Lecture 22

Scribe?

HW 5 out today

Last time

▷ Convergence guarantees for BFGS.
▷ Proof

Today

▷ L-BFGS
▷ Conjugate gradient method

L-BFGS

With BFGS, we solved multiple problems
+ We have descent.
+ Local superlinear convergence.
+ Only have to compute $\nabla f(x_k)$ per iter.
- However, we have a storage cost of $O(d^2)$. Thus it only works up to $d = 10^4 \sim 10^5$ (on personal computer).

To tackle higher sizes, we can forget far away iterates.
In HW 5 you'll show that BFGS updates
\[ B^{-1}_k = B_0^{-1} + \alpha_1 w w_1^T + \ldots + \alpha_k w_k w_k^T \]
Instead of keeping all \( 2k \) vectors we can just keep that last \( m \) where \( m \approx 2 - 30 \) (usually).

This leads to \( O(dm) \) memory when \( B_0 = I \) or any other simple to apply linear map.

Because
\[ B^{-1}_k \nabla f(x_k) = Df(x_k) + \sum_{j=k-m}^{k} \alpha_j (w_j \nabla f(x_k)) w_j \]

Conjugate gradient

Today we go back to least squares.

\[ \min_x \frac{1}{2} x^T A x - b^T x \]

where \( A > 0 \) and \( b \in \mathbb{R}^d \).

Optimality conditions say this is...
equivalent to solving \( Ax = b. \)

What would happened if I knew \( A = V \Lambda V^T? \)

I could simplify the problem (change of basis, spectral decomposition)

\[
\min_{x \in \mathbb{R}^d} \frac{1}{2} x^T A x - b^T x = \min_{y \in \mathbb{R}^d} \frac{1}{2} (V^T y)^T \Lambda (V^T y) - b^T V^T y
\]

\[
= \min_{y \in \mathbb{R}^d} \frac{1}{2} y^T \Lambda y - b^T V^T y
\]

This is a separable problem.

\[
\Rightarrow \minimized \text{ at } y_i^* = \arg\min y_i \lambda_i - b^T V_i y_i
\]

\[
= \frac{v_i^T b}{\lambda_i}
\]

\[
\minimized \text{ at } x^* = \sum V_i y_i^* = \sum \frac{v_i v_i^T b}{\lambda_i}
\]

Computing the spectral decomposition is too expensive.
Question: Is there a cheaper way to obtain a separable problem?

Conjugate vectors

Any $A > 0$ defines an inner product
\[
\langle x, y \rangle_A = x^T A y
\]

**Def.** 1. Two vectors $x \neq y$

$A$-conjugate if
\[
\langle x, y \rangle_A = 0.
\]

2. Given a linear subspace $L \subseteq \mathbb{R}^d$

$L^A = \{ y \mid \langle x, y \rangle_A = 0 \text{ for all } x \in L \}$.

3. The projection of $x$ onto $y$ w.r.t. $\langle \cdot, \cdot \rangle_A$ is
\[
\rho_y^A(x) = \frac{\langle x, y \rangle_A}{\langle y, y \rangle_A} y.
\]

**Lemma:** Let $s_1, \ldots, s_k$ be $A$-conjugate pairwise. Then, they are linearly independent and for all $x \in \text{span}\{s_1, \ldots, s_k\}$,
\[ x = \sum_{i=1}^{k} P_{s_i}(x). \]

Note that if \( s_1, \ldots, s_d \) are A-conjugate then
\[ S = (s_1, \ldots, s_d) \]
yields
\[ \min_{x \in \mathbb{R}^d} \frac{1}{2} x^T A x - b^T x = \min_{y \in \mathbb{R}^d} \frac{1}{2} (s y)^T A (s y) - b^T s y \]
\[ = \min \sum_{i,j} y_i s_i A s_j y_j \]
Decomposable \[ - \sum_{i} b^T s_i y_i \]
\[ = \min \sum_{i} y_i^2 s_i^T A s_i - b^T s_i y_i \]
\[ \Rightarrow y_i^* = \frac{b^T s_i}{s_i^T A s_i} \quad \Rightarrow \quad x^* = \sum_{i=1}^{d} \frac{s_i^T s_i^T b}{s_i^T A s_i} . \]

Question: How do we obtain conjugate vectors?

Gram-Schmidt Orthogonalization
Input: $A \succ 0$, and linearly independent $x_1, \ldots, x_k$
Output: $s_1, \ldots, s_k$ $A$-conjugates s.t.

$\text{span}\{s_i\} = \text{span}\{x_i\}$.

\begin{itemize}
  \item $S_1 = x_1$
  \item Recursively update
    \[ s_{i+1} = x_{i+1} - \sum P_{s_i}^A(x_{i+1}) \]
\end{itemize}

Check:

\begin{itemize}
  \item $\text{span}\{s_i\} = \text{span}\{x_i\}$
  \item $\langle s_i, s_j \rangle = 0$ \text{ \textforall } j < i+1$
  \item $s_{i+1} \neq 0$.
\end{itemize}

This algorithm is nice but when $k = d$, we have that we have to do $d$ steps each with complexity $O(d^2)$ $\Rightarrow$ $O(d^3)$ complexity.

Matrix multiplication to compute $\langle x, s_i \rangle_A$.

Question: can we find a good approximation of the solution of $Ax = b$ without doing $O(d^3)$ work?
Conjugate Gradient Method

Idea: Construct the basis \( s_1, \ldots, s_k \) using the residuals

\[
r_k = b - Ax_k = \nabla f(x_k).
\]

Then select

\[
(\star) \quad x_{k+1} = \text{arg min}_{s \in x_k + \text{span}(s_1, \ldots, s_k)} f(x)
\]

Let us see two supporting results.

**Lemma 3:** Let \( x_0 \) and \( s_1, \ldots, s_k \) be any vectors. Consider \( x_{k+1} \) given by \((\star)\), then \( \nabla f(x_{k+1}) \) is orthogonal (in the standard sense) to \( \text{span}(s_1, \ldots, s_k) \).

**Proof:** Equivalently

\[
y^* = \text{arg min}_{y \in \mathbb{R}^k} f(x_0 + Sy)
\]

By 1st-order optimality conditions.
\[
S^T \nabla f(x_0 + Sy^*) = 0 \\
\Rightarrow \nabla f(x_{k+1}) \text{ is orthogonal to span}\{s_1, \ldots, s_k\}.
\]

Thanks to separability:

**Lemma A:** Suppose that \( x_{k+1} \) is given by (A) and \( s_{k+1} \) is \( A \)-conjugate to each \( s_i \). Then,

\[
x_{k+2} \in \arg\min \ f(x) \\
\text{s.t.} \ x = x_{k+1} + \text{span}\{s_{k+1}\}
\]

is also a solution of

\[
x_{k+2} \in \arg\min \ f(x) \\
\text{s.t.} \ x = x_0 + \text{span}\{s_1, \ldots, s_{k+1}\}.
\]

**CG Method**

**Input:** \( x_0 \in \mathbb{R}^d, \ s_0 = r_0 = b - Ax_0 \)

**Update** \( i \leq d \):

\[
x_i = \arg\min \ f(x_i + \alpha s_i) \\
x_{i+1} = x_i + \alpha_i s_i \\
r_{i+1} = -\nabla f(x_{i+1}) = b - Ax_{i+1} \\
S_{i+1} = r_{i+1} - \sum P_{s_i}^A(r_{i+1})
\]

(Gram-Schmidt)