University of Wrocław: Algorithms for Big Data (Spring'22) 13/04/2022

Lecture 7: Regression, Low Rank Approximation

Lecturer: *Przemysław Uznański*

1 Linear Regression

In a regression problem we have predictor variables $a_1, \ldots, a_d$ and a measured variable b. In linear regression, we assume there is a relation $b \approx \sum_i a_i x_i$ for some $x_1, \ldots, x_d \in \mathbb{R}$.

We assume we received n batches $(a_{i,1}, \ldots, a_{i,d}, b_i), i = 1..n$. In the least square method we minimize the cost function

$$\sum_i (a_{i,1}x_1 + \ldots a_{i,d}x_d - b_i)^2.$$

Formally:

Definition 1. *On the input we have $A = \mathbb{R}^{n\times d}$ and $b \in \mathbb{R}^n$. Least square linear regression asks for $x \in \mathbb{R}^d$ so that*

$$\|Ax - b\|_2$$

is minimized.

1.1 Exact solution

Assume $b = Ax' + b'$ where b' is orthogonal to colsp(A) (column space of A) and Ax' is projection of b onto colsp(A). Then (by Pythagorean theorem)

$$\|Ax - b\|_2^2 = \|A(x - x') - b'\|_2^2 = \|A(x - x')\|_2^2 + \|b'\|_2^2$$

which is minimized when $x = x'$. The condition of Ax' being projection is equivalent to

$$A^T(Ax' - b) = A^T b' = 0$$

so equivalently we have a following condition

$$A^T A x' = A^T b. \tag{1}$$

If $(A^T A)$ is invertible (its rank is d), we can simply compute

$$x' = (A^T A)^{-1} A^T b.$$

Definition 2. *Let $A = U\Sigma V^T$ be SVD of A. Let $\Sigma^\dagger$ be defined as diagonal matrix where $\Sigma^\dagger_{i,i} = \frac{1}{\Sigma_{i,i}}$ if $\Sigma_{i,i} \neq 0$ and 0 otherwise. We then call $A^\dagger = V\Sigma^\dagger U^T$ a* pseudoinverse *of A.*

Theorem 3. *$x' = A^\dagger b$ satisfies condition (1) and has minimal L_2 norm among all the solutions.*

Proof. First part:

$$A^TAx' = A^TAA^\dagger b = (V\Sigma^TU^T)(U\Sigma V^T)(V\Sigma^\dagger U^T)b = V\Sigma^T\Sigma\Sigma^\dagger U^Tb = V\Sigma^TU^Tb = A^Tb$$

(note, $\Sigma^T\Sigma\Sigma^\dagger = \Sigma^T$ even though $\Sigma\Sigma^\dagger \neq I$ generally).

Second part: any solution is of the form

$$x'' = A^\dagger b + z$$

where $A^TAz = 0$, or equivalently $V\Sigma^T\Sigma V^Tz = 0$ or $\Sigma^T\Sigma V^Tz = 0$ (since V is orthonormal) or $\Sigma V^Tz = 0$ (since $\mathrm{Ker}(\Sigma^T\Sigma) = \mathrm{Ker}(\Sigma)$) or $V^Tz \in \mathrm{Ker}(\Sigma)$ or $z \in V \cdot \mathrm{Ker}(\Sigma)$.

We have $A \dagger b = V\Sigma^\dagger U^Tb \in V \cdot \mathrm{Im}(\Sigma^\dagger) = V \cdot \mathrm{Im}(\Sigma)$.

Since $\mathrm{Ker}(\Sigma) \perp \mathrm{Im}(\Sigma^\dagger)$, and since V is orthonormal, $V \cdot \mathrm{Ker}(\Sigma) \perp V \cdot \mathrm{Im}(\Sigma^\dagger)$. So by the Pythagorean theorem,

$$\|x''\|_2^2 = \|A^\dagger b\|_2^2 + \|z\|_2^2 \geq \|A^\dagger b\|_2^2$$

which proves optimality. □

Downside: time to compute SVD is $\mathcal{O}(\min(n^2d, nd^2))$ which can be prohibitive.

1.2 Approximate solution

Instead of solving exact regression, we pick a projection $\Pi \in \mathbb{R}^{m\times n}$ and solve a problem of smaller dimensionality (m instead of n), where we have ΠA and Πb instead of A and b:

$$\text{minimize} \quad \|\Pi Ax - \Pi b\|_2$$

It is enough to use subspace embedding Π for space spanned on columns of A + single vector b. Thus we can pick oblivious subspace embedding for $m = \mathcal{O}(d/\varepsilon^2)$, and have

$$\forall_{x\in\mathbb{R}^d} \|Ax - b\|_2 \leq \|\Pi Ax - \Pi b\|_2 \leq (1+\varepsilon)\|Ax - b\|_2.$$

Thus minimizing projected problem provides $1+\varepsilon$ approximation to original regression problem.

Total computation time is $\mathcal{O}(mn + \min(m^2d, md^2)) = \mathcal{O}(nd/\varepsilon^2 + d^3/\varepsilon^2)$.

2 Low rank approximation

Consider input matrix $A \in \mathbb{R}^{n\times d}$. The goal of the low-rank approximation is the following: find B such that B has small rank and $B \approx A$.

Denote such $B = C \times D$, where $C \in \mathbb{R}^{n\times k}$ and $D \in \mathbb{R}^{k\times d}$. Motivation (assume k is small)

- B requires much less space to store: $nk + kd$ vs nd.
- matrix-vector multiplication involving B is much faster: $B \cdot v$ takes $\mathcal{O}(nk + kd)$ time vs $\mathcal{O}(nd)$ time of $A \cdot v$.
- matrix-matrix multiplication: for $X \in \mathbb{R}^{d\times m}$, $B \cdot X$ takes $\mathcal{O}(kdm + nkm)$, vs $\mathcal{O}(ndm)$ time of $A \cdot X$
- A might have low-rank natural structure but we measured it with noise. Then B is the denoising of A

2.1 Exact solution

We are looking at

$$\arg \min_{B:\mathrm{rank}(B)\leq k} \|A - B\|$$

and denote it as A_k, best rank-k approximation of A.

How to find such A_k? Following theorem holds for both $\|\cdot\|_F$ and $\|\cdot\|_2$ norms.

Theorem 4 (Eckart-Young theorem)**.** *Consider SVD of $A = U\Sigma V^T$. Let Σ_k be Σ where only k largest in absolute value singular values are preserved, and every other value is zeroed.*

$$A_k = U\Sigma_k V^T \tag{2}$$

Unfortunately the time is dominated by SVD computation $\mathcal{O}(\min(nd^2, n^2d))$.

2.2 Approximate solution - projection

Approximate low rank approximation: we are looking for rank-k A'_k such that:

$$\|A'_k - A\|_F = (1 \pm \varepsilon)\|A_k - A\|.$$

Obtaining good approximate solution is possible for this problem, using the same framework: we project our problem to smaller dimension and hope that solution in reduced dimension approximates good solution to original problem.

Specifically, we use projection matrix $S \in \mathbb{R}^{m\times n}$, for small m. So $m = \mathcal{O}(k/\varepsilon)$ (note, m is independent of dimensions of A, and depends on desired rank k).

First, we show that it is ok to limit ourselves to following:

Theorem 5.

$$\min_{Y:rank(Y)\leq k} \|YSA - A\|_F \leq (1+\varepsilon)\|A - A_k\|_F$$

that is it is enough to limit ourselves to rank-k matrices in a row-space of SA.

Proof. Consider the regression problem:

$$\min_X \|A_kX - A\|_F$$

We have[1]

$$\|SA_kX - SA\|_F = (1 \pm \varepsilon)\|A_kX - A\|_F.$$

Left side is minimized for $X = (SA_k)^\dagger SA$ (see exercises), while right side is always at least $\|A_k - A\|_F$ (since A_kX is rank at most k and A_k was the best approximation of A for rank-k), so

$$\|SA_k(SA_k)^\dagger SA - SA\|_F = (1 \pm \varepsilon)\|A_k - A\|_F$$

and since S is an affine embedding, we can skip S on the left side at the cost of an extra $1 \pm \varepsilon$ factor. Thus we see that it is enough to set $Y = A_k(SA_k)^\dagger$ to have proper guarantees, and such Y is of rank at most k. □

[1] we picked S dimension to have a property of *affine embedding*: an approximate norm preserving projection for matrices, proof of this fact is outside of scope of this lecture

We choose a second affine embedding $R \in \mathbb{R}^{n \times m}$, so that

$$\|YSAR - AR\|_F = (1 \pm \varepsilon)\|YSA - A\|_F.$$

And so we reduced our problem to a following form: find rank-k Y that minimizes

$$\|YSAR - AR\|_F$$

and output YSA (preferably in a factorized form).

We now observe that[2], by pythagorean theorem and properties of projections:

$$\|YSAR - A\|_F^2 = \|YSAR - A(SAR)^\dagger SAR\|_F^2 + \|A(SAR)^\dagger SAR - A\|_F^2$$

thus we need to find rank-k Y that minimizes

$$\|YSAR - A(SAR)^\dagger SAR\|_F$$

however if Y is rank $\leq k$, then so is $YSAR$, so it is enough to find rank-k Z that minimizes:

$$\|Z - A(SAR)^\dagger SAR\|_F$$

and (by the structure of optimization problem) have $Z = YSAR$ guaranteed. Now we can apply Eckart-Young theorem to $A(SAR)^\dagger SAR$. It requires SVD, but the matrix we run it on is of dimension $n \times m$ so the cost is $nm^2 = n\frac{k^2}{\varepsilon^2}$. Thus we obtain $Z = CD$ where $C \in \mathbb{R}^{n \times k}$ and $D \in \mathbb{R}^{k \times m}$, and we can output $Z(SAR)^\dagger SA = C(D(SAR)^\dagger SA)$ with factors being C and $D(SAR)^\dagger SA$. Relevant computations of SA, SAR and $(SAR)^\dagger$ are all tractable (in time $n\frac{k^2}{\varepsilon^2}$).

3 Sparse Fourier

Fourier transform: signal → frequencies.

Definition 6. *Assume $a = (a_0, \ldots, a_{n-1})$ is a signal. Let ω be n-th root of unity, that is $\omega = e^{\frac{2\pi}{n}i}$. Let F be such that $F_{ij} = \frac{1}{\sqrt{n}}\omega^{ij}$. Then $\hat{a} = Fa$ is a (Discrete) Fourier transform of a.*

DFT can be computed in $\mathcal{O}(n \log n)$ time. However, for some applications this time can already be prohibitive. Consider a following scenario (of signal compression):

- Take input signal a and compute $\hat{a}$.
- Let $\hat{a}_k$ be $\hat{a}$ with only k largest magnitude elements kept (rest is zeroed).
- Output $a_k = F^{-1}\hat{a}_k$.

If we consider complexity measure of Fourier support $fs(a) = \|\hat{a}\|_0$ that is number of non-zero Fourier coefficients, then actually there is

$$a_k = \arg \min_{x: fs(x) \leq k} \|a - x\|_2$$

(proof: exercise)

[2] to project b to subspace of Ax we need to set $x = A^\dagger b$, similarly to project to subspace xA we need to set $x = bA^\dagger$, and this generalizes to matrices, see exercise

If we assume that a comes from real-life scenarios (photos, audio recording), then it should have only few "strong" frequencies, rest is noise. Since a_k has much simpler representation (namely, $\hat{a}_k$ which takes $\mathcal{O}(k \log n)$ bits), this is a lossy compression scheme.

How do we compute $\hat{a}_k$ efficiently? (Assumption is that we have random access to a, otherwise just *reading* the input would dominate the computation time.)

Simpler question: can we recover $\hat{a}$ if we know that $fs(a) = \|\hat{a}\|_0 \leq k$ (so there are only k non-zero frequencies)?