Feb 4, 2020

Last time: gradient descent $x_{k+1} = x_k - \alpha_k \underbrace{\nabla f(x_k)}_{g_k}$

error analysis: $e_k = x_k - x_*$ $\left(f(x) = \frac{1}{2} x^T A x + x^T b + c\right)$

$\|e_{k+1}\| < a\|e_k\|$ $a < 1$ $\|e_{k+1}\| \leq O(a^k)$

Newton: $x_{k+1} = x_k - H_k^{-1} g_k$

Stochastic gradient descent (SGD) $x_{k+1} = x_k - \alpha_k g_{Jk}$

$f(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x)$ $g(x) = \frac{1}{N} \sum_{i=1}^{N} g_i(x)$

$J \subseteq \{1, \ldots, N\}$ $g_{Jk} = \frac{1}{|J|} \sum_{j \in J} g_j(x)$ $\mathbb{E}(g_{Jk}) = g_k$

Should we expect SGD to converge?

$$x_{k+1} = x_k - \alpha_k (g_k + u_k) \quad \text{error/noise}$$

$$f(x) = \frac{1}{2} x^T A x + x^T b + c \qquad g(x) = Ax + b$$

$b = -Ax^*$

$$x_{k+1} - x^* = x_k - x^* - \alpha (A x_k + b + u_k)$$

$$e_{k+1} = e_k - \alpha (A(x_k - x^*)) - \alpha u_k$$

$$= (I - \alpha A) e_k - \alpha u_k$$

$$e_1 = (I - \alpha A) e_0 - \alpha u_0$$

$$e_2 = (I - \alpha A)^2 e_0 - \alpha (I - \alpha A) u_0 - \alpha u_1$$

$$\vdots$$

$$e_{k+1} = (I - \alpha A)^{k+1} e_0 - \sum_{j=0}^{k} (I - \alpha A)^{k-j} u_j$$

control as before

$O(\alpha^k)$

$$\left\| \sum_{j=0}^{k} (I - \alpha A)^{k-j} u_j \right\| \qquad \|u_j\| \le C\gamma^{-j} = C\gamma^{-k}\gamma^{k-j}$$

$$\le \sum_{j=0}^{k} \|u_j\| \, \|I - \alpha A\|^{k-j}$$

$$\le C\gamma^{-k} \sum_{j=0}^{k} \gamma^{k-j} \|I - \alpha A\|^{k-j} \qquad \ell \qquad \|e_{k+1}\|$$

$$\le C\gamma^{-k} \sum_{\ell=0}^{\infty} (\gamma \|I - \alpha A\|)^{\ell} \le \frac{C\gamma^{-k}}{1 - \gamma\|I - \alpha A\|}$$

gradients become more accurate over time, can still converge!

SGD: $x_{k+1} = x_k - \alpha g_{Jk}$ $|J| = 1$

If f is "nice enough", α is small enough

$$\mathbb{E}(f_k - f_*) \leq c_1 \alpha + (1 - c_2\alpha)^{-k}(f_0 - f_*)$$

$f_k = f(x_k)$

$f_* = f(x_*)$

↑ reduce to get closer to OPT

↑ smaller α makes convergence slower

Idea: r steps with $\alpha_0 \Rightarrow O(\alpha_0)$ error

$2r$ steps with $\alpha_0/2 \Rightarrow O(\alpha_0/2)$ error

$4r$ steps $\alpha_0/4$ $O(\alpha_0/4)$

learning rate sched.

Convergence looks like $O(1/r)$

GD: $O(a^r)$

still have problems w/ ill-conditioning

Common SGD heuristic:

keep g_{jk} same scale on all components

$$R_k \approx |g_{jk}|^2 \text{ (entry-wise)}$$

$$x_{k+1} = x_k - \alpha g_{jk}$$

$$(x_{k+1})_i = (x_k)_i - \frac{\alpha}{\sqrt{(R_k)_i + u}} (g_{jk})_i$$

RMSProp: $R_k = (1-\lambda) R_k + \lambda g_{jk}^2$ (entry-wise)

Adagrad: $R_k = R_{k-1} + g_{jk}^2$

Adam, Adadelta

(5) GD with momentum

$$x_{k+1} = x_k - \alpha_k g_k + \beta_k (x_k - x_{k-1})$$

$\alpha_k = \alpha \quad \beta_k = \beta$ (heavy ball)

$$x_1 = x_0 - \alpha g_0$$

$$x_2 = x_1 - \alpha g_1 + \underbrace{\beta (x_1 - x_0)}_{-\alpha g_0} =$$

$$= x_1 - \alpha g_1 - \beta \alpha g_0$$

$$\vdots$$

$$x_{k+1} = x_k - \alpha \sum_{j=1}^{k} \beta^{k-j} g_j$$

(S) GD with acceleration

$$\bar{x}_k = x_k + \beta_k (x_k - x_{k-1})$$

$$x_{k+1} = \bar{x}_k - \alpha_k g(\bar{x}_k)$$

$$x_{k+1} = x_k - \alpha_k g(x_k + \beta_k (x_k - x_{k-1})) + \beta_k (x_k - x_{k-1})$$

Nesterov acceleration

can choose α_k, β_k to get best possible convergence for gradient-based methods for smooth cvx functions

Coordinate descent (CD)

$$\min f(x) \qquad f: \mathbb{R}^n \to \mathbb{R}$$

Idea: only opt. one x_i at a time

for $i = 1, 2, \dots n$

$$x_{k+1} = \min_{\alpha} f(x_k + \alpha e_i)$$

$$\left(x_{k+1} = x_k - \alpha_k \left[\frac{\partial f}{\partial x_i}(x) \right] e_i \right)$$

} repeat

Next topic: latent factor models / dim. redux

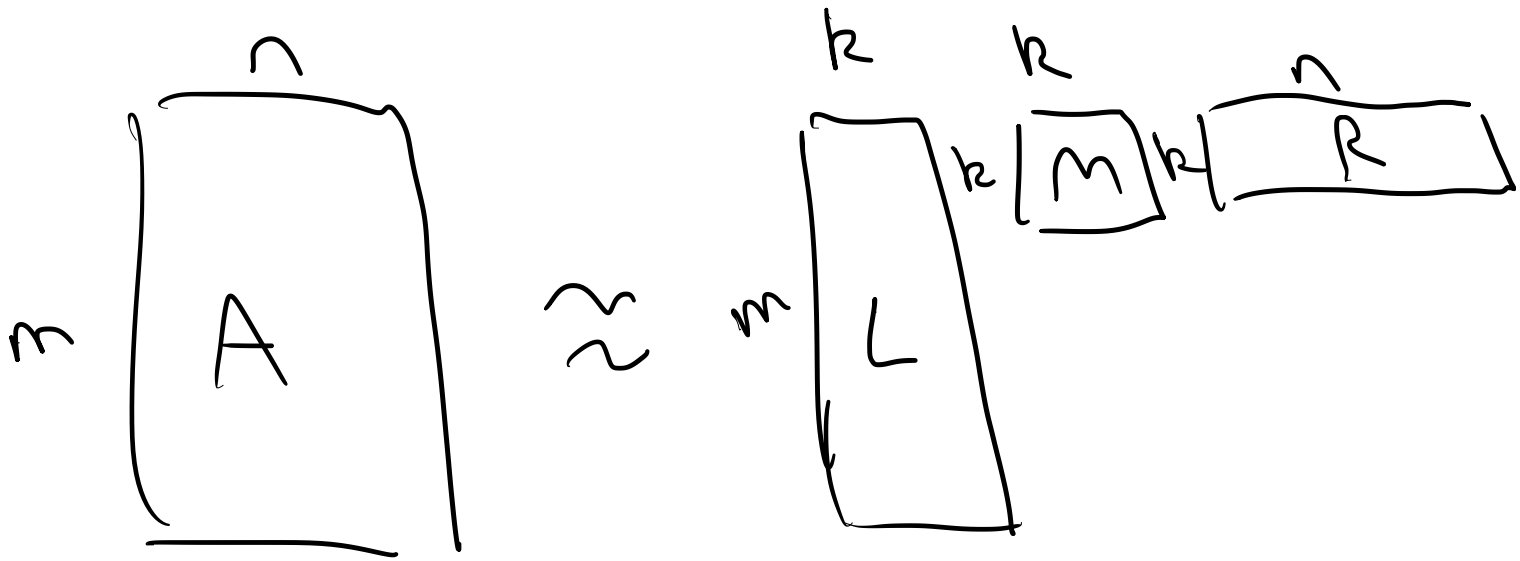
LLS: (A, b) $a_i^T \xrightarrow{f} b_i$

$\hat{x} = (A^T A)^{-1} A^T b$ $\quad f(a_i^T) = a_i^T \hat{x}$

Not just function fitting

- look for relationships b/w data pts
- don't know all features, fill in
- clustering / outliers
- interpretable explanations for what
  - generates data
- denoise

Matrix methods



$$\min_{Z} \|A - Z\|_F^2 \quad \text{s.t. } \operatorname{rank}(Z) = k$$

$$Z = U_k \Sigma_k V_k^T$$

PCA next lecture

Example: k-means clustering

points $a_1^T, \ldots, a_m^T \in \mathbb{R}^n$

$C : \{1, \ldots, m\} \to \{1, \ldots, k\}$

$C_j = \{ i \mid C(i) = j \}$

$$\min_C \sum_{j=1}^{k} \sum_{i \in C_j} \|a_i - r_j\|_2^2 \qquad r_j = \frac{1}{|C_j|} \sum_{i \in C_j} a_i$$

Lloyd's alg:

1. Assign point i to C_j if r_j is the closest mean to point i
2. re-compute r_j

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix} \quad n \times d \qquad L \quad n \times k \qquad L_{ij} = \begin{cases} 1 & \text{if point } i \text{ in } C_j \\ 0 & \text{o/w} \end{cases}$$

$$R = \begin{bmatrix} r_1^T \\ \vdots \\ r_k^T \end{bmatrix} \quad k \times d \qquad A \approx LR$$

$$e_i^T A = a_i^T \approx e_i^T (LR) = (e_i^T L) R = r_j^T \quad j = C(i)$$

$[0 \cdots 0\ 1\ 0 \cdots 0]$

$$\|A - LR\|_F^2$$

Alternatin min

① Fix L

$R = \min_S \|A - LS\|_F^2$

R = means

② $L = \min_W \|A - WR\|_F^2$

s.t. $W1 = 1$

$W_{ij} \in \{0, 1\}$