Gradient Methods

- Problem Formulation
- Gradient Methods
- Conjugent Gradient Methods

Contents

Problem Formulation

We are interested in solving "large" linear equation systems of the form

$$Ax = b \; ,$$

where $A \in \mathbb{R}^{n \times n}$ is a positive definite matrix.

For example needed to implement Newton's method unconstrained optimization!

Problem Formulation

We are interested in solving “large” linear equation systems of the form

$$Ax = b \ ,$$

where $A \in \mathbb{R}^{n \times n}$ is a positive definite matrix.

For example needed to implement Newton's method unconstrained optimization!

Iterative versus Direct Methods

Examples for direct LA methods are

- Gauss-elimination; LR decomposition)
- Gram-Schmidt methods; QR decomposition
- Cholesky factorization; tailored for positive (semi-)definite matrices)

These and similar methods

1. have complexity $O(n^3)$ (if sparsity is not exploited)
2. find x up to small rounding errors, if A is well-conditioned.

Iterative versus Direct Methods

Examples for direct LA methods are

- Gauss-elimination; LR decomposition)
- Gram-Schmidt methods; QR decomposition
- Cholesky factorization; tailored for positive (semi-)definite matrices)

These and similar methods

1. have complexity $O(n^3)$ (if sparsity is not exploited)
2. find x up to small rounding errors, if A is well-conditioned.

Iterative versus Direct Methods

In contrast to direct elimination/decomposition methods, we concentrate in this lecture on iterative algorithms which

- converge to a solution x of the equation $Ax = b$,
- improve an iterate for $x_k \approx x$ at every step,
- are stopped whenever a "sufficiently accurate solution" is found or if we run out of time.

Contents

An optimization perspective

Notice that solving the equation $Ax = b$, $A \succ 0$, is equivalent to solving the quadratic optimization problem

$$\min_x \; F(x) \qquad \text{with} \qquad F(x) = \frac{1}{2} x^T A x - x^T b \; .$$

Proof: The function F is strictly convex (since $F''(x) = A \succ 0$) and the gradient $\nabla F(x) = Ax - b$ is zero if and only if x satisfies $Ax = b$.

An alternative approach could start by solving $\min_x \; \|Ax - b\|_2^2$, but then we would square the condition number.

An optimization perspective

Notice that solving the equation $Ax = b$, $A \succ 0$, is equivalent to solving the quadratic optimization problem

$$\min_x \; F(x) \qquad \text{with} \qquad F(x) = \frac{1}{2} x^T A x - x^T b \, .$$

Proof: The function F is strictly convex (since $F''(x) = A \succ 0$) and the gradient $\nabla F(x) = Ax - b$ is zero if and only if x satisfies $Ax = b$.

An alternative approach could start by solving $\min_x \; \|Ax - b\|_2^2$, but then we would square the condition number.

An optimization perspective

Notice that solving the equation $Ax = b$, $A \succ 0$, is equivalent to solving the quadratic optimization problem

$$\min_x \; F(x) \qquad \text{with} \qquad F(x) = \frac{1}{2} x^T A x - x^T b \; .$$

Proof: The function F is strictly convex (since $F''(x) = A \succ 0$) and the gradient $\nabla F(x) = Ax - b$ is zero if and only if x satisfies $Ax = b$.

An alternative approach could start by solving $\min_x \; \|Ax - b\|_2^2$, but then we would square the condition number.

Newton-Type Methods for Linear Systems

Does it make sense to apply a Newton-type method to solve linear equations?

Yes, if the Hessian approximation $M \approx A$ is easy to invert!

Our orignal problem has the form

$$\min_x \; F(x) \qquad \text{with} \qquad F(x) = \frac{1}{2} x^T A x - x^T b \; .$$

If we apply Newton-type methods, the iterates are

$$x_{k+1} \; = \; x_k - M_k^{-1} \nabla F(x_k) \; = \; x_k - M_k^{-1} (Ax - b) \; .$$

Newton-Type Methods for Linear Systems

Does it make sense to apply a Newton-type method to solve linear equations?

Yes, if the Hessian approximation $M \approx A$ is easy to invert!

Our orignal problem has the form

$$\min_x \; F(x) \qquad \text{with} \qquad F(x) = \frac{1}{2} x^T A x - x^T b \,.$$

If we apply Newton-type methods, the iterates are

$$x_{k+1} \;=\; x_k - M_k^{-1} \nabla F(x_k) \;=\; x_k - M_k^{-1}(Ax - b) \,.$$

Newton-Type Methods for Linear Systems

Does it make sense to apply a Newton-type method to solve linear equations?

Yes, if the Hessian approximation $M \approx A$ is easy to invert!

Our orignal problem has the form

$$\min_x \; F(x) \qquad \text{with} \qquad F(x) = \frac{1}{2} x^T A x - x^T b \; .$$

If we apply Newton-type methods, the iterates are

$$x_{k+1} \; = \; x_k - M_k^{-1} \nabla F(x_k) \; = \; x_k - M_k^{-1}(Ax - b) \; .$$

Gradient Methods

If the matrix A is well-conditioned, we may choose the Hessian approximation $M = I$. This yields the so-called gradient method

$$x_{k+1} = x_k - \alpha_k \nabla F(x_k) = x_k - \alpha_k (A x_k - b) .$$

The line search parameter α_k can be found "exact line search", i.e., by solving

$$\min_{\alpha_k} F(x_k - \alpha_k (A x_k - b)) .$$

Gradient Methods

If the matrix A is well-conditioned, we may choose the Hessian approximation $M = I$. This yields the so-called gradient method

$$x_{k+1} \;=\; x_k - \alpha_k \nabla F(x_k) \;=\; x_k - \alpha_k (Ax_k - b)\ .$$

The line search parameter α_k can be found "exact line search", i.e., by solving

$$\min_{\alpha_k}\; F(x_k - \alpha_k(Ax_k - b))\ .$$

Gradient Methods

In order to work out explicitly the solution of the line search problem

$$\min_{\alpha_k} \; F(x_k - \alpha_k(Ax_k - b)) \; ,$$

we denote with $d_k = Ax_k - b$ the search direction and elimate α_k from the optimality conditions

$$0 = \nabla F(x_k - \alpha_k d_k)^T d_k = d_k^T d_k + \alpha_k d_k^T A d_k$$

which yields

$$\alpha_k = \frac{d_k^T d_k}{d_k^T A d_k} \; .$$

Summary: Gradient Method

1. **Input:** An intitial guess x_0 and tolerance $\epsilon > 0$.
2. **Repeat:**
   2.1 compute the step direction $d_k = Ax_k - b$,
   2.2 if $\|d\| < \epsilon$, stop.
   2.3 compute the line search parameter $\alpha_k = \frac{d_k^T d_k}{d_k^T A d_k}$,
   2.4 set $x_{k+1} = x_k - \alpha_k d_k$ and increase the counter $k \leftarrow k + 1$.
3. **Output:** A numerical approximation $x_k \approx x$ of the solution vector.

Convergence Analysis of Gradient Methods

Most of the convergence proofs for gradient methods first show that we get in every step a sufficient descent of the "Lyapunov function"

$$L(y) = (y - x)^T A (y - x) \; .$$

One way to show this is by using the equation

$$\begin{aligned} L(x_{k+1}) &= (x_k - x - \alpha_k d_k)^T A (x_k - x - \alpha_k d_k)^T \\ &= L(x_k) - 2\alpha_k (x_k - x)^T A d_k + \alpha_k^2 d_k^T A d_k \; . \end{aligned}$$

Since $d_k = Ax_k - b = A(x_k - x)$ this can be simplified further to

$$L(x_{k+1}) = L(x_k) - 2\alpha_k d_k^T d_k + \alpha_k^2 d_k^T A d_k = L(x_k) - \underbrace{\frac{\|d_k\|_2^4}{d_k^T A d_k}}_{>0} \; ,$$

which proves that we get a strict descent.

Convergence Analysis of Gradient Methods

Most of the convergence proofs for gradient methods first show that we get in every step a sufficient descent of the "Lyapunov function"

$$L(y) = (y-x)^T A(y-x)\ .$$

One way to show this is by using the equation

$$\begin{aligned} L(x_{k+1}) &= (x_k - x - \alpha_k d_k)^T A\,(x_k - x - \alpha_k d_k)^T \\ &= L(x_k) - 2\alpha_k (x_k - x)^T A d_k + \alpha_k^2 d_k^T A d_k\ . \end{aligned}$$

Since $d_k = Ax_k - b = A(x_k - x)$ this can be simplified further to

$$L(x_{k+1}) = L(x_k) - 2\alpha_k d_k^T d_k + \alpha_k^2 d_k^T A d_k = L(x_k) - \underbrace{\frac{\|d_k\|_2^4}{d_k^T A d_k}}_{>0}\ ,$$

which proves that we get a strict descent.

Convergence Analysis of Gradient Methods

Most of the convergence proofs for gradient methods first show that we get in every step a sufficient descent of the "Lyapunov function"

$$L(y) = (y-x)^T A(y-x) \; .$$

One way to show this is by using the equation

$$\begin{aligned} L(x_{k+1}) &= (x_k - x - \alpha_k d_k)^T A \, (x_k - x - \alpha_k d_k)^T \\ &= L(x_k) - 2\alpha_k (x_k - x)^T A d_k + \alpha_k^2 d_k^T A d_k \; . \end{aligned}$$

Since $d_k = Ax_k - b = A(x_k - x)$ this can be simplified further to

$$L(x_{k+1}) = L(x_k) - 2\alpha_k d_k^T d_k + \alpha_k^2 d_k^T A d_k = L(x_k) - \underbrace{\frac{\| d_k \|_2^4}{d_k^T A d_k}}_{>0} \; ,$$

which proves that we get a strict descent.

Convergence Analysis of Gradient Methods

In order to finally prove convergence, we have to analyze the equation

$$L(x_{k+1}) \quad = \quad L(x_k) - \frac{\|d_k\|_2^4}{d_k^T A d_k} \; ,$$

a bit further. For this aim, we estimate the term

$$\frac{\|d_k\|_2^4}{d_k^T A d_k L(x_k)} = \frac{\|d_k\|_2^4}{d_k^T A d_k \, d_k^T A^{-1} d_k} \geq \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)} = \frac{1}{\mathsf{cond}_2(A)} \; .$$

This proves that the gradient method converges with linear rate

$$L(x_{k+1}) \;\leq\; \left(1 - \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}\right) L(x_k)$$

Convergence Analysis of Gradient Methods

In order to finally prove convergence, we have to analyze the equation

$$L(x_{k+1}) \quad = \quad L(x_k) - \frac{\| d_k \|_2^4}{d_k^T A d_k} \ ,$$

a bit further. For this aim, we estimate the term

$$\frac{\| d_k \|_2^4}{d_k^T A d_k L(x_k)} = \frac{\| d_k \|_2^4}{d_k^T A d_k \, d_k^T A^{-1} d_k} \geq \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)} = \frac{1}{\mathsf{cond}_2(A)} \ .$$

This proves that the gradient method converges with linear rate

$$L(x_{k+1}) \ \leq \ \left(1 - \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)} \right) L(x_k)$$

Convergence Analysis of Gradient Methods

The convergence rate estimate $1 - \frac{\lambda_{\min}(A)}{\lambda_{\max}(A)}$ indicates that

- Gradient methods work very well, if $\text{cond}_2(A)$ is close to 1.
- The other way round, if A is ill-conditioned the gradient method converges very slowly.

These theoretical prediction are confirmed in numerical experiments.

Convergence of Gradient Methods in Practice

In practice, if we plot the iterates of a gradient method, we typically observe a "zig-zag" behavior. This is due to the fact that subsequent search directions of the gradient method are orthogonal to each other

$$d_{k+1}^T d_k = \left(d_k - \alpha_k A d_k\right)^T d_k = d_k^T d_k - \alpha_k d_k^T A d_k = 0 \; .$$

Remark: the convergence rate estimate of the gradient methods can be improved by using "Kantorovich's inequality" which yields

$$\sqrt{L(x_{k+1})} \;\leq\; \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} \sqrt{L(x_k)} \; .$$

Convergence of Gradient Methods in Practice

In practice, if we plot the iterates of a gradient method, we typically observe a "zig-zag" behavior. This is due to the fact that subsequent search directions of the gradient method are orthogonal to each other

$$d_{k+1}^T d_k = (d_k - \alpha_k A d_k)^T d_k = d_k^T d_k - \alpha_k d_k^T A d_k = 0 \, .$$

Remark: the convergence rate estimate of the gradient methods can be improved by using "Kantorovich's inequality" which yields

$$\sqrt{L(x_{k+1})} \;\leq\; \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} \sqrt{L(x_k)} \, .$$

Contents

Conjugent Vectors

Two vectors u and v are called conjugent (or "A-orthogonal") with respect to a matrix A, if

$$u^T A v = 0 \ .$$

- for gradient methods only successive search directions are orthogonal
- main idea of conjugent gradient methods: maintain A-orthogonality of all search directions.

Conjugent Vectors

Two vectors u and v are called conjugent (or "A-orthogonal") with respect to a matrix A, if

$$u^T A v = 0 \ .$$

- for gradient methods only successive search directions are orthogonal
- main idea of conjugent gradient methods: maintain A-orthogonality of all search directions.

Krylov subspaces

The affine vector spaces

$$K_i(A, d_0) = \mathsf{span}\left(d_0, Ad_0, A^2 d_0, \ldots, A^i d_0\right)$$

are called Krylov subspaces.

Construction of Conjugent Gradient Methods

The main idea is to construct iterates of the form

$$x_i = x_0 + \sum_{j=0}^{i-1} \beta_j d_j$$

such that the coefficients β_j are minimizers of

$$\min_{\beta} \; F\left(x_0 + \sum_{j=0}^{j-1} \beta_j d_j\right) \quad \text{with} \quad F(x) = \frac{1}{2} x^T A x - b^T x \, ,$$

which gives the optimality conditions $(Ax_i - b)^T d_j = 0$ for $j = 1, \ldots, i-1$.

Construction of Conjugent Gradient Methods

Let the previous search directions $d_0 = Ax_0 - b, d_1, \ldots, d_{i-1}$ be an A-orthogonal basis of the Krylov space $K_i(A, d_0)$. We may assume $Ax_i - b \notin K_i(A, d_0)$ as we would have $x_i = x$ otherwise. This motivates to construct the next search direction $d_i \in K_{i+1}(A, d_0)$ from

$$d_i = -(Ax_i - b) + \beta_{i-1} d_{i-1} \ .$$

This direction satisfies the orthogonality condition

$$d_i^T A d_j = -(Ax_i - b)^T A d_j + \beta_{i-1} d_{i-1}^T A d_j = 0$$

for $j = 1, \ldots, i-2$ by construction.

Construction of Conjugent Gradient Methods

The parameter β_{i-1} is then constructed in such a way that we have

$$0 = d_i^T A d_{i-1} = -(Ax_i - b)^T A d_{i-1} + \beta_{i-1} d_{i-1}^T A d_{i-1}$$

$$\Longrightarrow \quad \beta_{i-1} = \frac{(Ax_i - b)^T A d_{i-1}}{d_{i-1}^T A d_{i-1}} \ .$$

Thus yields the recursion law for the conjugent gradient method

$$\begin{aligned} g_{k+1} &= g_k + \alpha_k A g_k \\ x_{k+1} &= x_k + \alpha_k d_k \\ d_{k+1} &= -g_k + \beta_k d_k \end{aligned}$$

with $\alpha_k = \frac{g_k^T d_k}{d_k^T A d_k} = \frac{\|g_k\|_2^2}{d_k^T A d_k}$ and $\beta_k = \frac{\|g_{k+1}\|_2^2}{\|g_k\|_2^2}$. The method is started with $d_0 = -g_0 = b - Ax_0$.

Construction of Conjugent Gradient Methods

The parameter β_{i-1} is then constructed in such a way that we have

$$0 = d_i^T A d_{i-1} = -(Ax_i - b)^T A d_{i-1} + \beta_{i-1} d_{i-1}^T A d_{i-1}$$

$$\implies \quad \beta_{i-1} = \frac{(Ax_i - b)^T A d_{i-1}}{d_{i-1}^T A d_{i-1}} \; .$$

Thus yields the recursion law for the conjugent gradient method

$$\begin{aligned} g_{k+1} &= g_k + \alpha_k A g_k \\ x_{k+1} &= x_k + \alpha_k d_k \\ d_{k+1} &= -g_k + \beta_k d_k \end{aligned}$$

with $\alpha_k = \frac{g_k^T d_k}{d_k^T A d_k} = \frac{\|g_k\|_2^2}{d_k^T A d_k}$ and $\beta_k = \frac{\|g_{k+1}\|_2^2}{\|g_k\|_2^2}$. The method is started with $d_0 = -g_0 = b - Ax_0$.

Convergence of Conjugent Gradient Methods

Since we are constructing an A-orthogonal basis, the conjugent gradient method terminates after at most n steps.

- If we run the conjugent gradient method for n steps, it is a "direct method".
- In practice, the conjugent gradient method is terminated whenever sufficient accuracy is achcieved, e.g., if $\|g_k\| \leq \epsilon$.

Convergence of Conjugent Gradient Methods

Since we are constructing an A-orthogonal basis, the conjugent gradient method terminates after at most n steps.

- If we run the conjugent gradient method for n steps, it is a "direct method".
- In practice, the conjugent gradient method is terminated whenever sufficient accuracy is achcieved, e.g., if $\|g_k\| \leq \epsilon$.