Statistics and optimization
Unconstrained Optimization
Gradient-based iterative algorithms

Sylvain Sardy

Université de Genève

1. Iterative descent algorithms

Consider unconstrained optimization of

$$f : \mathbb{R}^N \to \mathbb{R},$$

a continuously differentiable function.

Iterative descent algorithms
Start with $\mathbf{x}^0 \in \mathbb{R}^N$ (a good guess may reveal crucial), then generate a sequence $\mathbf{x}^1, \mathbf{x}^2, \ldots,$ such that

$$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k), \tag{1}$$

so as to decrease the cost at each iteration and aim to converge to a local minimum.

Motivation for descent-based algorithm

Let us consider $\mathbf{x}_\alpha$ be a point near $\mathbf{x}^k$.
From Taylor first order expansion near $\mathbf{x}^k$ we have

$$f(\mathbf{x}_\alpha) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^{\mathrm{T}}(\mathbf{x}_\alpha - \mathbf{x}^k) + o(\|\mathbf{x}_\alpha - \mathbf{x}^k\|)$$

Hence if $\mathbf{d}^k = -\nabla f(\mathbf{x}^k)$ and $\mathbf{x}_\alpha = \mathbf{x}^k + \alpha \mathbf{d}^k$, then

$$f(\mathbf{x}_\alpha) = f(\mathbf{x}^k) - \alpha \|\mathbf{d}^k\|_2^2 + o(\alpha)$$

Hence for $\alpha > 0$ small enough, (1) is guaranteed with $\mathbf{x}^{k+1} = \mathbf{x}_\alpha$, provided we are not already at a stationary point.

Note that we can be less restrictive to guarantee a descent sequence.
Indeed if $\mathbf{x}_\alpha = \mathbf{x}^k + \alpha \mathbf{d}$ with $\mathbf{d}^{\mathrm{T}} \nabla f(\mathbf{x}^k) < 0$, then

$$f(\mathbf{x}_\alpha) = f(\mathbf{x}^k) + \alpha \nabla f(\mathbf{x}^k)^{\mathrm{T}} \mathbf{d} + o(\alpha),$$

which would also lead to a descent sequence for $\alpha > 0$ small enough.

So if a descent direction $\mathbf{d}$ makes an angle greater than 90 degree with the gradient, then $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \mathbf{d}$ leads to a descent sequence for $\alpha > 0$ small enough.

Iterative gradient-based algorithms

Start with $\mathbf{x}^0 \in \mathbb{R}^N$, then generate a sequence $\mathbf{x}^1, \mathbf{x}^2, \ldots,$ according to

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{d}^k \quad \text{with} \quad \nabla f(\mathbf{x}^k)^T \mathbf{d}^k < 0, \ \alpha^k > 0,$$

Note that if $\mathbf{d}^k = -D^k \nabla f(\mathbf{x}^k)$ with $D^k > 0$, then $\nabla f(\mathbf{x}^k)^T \mathbf{d}^k < 0$ since

$$\nabla f(\mathbf{x}^k)^T \mathbf{d}^k = -\nabla f(\mathbf{x}^k)^T D^k \nabla f(\mathbf{x}^k) < 0$$

provided $\nabla f(\mathbf{x}^k) \neq \mathbf{0}$.

Most gradient-based algorithms select the stepsize $\alpha^k > 0$ such that the algorithm is also a descent one.

2. Selecting the stepsize α^k

Minimization rule (line search)
Minimize the cost function in direction $\mathbf{d}^k$

$$\min_{\alpha \geq 0} f(\mathbf{x}^k + \alpha \mathbf{d}^k).$$

The univariate problem can be solved approximately and expensively with a golden section search, Newton, etc., or may have a closed form expression.

Note: at the solution $\alpha^\star$ we have $\nabla f(\mathbf{x}^k + \alpha^\star \mathbf{d}^k)^{\mathrm{T}} \mathbf{d}^k = 0$, or

$$\nabla f(\mathbf{x}^{k+1})^{\mathrm{T}} \mathbf{d}^k = 0. \tag{2}$$

Example: Consider the ridge regression estimator of regression coefficients $\mathbf{x}$ solution to

$$\min_{\mathbf{x}\in\mathbb{R}^{\mathrm{P}}} \frac{1}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_2^2,$$

where $\mathbf{y} \in \mathbb{R}^{\mathrm{N}}$, A is an $N \times P$ regression matrix and $\lambda \geq 0$ is the regularization parameter. A descent algorithm with the minimization rule will iteratively search the scalar α^k solution to

$$\min_{\alpha\geq 0} \frac{1}{2}\|\mathbf{y} - A(\mathbf{x}^k + \alpha\mathbf{d}^k)\|_2^2 + \lambda\|\mathbf{x}^k + \alpha\mathbf{d}^k\|_2^2.$$

Here the univariate problem has a closed form expression.

Armijo rule

Fix scalars s, β, σ, with $0 < \beta < 1$ and $0 < \sigma < 1$.

Then choose $\alpha^{(k)} = \beta^m$, where m is the first integer for which

$$f(\mathbf{x}^{(k)}) - f(\mathbf{x}^{(k)} + \beta^m s\mathbf{d}^{(k)}) \geq -\sigma\beta^m s\nabla f(\mathbf{x}^{(k)})^{\mathrm{T}}\mathbf{d}^{(k)}.$$

Typically $\sigma \in [10^{-5}, 10^{-1}]$, $\beta \in [1/10, 1/2]$, $s = 1$.

Armijo is the most accepted method.

3. Selecting the descent direction $\mathbf{d}^k$

Consider iterates of the form

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k D^k \nabla f(\mathbf{x}^k),$$

where D^k a symmetric positive definite matrix.

Various methods exist depending on the choice of the matrix D^k.

Steepest descent: $D^k = I$
Simple, no overhead calculations, but often slow.

Analysis of convergence rate for quadratic functions: suppose

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}} Q \mathbf{x},$$

where Q is a symmetric positive definite matrix with smallest eigenvalue λ_1 and largest eigenvalue λ_N.
The solution is $\mathbf{x}^\star = \mathbf{0}$ and $f(\mathbf{x}^\star) = 0$.

The steepest descent iterates are

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \nabla f(\mathbf{x}^k) = (I - \alpha^k Q)\mathbf{x}^k.$$

Stepwise rule:

- constant: If $\alpha = \frac{2}{\lambda_N + \lambda_1}$, then $\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|}{\|\mathbf{x}^k - \mathbf{x}^\star\|} \leq \frac{\lambda_N - \lambda_1}{\lambda_N + \lambda_1}$.
  This is the best convergence rate bound for steepest descent with constant step size α.

Proof:

- optimal: If α^k is chosen to minimize the line search at each iteration, then $\frac{f(\mathbf{x}^{k+1})-f(x^\star)}{f(\mathbf{x}^k)-f(x^\star)} \leq \left(\frac{\lambda_N-\lambda_1}{\lambda_N+\lambda_1}\right)^2$.

Proof:

Newton's method: $D^k = (\nabla^2 f(\mathbf{x}^k))^{-1}, k = 0, 1, \ldots$
As explained on page 5, it requires $\nabla^2 f(\mathbf{x}^k) > 0$.
Quadratic approximation of f around current iterate $\mathbf{x}^k$ based on Taylor expansion:

$$f(\mathbf{x}) \approx f^k(\mathbf{x}) = f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^{\mathrm{T}}(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^{\mathrm{T}} \nabla^2 f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k).$$

Recall: if $f : \mathbb{R}^{\mathrm{N}} \to \mathbb{R}$ is twice continuously differentiable on an open ball X centered at $\mathbf{x}$, then

- for all $\mathbf{y}$ such that $\mathbf{x} + \mathbf{y} \in X$, then

$$f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{y} + \frac{1}{2}\mathbf{y}^T \nabla^2 f(\mathbf{x})\mathbf{y} + o(\|\mathbf{y}\|^2).$$

- Moreover there exists $\alpha \in [0, 1]$ such that

$$f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{y} + \frac{1}{2}\mathbf{y}^T \nabla^2 f(\mathbf{x} + \alpha \mathbf{y})\mathbf{y}.$$

Hence, to find the next iterate, solve first order optimality conditions for the approximation: $\nabla f^k(\mathbf{x}) = \mathbf{0}$, i.e.,

$$\nabla f(\mathbf{x}^k) + \nabla^2 f(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) = \mathbf{0}$$

or

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k) \quad \text{with} \quad \alpha^k = 1.$$

This is the pure Newton method.

Example: Generalized linear model for independent Poisson data with canonical log-link

$$Y_n \sim \mathrm{Poisson}(\mu_n) \quad \text{with} \quad \begin{cases} \mu_n = \exp(\eta_n) \\ \eta_n = \alpha_0 + \sum_{p=1}^{P} \alpha_p x_{np} \end{cases} .$$

The maximum likelihood estimate solves

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{\mathrm{P}}} \sum_n -y_n \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{x}_n + \exp(\boldsymbol{\alpha}^{\mathrm{T}} \mathbf{x}_n),$$

which is a strictly convex cost (provided $\mathrm{Ker}(X) = \{\mathbf{0}\}$) defined on an open and convex domain.

To apply Newton's method, the gradient and Hessian are

$$\begin{aligned}
\nabla_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}^k) &= \sum_n -y_n \mathbf{x}_n + \exp(\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\alpha}^k)\mathbf{x}_n \\
&= X^{\mathrm{T}}(\exp(X\boldsymbol{\alpha}^k) - \mathbf{y}) \\
&= X^{\mathrm{T}}(\boldsymbol{\mu}^k - \mathbf{y}) \\
\nabla^2 f(\boldsymbol{\alpha}^k) &= \sum_n \exp(\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\alpha}^k)\mathbf{x}_n \mathbf{x}_n^{\mathrm{T}} \\
&= X^{\mathrm{T}} W^k X, \quad \text{with} \quad W^k = \mathrm{diag}(\exp(\mathbf{x}_n^{\mathrm{T}} \boldsymbol{\alpha}^k)) > 0.
\end{aligned}$$

So the pure Newton's iterates are

$$\boldsymbol{\alpha}^{k+1} = \boldsymbol{\alpha}^k - (XW^kX^{\mathrm{T}})^{-1}(X^{\mathrm{T}}(\boldsymbol{\mu}^k - \mathbf{y})).$$

This is the so-called Iterated Reweighted Least Squares.

Some properties of Newton's method:

- Finds the global minimum of a positive definite quadratic function in a single iteration.
- The pure Newton method is not a descent method in general, and may be attracted to local maxima as well since it just tries to solve the system of equations

$$\nabla f(\mathbf{x}) = \mathbf{0}.$$

- The cost function must be twice differentiable.

- Typically converges very fast and does not exhibit the zig-zagging behavior of steepest descent:

  Superlinear convergence of Newton-like methods: Let f be twice continuously differentiable. Consider $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha^k \mathbf{d}_k$ and suppose that

  $$\mathbf{x}^k \to \mathbf{x}^\star, \quad \nabla f(\mathbf{x}^\star) = \mathbf{0}, \quad \nabla^2 f(\mathbf{x}^\star) \text{ is positive definite.}$$

  Assume further that $\nabla f(\mathbf{x}^k) \neq \mathbf{0}$ for all k and

  $$\lim_{k\to\infty} \frac{\|\mathbf{d}^k - (-\nabla^2 f(\mathbf{x}^k))^{-1} \nabla f(\mathbf{x}^k))\|}{\|\nabla f(\mathbf{x}^k)\|} = 0.$$

  Then if α^k is chosen with the Armijo rule with $s = 1$ and $\sigma < 1/2$, we have

  $$\lim_{k\to\infty} \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^\star\|}{\|\mathbf{x}^k - \mathbf{x}^\star\|} = 0.$$

- Let f be a function with a stationary point $\mathbf{x}^\star$. Suppose there is a ball $B_{\tilde{\delta}}(\mathbf{x}^\star)$ of radius $\tilde{\delta}$ around $\mathbf{x}^\star$ where f is twice continuously differentiable.
  Suppose also that $\nabla^2 f(\mathbf{x}^\star) > 0$.

  Then there exists $\delta > 0$ such that if $\mathbf{x}^0 \in B_\delta(\mathbf{x}^\star)$, then the pure Newton iterates

  $$\mathbf{x}^{k+1} = \mathbf{x}^k - [\nabla^2 f(\mathbf{x}^k)]^{-1} \nabla f(\mathbf{x}^k)$$

  are defined, belong to $B_\delta(\mathbf{x}^\star)$, and converge to $\mathbf{x}^\star$.

The difficulty is in the choice of $\mathbf{x}^0$ and in the calculation of $(\nabla^2 f(\mathbf{x}^k))^{-1}$ and guaranteeing its positive definiteness at each iteration (e.g., f may be linear in some regions).
Here are some cheap alternatives:

- Diagonally scaled steepest descent. Take

$$D^k = [\mathrm{diag}(\nabla^2 f(\mathbf{x}^k))]^{-1}.$$

  It is often surprisingly effective in practice.
- Modified Newton's method. Update $[\nabla^2 f(\mathbf{x}^k)]^{-1}$ regularly every K iterations.
- Discretized Newton method. Take $D^k = [H(\mathbf{x}^k)]^{-1}$, where H approximates $\nabla^2 f(\mathbf{x}^k)$ based on finite difference of the gradient or values of f.
- Replace Newton iteration by steepest descent when Newton direction is not defined or not a descent direction.

Quasi-Newton's method:

$$\left\{ \begin{array}{rcl} \mathbf{x}^{k+1} & = & \mathbf{x}^k + \alpha^k \mathbf{d}^k, \\ \mathbf{d}^k & = & -D^k \nabla f(\mathbf{x}^k), \end{array} \right.$$

where D^k is a positive definite matrix such that $\mathbf{d}^k$ tends to the Newton direction.

Idea: Successive iterates $\{\mathbf{x}^k, \mathbf{x}^{k+1}, \nabla f(\mathbf{x}^k), \nabla f(\mathbf{x}^{k+1})\}$ contain information about the curvature since

$$\mathbf{q}^k \approx \nabla^2 f(\mathbf{x}^{k+1})\mathbf{p}^k \quad \text{with} \quad \left\{ \begin{array}{rcl} \mathbf{p}^k & = & \mathbf{x}^{k+1} - \mathbf{x}^k \\ \mathbf{q}^k & = & \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) \end{array} \right. .$$

Hence after n (linearly independent) iteration increments $\mathbf{p}^0, \ldots, \mathbf{p}^{n-1}$, we have that

$$\nabla^2 f(\mathbf{x}^n) \approx [\mathbf{q}^0 \ldots \mathbf{q}^{n-1}][p^0 \ldots \mathbf{p}^{n-1}]^{-1}$$

or

$$[\nabla^2 f(\mathbf{x}^n)]^{-1} \approx [\mathbf{p}^0 \ldots \mathbf{p}^{n-1}][q^0 \ldots \mathbf{q}^{n-1}]^{-1},$$

a relation that is exact if the cost is quadratic.

Davidon-Fletcher-Powell ($\xi^k = 0$) and Broyden-Fletcher-Goldfarb-Shanno ($\xi^k = 1$) algorithms

Initiate D^0 as a positive definite matrix, e.g., $D^0 = [\mathrm{diag}(\nabla^2 f(\mathbf{x}^0))]^{-1}$. Then approximate $[\nabla^2 f(\mathbf{x}^{k+1})]^{-1}$ at each iteration by

$$D^{k+1} = D^k + \frac{\mathbf{p}^k \mathbf{p}^{k^{\mathrm{T}}}}{\mathbf{p}^{k^{\mathrm{T}}} \mathbf{q}^k} - \frac{D^k \mathbf{q}^k \mathbf{q}^{k^{\mathrm{T}}} D^k}{\mathbf{q}^{k^{\mathrm{T}}} D^k \mathbf{q}^k} + \xi^k \tau^k \mathbf{v}^k \mathbf{v}^{k^{\mathrm{T}}},$$

where

$$\begin{aligned} \mathbf{v}^k &= \frac{\mathbf{p}^k}{\mathbf{p}^{k^{\mathrm{T}}} \mathbf{q}^k} - \frac{D^k \mathbf{q}^k}{\tau^k}, \\ \tau^k &= \mathbf{q}^{k^{\mathrm{T}}} D^k \mathbf{q}^k, \\ \xi^k &\in [0, 1]. \end{aligned}$$

Property 1: If $D^k > 0$ and the stepsize α^k is chosen so that $\mathbf{x}^{k+1}$ satisfies

$$\nabla f(\mathbf{x}^k)^{\mathrm{T}}\mathbf{d}^k < \nabla f(\mathbf{x}^{k+1})^{\mathrm{T}}\mathbf{d}^k,$$

then $D^{k+1} > 0$.
Proof:

Property 2: If the cost function $f : \mathbb{R}^{\mathrm{n}} \to \mathbb{R}$ is quadratic

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}Q\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{x}$$

and the stepsize is chosen optimally, then:

- $D^n = Q^{-1}$
- $\mathbf{d}^0, \mathbf{d}^1, \ldots, \mathbf{d}^{n-1}$ are Q-conjugate.

In `R`:

```
> optim(..., method="BFGS"):
```

- option "BFGS" in `optim`.
  Property: best general purpose quasi-Newton method known, but requires storage of D_k.
  Provide an approximate to the inverse of the Hessian: can be used to calculate the covariance matrix of the MLE.
- option "Nelder-Mead" in `optim`.
  Property: not a gradient method, weak theoretical convergence, works reasonable well in small dimension.
- option "CG" (conjugate gradient) in `optim`.
  Property: no storage of D_k, but fragile (conjugacy progressively lost).

Conjugate gradient originally developed to solve a quadratic cost

$$\min_{\mathbf{x}\in\mathbb{R}^{\mathrm{N}}} \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x},$$

where $Q > 0$, or equivalently, to solve the linear system

$$Q\mathbf{x} = \mathbf{b}.$$

Idea: choose $\mathbf{d}^k$ to be the closest vector to the steepest descent direction $-\nabla f(\mathbf{x}^k)$ that is Q-conjugate to the previous directions $\mathbf{d}^1, \ldots, \mathbf{d}^{k-1}$, i.e.,

$$\mathbf{d}^{k^{\mathrm{T}}} Q\mathbf{d}^i = 0, \quad i = 1, \ldots, k-1.$$

This can be achieved by a Gram-Schmidt operation:

$$\mathbf{d}^k = -\nabla f(\mathbf{x}^k) - \sum_{j=0}^{k-1} \frac{(-\nabla f(\mathbf{x}^k))^{\mathrm{T}} Q\mathbf{d}^j}{(\mathbf{d}^j)^{\mathrm{T}} Q\mathbf{d}^j}\mathbf{d}^j.$$

CG algorithm

Set $\mathbf{x}^0$, $\mathbf{g}^0 = \nabla f(\mathbf{x}^0)$ and $\mathbf{d}^0 = -\mathbf{g}^0$.

Then $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^{k+1}\mathbf{d}^k$ with

- $\alpha^{k+1} = \frac{\mathbf{g}^{k\mathrm{T}}\mathbf{g}^k}{\mathbf{d}^{k\mathrm{T}}A\mathbf{d}^k}$
- if $\|\mathbf{g}^k\| < \epsilon$ stop; otherwise $k = k + 1$ and calculate $\mathbf{g}^k = \nabla f(\mathbf{x}^k) = Q\mathbf{x}^k - \mathbf{b}$, $\beta^k = \frac{\mathbf{g}^{k\mathrm{T}}\mathbf{g}^k}{\mathbf{g}^{k-1\mathrm{T}}\mathbf{g}^{k-1}}$ and $\mathbf{d}^k = -\mathbf{g}^k + \beta^k\mathbf{d}^{k-1}$.

Properties:

In theory, the maximum number of iterations is equal to the number of different eigenvalues of Q.

In practice, to get the solution to machine precision urequires more than N iterations.

Good approximation is often reached much earlier.

Useful when Q is large and sparse, or a fast operator like FFT or DWT.

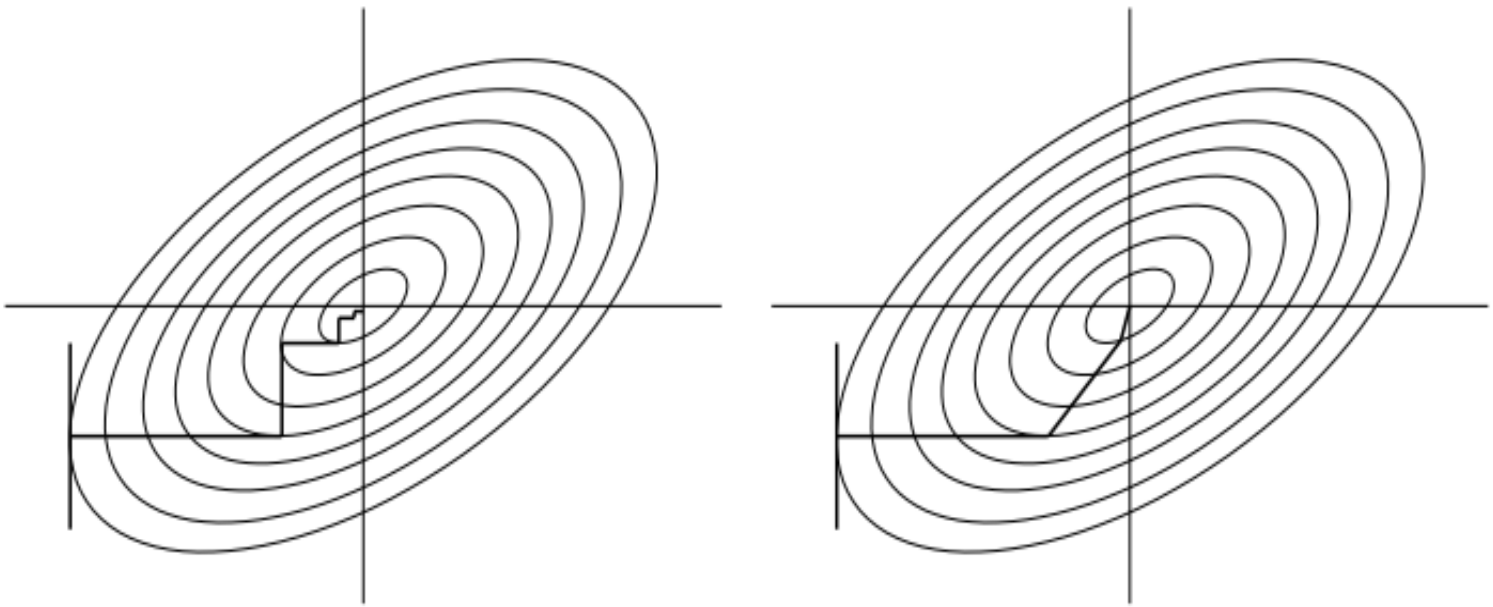

Figure 6.14: Steepest descent vs. conjugate gradient.

5. Non gradient-based iterative algorithm

Coordinate descent (Relaxation)
At each iteration solve a univariate problem

$$\min_{\xi \in X_i} f(x_1^k, \ldots, x_{i-1}^k, \xi, x_{i+1}^k, \ldots, x_N^k), \tag{3}$$

and iterate in a cyclic manner $i = 1, \ldots, N, 1, \ldots, N, \ldots$

Proposition: Suppose f is continuously differentiable over $X = X_1 \times \ldots \times X_N$, where each X_n is closed and convex, and suppose that (3) has a unique solution, then coordinate descent converges to a stationary point.

Properties

- easy to implement
- fast if closed form solution at each iteration
- can sometimes use parallel computing
- similar convergence properties as steepest descent
- can iterate in blocks as well (BCR)
- various rules to select next i, cyclic or not
- can handle certain constraints and non-differentiability

Paul Tseng worked on relaxion of these assumptions.

5. Stopping criterion

Many rules: fix $\epsilon > 0$, then stop iterations when

- $\|\nabla f(\mathbf{x}^k)\| \leq \epsilon$.
- $\|\nabla f(\mathbf{x}^k)/N\| \leq \epsilon$.
- $\|\nabla f(\mathbf{x}^k)\|/\|\nabla f(\mathbf{x}^0)\| \leq \epsilon$.
- $\|\mathbf{d}^k\| \leq \epsilon$, assuming the directions $\mathbf{d}^k$ captures the relative rescaling of the variables.

The difficulty remains in the choice of ϵ, and is quite problem dependent.

Table : Simulation results to compare two estimators

ϵ	Waveshrink	MRF-ℓ_1 large ϵ	MRF-ℓ_1 small ϵ
blocks $N = 1024$	0.22	0.13	0.07
bumps $N = 1024$	0.52	0.62	0.41
heavisine $N = 1024$	0.31	0.13	0.13
Doppler $N = 1024$	0.56	0.63	0.41

Illustration of the danger of not solving the optimization problem “well enough".

6. Convergence

The most we can expect from a gradient-based method is that it converges to a stationary point, including a local maximum.

Such a point is a global minimum if the cost function is convex and differentiable.
Otherwise, without knowing much about uniqueness of stationary point, we can run a gradient-based algorithm from multiple starting points.

7. Conclusions

Algorithms have their specificities and should be carefully chosen and employed based on the specificity of your optimization problem (differentiability, convexity, ...)

Black box tools are convenient but dangerous.

Initial point $\mathbf{x}^0$ and convergence threshold ϵ are also important quantities to choose.