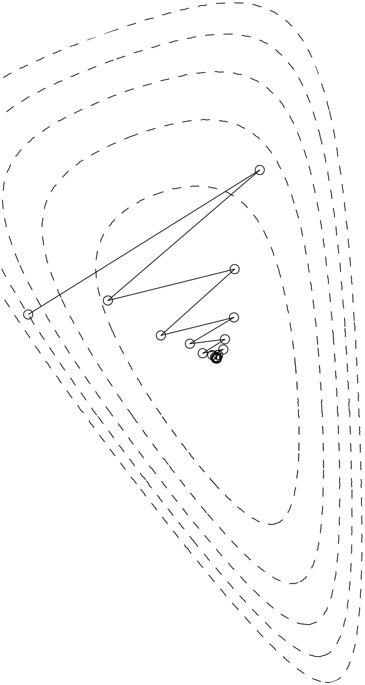


Introduction to Optimization

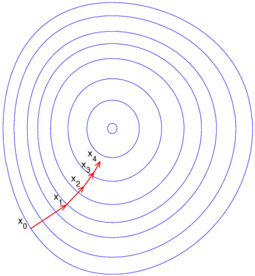
Gradient-based Methods

Marc Toussaint
U Stuttgart

Gradient descent methods

- Plain gradient descent (with adaptive stepsize)
- Steepest descent (w.r.t. a known metric)
- Conjugate gradient (requires line search)
- Rprop (heuristic, but quite efficient)

Gradient descent



- Notation:
  objective function: $f : \mathbb{R}^n \to \mathbb{R}$
  gradient vector: $\nabla f(x) = \left[\frac{\partial}{\partial_x} f(x) \right]^\top \in \mathbb{R}^n$

- Problem:

$$\min_x f(x)$$

  where we can evaluate $f(x)$ and $\nabla f(x)$ for any $x \in \mathbb{R}^n$

- Gradient descent:
  Make iterative steps in the direction $-\nabla f(x)$.

Plain Gradient Descent

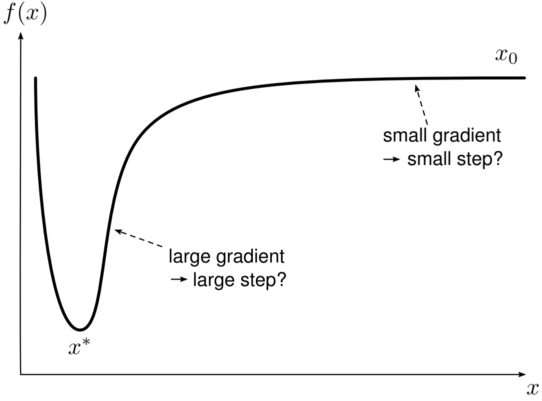
Fixed stepsize

BAD! gradient descent:

Input: initial $x \in \mathbb{R}^n$, function $\nabla f(x)$, stepsize α, tolerance θ
Output: x

repeat
    $x \leftarrow x - \alpha \nabla f(x)$
until $|\Delta x| < \theta$ [perhaps for 10 iterations in sequence]

Making steps proportional to $\nabla f(x)$??



NO!

We need methods indep. of $|\nabla f(x)|$, invariant of scaling of f and x!

How can we become independent of $|\nabla f(x)|$?

- Line search — which we'll discuss briefly later
- Stepsize adaptation

Gradient descent with stepsize adaptation

Input: initial $x \in \mathbb{R}^n$, functions $f(x)$ and $\nabla f(x)$, initial stepsize α, tolerance θ

Output: x

repeat
    $y \leftarrow x - \alpha \frac{\nabla f(x)}{|\nabla f(x)|}$
    if [**then**step is accepted]$f(y) \leq f(x)$
        $x \leftarrow y$
        $\alpha \leftarrow 1.2\alpha$ *// increase stepsize*
    else[step is rejected]
        $\alpha \leftarrow 0.5\alpha$ *// decrease stepsize*
    end if
until $|y - x| < \theta$ [perhaps for 10 iterations in sequence]

("magic numbers")

α determins the absolute stepsize

stepsize is automatically adapted

- Guaranteed monotonicity (by construction)

  If f is convex $\Rightarrow$ convergence
  For typical non-convex bounded f $\Rightarrow$ convergence to local optimum

Steepest Descent

Steepest Descent

- The gradient $\nabla f(x)$ is sometimes called *steepest descent direction*

Is it really?

Steepest Descent

- The gradient $\nabla f(x)$ is sometimes called *steepest descent direction*

Is it really?

- Here is a possible definition:

The steepest descent direction is the one where, when I make a step of length 1, I get the largest decrease of f in its linear approximation.

$$\operatorname*{argmin}_{\delta} \nabla f(x)^\top \delta \qquad \text{s.t. } \|\delta\| = 1$$

Steepest Descent

- But the norm $\|\delta\|^2 = \delta^\top A \delta$ depends on the metric A!

  Let $A = B^\top B$ (Cholesky decomposition) and $z = B\delta$

$$\begin{aligned}
\delta^* &= \operatorname*{argmin}_\delta \nabla f^\top \delta \qquad \text{s.t. } \delta^\top A \delta = 1 \\
&= B^{\text{-}1} \operatorname*{argmin}_z (B^{\text{-}1} z)^\top \nabla f \qquad \text{s.t. } z^\top z = 1 \\
&= B^{\text{-}1} \operatorname*{argmin}_z z^\top B^{\text{-}\top} \nabla f \qquad \text{s.t. } z^\top z = 1 \\
&= B^{\text{-}1} [-B^{\text{-}\top} \nabla f] = -A^{\text{-}1} \nabla f
\end{aligned}$$

The steepest descent direction is $\delta = -A^{\text{-}1} \nabla f$

Behavior under linear coordinate transformations

- Let B be a matrix that describes a linear transformation in coordinates

- A coordinate vector x transforms as $z = Bx$
- The gradient vector $\nabla_x f(x)$ transforms as $\nabla_z f(z) = B^{-\top} \nabla_x f(x)$
- The metric A transforms as $A_z = B^{-\top} A_x B^{-1}$
- The steepest descent transforms as $A_z^{-1} \nabla_z f(z) = B A_x^{-1} \nabla_x f(x)$

  The steepest descent transforms like a normal coordinate vector (covariant)

(Nonlinear) Conjugate Gradient

Conjugate Gradient

- The "Conjugate Gradient Method" is a method for solving large linear eqn. systems $Ax + b = 0$
  We mention its extension for optimizing nonlinear functions $f(x)$

- A key insight:

  – at x_k we computed $\nabla f(x_k)$
  – we made a (line-search) step to x_{k+1}
  – at x_{k+1} we computed $\nabla f(x_{k+1})$

What conclusions can we draw about the "local quadratic shape" of f?

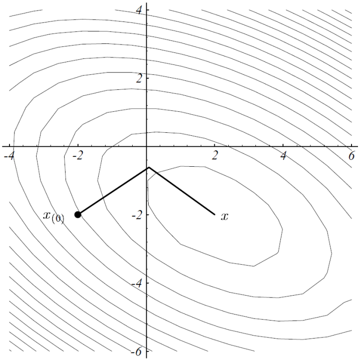
Conjugate Gradient

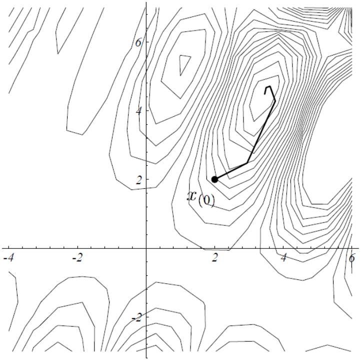
Input: initial $x \in \mathbb{R}^n$, functions $f(x), \nabla f(x)$, tolerance θ
Output: x

initialize descent direction $d = g = -\nabla f(x)$
repeat
    $\alpha \leftarrow \operatorname{argmin}_\alpha f(x + \alpha d)$ *// line search*
    $x \leftarrow x + \alpha d$
    $g' \leftarrow g,\ g = -\nabla f(x)$ *// store and compute grad*
    $\beta \leftarrow \max \left\{ \frac{g^\top (g - g')}{g'^\top g'}, 0 \right\}$
    $d \leftarrow g + \beta d$ *// conjugate descent direction*
until $|\Delta x| < \theta$

- Notes:
  – $\beta > 0$: The new descent direction always adds a bit of the old direction!
  – This essentially provides 2nd order information
  – The equation for β is by Polak-Ribière: On a quadratic function $f(x) = x^\top A x$ this leads to **conjugate** search directions, $d'^\top A d = 0$.
  – All this really only works with **line search**

Conjugate Gradient





- For quadratic functions CG converges in n iterations. But each iteration does *line search*!

Conjugate Gradient

- Useful tutorial on CG and **line search**:

  J. R. Shewchuk: *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*

Rprop

Rprop

"Resilient Back Propagation" (outdated name from NN times...)

Input: initial $x \in \mathbb{R}^n$, function $f(x), \nabla f(x)$, initial stepsize α, tolerance θ
Output: x

initialize $x = x_0$, all $\alpha_i = \alpha$, all $g_i = 0$
repeat
  $g \leftarrow \nabla f(x)$
  $x' \leftarrow x$
  for $i = 1 : n$ **do**
    if [**then**same direction as last time]$g_i g_i' > 0$
      $\alpha_i \leftarrow 1.2\alpha_i$
      $x_i \leftarrow x_i - \alpha_i \ \text{sign}(g_i)$
      $g_i' \leftarrow g_i$
    else if [**then**change of direction]$g_i g_i' < 0$
      $\alpha_i \leftarrow 0.5\alpha_i$
      $x_i \leftarrow x_i - \alpha_i \ \text{sign}(g_i)$
      $g_i' \leftarrow 0$ *// force last case next time*
    else
      $x_i \leftarrow x_i - \alpha_i \ \text{sign}(g_i)$
      $g_i' \leftarrow g_i$
    end if
    optionally: cap $\alpha_i \in [\alpha_{\text{min}}\, x_i, \alpha_{\text{max}}\, x_i]$
  end for
until $|x' - x| < \theta$ for 10 iterations in sequence

Rprop

- Rprop is a bit crazy:
  - stepsize adaptation in each dimension *separately*
  - it not only ignores $|\nabla f|$ but also its exact direction
    step directions may differ up to $< 90^\circ$ from ∇f
  - Often works very robustly
  - Guarantees? See work by Ch. Igel

- If you like, have a look at:
  Christian Igel, Marc Toussaint, W. Weishui (2005): Rprop using the natural gradient compared to Levenberg-Marquardt optimization. In Trends and Applications in Constructive Approximation. International Series of Numerical Mathematics, volume 151, 259-272.

Appendix

Two little comments on stopping criteria & costs...

Appendix: Stopping Criteria

- Standard references (Boyd) define stopping criteria based on the "change" in $f(x)$, e.g. $|\Delta f(x)| < \theta$ or $|\nabla f(x)| < \theta$.

- Throughout I will define stopping criteria based on the change in x, e.g. $|\Delta x| < \theta$! In my experience this is in many problems more meaningful, and invariant of the scaling of f.

Appendix: Optimization Costs

- Standard references (Boyd) assume line search is cheap and measure optimization costs as the number of iterations (counting 1 per line search).

- Throughout I will assume that every evaluation of $f(x)$ or $(f(x), \nabla f(x))$ or $(f(x), \nabla f(x), \nabla^2 f(x))$ is equally expensive!