star_{\mathrm{mcs}})$ are random variables (hopefully close to $x^\star$ and $p^\star$, optimal value of original problem)

- theory says
  - (with some technical conditions) as $N \to \infty$, $x^\star_{\mathrm{mcs}} \to x^\star$
  - $\mathbf{E}\, \hat{F}_0(x^\star_{\mathrm{mcs}}) \leq p^\star$

Out-of-sample validation

- a practical method to check if N is 'large enough'
- use a second set of samples ('validation set') $\omega_1^{\mathrm{val}}, \ldots, \omega_M^{\mathrm{val}}$, with probabilities $\pi_1^{\mathrm{val}}, \ldots, \pi_M^{\mathrm{val}}$ (usually $M \gg N$)

  (original set of samples called 'training set')
- evaluate

$$\hat{F}_i^{\mathrm{val}}(x^\star_{\mathrm{mcs}}) = \sum_{j=1}^{M} \pi_j^{\mathrm{val}} f_i(x^\star_{\mathrm{mcs}}, \omega_j^{\mathrm{val}}), \quad i = 0, \ldots, m$$

- if $\hat{F}_i(x^\star_{\mathrm{mcs}}) \approx \hat{F}_i^{\mathrm{val}}(x^\star_{\mathrm{mcs}})$, our confidence that $x^\star_{\mathrm{mcs}} \approx x^\star$ is enhanced
- if not, increase N and re-compute $x^\star_{\mathrm{mcs}}$

Example

- we consider problem

$$\begin{array}{ll} \text{minimize} & F_0(x) = \mathbf{E} \max_i (Ax+b)_i \\ \text{subject to} & F_1(x) = \mathbf{E} \max_i (Cx+d)_i \leq 0 \end{array}$$

  with optimization variable $x \in \mathbf{R}^n$

  $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, $C \in \mathbf{R}^{k \times n}$, $d \in \mathbf{R}^k$ are random

- we consider instance with $n = 10$, $m = 20$, $k = 5$
- certainty-equivalent optimal value yields lower bound 19.1
- we use Monte Carlo sampling with $N = 10,\ 100,\ 1000$
- validation set uses $M = 10000$

	$N = 10$	$N = 100$	$N = 1000$
F_0 (training)	51.8	54.0	55.4
F_0 (validation)	56.0	54.8	55.2
F_1 (training)	0	0	0
F_1 (validation)	1.3	0.7	-0.03

we conclude:

- $N = 10$ is too few samples
- $N = 100$ is better, but not enough
- $N = 1000$ is probably fine

Production planning with uncertain demand

- manufacture quantities $q = (q_1, \ldots, q_m)$ of m finished products
- purchase raw materials in quantities $r = (r_1, \ldots, r_n)$ with costs $c = (c_1, \ldots, c_n)$, so total cost is $c^T r$
- manufacturing process requires $r \succeq Aq$

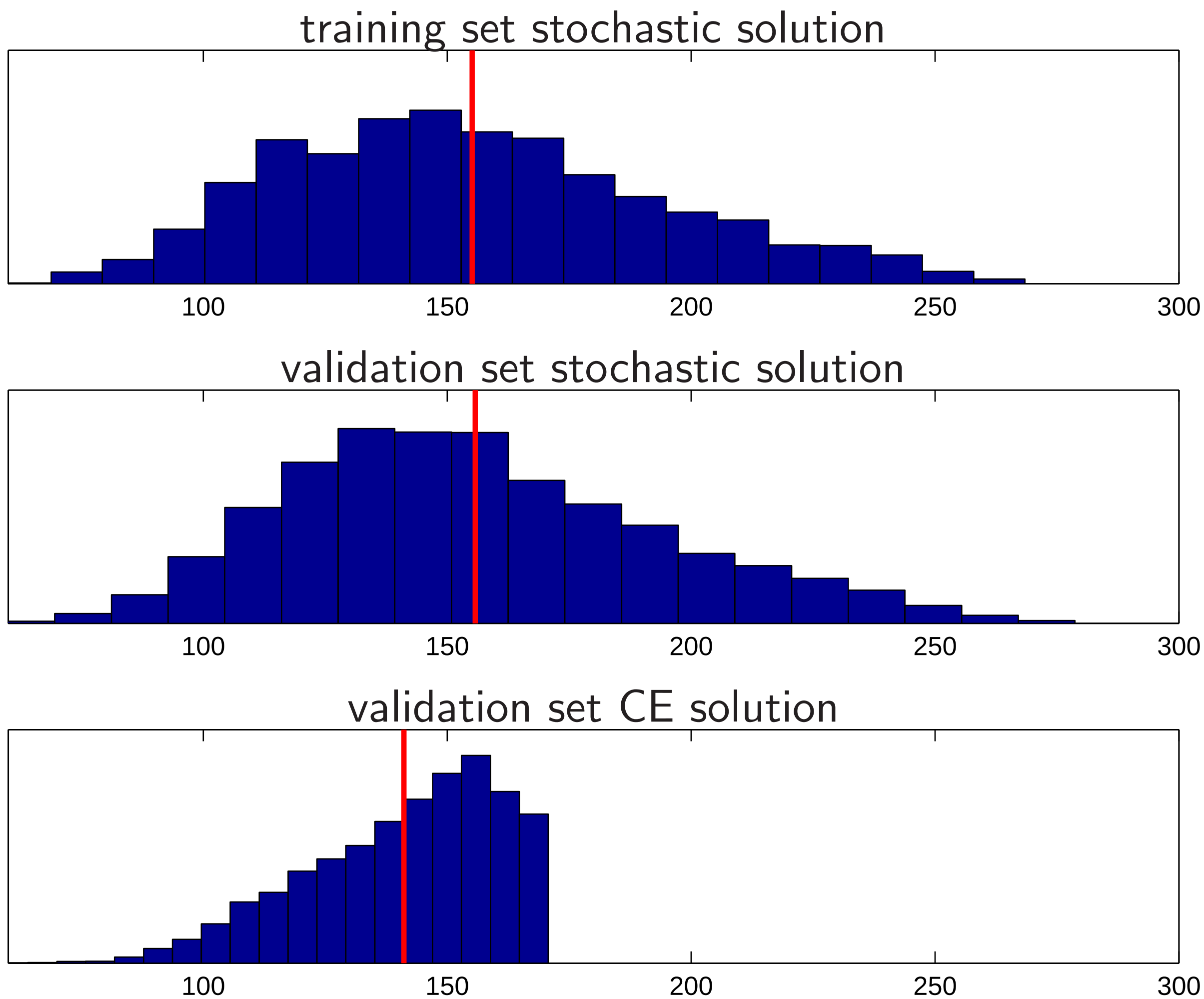
  A_{ij} is amount of raw material i needed per unit of finished product j
- product demand $d = (d_1, \ldots, d_m)$ is random, with known distribution
- product prices are $p = (p_1, \ldots, p_m)$, so total revenue is $p^T \min(d, q)$
- maximize (expected) net revenue (over optimization variables q, r):

$$\begin{array}{ll} \text{maximize} & \mathbf{E}\, p^T \min(d, q) - c^T r \\ \text{subject to} & r \succeq Aq, \quad q \succeq 0, \quad r \succeq 0 \end{array}$$

Problem instance

- problem instance has $n = 10$, $m = 5$, d log-normal
- certainty-equivalent problem yields upper bound 170.7
- we use Monte Carlo sampling with $N = 2000$ training samples
- validated with $M = 10000$ validation samples

	F_0
training	155.7
validation	155.1
CE (using $\bar{d}$)	170.7
CE validation	141.1


training set stochastic solution
100
150
200
250
300
validation set stochastic solution
100
150
200
250
300
validation set CE solution
100
150
200
250
300

Minimum average loss prediction

- $(x, y) \in \mathbf{R}^n \times \mathbf{R}$ have some joint distribution

- find weight vector $w \in \mathbf{R}^n$ for which $w^T x$ is a good estimator of y

- choose w to minimize expected value of a convex *loss function* l

$$J(w) = \mathbf{E}\, l(w^T x - y)$$

  - $l(u) = u^2$: mean-square error
  - $l(u) = |u|$: mean-absolute error

- we do not know joint distribution, but we have independent samples ('training data')

$$(x_i, y_i), \quad i = 1, \ldots, N$$

- Monte Carlo sampling method (called *training*):
  choose w to minimize sample average loss

$$w_{\rm sa} = \mathop{\rm argmin}_w \left(\frac{1}{N} \sum_{i=1}^{N} l(w^T x_i - y_i) \right)$$

  with associated sample average loss $J_{\rm sa}$

- validate predictor $y \approx w_{\rm sa}^T x$ on a different set of M samples:

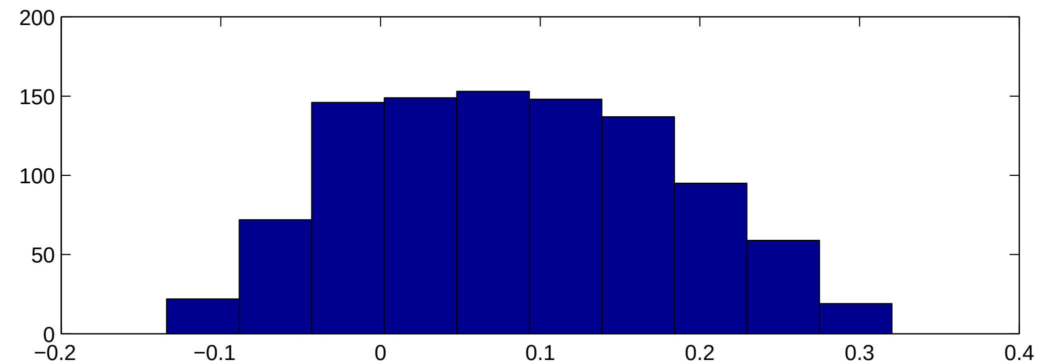
$$J_{\rm val} = \frac{1}{M} \sum_{i=1}^{M} l(w_{\rm sa}^T x_i^{\rm val} - y_i^{\rm val})$$

- if $J_{\rm sa} \approx J_{\rm val}$ (and M is large enough), we say predictor *generalizes*

Example

- $n = 10$; $N = 1000$ training samples; $M = 10000$ validation samples
- $l(u) = (u)_+ + 4(u)_-$ (under-predicting $4\times$ more expensive)

training set prediction errors



validation set prediction errors