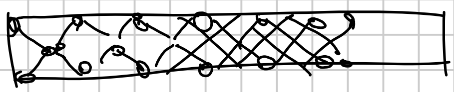
Note Title 2024-10-04

$A \in \mathbb{R}^{m \times m}$ $\quad m = 100{,}000$

$O(m^3)$ time, $O(m^2)$ space is unfeasible



Format: list of (i, j, A_{ij}) with $A_{ij} \neq 0$

Also: CSC, CSR (compressed sparse row/column)

Operations in this format: matrix-vector product:

$$w = Av \qquad w_i = \sum_{\substack{j=1 \\ A_{ij} \neq 0}}^{m} A_{ij} v_j$$

$w = 0;$

for (i, j, A_{ij}) in A:

$\quad w[i] \leftarrow w[i] + A[i,j] * v[j]$

end

$O(\mathrm{nnz}(A))$

Solving linear systems?

Using optimization ideas, iteratively / approximately

$$f(x) = \frac{1}{2} x^T Q x + q^T x + \text{const} \qquad Q \succ 0 \text{ pos. def.}$$

minimum in $x_* = -Q^{-1} q \qquad v := -q$

Let us start from a simple case $Q = I$

$$\min_{y \in \mathbb{R}^m} \frac{1}{2} \|y - w\|^2 = \frac{1}{2}(y-w)^T(y-w) = \frac{1}{2} y^T y - \underbrace{\frac{1}{2} w^T y - \frac{1}{2} y^T w}_{\text{equal}} + \frac{1}{2} w^T w$$

$$= \frac{1}{2} y^T y - w^T y + \text{const}$$

$$= \frac{1}{2}\left(y_1^2 + \dots + y_m^2\right) - \left(w_1 y_1 + \dots + w_m y_m\right) + \text{const}$$

In this case, solution given by $y = w$.

$y^0 = 0$

$$y^1 = \begin{bmatrix} w_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad y^2 = \begin{bmatrix} w_1 \\ w_2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad y^3 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \dots \quad y^m = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ w_m \end{bmatrix}$$

Essentially, at each step we are doing a line search along a direction of the canonical basis $e^k = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow$ pos. k

$$\varphi_{y^k, e^k} = y^k + \alpha e^{k+1} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \\ \alpha \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

minimum in

$$y^{k+1} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \\ w_{k+1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

<u>Note</u> at each step, we are solving a problem on a subspace of dimension 1:

$$y_{k+1} = \min f(y) \text{ over the vectors of the form } \begin{bmatrix} w_1 \\ \vdots \\ w_k \\ * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

but also

$$y_{k+1} = \min f(y) \text{ over the vectors of form } \begin{bmatrix} * \\ * \\ \vdots \\ * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} * \\ * \\ \vdots \\ * \end{matrix}}\right\} k+1 \\ \left.\vphantom{\begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix}}\right\} \text{zeros} \end{matrix}$$

The same idea works for any orthogonal basis

$$U = \begin{bmatrix} u^1 & u^2 & \cdots & u^m \end{bmatrix}$$

$$w = u^1 c_1 + u^2 c_2 + \cdots + u^m c_m \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

$$= U c$$

<u>Algorithm:</u> Given $u^1, u^2, \ldots u^m$ orthonormal basis,

$y^0 = 0$

for $k = 1, 2, \ldots m$

$y^{k+1} = \arg\min \frac{1}{2} \|y - w\|^2 + \text{const}$ over $\{y^k + \alpha u^k\}$;

(minimum of a quadratic function in α / parabola)

end

Essentially, you are computing one entry of c at a time:

$w = Uc$

$$y^k = U \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

At each step,

$$y^{k+1} = \arg\min f(y) \quad \text{over} \quad U \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \\ * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow \text{unknown}$$

$$= \arg\min f(y) \quad \text{over} \quad U \begin{bmatrix} * \\ * \\ \vdots \\ * \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \Big\}_{k+1} = \operatorname{span}(u^1, \ldots u^k)$$

converges exactly after m steps.

Take any invertible matrix R, $\quad y = Rx \quad x = R^{-1} y$

$$f(y) = \frac{1}{2} y^T y - W^T y + \text{const} = \frac{1}{2} x^T \underbrace{R^T R}_{Q} x - \underbrace{W^T R}_{v^T} x + \text{const}$$

$$= \frac{1}{2} x^T Q x - v^T x + \text{const}$$

With suitable choices of R, w, we can obtain any $Q \succ 0, v$

y space

Given orthogonal search dirs u

$y^0 = 0$

for $k = 1, 2, \dots m$

$\quad y^{k+1} = \arg\min \frac{1}{2} y^T y - W^T y + \text{const}$

$\quad$ over $\{ y^k + \alpha u^k \}$ $\quad Rx^k + \alpha R d^k$

end

x space

Given directions $d^1, d^2, \dots d^m$

$\boxed{R d^k = u^k}$

$y = Rx$

$u = Rd$

$x^0 = 0$

for $k = 1, 2, \dots m$

$\quad x^{k+1} = \arg\min \frac{1}{2} x^T \underbrace{R^T R}_{Q} x - \underbrace{W^T R}_{v^T} x + \text{const}$

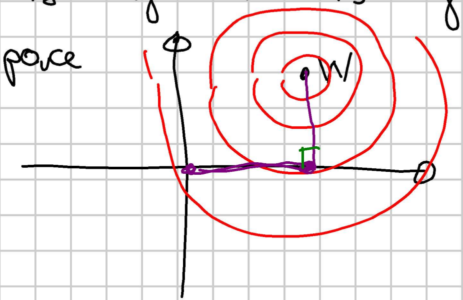
$\quad$ over $\{ x^k + \alpha d^k \}$

end

At each step, we can still solve the internal problem, univariate problem in α (line search), knowing only Q, v

This algorithm is guaranteed to converge in m steps!

y space



x space

The directions $Rd^j = u^j$ satisfy

$$u_i^T u_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \qquad d_i R^T R d_j = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

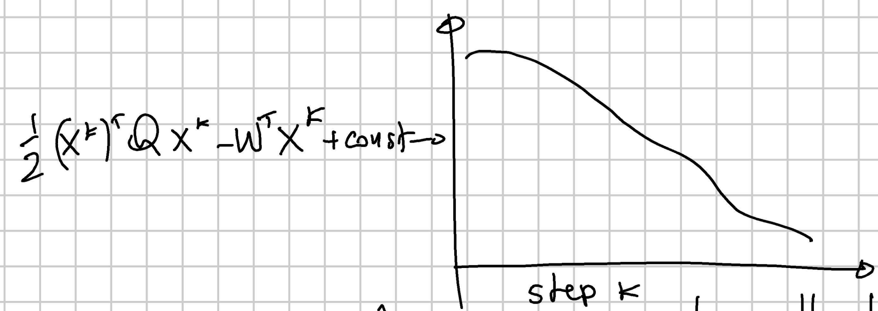
$$\parallel$$

$$d_i^T Q d_j = \begin{cases} 0 \\ 1 \end{cases}$$

These directions are <u>Q-orthogonal</u> or <u>Q-conjugate</u>, i.e., $d_i^T Q d_j = \begin{cases} 0 \\ 1 \end{cases}$

At each step, we compute x^{k+1} by solving arg min over $\{x^k + \alpha d^k\}$, but we get also

$$x^{k+1} = \arg\min \text{ over } \{d^1, d^2, \dots d^k\}$$



decrease monotonically!

How to get directions d^j?

At step 0, just take $d^0 = v - Qx^0$
(negative gradient)

<u>Def</u>: the <u>residual</u> of x^k is $v - Qx^k = r^k$
(negative gradient)

At step 1, we start from $r^1 = v - Qx^1$, and orthogonalize it:

$$d^1 = r^1 + \beta d^0, \text{ where}$$

β is chosen such that $0 = (d^0)^T Q d^1 = (d^0)^T Q (r^1 + \beta d^0)$

$$\Leftrightarrow \beta = \frac{(d^0)^T Q r^1}{(d^0)^T Q d^0}$$

At each step k, it is sufficient to take

$$d^k = r^k - \beta d^{k-1} \quad \text{with } \beta \text{ chosen s.t. } (d^{k-1})^T Q d^k = 0$$

and (magically) it is also Q-orthogonal to <u>all</u> previous directions

---

Algorithm (conjugate gradient)

$x^0 = 0 \qquad r^0 = v - Q \cdot 0 = v \qquad d^0 = v$

for $k = 1, 2, 3, \dots m$

$$\alpha_k = ((r^{k-1})^T r^{k-1}) / ((d^{k-1})^T \boxed{Q d^{k-1}}).$$

compute once!

$$x^k = x^{k-1} + \alpha_k d^{k-1} \qquad \text{(line search)}$$

$$r^k = v - Qx^k = v - Q(x^{k-1} + \alpha_k d^{k-1}) = r^{k-1} + \alpha_k \boxed{Q d^{k-1}}$$

$$\beta_k = ((r^k)^T r^k) / ((r^{k-1})^T (r^{k-1})) = \frac{\|r_k\|^2}{\|r_{k-1}\|^2}$$

$$d^k = r^k + \beta_k d^{k-1}$$

end

<u>Storage</u>: $\sim 3m$ (the vectors x^k, d^k, r^k at each step)

<u>Cost</u>: $\underbrace{1 \text{ matvec}}_{O(\text{nnz}(A))} + O(m)$ operations per step.

"black box algorithm": if we have a fast implementation of $\lambda x. Qx$, we just need to call that once no very efficient for sparse matrices!

Def: Given $Q \in \mathbb{R}^{m \times m}$, $v \in \mathbb{R}^m$, $n \leq m$, define

$$K_n(Q,v) = \text{span}(v, Qv, Q^2v, \dots Q^{n-1}v)$$

<u>Krylov space</u>

← matrix powers!

$$Q^3 = Q \cdot Q \cdot Q$$

$$w = vc_1 + Qvc_2 + Q^2vc_3 + \dots + Q^{n-1}vc_n$$

$$= (\underbrace{c_1 I + c_2 Q + \dots + c_n Q^{n-1}}_{p(Q)})v$$

$p(Q)$, p polynomial of degree $< n$

$K_1(Q,v)$ = multiples of v, $K_2(Q,v)$ = l.c. of v and Qv

If $v, Qv, \dots Q^{n-1}v$ are lin. ind., then for every $w \in K_n(Q,v)$, the c_i are <u>unique</u>

The <u>degree</u> of w (with respect to Q,v) is well defined

If w has degree $n-1$, then $w \in K_n(Q,v) \setminus K_{n-1}(Q,v)$

If w has degree $n-1$, then Qw has degree n

<u>Ex</u>:

$$w = v + Qv \cdot 2 - Q^2v \cdot 3 \quad \in K_3(Q,v)$$

$$Qw = Qw + Q^2v \cdot 2 - Q^3v \cdot 3 \quad \in K_4(Q,v) \qquad w \in \mathbb{R}^m$$

<u>Obs</u>: the iterates of the gradient method

belong to Krylov subspaces:

$$f(x) = \frac{1}{2} x^T Q x - v^T x$$

$$x^0 = 0$$

$$x^1 = -g^0 \alpha = (v - Q x_0) \cdot \alpha = v\alpha \quad \in K^1(Q, v)$$

$$x^2 = x^1 - g^1 \beta = v\alpha - (v - Qv\alpha)\beta \quad \in K^2(Q, v)$$

$$x^3 = x^2 - g^2 \gamma = x^2 - (v - Q x^2)\gamma \quad \in K^3(Q, v)$$

The same holds for CG.