EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 20

Overview

- Steepest descent
- Conjugate gradient

Reading

- Chapter 38 of *Numerical Linear Algebra* by Llyod Trefethen and David Bau
- Chapter 10 of *Matrix Computations* by Gene Golub and Charles Van Loan
- "An Introduction to Conjugate Gradient Method Without the Agonizing Pain" by Jonathan Shewchuk

Quadratic form

- For real symmetric A, a quadratic form is simply a scalar

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top \mathbf{x} + c$$

- Setting the gradient to zero

$$\nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = 0 \Longrightarrow A\mathbf{x} = \mathbf{b}$$

- The solution to $A\mathbf{x} = \mathbf{b}$, $\mathbf{x}_*$, is a critical point of $f(\mathbf{x})$
- If A is positive definite as well, then at arbitrary point $\mathbf{p}$

$$f(\mathbf{p}) = f(\mathbf{x}_*) + \frac{1}{2}(\mathbf{p} - \mathbf{x}_*)^\top A(\mathbf{p} - \mathbf{x}_*) \geq 0$$

  and the latter term is positive for all $\mathbf{p} \neq \mathbf{x}_*$ (and $\mathbf{x}_*$ is a global minimum of f)

Quadratic form (cont'd)

- $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top\mathbf{x} + c$
- At $\mathbf{x}_*$, $A\mathbf{x}_* = \mathbf{b}$
- Let $\mathbf{e} = \mathbf{p} - \mathbf{x}_*$

$$
\begin{aligned}
f(\mathbf{p}) = f(\mathbf{x}_* + \mathbf{e}) &= \tfrac{1}{2}(\mathbf{x}_* + \mathbf{e})^\top A(\mathbf{x}_* + \mathbf{e}) - \mathbf{b}^\top(\mathbf{x}_* + \mathbf{e}) + c \\
&= \tfrac{1}{2}\mathbf{x}_*^\top A\mathbf{x}_* + \mathbf{e}^\top A\mathbf{x}_* + \tfrac{1}{2}\mathbf{e}^\top A\mathbf{e} - \mathbf{b}^\top\mathbf{e} + c \\
&= \tfrac{1}{2}\mathbf{x}_*^\top A\mathbf{x}_* - \mathbf{b}^\top\mathbf{x}_* + c + \mathbf{e}^\top\mathbf{b} + \tfrac{1}{2}\mathbf{e}^\top A\mathbf{e} - \mathbf{b}^\top\mathbf{e} \\
&= f(\mathbf{x}_*) + \tfrac{1}{2}\mathbf{e}^\top A\mathbf{e}
\end{aligned}
$$

Steepest descent

- Start at an arbitrary point $\mathbf{x}_0$ and slide down to the bottom of the paraboloid by taking a series of steps $\mathbf{x}_1, \mathbf{x}_2, \ldots$ until we are satisfied that we are close enough to the solution $\mathbf{x}_*$
- Choose the direction which f decreases most quickly, i.e., the opposite of $\nabla f(\mathbf{x}_i)$

$$-\nabla f(\mathbf{x}_i) = \mathbf{b} - A\mathbf{x}_i$$

- The error $\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_*$ is a vector that indicates how far we are from the solution
- The residual $\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i$ indicates how far we are from the correct value of $\mathbf{b}$
- It is easy to see that $\mathbf{r}_i = -A\mathbf{e}_i$, and residual is being the error transformed by A into the same space as $\mathbf{b}$
- More importantly,

$$\mathbf{r}_i = -\nabla f(\mathbf{x}_i)$$

- Can think of residual as the direction of steepest descent

Steepest descent (cont'd)

- After finding the direction, move to the next point
$$\mathbf{x}_i = \mathbf{x}_{i-1} + \alpha \mathbf{r}_{i-1}$$
- How big is the step?
- A line search is a procedure that chooses α to minimize f along a line
- From basic calculus, α minimizes f when the directional derivative $\frac{d}{d\alpha} f(\mathbf{x}_i)$ is equal to zero
$$\frac{d}{d\alpha} f(\mathbf{x}_i) = \nabla f(\mathbf{x}_i)^\top \frac{d}{d\alpha}\mathbf{x}_i = \nabla f(\mathbf{x}_i)^\top \mathbf{r}_{i-1} = -\mathbf{r}_i^\top \mathbf{r}_{i-1}$$
- To determine α
$$\begin{aligned}
\mathbf{r}_i^\top \mathbf{r}_{i-1} &= 0 \\
(\mathbf{b} - A\mathbf{x}_i)^\top \mathbf{r}_{i-1} &= 0 \\
(\mathbf{b} - A(\mathbf{x}_{i-1} + \alpha \mathbf{r}_{i-1}))^\top \mathbf{r}_{i-1} &= 0 \\
(\mathbf{b} - A\mathbf{x}_{i-1})^\top \mathbf{r}_{i-1} - \alpha (A\mathbf{r}_{i-1})^\top \mathbf{r}_{i-1} &= 0 \\
(\mathbf{b} - A\mathbf{x}_{i-1})^\top \mathbf{r}_{i-1} &= \alpha (A\mathbf{r}_{i-1})^\top \mathbf{r}_{i-1} \\
\mathbf{r}_{i-1}^\top \mathbf{r}_{i-1} &= \alpha \mathbf{r}_{i-1}^\top (A\mathbf{r}_{i-1}) \\
\alpha &= \frac{\mathbf{r}_{i-1}^\top \mathbf{r}_{i-1}}{\mathbf{r}_{i-1}^\top A\mathbf{r}_{i-1}}
\end{aligned}$$

Steepest descent (cont'd)

- Put it all together,

$$\begin{aligned}\mathbf{r}_{i-1} &= \mathbf{b} - A\mathbf{x}_{i-1} \\ \alpha_i &= \frac{\mathbf{r}_{i-1}^\top \mathbf{r}_{i-1}}{\mathbf{r}_{i-1}^\top A\mathbf{r}_{i-1}} \\ \mathbf{x}_i &= \mathbf{x}_{i-1} + \alpha_i \mathbf{r}_{i-1}\end{aligned}$$

- Can save computation by multiply $-A$ to the above equation and adding $\mathbf{b}$ on both sides

$$\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_i A\mathbf{r}_{i-1}$$

- Steepest descent often finds itself taking steps in the same directions as earlier steps
- Convergence rate depends on the conditional number, κ
- Steepest descent can converge quickly if a fortunate starting point is chosen, but is usually at worst when κ is large

Conjugate gradient as a direct method

- We want to solve a system of linear systems: $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{R}^{m \times m}$ is symmetric and positive
- Two non-zero vectors $\mathbf{u}$ and $\mathbf{v}$ are conjugate with respect to A if

$$\mathbf{u}^\top A\mathbf{v} = 0$$

- Since A is symmetric and positive definite

$$\langle \mathbf{u}, \mathbf{v} \rangle_A = \langle A^\top \mathbf{u}, \mathbf{v} \rangle = \langle A\mathbf{u}, \mathbf{v} \rangle = \langle \mathbf{u}, A\mathbf{v} \rangle = \mathbf{u}^\top A\mathbf{v}$$

- Suppose $\{\mathbf{p}_n\}$ is a sequence of m mutually conjugate directions, then the $\mathbf{p}_n$ form a basis of $\mathbb{R}^m$ and we can expand the solution $\mathbf{x}_*$ (unique solution of $A\mathbf{x} = \mathbf{b}$) in this basis

$$\mathbf{x}_* = \sum_{i=1}^{m} \alpha_i \mathbf{p}_i$$

- The coefficients are given by

$$\begin{aligned} \mathbf{b} &= A\mathbf{x}_* = \textstyle\sum_{i=1}^m \alpha_i A\mathbf{p}_i \\ \mathbf{p}_n^\top \mathbf{b} &= \mathbf{p}_n^\top A\mathbf{x}_* = \textstyle\sum_{i=1}^m \alpha_i \mathbf{p}_n^\top A\mathbf{p}_i = \alpha_n \mathbf{p}_n^\top A\mathbf{p}_n \\ \alpha_n &= \frac{\mathbf{p}_n^\top \mathbf{b}}{\mathbf{p}_n^\top A\mathbf{p}_n} = \frac{\langle \mathbf{p}_n, \mathbf{b} \rangle}{\langle \mathbf{p}_n, \mathbf{p}_n \rangle_A} = \frac{\langle \mathbf{p}_n, \mathbf{b} \rangle}{\|\mathbf{p}_n\|_A^2} \end{aligned}$$

Conjugate gradient as a direct method (cont'd)

- First find a sequence of n conjugate directions and then compute the coefficients (require only inner products)
- How to find conjugate directions?
- Gram-Schmidt conjugations: Start with n linearly independent vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$
- For each vector, subtract those parts that are not A-orthogonal to the other processed vectors

$$\begin{aligned} \mathbf{p}_n &= \mathbf{u}_n + \textstyle\sum_{k=1}^{n} \beta_{nk} \mathbf{p}_k \\ \beta_{nj} &= -\frac{\mathbf{u}_n^\top A \mathbf{p}_j}{\mathbf{p}_j^\top A \mathbf{p}_j} \end{aligned}$$

- Problem: Gram-Schmidt conjugation is slow and we have to store all the vectors that we have created

Conjugate gradient as an iterative method

- If we choose the conjugate vectors $\mathbf{p}_n$ carefully, we may not need all of them to obtain a good approximation
- Also, the direct method does not scale well when m is large
- Without loss of generality, assume the initial guess $\mathbf{x}_0 = 0$
- Need a metric to tell us whether we are getting closer to $\mathbf{x}_*$

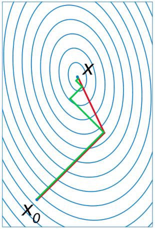
$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top A\mathbf{x} - \mathbf{b}^\top \mathbf{x} \quad , \mathbf{x} \in \mathbb{R}^m$$

  where $\nabla f(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$
- This suggests taking the first basis vector $\mathbf{p}_1$ to be the gradient of f at $\mathbf{x}_0$, i.e., $A\mathbf{x}_0 - \mathbf{b}$
- Since $\mathbf{x}_0 = 0$, $\mathbf{p}_1 = -\mathbf{b}$
- The other direction vectors in the basis will be conjugate to the gradient, hence the name conjugate gradient method

Conjugate gradient as an iterative method (cont'd)

- Let $\mathbf{r}_n$ be the residual at n-th step: $\mathbf{r}_n = \mathbf{b} - A\mathbf{x}_n$
- Note that $\mathbf{r}_n$ is the negative gradient of f at $\mathbf{x} = \mathbf{x}_n$, so the gradient descent method would be to move in the direction of $\mathbf{r}_n$
- Here we insist the directions $\mathbf{p}_n$ are conjugate to each other, so we take the direction closest to the gradient $\mathbf{r}_n$ under the conjugacy constraint

$$\mathbf{p}_{n+1} = \mathbf{r}_n - \frac{\mathbf{p}_n^\top A\mathbf{r}_n}{\mathbf{p}_n^\top A\mathbf{p}_n}\mathbf{p}_n$$



(green: steepest descent, red: conjugate gradient descent)

Conjugate gradients

- The conjugate gradient (CG) iteration is the "original" Krylov subspace iteration
- The most famous of these methods and one the mainstays of scientific computing
- Discovered by Hestenes and Stiefel in 1952, it solves symmetric positive definite systems of equations amazingly quickly if the eigenvalues are well distributed
- Consider the case of 2-norm in solving a nonsingular system of equations $A\mathbf{x} = \mathbf{b}$ with exact solution $\mathbf{x}_* = A^{-1}\mathbf{b}$
- Let $\mathcal{K}_n$ denote the n-th Krylov subspace generated by $\mathbf{b}$

$$\mathcal{K}_n = \langle \mathbf{b}, A\mathbf{b}, \ldots, A^{n-1}\mathbf{b} \rangle$$

- One approach to minimize 2-norm of the residual is based on the Krylov subspace is GMRES

Minimizing the 2-norm of the residual

- In GMRES, at step n, $\mathbf{x}_*$ is approximated by the vector $\mathbf{x}_n \in \mathcal{K}_n$ that minimizes $\|\mathbf{r}_n\|_2$ where $\mathbf{r}_n = \mathbf{b} - A\mathbf{x}_n$
- The usual GMRES algorithm does more work than is necessary for minimizing $\|\mathbf{r}_n\|_2$
- When A is symmetric, faster algorithms are available based on three-term instead of $(n+1)$-term recurrences at step n
- One of these goes by the names of conjugate residuals or MINRES (minimal residuals)

Minimize the A-norm of the error

- Assume that A is real, symmetric, and positive definite
- That means the eigenvalues of A are all positive or equivalently, that $\mathbf{x}^\top A\mathbf{x} > 0$ for every nonzero $\mathbf{x} \in \mathbb{R}^m$
- Under this assumption, the function $\|\cdot\|_A$ defined by

$$\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^\top A\mathbf{x}}$$

  is the A-norm on $\mathbb{R}^m$
- The vector whose A-norm will concern us is $\mathbf{e}_n = \mathbf{x}_* - \mathbf{x}_n$, the error at step n
- The conjugate gradient iteration is a system of recurrence formulas that
  - generated the unique sequence of iterates $\{\mathbf{x} \in \mathcal{K}_n\}$
  - with the property that at step n, $\|\mathbf{e}_n\|_A$ is minimized
- Will reveal the use of orthogonality in minimizing $\|\mathbf{e}_n\|_A$

The conjugate gradient iteration

- To solve $A\mathbf{x} = \mathbf{b}$
- Algorithm:

$$
\begin{aligned}
&\mathbf{x}_0 = \mathbf{0},\ \mathbf{r}_0 = \mathbf{b},\ \mathbf{p}_0 = \mathbf{r}_0 \\
&\textbf{for } n = 1, 2, 3, \ldots \textbf{ do} \\
&\quad \alpha_n = (\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1})/(\mathbf{p}_{n-1}^\top A\mathbf{p}_{n-1}) \quad \text{// step length} \\
&\quad \mathbf{x}_n = \mathbf{x}_{n-1} + \alpha_n \mathbf{p}_{n-1} \quad \text{// approximate solution} \\
&\quad \mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n A\mathbf{p}_{n-1} \quad \text{// residual} \\
&\quad \beta_n = (\mathbf{r}_n^\top \mathbf{r}_n)/(\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1}) \quad \text{// improvement this step} \\
&\quad \mathbf{p}_n = \mathbf{r}_n + \beta_n \mathbf{p}_{n-1} \quad \text{// search direction} \\
&\textbf{end for}
\end{aligned}
$$

- Very simple - programmable in a few lines of MATLAB
- Deals only with m-vectors, not with individual entries of vectors or matrices
- The only complication is the choice of a convergence criterion

The conjugate gradient iteration (cont'd)

- At each step, the conjugate gradient iteration involves several vector manipulation and one matrix-vector product, the computation of $A\mathbf{p}_{n-1}$
- If A is dense and unstructured, the matrix-vector product dominates the operation count, $O(m^2)$ flops for each step
- If A is sparse or has other structure that can be exploited, the operation count may be as low as $O(m)$ flops per step

The conjugate gradient iteration (cont'd)

Theorem

Let the conjugate gradient iteration be applied to a symmetric positive definite matrix problem $A\mathbf{x} = \mathbf{b}$. As long as the iteration has not yet converged (i.e., $\mathbf{r}_{n-1} \neq 0$), the algorithm proceeds without divisions by zero, and we have the following identities of subspaces:

$$\begin{aligned} \mathcal{K}_n &= \langle \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \rangle = \langle \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1} \rangle \\ &= \langle \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{n-1} \rangle = \langle \mathbf{b}, A\mathbf{b}, \ldots, A^{n-1}\mathbf{b} \rangle \end{aligned} \tag{1}$$

Moreover, the residuals are orthogonal

$$\mathbf{r}_n^\top \mathbf{r}_j = 0 \quad (j < n) \tag{2}$$

and the search directions are A-conjugate

$$\mathbf{p}_n^\top A \mathbf{p}_j = 0 \quad (j < n) \tag{3}$$

The conjugate gradient iteration (cont'd)

- (Proof by induction) From the initial guess $\mathbf{x}_0 = 0$ and the formula $\mathbf{x}_n = \mathbf{x}_{n-1} + \alpha_n \mathbf{p}_{n-1}$, it follows by induction that $\mathbf{x}_n$ belongs to $\langle \mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{n-1} \rangle$
- From $\mathbf{p}_n = \mathbf{r}_n + \beta_n \mathbf{p}_{n-1}$, it follows that this is the same as $\langle \mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_{n-1} \rangle$
- From $\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n A\mathbf{p}_{n-1}$, it follows that this is the same as $\langle \mathbf{b}, A\mathbf{b}, \ldots, A^{n-1}\mathbf{b} \rangle$
- To prove (2), apply the formula $\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n A\mathbf{p}_{n-1}$ and the identity $(A\mathbf{p}_{n-1})^\top = \mathbf{p}_{n-1}^\top A$ to compute
$$\mathbf{r}_n^\top \mathbf{r}_j = \mathbf{r}_{n-1}^\top \mathbf{r}_j - \alpha_n \mathbf{p}_{n-1}^\top A\mathbf{r}_j$$
- If $j < n-1$, both terms on the right are zero by induction
- If $j = n-1$, the difference on the right is zero provided $\alpha_n = (\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1})/(\mathbf{p}_{n-1}^\top A\mathbf{r}_{n-1})$
- Note it is almost the same as the line $\alpha_n = (\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1})/(\mathbf{p}_{n-1}^\top A\mathbf{p}_{n-1})$
- Since $\mathbf{p}_{n-1}$ and $\mathbf{r}_{n-1}$ differ by $\beta_{n-1}\mathbf{p}_{n-2}$, the effect of this replacement is to change the denominator by $\beta_{n-1}\mathbf{p}_{n-1}^\top A\mathbf{p}_{n-2}$, which is zero by induction

The conjugate gradient iteration (cont'd)

- To prove (3), we apply the formula $\mathbf{p}_n = \mathbf{r}_n + \beta_n \mathbf{p}_{n-1}$ to compute
$$\mathbf{p}_n^\top A\mathbf{p}_j = \mathbf{r}_n^\top A\mathbf{p}_j + \beta_n \mathbf{p}_{n-1}^\top A\mathbf{p}_j$$
- If $j < n-1$, both terms on the right are again zero by induction
- If $j = n-1$, the sum on the right is zero provided $\beta_n = -(\mathbf{r}_n^\top A\mathbf{p}_{n-1})/(\mathbf{p}_{n-1}^\top A\mathbf{p}_{n-1})$, which we can write equivalently in the from $\beta_n = (-\alpha_n \mathbf{r}_n^\top A\mathbf{p}_{n-1})/(\alpha_n \mathbf{p}_{n-1}^\top A\mathbf{p}_{n-1})$
- Recall $\mathbf{r}_n = \mathbf{r}_{n-1} + \alpha_n \mathbf{p}_{n-1}$ and $\mathbf{r}_n^\top \mathbf{r}_n = \mathbf{r}_n^\top(\mathbf{r}_{n-1} + \alpha_n \mathbf{p}_{n-1}) = \alpha_n \mathbf{r}_n^\top \mathbf{p}_{n-1}$
- Likewise, use $\mathbf{r}_n = \mathbf{r}_{n-1} - \alpha_n A\mathbf{p}_{n-1}$ and $\mathbf{p}_n = \mathbf{r}_n + \beta_n \mathbf{p}_{n-1}$ to show $\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1} = \mathbf{r}_{n-1}^\top(\mathbf{r}_n + \alpha_n A\mathbf{p}_{n-1}) = (\mathbf{p}_{n-1} - \beta_{n-1}\mathbf{p}_{n-2})^\top \alpha_n A\mathbf{p}_{n-1} = \mathbf{p}_{n-1}^\top(\alpha_n A\mathbf{p}_{n-1})$
- This is the same as the line $\beta_n = (\mathbf{r}_n^\top \mathbf{r}_n)/(\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1})$ except that $\mathbf{r}_n^\top \mathbf{r}_n$ has been replaced by $\mathbf{r}_n^\top(-\alpha_n A\mathbf{p}_{n-1})$ and $\mathbf{r}_{n-1}^\top \mathbf{r}_{n-1}$ has been replaced by $\mathbf{p}_{n-1}^\top(\alpha_n A\mathbf{p}_{n-1})$