Optimization

Màster de Fonaments de Ciència de Dades

Lecture X. Stochastic optimization methods

Gerard Gómez

Contents

Stochastic optimization problems

The problem of minimizing an objective function $f(\boldsymbol{x})$ can be formulated by finding the set

$$\boldsymbol{X}^* = \arg\min_{\boldsymbol{x}\in\boldsymbol{X}} f(\boldsymbol{x}) = \{\boldsymbol{x}^* \in \boldsymbol{X} \mid f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \boldsymbol{X}\}$$

Deterministic optimization assumes that perfect information is available about the objective function (and derivatives, if required). This information is used to determine the search direction and step size in a deterministic manner at each iteration of the procedure

Stochastic optimization problems are problems with **inherent uncertainty**, and that arise in areas such as:

- Situations where there are experimental random errors in the measurements
- Optimization procedures where Monte Carlo simulations are required to estimate the state or the parameters of a certain system

These problems are inappropriate for classical deterministic methods of optimization

Stochastic optimization problems

Some stochastic optimization methods **applications** include:

- **Engineering:** running computer simulations to refine the design of a car or an aircraft
- **Medicine:** designing laboratory experiments to extract the maximum information about the efficacy of a new drug or procedure
- **Traffic engineering:** setting the timing for the signals in a traffic network
- **Business:** making short- and long-term investment decisions in order to increase profit

Stochastic optimization problems and methods

The name **stochastic optimization** is applied when:

I. There is random noise in the measurements of $f(\boldsymbol{x})$

– and/or –

II. There is a random choice made in the search direction as the algorithm iterates toward a solution

We represent the random noise in the measurements of f at a given $\boldsymbol{x}$ as

$$F(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon(\boldsymbol{x})$$

where $\epsilon(\boldsymbol{x})$ represents the noise term

Note that the noise terms depend on $\boldsymbol{x}$. It implies that the common statistical assumption of independent, identically distributed (iid) noise does not necessarily apply since $\boldsymbol{x}$ will be changing as the search process proceeds

Noise in the objective function measurements arises in almost any case where physical system measurements or computer simulations are used

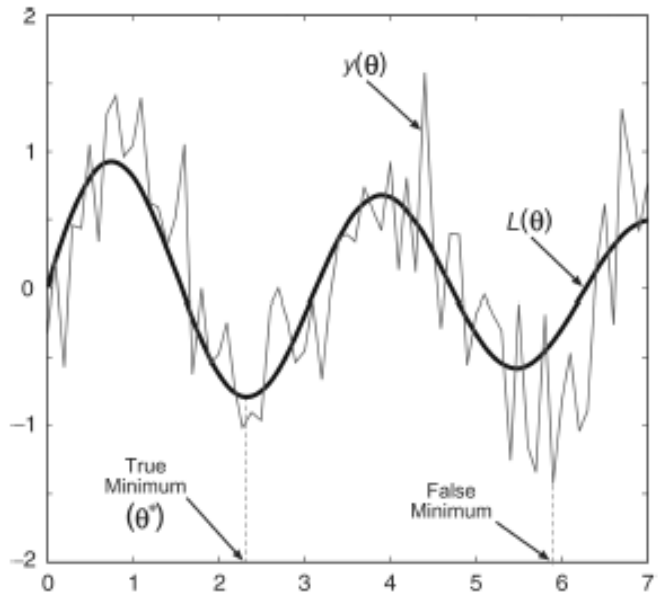
The random noise in the measurements of $f(\boldsymbol{x})$, fundamentally alters the search and optimization process because the algorithm is getting potentially misleading information

Stochastic optimization problems. Example

Consider the following cost function $f(x) = e^{-0.5x}\sin(2x)$

For $x \in [0, 7]$, the (unique) minimum occurs at $x^* = 3\pi/4 \approx 2.36$

Suppose that we are not able to calculate $f(x)$, obtaining instead only noisy measurements $F(x) = f(x) + \epsilon$ where the noises ϵ are iid with distribution $\mathcal{N}(0, 0.5^2)$



(Notation: $L(\theta) \equiv f(x)$, and $y(\theta) \equiv f(x) + \epsilon$)

We use an algorithm to find x^*, as well as a sample method of collecting one measurement at each increment of 0.1 over the interval [0,7]. The resut will be that the (false) minimum is at $x = 5.9$ far from the actual x^*

Stochastic optimization methods

- Stochastic optimization methods include methods with randomness in the search direction, such as genetic algorithms
- Concerning the randomness in the search direction, it is sometimes beneficial to deliberately introduce it into the search process as a means of speeding convergence and making the algorithm less sensitive to modeling errors
- The introduced randomness is usually created via computer-based pseudo random number generators
- One of the roles of injected randomness in stochastic optimization is to allow for "surprise" movements to unexplored areas of the search space that may contain an unexpectedly good $\boldsymbol{x}$ value. This is especially relevant in seeking out a global optimum among multiple local solutions

Stochastic optimization methods

- **Stochastic optimization methods** generate and/or use **random variables**[1]

  The **random variables** can **appear** in the **formulation** of the optimization problem itself, which involve **random objective functions** or **random constraints**

  These methods **generalize deterministic methods** for deterministic problems

[1] A **random (aleatory, stochastic) variable** is a function X whose values depend in some deterministic way on the set Ω of possible outcomes of a random event $X : \Omega \to \mathbb{R}$

Stochastic optimization methods

- Stochastic optimization **combines** both meanings of stochastic optimization: **stochastic problems** and **stochastic methods**
- In stochastic problems, knowledge that the objective function values are contaminated by random noise leads naturally to algorithms that use statistical tools (such as the computation of the espectation) to estimate the values of the function
- Like deterministic optimization, there is **no single solution method** that works well for all problems. Some assumptions, such as convexity, or limits on the size of the decision and outcome spaces, are needed to make problems tractable

Simple stochastic optimization methods. Direct random search methods

- Consider the problem of trying to find an optimal $x^* \in X$ based on noise-free measurements of $f(x)$
- Direct random search methods are the simplest methods of stochastic optimization
- These methods have a number of advantages relative to other search methods that include:
  1. Relative ease coding
  2. The need to only obtain f measurements (versus gradients or other information)
  3. Reasonable computational efficiency (especially for those direct search algorithms that make use of some local information in their search)
  4. Broad applicability to non-trivial objective functions and/or to x that may be continuous, discrete, or some hybrid form
  5. A strong theoretical foundation

Simple stochastic optimization methods. Blind random search

Blind random search. This is the simplest random search method, where the current sampling for $\boldsymbol{x}$ does not take into account the previous samples

The method can be implemented in recursive form as follows:

1. *Step 0 (Initialization).* Choose an initial value of $\boldsymbol{x}$, say $\boldsymbol{x}^0 \in \boldsymbol{X}$, either randomly or deterministically. (If random, usually a uniform distribution on $\boldsymbol{X}$ is used.) Calculate $f(\boldsymbol{x}^0)$, and $k = 0$

2. *Step 1.* Generate a new independent value $\boldsymbol{x}_{new}^{k+1} \in \boldsymbol{X}$, according to the chosen probability distribution. If $f(\boldsymbol{x}_{new}^{k+1}) < f(\boldsymbol{x}^k)$, set $\boldsymbol{x}^{k+1} = \boldsymbol{x}_{new}^{k+1}$, else $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$

3. *Step 2.* Stop if the maximum number of f evaluations has been reached, or the user is otherwise satisfied with the current estimate for $\boldsymbol{x}$ via appropriate stopping criteria; else, return to *Step 1* with the new k set to the former $k + 1$

Simple stochastic optimization methods. Blind random search

- The above algorithm converges almost surely (a.s.) to $\boldsymbol{X}^*$ under very general conditions
- Convergence alone is an incomplete indication of the performance of the algorithm, and it is also important to examine the rate of convergence
- Blind random search is a reasonable algorithm when $\boldsymbol{X}$ is low dimensional, but it can be shown that the method is generally a very slow algorithm for even moderately dimensioned $\boldsymbol{X}$. This is a direct consequence of the exponential increase in the size of the search space as the dimension p of $\boldsymbol{x}$, increases
- As an illustration, consider a case where $\boldsymbol{X} = [0, 1]^p$ (the p-dimensional hypercube with minimum and maximum values of 0 and 1 for each component of $\boldsymbol{x}$) and where one wishes to guarantee with probability 0.90 that each element of $\boldsymbol{x}$ is within 0.04 units of the optimal value

  As p increases from 1 to 10, there is an approximate 10^{10}-fold increase in the number of objective function evaluations required

Simple stochastic optimization methods. Localized random search

The localized random search algorithm changes the sampling strategy, which does not imply that the algorithm is only useful for local optimization. In fact, it has has global convergence properties. As with blind search, the algorithm may be used for continuous or discrete problems

1. *Step 0 (Initialization).* Choose an initial value of $\boldsymbol{x}$, say $\boldsymbol{x}^0 \in \boldsymbol{X}$, either randomly or deterministically. Set $k = 0$

2. *Step 1.* Generate an independent random vector $\boldsymbol{d}^k \in \mathbb{R}^p$ and add it to the current $\boldsymbol{x}$ value, $\boldsymbol{x}^k$. Check if $\boldsymbol{x}^k + \boldsymbol{d}^k \in \boldsymbol{X}$. If $\boldsymbol{x}^k + \boldsymbol{d}^k \notin \boldsymbol{X}$, generate a new $\boldsymbol{d}^k$ and repeat or, alternatively, move $\boldsymbol{x}^k + \boldsymbol{d}^k$ to the nearest valid point within $\boldsymbol{X}$. Let $\boldsymbol{x}_{new}^{k+1}$ equal $\boldsymbol{x}^k + \boldsymbol{d}^k \in \boldsymbol{X}$ or the afore mentioned nearest valid point in $\boldsymbol{X}$

3. *Step 2.* If $f(\boldsymbol{x}_{new}^{k+1}) < f(\boldsymbol{x}^k)$, set $\boldsymbol{x}^{k+1} = \boldsymbol{x}_{new}^{k+1}$, else $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k$

4. *Step 3.* Stop if the maximum number of f evaluations has been reached or the user is otherwise satisfied with the current estimate for $\boldsymbol{x}$ via appropriate stopping criteria; else, return to *Step 1* with the new k set to the former $k + 1$

Simple stochastic optimization methods. Localized random search

- For continuous problems, usually the (multivariate) normal distribution for generating the $\boldsymbol{d}^k$ is used, but other distributions can be used
- The distribution should have zero mean, and each component should have a variation (e.g., standard deviation) consistent with the magnitudes of the corresponding $\boldsymbol{x}$ elements. This allows the algorithm to assign roughly equal weight to each of the components of $\boldsymbol{x}$ as it moves through the search space
- Although not formally allowed in the convergence theory, it is often advantageous in practice if the variability in $\boldsymbol{d}^k$ is reduced as k increases
- The convergence theory for the localized algorithms tends to bemore restrictive than the theory for blind search

  The convergence is in the "in probability" sense, and the convergence theorem allows for more than one global minimum to exist in $\boldsymbol{X}$. Therefore, in general, the result provides no guarantee of $\boldsymbol{x}^k$ ever settling near any one value $\boldsymbol{x}^*$

Example. The blood-testing problem as a stochastic optimization problem

- **Data**
  - A large number N of individuals are subjected to a blood test
  - The probability that the test is positive is the same for all individuals and equal to p: $P(X = 1) = p$ and $P(X = 0) = 1 - p$, where X denotes the random variable
  - Individuals are stochastically independent
- **Blood-testing method**

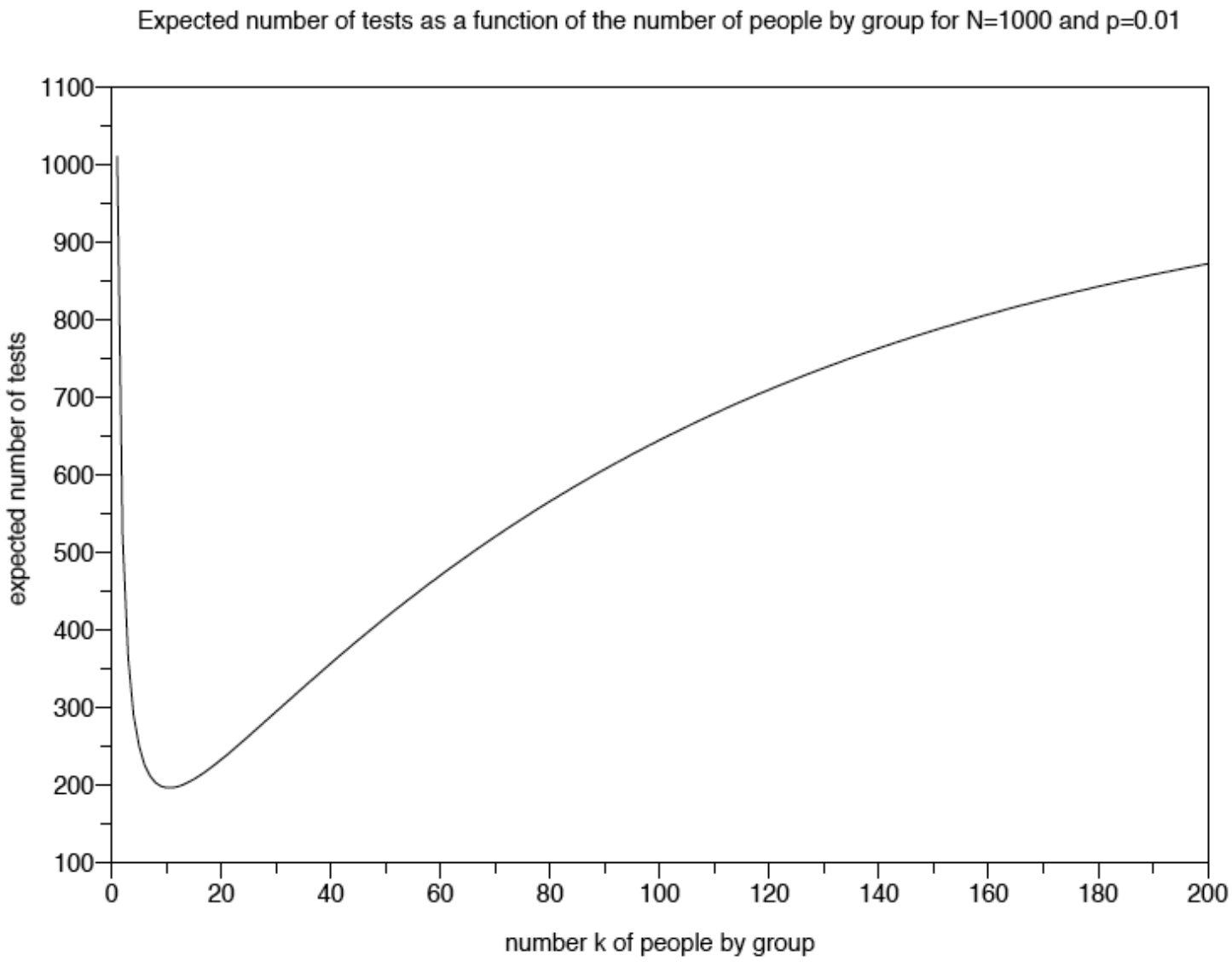
  The blood **samples of k individuals are pooled and analyzed together**
  - If the **test is negative**, this **one test suffices for the k individuals**
  - If the **test is positive**, each of the k persons must be tested separately, and $k + 1$ **tests are required** in all
- **Optimization problem**
  - Find the **value of k which minimizes the expected number of tests**
  - Find the minimal expected number of tests

What is the optimal value of k that minimizes the expected number of tests?

- For the first pool, $\{1, ..., k\}$, the test is
  - Negative, with probability $(1-p)^k$ (by independence) $\Rightarrow$ 1 test
  - Positive, with probability $1-(1-p)^k \Rightarrow k+1$ tests
- When the pool size k is small, compared to the number N of individuals, the blood samples $\{1, ..., N\}$ are split in approximately N/k groups, so that the expected number of tests is

$$J(k) \approx \frac{N}{k}[1 \times (1-p)^k + (k+1) \times (1-(1-p)^k)]$$

The expected number of tests displays a marked hollow



In practice, R. Dorfman achieved savings up to 80%

- As we have seen, the expected number of tests is

$$J(k) \approx \frac{N}{k}[1 \times (1-p)^k + (k+1) \times (1-(1-p)^k)]$$

- For small p: $(1-p)^k \approx 1 - kp$, $1-(1-p)^k \approx kp$, so

$$\frac{J(k)}{N} \approx \frac{1}{k}[1 - kp + (k+1)kp] = \frac{1}{k}[1 + k^2 p] = \frac{1}{k} + kp$$

  So

  - The optimal number of individuals per group is $k^* \approx \dfrac{1}{\sqrt{p}}$
  - The minimal expected number of tests is about

$$J^* = J(k^*) \approx N\left(\sqrt{p} + \frac{p}{\sqrt{p}}\right) = 2N\sqrt{p} < N$$

- In practice, R. Dorfman achieved savings up to 80%, compared to making N tests

$$p = 1/100, \quad \Rightarrow \quad k^* \approx 1/\sqrt{1/100} = 10 \quad \Rightarrow \quad J^* \approx 0.2\,N = N/5$$

Minimizing the tail value at a certain risc level

Remarks

- The **optimal number of individuals per group** k^* can also be considered a **random variable** X. Then we have two options
- Minimize the mathematical expectation $\mathbb{E}$ of the number of tests. It can be shown that the optimal number of individuals per group is 11
- Minimize the Tail Value at Risk of the number of tests at a certain risc level. It can be shown that for a risc level equal to 5%, the optimal number of individuals per group is 5

The Value at Risk of the random variable X at risk level $\lambda \in (0,1)$ is defined as

$$VR_\lambda(X) = \inf\{x \in \mathbb{R} \mid P(X > x) < \lambda\}$$

Intuitively, saying that the $VR_5(X) = k$ means that $X > k$ with probability of at most 5%

The Tail Value at Risk of X at level $\lambda \in (0,1)$ is

$$TVR_\lambda(X) = \mathbb{E}\left(X \mid X > VR_\lambda(X)\right)$$

Single and multiple stage stochastic optimization problems

In order to construct stochastic models of optimization problems, it is necessary to know

- The statistical characteristics of the random parameters of the problem
- Information on the **order** in which information enters, is stored, and is used

These produces two kinds of stochastic optimization problems: single and multiple stage problems

- Single stage problems (problems with a single time period)

  In single-stage problems, the dynamics of entering the initial information does not play a role, and the solution is accepted once and is not corrected

  Single stage problems **try to find a single optimal decision**, such as the best set of parameters for a statistical model given data

  Single stage problems are **usually solved with modified deterministic optimization methods**

Example of single stage problem

- Each morning, the newsvendor must decide **how many copies** $u \in U = \{0, 1, 2...\}$ **of the day's paper to order**. Here u is the **decision variable**
- The newsvendor will meet a **demand** $w \in W = \{0, 1, 2...\}$. The variable w is the **uncertainty**
- The newsvendor faces an economic tradeoff
  - He pays the unitary **purchasing cost** c per copy
  - He **sells a copy at price** p
  - If he remains with an unsold copy, it is worthless (perishable good)
- The newsvendor's **cost function** $J(u, w)$ depends both on the decision u and on the uncertainty w (difference between the purchasing cost and the selling):

$$J(u, w) = c\,u - \min\{u, w\}\,p = \max\{c\,u - p\,u, c\,u - p\,w\}$$

- The solution of

$$\min_{u \in U} J(u, w),$$

depends on the unknown quantity w!

Example of single stage problem (cont.)

- In the robust (or pessimistic) approach, the newsvendor **minimizes the worst value of the cost function,** $\max_{w \in W} J(u, w)$, this is

$$\min_{u \in U} \left(\max_{w \in W} J(u, w) \right)$$

- In the stochastic (or expected) approach, the newsvendor **minimizes the expected cost** $\mathbb{E}_W[J(u, W)]$[2], solving

$$\min_{u \in U} \mathbb{E}_W[J(u, W)]$$

as if nature played stochastically

[2] If X is a random variable, $\mathbb{E}[X]$ (also denoted by $\mathbb{E}\,X$) denotes the **expected value of** X (also called expectation, average, mean value, mean, or first moment). $\mathbb{E}[X]$ **is the probability-weighted average of all possible values of** X. If X is **discrete**, each possible value the random variable can assume is multiplied by its probability of occurring, and the resulting products are summed to produce the expected value. For a **continuous** random variable, an integral of the variable with respect to its probability density replaces the sum of the above definition

Example of single stage problem (cont.)

If the newsvendor minimizes the worst costs: $\min_{u \in U} (\max_{w \in W} J(u, w))$

Assume that:

- The demand w belongs to a set $W = [w^l, w^u]$
- The newsvendor knows this set $W = [w^l, w^u]$
- The worst costs is

$$\tilde{J}(u) = \max_{w \in W} J(u, w) = \max_{w \in W} \{cu - \min_{u \in U} \{u, w\}\, p\} = cu - \min_{u \in U} \{u, w^l\}\, p$$

- It can be seen that $u^* = w^l$ minimizes the above expression
- Once the newsvendor makes the optimal order, $u^* = w^l$, the optimal cost is

$$\begin{array}{rccc} J(u^*, \cdot): & W = [w^l, w^u] & \longrightarrow & \mathbb{R} \\ & w & \longrightarrow & (c - p) w^l \end{array}$$

which is no longer uncertain

Example of single stage problem (cont.)

If the newsvendor minimizes the expected costs: $\min_{u \in U} \mathbb{E}_W[J(u, W)]$

Assume that

- The demand is a random variable denoted by W
- The newsvendor knows the probability distribution P_W of the demand W
- The expected costs are

$$\tilde{J}(u) = \mathbb{E}_W[J(u, W)] = \mathbb{E}_W[cu - \min_{u \in U}\{u, w\}\, p]$$

- Then, is possible to compute the order u^* which minimizes $J(u)$

  This is done calculating $\tilde{J}(u+1) - \tilde{J}(u)$ and using the decumulative distribution function $u \to P_W(W > u)$

Single stage stochastic optimization

Formal concepts and notation

Let $\mathcal{X}$ be the domain of all **feasible decisions** and $x \in \mathcal{X}$ a specific decision

We would like to search over $\mathcal{X}$ to **find a decision that minimizes a cost function** F

Let ξ denote **random information** that is **available** only after the decision is made. The cost function, $F(x, \xi)$, will depend on x and ξ

Since we cannot directly optimize $F(x, \xi)$ we instead **minimize the expected value,** $\mathbb{E}[F(x, \xi)]$

The **general single stage stochastic optimization problem** becomes: find ζ^* such that

$$\zeta^* = \min_{x \in \mathcal{X}} \{ f(x) = \mathbb{E}[F(x, \xi)] \}$$

For single stage problems, it is **assumed that the decision space** $\mathcal{X}$ **is convex and the objective function** $F(x, \xi)$ **is convex in** x for any realization ξ

Sample average approximation for single stage stochastic optimization problems

Sample average approximation (SAA) is a two steps method that uses sampling and deterministic optimization to solve

$$\zeta^* = \min_{x \in \mathcal{X}} \{ f(x) = \mathbb{E}\left[F(x, \tilde{\xi}) \right] \}$$

- The first step in SAA is **sampling**. While directly computing the expected cost function $\mathbb{E}\left[F(x, \tilde{\xi}) \right]$ is not possible for most problems, it can be approximated through Monte Carlo sampling in some situations

  Let $\tilde{\xi}_i$, $i = 1, ..., n$ be a set of independent, identically distributed realizations of $\tilde{\xi} i$, and let $F(x, \tilde{\xi}_i)$ be the cost function realization for $\tilde{\xi}_i$. The expected cost function is approximated by the average of the realizations

$$\mathbb{E}[F(x, \xi)] \approx \frac{1}{n} \sum_{i=1}^{n} F(x, \tilde{\xi}_i)$$

- The second step in SAA is **search**. The right hand side of the above equation is deterministic, so deterministic optimization methods can be used to solve the approximate problem

$$\zeta_n^* = \min_{x \in \mathcal{X}} \left\{ f_n(x) = \frac{1}{n} \sum_{i=1}^{n} F(x, \tilde{\xi}_i) \right\}$$

Properties of sample average approximations

Like all stochastic optimization methods, SAA relies upon a collection of random variables to produce a statistical estimate

Most theoretical results follow directly from the the Law of Large Numbers and Central Limit Theorem due to the construction of SAA

- Under mild regularity conditions, for any fixed $\boldsymbol{x}$ **the limiting distribution of the SAA estimate is Gaussian** with mean $f(\boldsymbol{x})$ and variance $\sigma^2(\boldsymbol{x}) = \text{var}(F(\boldsymbol{x}, \boldsymbol{\xi}))/n < \infty$ is

$$\sqrt{n}\left[f_n(\boldsymbol{x}) - f(\boldsymbol{x})\right] \to \mathcal{N}(0, \sigma^2(\boldsymbol{x}))$$

- Under more restrictive conditions, including Lipschitz continuity of $F(\cdot, \boldsymbol{\xi})$, convexity of $F(\boldsymbol{x}, \boldsymbol{\xi})$, convexity of $\mathcal{X}$, and $f(\boldsymbol{x})$ having a unique optimum, a **similar result holds for the optimal values**

$$\sqrt{n}\left[\zeta_n^* - \zeta^*\right] \to \mathcal{N}(0, \sigma^2(\boldsymbol{x}))$$

The limiting Gaussian distribution can be used to determine the number of samples needed to generate an ϵ-optimal solution with at least probability $1 - \alpha$

Multistage problems

- **Multistage problems** (problems with multiple time periods)

  Multistage problems try to **find an optimal sequence of decisions**

  $$x_t, \quad t = 0, ..., T$$

  that minimize an expected cost function. The subscript t denotes the time at which decision x_t is made

  Usually decisions and random outcomes at time t affect the value of future decisions

  An **example** would be making a **move in a chess game**. With a move, the player may capture one of his opponent's pieces, change his board position, and alter his possible future moves. He needs to account for these issues to select the move that maximizes his probability of winning

  Two-stage problems are the most widespread models of control processes

  The dependence of future decisions on random outcomes, makes **direct modification of deterministic methods difficult in multistage problems**

  Multistage methods are more reliant on statistical approximation and strong assumptions about problem structure

Two-stage linear stochastic programming problem

The **linear programming problem**

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \textbf{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \tilde{T}\boldsymbol{x} = \tilde{\boldsymbol{\xi}} \end{aligned}$$

with $\boldsymbol{c}, \boldsymbol{x} \in \mathbb{R}^n$, A is a $m \times n$ matrix, $\boldsymbol{b} \in \mathbb{R}^m$, $\tilde{T}$ is a $d \times n$ matrix, and $\tilde{\boldsymbol{\xi}} \in \mathbb{R}^d$

The parameters with a **tilde over them are stochastic variables**. We know their **probability distribution, but we don't know their values exactly**

This problem is **ill-defined**, since a solution $\boldsymbol{x}$ that is optimal for one realisation of $\tilde{T}$ and $\tilde{\boldsymbol{\xi}}$ may even be infeasible for another

Two-stage linear stochastic programming problem

In the 2-stage stochastic programming approach, one should think of the decision process taking place in two stages

- In the first, values for the first stage variables $\boldsymbol{x}$ are chosen
- In the second, upon a realisation of the random parameters, a recourse action is taken in case of infeasibilities

  – To choose the optimal action given the infeasibilities, costs are attached to the various possible recourse actions (second stage problem)

  – **The expected cost of the optimal recourse action is added to the objective function**

Two-stage linear stochastic programming problem

A **generic mathematical programming formulation for this linear problem is**

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} + Q(\boldsymbol{x}) \\ \textbf{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

with

$$Q(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\min_{\boldsymbol{y} \in \mathbb{R}^p} \{\tilde{\boldsymbol{q}}^T \boldsymbol{y} \mid W\boldsymbol{y} = \tilde{T}\boldsymbol{x} - \tilde{\boldsymbol{\xi}}\}]$$

Here

- $\tilde{\boldsymbol{q}} \in \mathbb{R}^p$, W is a $d \times p$ matrix
- $\boldsymbol{x} \in \mathbb{R}^n$ is the first-stage decision variable vector
- $\boldsymbol{y} \in \mathbb{R}^m$ is the second-stage decision variable vector
- $\boldsymbol{\xi}(\boldsymbol{q}, \tilde{T}, W)$ contains the data of the second-stage problem

We assume that W is such that for any $\boldsymbol{x}$ and any realisation of $\tilde{T}$ and $\tilde{\boldsymbol{\xi}}$ there exists a feasible solution $\boldsymbol{y}$ in the second stage problem

Two stage stochastic optimization

The **general two-stage stochastic optimization problem** can be formulated as

$$\min_{x \in \mathcal{X}} \{ f(x) + \mathbb{E}_{\tilde{\xi}}[Q(x, \tilde{\xi})] \}$$

where

$$Q(x, \tilde{\xi}) = \min_{y \in \mathbb{R}^p} \{ \tilde{q}(y, \tilde{\xi}) \mid \tilde{T}(\tilde{\xi})x + W(\tilde{\xi})y = h(\tilde{\xi}) \}$$

is the optimal value of the second-stage problem

In this formulation

- At the first stage we have to **make a "here-and-now" decision x** before the realization of the uncertain data $\tilde{\xi}$, viewed as a random vector, is known
- At the second stage, after a realization of $\tilde{\xi}$ becomes available, we optimize our behavior by solving an appropriate optimization problem

Multistage stochastic optimization

- Mathematically, we can describe **multistage stochastic optimization problems** as an iterated expectation

$$\zeta^* = \min_{x_0 \in \mathcal{X}_0} \mathbb{E}\left[\inf_{x_1 \in \mathcal{X}_1(x_0, \xi_1)} F_1(\boldsymbol{x}_1, \boldsymbol{\xi}_1) + \mathbb{E}\left[\ldots + \mathbb{E}\left[\inf_{x_T \in \mathcal{X}_T(x_{0:T-1}, \xi_{1:T})} F_T(\boldsymbol{x}_T, \boldsymbol{\xi}_T)\right]\right]\right]$$

  - T is the number of time periods
  - $\boldsymbol{x}_{0:t}$ is the collection of all decisions between 0 and t
  - $\boldsymbol{\xi}_t$ is a random outcome observable at time t
  - $\mathcal{X}_t(\boldsymbol{x}_{0:t-1}, \boldsymbol{\xi}_{1:t})$ is a decision set that depends on all decisions and random outcomes between times 0 and t
  - $F_t(\boldsymbol{x}_t, \boldsymbol{\xi}_t)$ is a cost function for time period t that depends on the decision and random outcome for period t
  - The time horizon T may be either finite or infinite

- Unlike the methods for single stochastic optimization, there are no multistage solution methods that work well for all problems within a broad class, like convex problems or Markov decision processes

Summary of multistage stochastic optimization

- General two-stage linear programming problem

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} + Q(\boldsymbol{x}) \\ \text{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \end{aligned}$$

with

$$Q(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\xi}}[\min_{\boldsymbol{y} \in \mathbb{R}^p} \{\tilde{\boldsymbol{q}}^T \boldsymbol{y} \mid W\boldsymbol{y} = \tilde{T}\boldsymbol{x} - \tilde{\boldsymbol{\xi}}\}]$$

- General two-stage stochastic optimization problem

$$\min_{\boldsymbol{x} \in X} \{g(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{\xi}}[Q(\boldsymbol{x}, \boldsymbol{\xi})]\}$$

with

$$Q(\boldsymbol{x}, \boldsymbol{\xi}) = \min_{\boldsymbol{y}} \{\boldsymbol{q}(\boldsymbol{y}, \boldsymbol{\xi}) \mid T(\boldsymbol{\xi})\boldsymbol{x} + W(\boldsymbol{\xi})\boldsymbol{y} = h(\boldsymbol{\xi})\}$$

is the optimal value of the second-stage problem

- General multistage stochastic optimization problems

$$\zeta^* = \min_{x_0 \in \mathcal{X}_0} \mathbb{E}\left[\inf_{x_1 \in \mathcal{X}_1(x_0, \xi_1)} F_1(\boldsymbol{x}_1, \boldsymbol{\xi}_1) + \mathbb{E}\left[\ldots + \mathbb{E}\left[\inf_{x_T \in \mathcal{X}_T(x_{0:T-1}, \xi_{1:T})} F_T(\boldsymbol{x}_T, \boldsymbol{\xi}_T)\right]\right]\right]$$

The stochastic gradient method

- We want to minimize the function $f(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$
- A deterministic gradient method computes the gradient exactly

$$x_{k+1} = x_k - \alpha_k \nabla f_i(x_k) = x_k - \alpha_k \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_k)$$

  - Computing the exact gradient is $\mathcal{O}(N)$
  - We can get convergence with constant α_k or using line-search

- A stochastic gradient method estimates the gradient from a sample $j_k \in \{1, 2, ..., N\}$

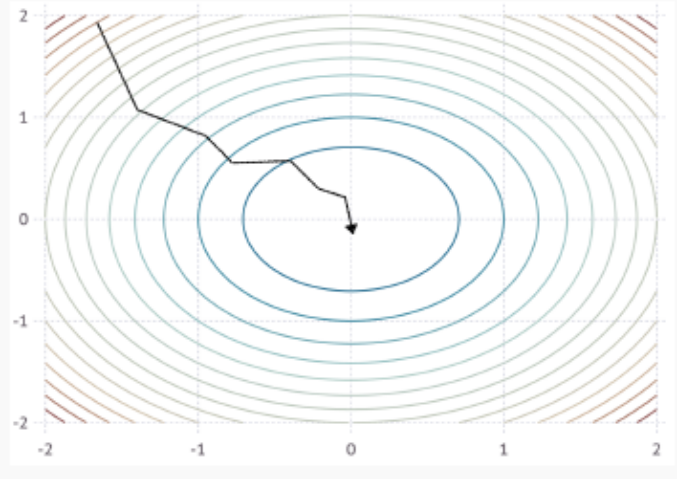
$$x_{k+1} = x_k - \alpha_k \nabla f_i(x_k) = x_k - \alpha_k \frac{1}{n}\sum_{j_k=1}^{n} \nabla f_{j_k}(x_k)$$

  - Note that this gives an unbiased estimate of the gradient

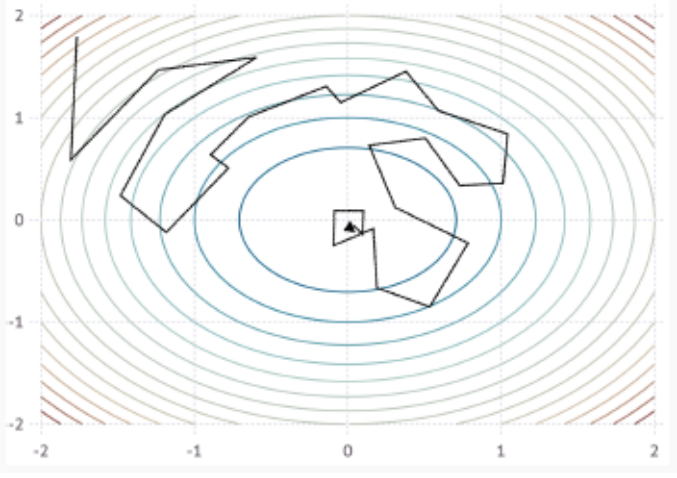
$$\mathbb{E}[f'_{j_k}(x)] = \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(x_i) = \nabla f(x)$$

  - The iteration cost no longer depends on N
  - Convergence requires $\alpha_k \to 0$

The stochastic gradient method



Deterministic gradient method



Stochastic gradient method

The stochastic subgradient method. Noisy unbiased subgradients

Noisy unbiased subgradients

Let

$$f : \mathbb{R}^m \to \mathbb{R}$$

be a **convex function**

We say that a random variable vector $\tilde{\boldsymbol{g}} \in \mathbb{R}^n$ is a **noisy unbiased subgradient** of f at $\boldsymbol{x}$ if

$$\mathbb{E}\left[\tilde{\boldsymbol{g}}\right] = \boldsymbol{g} \in \partial f(\boldsymbol{x})$$

According to the definition of a subgradient, this means

$$f(\boldsymbol{z}) \geq f(\boldsymbol{x}) + \left(\mathbb{E}\left[\tilde{\boldsymbol{g}}\right]\right)^T (\boldsymbol{z} - \boldsymbol{x}), \quad \forall \boldsymbol{z} \in \mathbb{R}^m$$

Equivalently, $\tilde{\boldsymbol{g}} \in \mathbb{R}^n$ is a noisy unbiased subgradient of f at $\boldsymbol{x}$ if it can be written as

$$\tilde{\boldsymbol{g}} = \boldsymbol{g} + \boldsymbol{v}$$

where $\boldsymbol{g} \in \partial f(\boldsymbol{x})$ and $\boldsymbol{v}$ **has zero mean**

The stochastic subgradient method. Noisy subgradients

If x is also a random variable, then we say that $\tilde{g}$ is a **noisy subgradient** of f at x (which is random) if for all z

$$f(z) \geq f(x) + (\mathbb{E}(\tilde{g} \mid x))^T (z - x)$$

holds almost surely (it happens with probability one or, in other words, the set of possible exceptions may be non-empty, but it has probability zero)

We can write this compactly as $\mathbb{E}(\tilde{g} \mid x) \in \partial f(x)$

The noise v can represent

- Error in computing a true subgradient
- Error that arises in Monte Carlo evaluation of a function defined as an expected value
- Measurement error

Remark. The conditional expectation of a random variable X, given another random variable Y, is another random variable equal to the average of the former over all, or eventually one, possible outcomes of Y: $\mathbb{E}[(X \mid Y)] \equiv \mathbb{E}(X \mid Y)$, $\mathbb{E}[(X \mid Y = y)] \equiv \mathbb{E}(X \mid Y = y) \equiv \mathbb{E}(X \mid y)$

The stochastic subgradient method

The **stochastic subgradient method** is essentially the subgradient method, but **using noisy subgradients** and a more **limited set of step size rules**

- The simplest case corresponds to **unconstrained minimization of a convex function** $f : \mathbb{R}^m \to \mathbb{R}$. In this case, the stochastic subgradient method uses the **standard update**

$$x^{k+1} = x^k - \alpha_k \tilde{g}^k$$

where x^k is the k-th iterate, $\alpha_k > 0$ is the k-th step size, and $\tilde{g}^k$ is a noisy subgradient of f at x^k

$$\mathbb{E}(\tilde{g}^k \mid x^k) = g^k \in \partial f(x^k)$$

- As in the ordinary subgradient method, we can have $f(x^k)$ **increase** during the algorithm, so we **keep track of the best point found so far**, and the associated function value

$$f^k_{best} = \min\{f(x^1), ..., f(x^k)\}$$

The stochastic subgradient method. Convergence

We will give **convergence** results, of the stochastic subgradient method, using **step sizes that are square-summable but not summable**

$$\alpha_k \geq 0, \quad \sum_{k=1}^{\infty} \alpha_k^2 = \|\alpha\|_2^2 < \infty \quad \sum_{k=1}^{\infty} \alpha_k = \infty$$

We will assume that

- There is an x^* that minimizes f
- $\mathbb{E}\left[\|x^1 - x^*\|_2^2\right] \leq R^2$ for a certain $R \geq 0$
- There is a G such that $\mathbb{E}\left[\|g^k\|_2^2\right] \leq G^2$ for all k

Under these ussumptions, we will see that we have **convergence in expectation**, this is

$$\mathbb{E}[f_{best}^k] \equiv \mathbb{E}\left[\min\{f(x^1), ..., f(x^k)\}\right] \to f^*$$

as $k \to \infty$

We also have **convergence in probability**: for any $\epsilon > 0$

$$\lim_{k \to \infty} Prob\left(f_{best}^k \geq f^* + \epsilon\right) = 0$$

Convergence

Proof:

We have

$$\begin{aligned}
\mathbb{E}\left[\left(\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\|_2^2 \mid \boldsymbol{x}^k\right)\right] &= \mathbb{E}\left[\left(\|\boldsymbol{x}^{k}-\alpha_k\tilde{\boldsymbol{g}}^k-\boldsymbol{x}^*\|_2^2 \mid \boldsymbol{x}^k\right)\right] \\
&= \|\boldsymbol{x}^k-\boldsymbol{x}^*\|_2^2 - 2\alpha_k\left[\mathbb{E}\left((\tilde{\boldsymbol{g}}^k)^T(\boldsymbol{x}^k-\boldsymbol{x}^*) \mid \boldsymbol{x}^k\right)\right] + \alpha_k^2\mathbb{E}\left[\left(\|\tilde{\boldsymbol{g}}^k\|_2^2 \mid \boldsymbol{x}^k\right)\right] \\
&= \|\boldsymbol{x}^k-\boldsymbol{x}^*\|_2^2 - 2\alpha_k\mathbb{E}\left[(\tilde{\boldsymbol{g}}^k \mid \boldsymbol{x}^k)^T(\boldsymbol{x}^k-\boldsymbol{x}^*)\right] + \alpha_k^2\mathbb{E}\left[\left(\|\tilde{\boldsymbol{g}}^k\|_2^2 \mid \boldsymbol{x}^k\right)\right] \\
&\leq \|\boldsymbol{x}^k-\boldsymbol{x}^*\|_2^2 - 2\alpha_k(f(\boldsymbol{x}^k)-f^*) + \alpha_k^2\mathbb{E}\left[\left(\|\tilde{\boldsymbol{g}}^k\|_2^2 \mid \boldsymbol{x}^k\right)\right]
\end{aligned}$$

where the inequality holds almost surely, and follows from

$$\mathbb{E}\left[(\tilde{\boldsymbol{g}}^k \mid \boldsymbol{x}^k)\right] \in \partial f(\boldsymbol{x}^k)$$

From the above inequality, and using the assumption $\mathbb{E}\left[\|\boldsymbol{g}^k\|_2^2\right] \leq G^2$, we get

$$\mathbb{E}\left[\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\|_2^2\right] \leq \mathbb{E}\left[\|\boldsymbol{x}^{k}-\boldsymbol{x}^*\|_2^2\right] - 2\alpha_k\left(\mathbb{E}\left[f(\boldsymbol{x}^k)-f^*\right]\right) + \alpha_k^2 G^2$$

Recursively applying this inequality yields

$$\mathbb{E}\left[\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^*\|_2^2\right] \leq \mathbb{E}\left[\|\boldsymbol{x}^{1}-\boldsymbol{x}^*\|_2^2\right] - 2\sum_{i=1}^{k}\alpha_i\left(\mathbb{E}\left[f(\boldsymbol{x}^i)-f^*\right]\right) + G^2\sum_{i=1}^{k}\alpha_i^2$$

Convergence (cont.)

Using

$$\mathbb{E}\left[\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2^2\right] \geq 0, \quad \mathbb{E}\left[\|\boldsymbol{x}^1 - \boldsymbol{x}^*\|_2^2\right] \leq R^2, \quad \sum_{i=1}^{k} \alpha_i^2 \leq \|\alpha\|_2^2$$

we get

$$2\sum_{i=1}^{k} \alpha_i \left(\mathbb{E}\left[f(\boldsymbol{x}^i) - f^*\right]\right) \leq R^2 + G^2\|\alpha\|_2^2$$

therefore, we have

$$\min_{i=1,\ldots,k} \mathbb{E}\left[f(\boldsymbol{x}^i) - f^*\right] \leq \frac{R^2 + G^2\|\alpha\|_2^2}{2\sum_{i=1}^{k} \alpha_i},$$

which shows that $\min_{i=1,\ldots,k} \mathbb{E}\left[f(\boldsymbol{x}^i)\right]$ converges to f^*, since $\sum_{k=1}^{\infty} \alpha_k = \infty$

Finally, we note that by Jensen's inequality [3] and the concavity of the minimum function, we have

$$\mathbb{E}\left[f_{best}^k\right] = \mathbb{E}\left[\min_{i=1,\ldots,k} f(\boldsymbol{x}^i)\right] \leq \min_{i=1,\ldots,k} \mathbb{E}\left[f(\boldsymbol{x}^i)\right] \to f^*$$

so $\mathbb{E}\left[f_{best}^k\right]$ also converges to f^*

[3] If ϕ is a convex function and X a random variable, then $\phi\left(\mathbb{E}[X]\right) \leq \mathbb{E}[\phi(X)]$

Convergence (cont.)

To show **convergence in probability**, we use Markov's inequality to obtain, for a fixed $\epsilon > 0$

$$Prob\left(f_{best}^{k} - f^{*} \geq \epsilon\right) \leq \frac{\mathbb{E}\left[f_{best}^{k} - f^{*}\right]}{\epsilon}$$

The righthand side goes to zero as $k \to \infty$, so the lefthand side does as well

The stochastic subgradient method. Example

We want to **minimize** the function

$$f(\boldsymbol{x}) = \max_{i=1,\ldots,m} (\boldsymbol{a}_i^T \boldsymbol{x} + \boldsymbol{b}_i), \quad \text{with} \quad \boldsymbol{x}, \boldsymbol{a}_i \in \mathbb{R}^n \quad \text{and} \quad \boldsymbol{b}_i \in \mathbb{R}$$

We use a stochastic subgradient algorithm

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \alpha_k \tilde{\boldsymbol{g}}^k$$

with noisy subgradient

$$\tilde{\boldsymbol{g}}^k = \boldsymbol{g}^k + \boldsymbol{v}^k, \quad \boldsymbol{g}^k \in \partial f(\boldsymbol{x}^k)$$

where the $\boldsymbol{v}^k$ are independent zero mean random variables

We will take $n = 20$ variables, $m = 100$ terms, and the problem data $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ generated from a unit normal distribution

The norm of the vectors $\boldsymbol{a}_i$ used is on the order of $\sqrt{20} \approx 4.5$, so $\|\boldsymbol{g}\| = \|\boldsymbol{a}_i\| \approx 4.5$

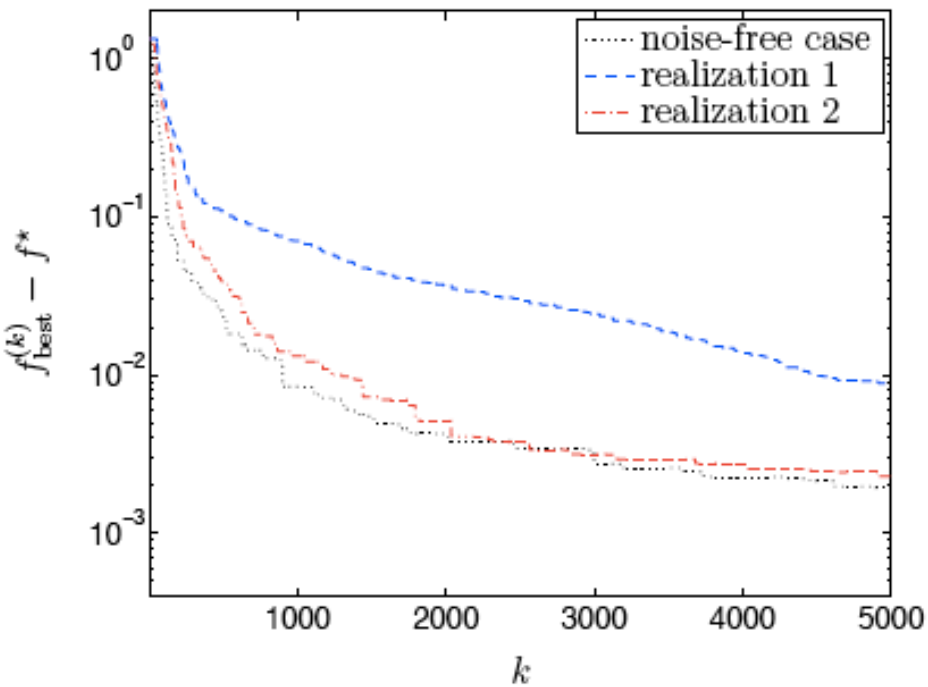
The noises $\boldsymbol{v}^k$ are independent and identically distributed normal random variables of 0 mean and variance $\sigma^2 = 0.5 \cdot Id$, and the subgradient noise is around the 25% of the true subgradient

The stochastic subgradient method. Example (cont.)

The initial point used is the vector $x^1 = 0$, and the step rule is the square summable but not summable rule $\alpha_k = 1/k$

The figure shows the convergence of the stochastic subgradient method for two realizations of the noisy subgradient process, together with the noise-free case for comparison

The figure also shows that convergence is only a bit slower with subgradient noise



The value of $f^* \approx 1.1$ has been obtained using linear programming

The stochastic subgradient method. Example (cont.)

For 100 realizations of the procedure for each value of k, the the error bars show the sample mean plus and minus one standard deviation of $\mathbb{E}\left[f_{best}^k - f^*\right]$ for k in multiples of 250