Math 381 - Fall 2022

Jay Newby

University of Alberta

Week 12

Last Time

1. Continuous optimization problems
2. Existence and uniqueness
3. Coercive functions
4. Level sets and sublevel sets
5. Convex sets
6. Convex functions
7. Critical points and classification of critical points

Today

1. Sensitivity and conditioning of unconstrained problems
2. One dimensional minimization algorithms
3. n-dimensional minimization algorithms
   - Newton's method
   - Gradient descent
   - Line search
   - Steepest descent

Optimization in one dimension

- Newton's method
- Golden Section Search (see reading)
- Successive Parabolic Search (see reading)
- Safeguard methods (see reading)

Unconstrained multidimensional optimization

Continuous optimization problem: unconstrained

Given a continuous and sufficiently smooth function $f : \mathbb{R}^n \to \mathbb{R}$, find

$$\min_x f(x).$$

- Gradient descent and stochastic gradient descent
- Steepest descent
- Conjugate gradient method
- Newton's method and quasi-Newton methods
- Secant updating method
- Infinite dimensional optimization problems

Root finding methods

Recall

Newton's Method

Let $g(x)$ be continuous and sufficiently smooth. Let $g(\hat{x}) = 0$ and $g'(\hat{x}) \neq 0$. Given an initial guess x_0 sufficiently close to $\hat{x}$, the iteration

$$x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$

Converges to $\hat{x}$ as $k \to \infty$.

Newton's method for minimization problems

We can apply Newton's method to solve $g(x) = f'(x) = 0$.

Newton's Method for minimization problems

Let $f(x)$ be continuous and sufficiently smooth. Let $f'(\hat{x}) = 0$ and $f''(\hat{x}) \neq 0$. Given an initial guess x_0 sufficiently close to $\hat{x}$, the iteration

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Converges to $\hat{x}$ as $k \to \infty$.

Newton's method in higher dimensions

Newton's method in higher dimensions

Given a current best guess for the minimizer x_k, we look within a local neighborhood for a better guess. Taylor expand around x_k with

$$f(x_k + s) \sim f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T H_f(x_k) s.$$

Let $s_k \in \mathbb{R}^n$ be the value of s that minimizes the above local quadratic approximation, which satisfies $H_f(x_k)s_k = -\nabla f(x_k)$. The next iterate is then given by

$$x_{k+1} = x_k + s_k = x_k - H_f(x_k)^{-1} \nabla f(x_k).$$

Newton's method

- Quadratic convergence if the initial guess x_0 is sufficiently close to a minimum
- Can be unpredictable and unstable otherwise
- Requires the gradient and the Hessian
- Many variants exist

Quasi-Newton methods are variations on Newton's method that sacrifice a little convergence speed for efficiency and robustness

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k),$$

where α_k is a line search parameter and B_k is some approximation to the Hessian.

Secant updating methods

BFGS method

Given an initial guess x_0 and an initial Hessian approximation B_0 (e.g., $B_0 = I$) the method proceeds as follows.

$$
\begin{aligned}
s_k &= -B_k^{-1} \nabla f(x_k) \\
x_{k+1} &= x_k + s_k \\
y_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \\
B_{k+1} &= B_k + \frac{1}{y_k^T s_k} y_k y_k^T - \frac{1}{s_k^T B_k s_k} B_k s_k s_k^T B_k
\end{aligned}
$$

In practice, a factorization of B_k is updated so that the linear system can be solved in $O(n^2)$ operations instead of $O(n^3)$.

Gradient Descent

Gradient Descent

Given a sufficiently smooth objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the gradient $\nabla f(x)$ the algorithm proceeds as follows. Let $r_k > 0$ be a step size on the kth iteration be given. At each iteration we have

$$x_{k+1} = x_k - r_k \nabla f(x_k).$$

The step size can depend on the number of iterations so that $r_k \to 0$ as $k \to \infty$.

- Powerful method when used in conjunction with automatic differentiation
- Scales well with dimension n and is mostly used on extremely high dimensional problems (training neural networks)

Stochastic gradient Descent

Stochastic gradient Descent

This version of gradient descent uses a random process ξ_k, where $E[\xi_k] = 0$, so that

$$x_{k+1} = x_k - r_k(\nabla f(x_k) + \xi_k).$$

This is the form used to train neural networks. Under "ideal" circumstances, during each training step the loss is calculated for all of the training examples in the training set. Instead each training iteration only uses a random subset of training examples. The gradient step for the full set and the random subset are not the same. The difference between the two is viewed as random noise.

- Allows for more robust convergence to global minima or higher quality local minima

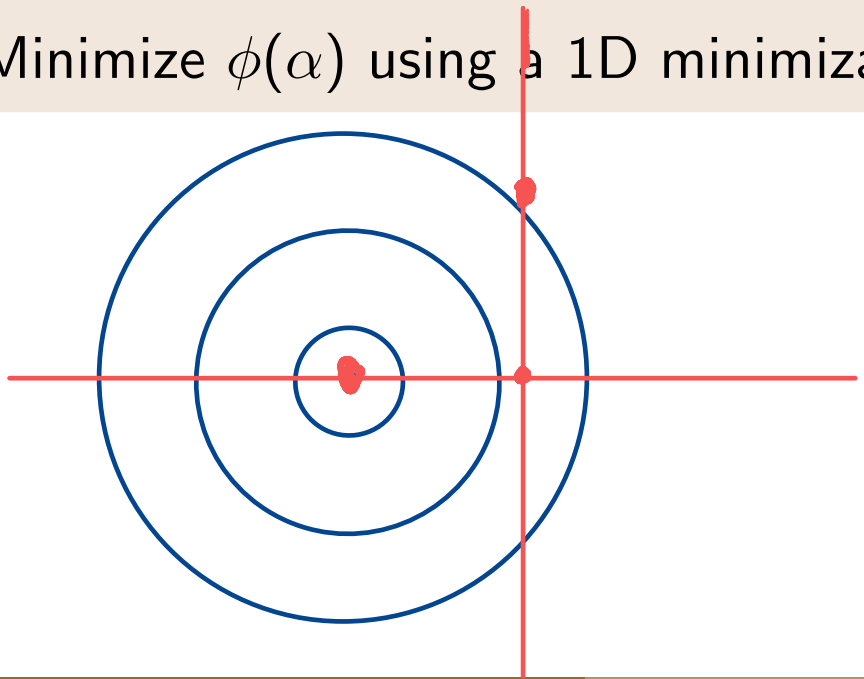
Line search methods

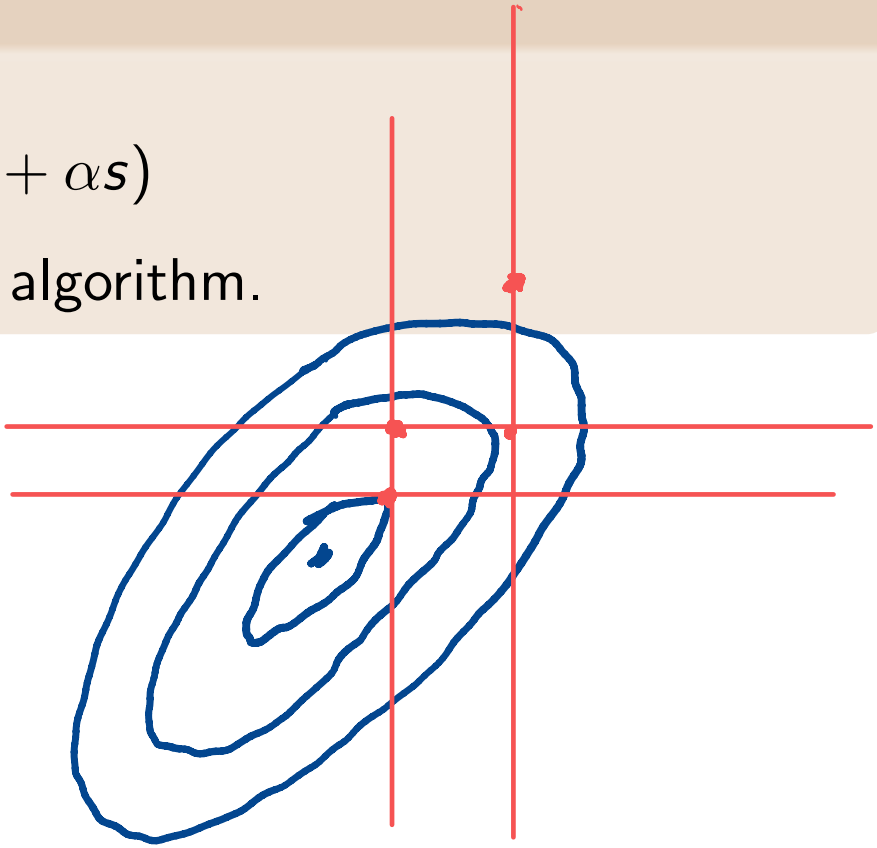
Line search methods

Let

$$\phi(\alpha) = f(x + \alpha s)$$

Minimize $\phi(\alpha)$ using a 1D minimization algorithm.





Coordinate descent method (related to ~~Gibbs sampling~~) stochastic gradient descent

Coordinate descent

Let $x \in \mathbb{R}^m$.

$$x_i^{(n)} = \arg\min_{x_i \in \mathbb{R}} f(x_1^{(n)}, x_2^{(n)}, \ldots, x_{i-1}^{(n)}, x_i, x_{i+1}^{(n-1)}, \ldots, x_m^{(n-1)})$$

Coordinate descent method (related to Gibbs sampling)

Coordinate descent

Let $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$. Consider the two step process

1. $x_n = \arg\min_{x \in \mathbb{R}^p} f(x, y_{n-1})$
2. $y_n = \arg\min_{y \in \mathbb{R}^q} f(x_n, y)$

Steepest descent method (not to be confused with gradient descent)

Steepest descent method

Given the gradient of the objective function $\nabla f(x)$ and an initial guess x_0, the method proceeds as follows. At each iteration $k > 0$, let $s_k = -\nabla f(x_k)$. Then

$$x_{k+1} = x_k + \alpha_k s_k, \qquad \alpha_k = \underset{\alpha \in \mathbb{R}}{\arg\min}\, f(x_k + \alpha s_k).$$

Steepest descent method converges linearly

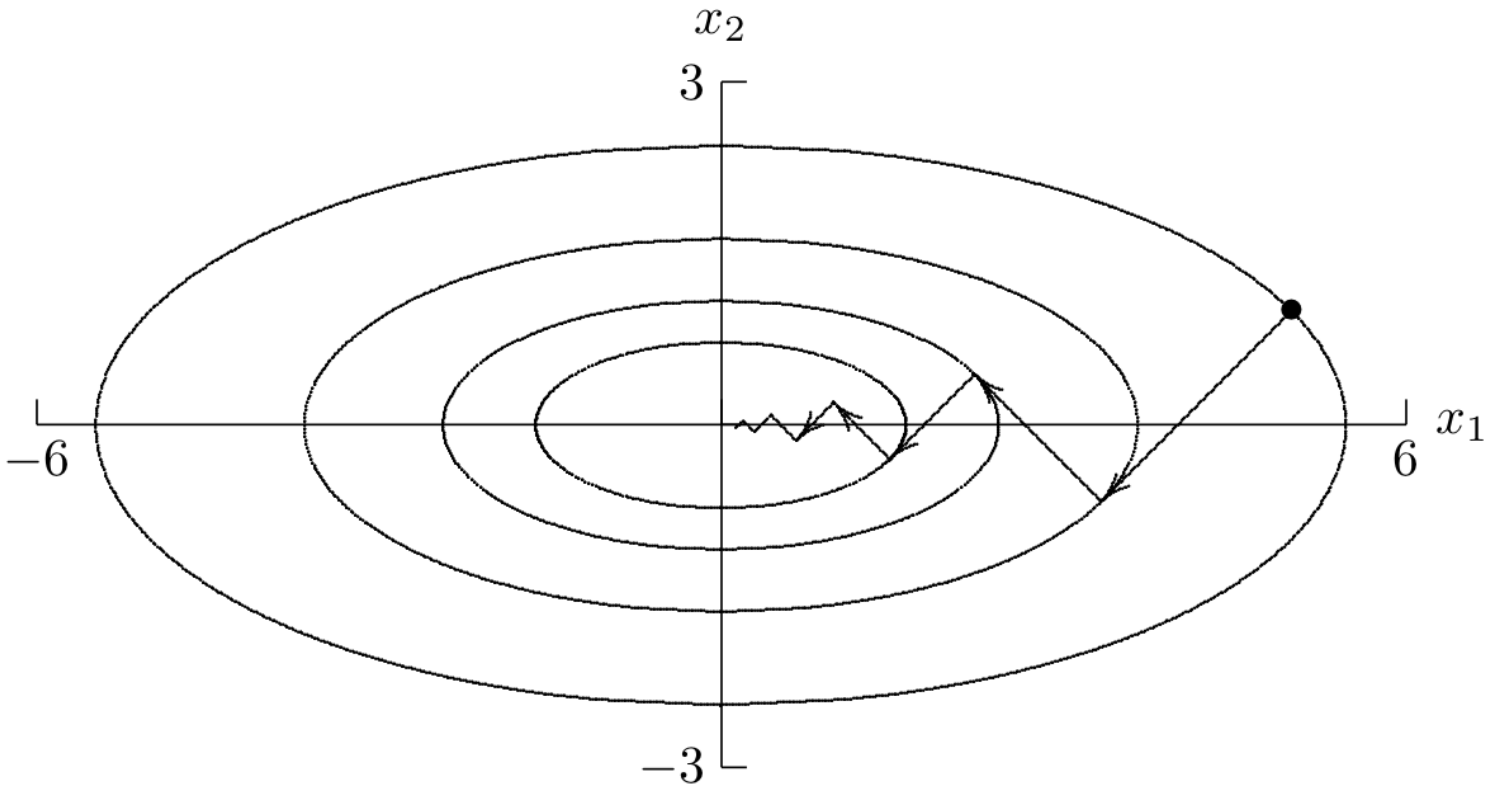


Figure 6.9: Convergence of steepest descent.

$$f(x) = \frac{1}{2}x_1^2 + \frac{5}{2}x_2^2$$

Conjugate gradient method

Conjugate gradient method

Given the gradient of the objective function $\nabla f(x)$ and an initial guess x_0, the method proceeds as follows. Initialize the search direction with $s_0 = -\nabla f(x_0)$. Then, for each $k = 0, 1, \ldots$

$\|Ax-b\|_2^2$

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}} f(x_k + \alpha s_k),$$

$$x_{k+1} = x_k + \alpha_k s_k,$$

$$g_{k+1} = \nabla f(x_{k+1}),$$

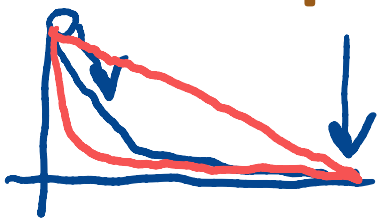
$$\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k},$$

$$s_{k+1} = -g_{k+1} + \beta_{k+1} s_k.$$

Conjugate gradient method

- If $f(x) = x^T Ax$ for square s.p.d matrix $A \in \mathbb{R}^{n \times n}$ then CG converges to the *exact* answer in n steps
- Used as an iterative method to solve linear systems with s.p.d square matrix (can get blazing speed with the right preconditioner)

Infinite dimensional optimization problems



Infinite dimensional optimization problem

Let S be a suitable function space. Let $F : S \to \mathbb{R}$ be a functional on S. Find

$$\min_{g \in S} F[g].$$

Various constraints are often imposed on the problem.

Example:

$$F[g] = \int_0^T L[t, g(t), g'(t)]dt$$

$$\frac{\delta F}{\delta g} = 0$$