Least Squares with Gradient Descent

Consider a least-squares problem with a matrix A and vector b. Formulate this as a quadratic minimization problem and use a gradient descent algorithm to solve the least-squares problem. The gradient descent algorithm should not involve any matrix inversions.

Analysis of the problem:

Algorithm of Gradient Descent Method:

---

given a starting point $x \in \mathbf{dom}\, f$.
repeat
  1. $\Delta x := -\nabla f(x)$.
  2. *Line search.* Choose step size t via exact or backtracking line search.
  3. *Update.* $x := x + t\Delta x$.
until stopping criterion is satisfied.

Least-square problem: $f_0 = \|Ax - b\|_2$
Formulate this Least-square problem as a quadratic minimization problem:
Objective function:

$$f(x) = \|Ax - b\|_2{}^2 = (Ax - b)^T(Ax - b) = [(Ax)^T - b^T](Ax - b) =$$
$$(Ax)^T Ax - b^T Ax - (Ax)^T b + b^T b = (Ax)^T Ax - b^T Ax - x^T A^T b + b^T b =$$
$$(Ax)^T Ax - b^T Ax - (b^T Ax)^T + b^T b = x^T A^T Ax - 2x^T A^T b + b^T b$$

Step 1: compute Δx:
Gradient: $\nabla f(x) = 2A^T Ax - 2A^T b$
Then $\Delta x = -\nabla f(x)$
Step 2: find t through exact line search:

For exact line search, $\mathrm{t} = \underset{t>0}{arg\,min}\, f(x + t\Delta x)$

$$\begin{aligned} f(x + t\Delta x) &= (x + t\Delta x)^T A^T A(x + t\Delta x) - 2(x + t\Delta x)^T A^T b + b^T b \\ &= x^T A^T Ax + t\Delta x^T A^T Ax + x^T A^T At\Delta x + t\Delta x^T A^T At\Delta x - 2x^T A^T b \\ &\quad - 2t\Delta x^T A^T b + b^T b \end{aligned}$$

Take derivative of $f(x + t\Delta x)$ with respect to t, making it vanish,

$$\begin{aligned} \nabla_t f(x + t\Delta x) &= \nabla_t (t\Delta x^T A^T Ax + x^T A^T At\Delta x + t\Delta x^T A^T At\Delta x - 2t\Delta x^T A^T b) \\ &= \Delta x^T A^T Ax + x^T A^T A\Delta x + 2t\Delta x^T A^T A\Delta x - 2\Delta x^T A^T b = 0 \end{aligned}$$

Solving it, we get: $\mathrm{t} = \frac{2\Delta x^T A^T b - \Delta x^T A^T Ax - x^T A^T A\Delta x}{2\Delta x^T A^T A\Delta x}$

Step 3: update $x = x + t\Delta x$

The stopping criterion:

We cannot stop until the function value of the current step *f(x)* is very close to the optimal value p^*. i.e. $f(x) - p^* < \varepsilon$

Assume $x, y \in \mathrm{dom} f$, the second-order Taylor approximation of f(y) (y is any arbitrary point in domf) at point x is:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x)$$

Note that f(x) is convex, therefore, Taylor approximation of f(y) is an under estimator of f(y).

At optimal point, $\nabla f(x^*) = 0$, since $f(x) = \|Ax - b\|_2{}^2$ is convex, we further assume it to be strong convex, i.e. $\nabla^2 f(x) \succcurlyeq mI,\ where\ m > 0$

We have: $f(y) \geq f(x) + \frac{m^2}{2}\|y - x\|_2{}^2, m > 0$

$f(x) - f(y) \leq \frac{m^2}{2}\|y - x\|_2{}^2$, which is an upper bound of $f(x) - p^*,$ if y is the optimal solution ($f(y) = p^*$). However, this upper bound is with respect to $\|y - x\|_2$, which is not satisfactory, therefore, we want to find an upper bound of $f(x) - p^*$, with respect to function value of f.

Since $\nabla f(x) = \frac{f(y) - f(x)}{\|y - x\|_2}$, plug it into the above function,

$$f(x) - f(y) \leq -\frac{m^2}{2}\|y - x\|_2{}^2 = -\frac{m^2}{2}[\frac{f(x) - f(y)}{\nabla f(x)}]^2, m > 0$$

let y be the optimal solution ($f(y) = p^*$), we have $f(x) - p^* \leq \frac{\|\nabla f(x)\|_2{}^2}{2m}$

$\frac{\|\nabla f(x)\|_2{}^2}{2m}$ is the upper bound of the gap between the final-step function value and optimal value, i.e. $f(x) - p^*$.

We want to limit it to be smaller than a predetermined threshold ε, $\frac{\|\nabla f(x)\|_2{}^2}{2m} < \varepsilon$,

then $\|\nabla f(x)\|_2 < (2m\varepsilon)^{\frac{1}{2}}$. Since m is a constant depending on coordinates; ε is a constant depending on accuracy requirement, we sometimes combine them, The stopping criterion is: $\|\nabla f(x)\|_2 < \varepsilon$

Implement in Matlab: