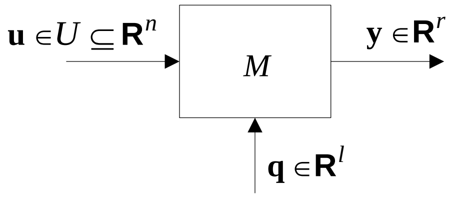
Optimization

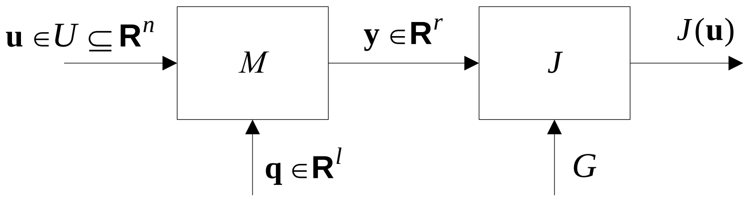
- We study parameter optimization algorithms.

(1) Generic problem definition

- Basic ingredients: *M, U, G*



- *Model* (*M*): (parameterized) mathematical representation of "plant"
- $\mathbf{y} = M(\mathbf{u})$ (deterministic optimization) or $\mathbf{y} = M(\mathbf{u}, \mathbf{q})$ (stochastic optimization with $\mathbf{q}$ random vector)
- *Admissible decisions* (*U*): admissible range of values of parameters



- *Goal* (*G*): a performance index or objective (function) expressed as a function of $\mathbf{u} \in U \subseteq \mathbf{R}^n$
- Performance index, $J : \mathbf{R}^n \to \mathbf{R}$ or $J : U \to \mathbf{R}$
- Generic optimization problem is

$$\min_{\mathbf{u} \in U} J(\mathbf{u}) .$$

(1) An element $\mathbf{u}^* \in U$ which achieves the min above is optimal and $J^* = J(\mathbf{u}^*)$ is the optimal or minimal value. We say $\mathbf{u}^* \in \arg\min_{\mathbf{u} \in U} J(\mathbf{u})$.

(2) $\mathbf{u}^* \in U$ is not necessarily unique.

(3) $\mathbf{u}^* \in U$ may not exist. If we are not sure about its existence, we write

$$J^* = \inf_{\mathbf{u} \in U} J(\mathbf{u}) \quad \text{or} \quad J^* = \sup_{\mathbf{u} \in U} J(\mathbf{u}) .$$

(4) $\min_{\mathbf{u} \in U} J(\mathbf{u}) = -\max_{\mathbf{u} \in U} \{-J(\mathbf{u})\}$ and $\max_{\mathbf{u} \in U} J(\mathbf{u}) = -\min_{\mathbf{u} \in U} \{-J(\mathbf{u})\}$.

(2) Existence of optimal elements

- (*<u>Compactness theorem</u>*) Suppose $U \subseteq \mathbf{R}^n$ is nonempty and *<u>compact</u>* and $J: U \to \mathbf{R}$ is *<u>continuous</u>*. Then an optimal element $\mathbf{u}^* \in U$ exists, i.e., $J^* = J(\mathbf{u}^*) = \min_{\mathbf{u} \in U} J(\mathbf{u})$.
- Without compactness we need coercivity or convexity for existence of the optimal elements.
- Suppose a function $J: \mathbf{R}^n \to \mathbf{R}$ is continuous. *J* is *<u>coercive</u>* if $\lim_{\|\mathbf{u}\| \to \infty} J(\mathbf{u}) \to +\infty$, i.e., *J* blows up uniformly in every direction.
- (*<u>Coercivity theorem</u>*) Suppose $U \subseteq \mathbf{R}^n$ is nonempty and *<u>closed</u>* and $J: U \to \mathbf{R}$ is *<u>continuous</u>* and *<u>coercive</u>*. Then an optimal element $\mathbf{u}^* \in U$ exists, i.e., $J^* = J(\mathbf{u}^*) = \min_{\mathbf{u} \in U} J(\mathbf{u})$.

(3) Uniqueness of the optimum

- (*<u>Convex set</u>*) A set $U \subseteq \mathbf{R}^n$ is convex if

$$\forall \mathbf{u}, \mathbf{v} \in U \text{ and } \forall \lambda \in [0,1],\ (1-\lambda)\mathbf{u} + \lambda \mathbf{v} \in U.$$

- (*<u>Convex function</u>*) A function $J: \mathbf{R}^n \to \mathbf{R}$ is (strictly, with <) convex if

$$\forall \mathbf{u}, \mathbf{v} \in U \text{ and } \forall \lambda \in [0,1],\ J\left[(1-\lambda)\mathbf{u} + \lambda \mathbf{v}\right] \le (1-\lambda) J(\mathbf{u}) + \lambda J(\mathbf{v}).$$

- *J* is *<u>concave</u>* if -*J* is convex.
- (*<u>Hessian lemma</u>*) Suppose *J* is continuously differentiable on an open convex set *U*. Then, *J*(**u**) is convex iff the Hessian matrix $\nabla^2 J(\mathbf{u})$ is positive semidefinite at each point $\mathbf{u} \in U$ where

$$\nabla^2 J(\mathbf{u}) := \begin{bmatrix} \frac{\partial^2 J}{\partial u_1^2} & \frac{\partial^2 J}{\partial u_1 \partial u_2} & \cdots & \frac{\partial^2 J}{\partial u_1 \partial u_n} \\ \frac{\partial^2 J}{\partial u_2 \partial u_1} & \frac{\partial^2 J}{\partial u_2^2} & \cdots & \frac{\partial^2 J}{\partial u_2 \partial u_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 J}{\partial u_n \partial u_1} & \frac{\partial^2 J}{\partial u_n \partial u_2} & \cdots & \frac{\partial^2 J}{\partial u_n^2} \end{bmatrix}.$$

- If *J* is convex, the optimum is unique.

(4) Local minimum

- Suppose $U \subseteq \mathbf{R}^n$ is open and $J : U \to \mathbf{R}$. A point $\mathbf{u}^* \in U$ is a *strong local minimum* of J over U if $J(\mathbf{u}) > J(\mathbf{u}^*)$ for all $\mathbf{u} \in U$ in some (open) neighbor of $\mathbf{u}^*$.
- A *weak local minimum* is defined with ≥.
- (*Theorem*) Suppose $U \subseteq \mathbf{R}^n$ is open and $J : U \to \mathbf{R}$ and $\mathbf{u}^* \in U$. $\mathbf{u}^*$ is a strong local minimum if $\nabla J(\mathbf{u}^*) = 0$ and $\nabla^2 J(\mathbf{u}) > 0$ (positive definite), where the gradient

$$\nabla J(\mathbf{u}^*) = \begin{bmatrix} \frac{\partial J}{\partial u_1} & \frac{\partial J}{\partial u_2} & \cdots & \frac{\partial J}{\partial u_n} \end{bmatrix}^T .$$

(5) Steepest descent algorithm

- Let $J : \mathbf{R}^n \to \mathbf{R}$ is differentiable and $\mathbf{v} \in \mathbf{R}^n$ is given with $\mathbf{v} \neq \mathbf{0}$. The *directional derivative* at a given point $\bar{\mathbf{u}} \in \mathbf{R}^n$ in the direction of $\mathbf{v}$ is

$$\left.\frac{dJ}{d\mathbf{v}}\right|_{\bar{\mathbf{u}}} = \left[\nabla J(\mathbf{u})\right]^T \hat{\mathbf{v}} \quad \text{where} \quad \hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}.$$

- Note that J is decreasing in the negative gradient direction since with $\mathbf{v} = -\nabla J(\mathbf{u})$,

$$\left.\frac{dJ}{d\mathbf{v}}\right|_{\bar{\mathbf{u}}} = -\left[\nabla J(\mathbf{u})\right]^T \frac{\nabla J(\mathbf{u})}{\|\nabla J(\mathbf{u})\|} = -\|\nabla J(\mathbf{u})\| \le 0 .$$

- $\mathbf{v} = -\frac{\nabla J(\mathbf{u})}{\|\nabla J(\mathbf{u})\|}$ is the *direction of maximal decrease*.
- (*Steepest descent algorithm*)
  (1) Initial guess, $\mathbf{u}^0$.
  (2) Iterate with $\mathbf{u}^{k+1} = \mathbf{u}^k - h\frac{\nabla J(\mathbf{u}^k)}{\|\nabla J(\mathbf{u}^k)\|}$ for $k = 1, 2, \ldots$ with $h > 0$ being the algorithm step size.
  (3) Stopping criterions may include

  smallness of $\left|J(\mathbf{u}^{k+1}) - J(\mathbf{u}^k)\right|$,

  smallness of $\left\|\mathbf{u}^{k+1} - \mathbf{u}^k\right\|$,

  smallness of $\left\|\nabla J(\mathbf{u}^k)\right\|$,

  condition on $\nabla^2 J(\mathbf{u}^k)$,
  some criteria for N consecutive times,
  and combinations of above.

- Step size
  (1) Fixed step size
  (2) Dynamic step size by line search (uniform search, sequential search, dichotomous search, golden section method, Fibonacci search)
- Criticism of steepest descent algorithm
  (1) Myopic
  (2) Possible oscillation
  (3) May require multiple restarts to avoid a local minimum
  (4) Slow convergence
- Good performance when $\mathbf{u}^0$ is far away from the optimum.

(6) Generalized Newton-Raphson

- Pretend J is quadratic near $\mathbf{u}^k$ and $\mathbf{u}^{k+1} = \mathbf{u}^k + \Delta\mathbf{u}$. Then,

$$J(\mathbf{u}^k + \Delta\mathbf{u}) = J(\mathbf{u}^k) + \left[\nabla J(\mathbf{u}^k)\right]^T \Delta\mathbf{u} + \frac{1}{2}(\Delta\mathbf{u})^T \nabla^2 J(\mathbf{u}^k)\Delta\mathbf{u} + \text{h.o.t.}$$

where h.o.t. is 0 from the quadratic assumption. Now we optimize w.r.t. $\Delta\mathbf{u}$, i.e., let

$$\frac{\partial J(\mathbf{u}^k + \Delta\mathbf{u})}{\partial \Delta\mathbf{u}} = \nabla J(\mathbf{u}^k) + \nabla^2 J(\mathbf{u}^k)\Delta\mathbf{u} = 0$$

and

$$\Delta\mathbf{u} = -\left[\nabla^2 J(\mathbf{u}^k)\right]^{-1} \nabla J(\mathbf{u}^k).$$

- (*<u>Generalized Newton algorithm</u>*)
  (1) Initial guess, $\mathbf{u}^0$.
  (2) Iterate with $\mathbf{u}^{k+1} = \mathbf{u}^k - \left[\nabla^2 J(\mathbf{u}^k)\right]^{-1} \nabla J(\mathbf{u}^k)$ for k = 1, 2, ...
  (3) Check the conditioning of the matrix $\left[\nabla^2 J(\mathbf{u}^k)\right]^{-1}$.
  (4) Stopping criterions
- Good performance near the optimum but computationally expensive

(7) Gauss-Newton algorithm

- Computation of Hessian is often too expensive or impossible.
- Approximate Hessian as $\left[\nabla^2 J(\mathbf{u}^k)\right] \approx \left[\nabla J(\mathbf{u}^k)\right]^T \nabla J(\mathbf{u}^k)$.
- In order to handle the ill-conditioning problem of $\left[\nabla J(\mathbf{u}^k)\right]^T \nabla J(\mathbf{u}^k)$, use *<u>Levenberg-</u>*

Marquardt method where $\left[\nabla^2 J(\mathbf{u}^k)\right] \approx \left\{\left[\nabla J(\mathbf{u}^k)\right]^T \nabla J(\mathbf{u}^k) + \delta\mathbf{I}\right\} = \tilde{\mathbf{H}}$ for some positive constant δ.

- (*Gauss-Newton algorithm*)
  (1) Initial guess, $\mathbf{u}^0$.
  (2) Iterate with $\mathbf{u}^{k+1} = \mathbf{u}^k - \tilde{\mathbf{H}}^{-1}\nabla J(\mathbf{u}^k)$ for k = 1, 2, ...
  (3) Check the conditioning of the matrix $\left\{\left[\nabla J(\mathbf{u}^k)\right]^T \nabla J(\mathbf{u}^k)\right\}^{-1}$ and adjust δ if necessary.
  (4) Stopping criterions

(8) Conjugate direction (gradient) algorithm

- Let $\mathbf{A}$ be an $n \times n$ positive definite symmetric matrix. A pair of nonzero vectors $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$ are *mutually conjugate* w.r.t. $\mathbf{A}$ if $\mathbf{x}^T\mathbf{A}\mathbf{y} = 0$.
- Orthogonality is a special case with $\mathbf{A} = \mathbf{I}$.
- Eigenvectors of $\mathbf{A}$ are mutually conjugate.
- If $\left\{\mathbf{v}^k\right\}_{k=0}^{n-1}$ are mutually conjugate, then they are linearly independent.
- Conjugate direction was found to be better than the steepest direction in many cases. There are many possible conjugate directions.
- (*Conjugate direction algorithm in general*)
  (1) Initial guess, $\mathbf{u}^0$.
  (2) Iterate with $\mathbf{u}^{k+1} = \mathbf{u}^k + h_k \dfrac{\mathbf{v}^k}{\left\|\mathbf{v}^k\right\|}$ for k = 1, 2, ... with $h_k > 0$ being the algorithm step size.
  (3) Stopping criterions
- (*Conjugate direction algorithm - Fletcher Reeves*)
  (1) Initial guess, $\mathbf{u}^0$ and $\mathbf{v}^0 = -\nabla J(\mathbf{u}^0)$.
  (2) Iterate with $\mathbf{u}^{k+1} = \mathbf{u}^k + h_k \dfrac{\mathbf{v}^k}{\left\|\mathbf{v}^k\right\|}$ for k = 1, 2, ... with $h_k > 0$ being the algorithm step size. Compute $\mathbf{v}^{k+1} = -\nabla J(\mathbf{u}^{k+1}) + \dfrac{\left\|\nabla J(\mathbf{u}^{k+1})\right\|^2}{\left\|\nabla J(\mathbf{u}^k)\right\|^2}\mathbf{v}^k$.
  (3) Stopping criterions

- In conjugate direction algorithm, we have the following condition.

$$\mathbf{v}^k \perp \nabla J(\mathbf{u}^{k+1}) \quad \text{or} \quad \left(\mathbf{v}^k\right)^T \nabla J\left(\mathbf{u}^k + h_k \frac{\mathbf{v}^k}{\left\|\mathbf{v}^k\right\|}\right) = 0 .$$