

## Lecture 7: Regression, Low Rank Approximation

Lecturer: *Przemysław Uznański*

Scribe: -

## 1 Linear Regression

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

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

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

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 column space of  $A$  and  $Ax'$  is projection of  $b$  onto column space of  $A$ . 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  $\Sigma_{i,i}^\dagger = \frac{1}{\Sigma_{i,i}}$  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:

$$A^T A x' = A^T A A^\dagger b = (V \Sigma^T U^T)(U \Sigma V^T)(V \Sigma^\dagger U^T) b = V \Sigma^T \Sigma \Sigma^\dagger U^T b = V \Sigma^T U^T b = A^T b$$

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

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

Both  $\text{Ker}(\Sigma)$  and  $\text{Im}(\Sigma^\dagger)$  are subspaces of  $\mathbb{R}^d$  - and by closer investigation since  $\Sigma$  is diagonal, both depend on zero/non-zero elements in  $\Sigma_{i,i}$ . Namely,  $\text{Ker}(\Sigma) \perp \text{Im}(\Sigma^\dagger)$ , and since  $V$  is orthonormal,  $V \cdot \text{Ker}(\Sigma) \perp V \cdot \text{Im}(\Sigma^\dagger)$ . 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.  $\square$

Downside: time to compute SVD is  $\mathcal{O}(\min(n^2 d, n d^2))$  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  $n$ ):

$$\text{minimize } \|\Pi A x - \Pi b\|_2$$

It is enough to use subspace embedding  $\Pi$  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

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

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

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

## 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 structure + noise,  $B$  is denoising of  $A$

## 2.1 Exact solution

We are looking at

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

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.** *Consider SVD of  $A = U\Sigma V^T$ . Let  $\Sigma_k$  be  $\Sigma$  where only  $k$  largest in absolute value singular values are preserved, and every other value is zeroed.*

$$A_k = U\Sigma_k V^T \quad (2)$$

*Proof.* TODO □

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

## 2.2 Approximate solution

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  $m = \mathcal{O}(k/\varepsilon^2)$  (note,  $m$  is independent of dimensions of  $A$ , and depends on desired rank  $k$ ).

We mention following theorem

**Theorem 5** (Clarkson and Woodruff, 2013). *There is algorithm that outputs  $A'_k$  of rank  $k$  in a factorized form, satisfying  $\|A'_k - A\|_F \leq \|A_k - A\|_F \cdot (1 + \varepsilon)$ . The time to compute is  $\mathcal{O}(nnz(A) + (n + d)\text{poly}(k/\varepsilon))$*

We skip the proof due to lack of time/space.

## 3 Sparse Fourier

Fourier transform: signal  $\rightarrow$  frequencies.

**Definition 6.** *Assume  $a = (a_0, \dots, a_{n-1})$  is a signal. Let  $\omega$  be  $n$ -th root of unity, that is  $\omega = e^{-\frac{2\pi}{n}}$ . 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)

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)?

### 3.1 Recovering sparse signal

Assume  $\log n$  is power of two.

Define  $p_{d,\ell}(x) = \sum_{i: i \bmod 2^\ell = d} \hat{a}_i x^i$ , and for a polynomial  $p(x)$  we write  $\|p\|^2 = \sum_i p_i^2$ .

First trick is how to estimate  $\|p_{d,\ell}\|_2^2$  with additive error. Given such blackbox, we can proceed:

1. Define  $S_\ell = \{i : \|p_{i,\ell}\|^2 > 0\}$
2. Given  $S_\ell$ , compute  $S_{\ell+1}$ : for each  $d \in S_\ell$ , test for  $e \in \{d, d + 2^\ell\}$  whether  $\|p_{e,\ell+1}\|^2 > 0$  and if so, add  $e$  to  $S_{\ell+1}$ .
3.  $p_{i,\log n} = \hat{a}_i$

The idea is to start at level 0 and proceed. The size of each  $S_\ell$  is bounded by  $k$ , thus the total number of steps (2) done is  $\mathcal{O}(k \log n)$ .