CIL

WEEK 4, 12-13 MAR 2020

MATRIX APPROXIMATION & RECONSTRUCTION

Problem Setup

Given a matrix $A \in \mathbb{R}^{m \times n}$ with observed entries $\mathcal{I} \subseteq [m] \times [n]$

Goal: Fill in the unobserved entries.

What makes it possible?

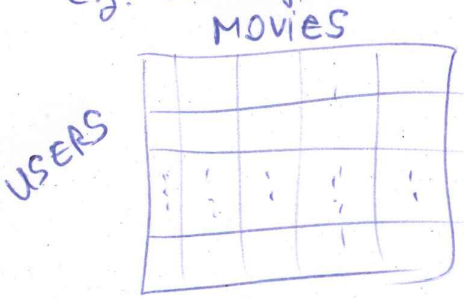
↳ Assume: there exist representations of the rows and the columns that require less than $m \times n$ parameters.
(in other words: "there is something to learn")

"collaborative filtering"

Examples

① recommender systems

e.g. the Netflix contest



$\mathcal{I} = \{(i,j) \mid \text{user } i \text{ rated movie } j\}$

Objectives

① $$\begin{aligned}&\text{minimize}_{X \in \mathbb{R}^{m\times n}} && \|A - X\|_{\mathcal{I}}^2 \\ &\text{subject to} && \text{rank}(X) \leq k\end{aligned}$$

(NP-) hard!
↓
"Is there a solution that achieves loss C?"

Notation: $\|X\|_{\mathcal{I}}^2 = \sum_{(i,j)\in\mathcal{I}} x_{ij}^2$

{Low-rank matrix recovery}

② $$\begin{aligned}&\text{minimize}_{X} && \text{rank}(X) \\ &\text{subject to} && \|A - X\|_{\mathcal{I}} = 0\end{aligned}$$

{Exact matrix recovery}

"simplest explanation"

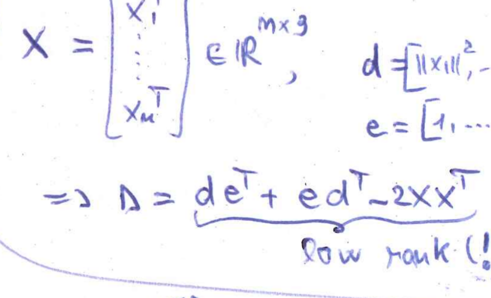
② sensor localization

→ n sensors: points $x_i \in \mathbb{R}^3$, $i = 1, \dots, n$

→ observe partial info about pairwise distances

$$D_{ij} = \|x_i - x_j\|^2 = \\ = \|x_i\|^2 + \|x_j\|^2 - 2x_i^T x_j$$

→ want to infer distances between all pairs of points.

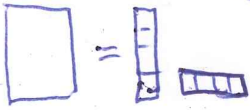


$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_n^T \end{bmatrix} \in \mathbb{R}^{n\times 3}$, $d = [\|x_1\|^2, \dots, \|x_n\|^2]^T$

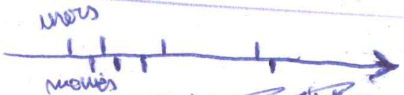
$e = [1, \dots, 1]^T$

$$\Rightarrow D = \underbrace{de^T + ed^T - 2XX^T}_{\text{low rank (!)}}$$

Q: Interpretation for the case $k=1$?



A: users → generosity
movies → quality



Statistical models?

Single-spike model

$Y = a^*(b^*)^T + W$, $W \sim \mathcal{N}(0,1)^{m\times n}$

① $\overset{\text{max}}{\Longrightarrow}$ likelihood

~~minimize $\|Y - a^*(b^*)^T$~~

$$\begin{aligned}&\text{minimize} && \|Y - ab^T\|_F^2 \\ &\text{subject to} && a, b \in \mathbb{R}^m\end{aligned}$$

Q: How many degrees of freedom in a rank-r matrix $\in \mathbb{R}^{m\times n}$?

$rm + (m-r)r$

Recall

<u>If</u> $\mathcal{I} = [m] \times [n]$ (full matrix observed)

<u>Then</u> problem ① has a closed-form solution:

<u>with</u> $A = U \Sigma V^T$ and $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$

$= \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i^T$ (for $k \le \text{rank}(A)$) (Eckart-Young Theorem)

<u>then</u> $\min_{\text{rank}(X)=k} \|A - X\|_F^2 = \|A - A_k\|_F^2 = \sum_{r=k+1}^{\text{rank}(A)} \sigma_r^2$

Back to problem ①

~~$\min\{\text{rank}(A), \text{rank}(B)\} \le$~~ $\text{rank}(A+B) \le \text{rank}(A) + \text{rank}(B)$

$$\min_{X:\,\text{rank}(X) \le k} \|A - X\|_{\mathcal{I}}^2$$

<u>Attempt I</u>: Optimise approximately with gradient descent + projections.
Can we project onto $\{X : \text{rank}(X) \le k\}$? (Yes, see above)
In practice, the steps are too chaotic. (Should sound familiar: PCA)

<u>Attempt II</u>: Manifold optimization: take steps "on the manifold".

<u>Attempt III</u>: Reparametrise:

$$\begin{array}{l} \text{minimize} \\ U \in \mathbb{R}^{k\times m} \\ V \in \mathbb{R}^{k\times m} \end{array} \|A - U^T V\|_{\mathcal{I}}^2$$

$U = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k\times m}$

$V = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_m \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{k\times m}$

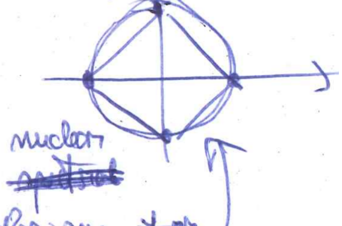
→ Regularised loss function:

$$L(U,V) = \sum_{(i,j)\in\mathcal{I}} (a_{ij} - u_i^T v_j)^2 + \lambda \sum_{i=1}^{m} \|u_i\|^2 + \lambda \sum_{j=1}^{n} \|v_j\|^2$$

Why regularise?

↳ Avoid overfitting. For instance, all user and items representations are constrained to a ball: (similar to ridge regression)

$\|U\|_F \le R$
$\|V\|_F \le R$



Could try ~~spectral~~ nuclear norm regularisers for sparsity!

→ "Solve" it optimistically with gradient descent? Fix one, backprop for the other?

↳ Notice that we have <u>closed form</u> solution when one argument is fixed.

$\boxed{\text{Solution problem ②}}$

① $L(U,V)$ is not convex!

Counter example: $m = n = 1$, $u \neq 0, v \neq 0$

(or, in higher dimensions,

$$U = \begin{bmatrix} u & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}; \quad V = \begin{bmatrix} v & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

$$L(u,v) = (a - uv)^2 + \lambda u^2 + \lambda v^2$$

~~Suffices to~~ Show $L'(u,v) = (a-uv)^2$ is not convex. (for simplicity)

$$\nabla^2 L'(u,v) = \begin{bmatrix} 2v^2 & 4uv - 2a \\ 4uv - 2a & 2u^2 \end{bmatrix}$$

First: check twice-differentiability.

$$|\nabla^2 L'(u,v)| = 4u^2v^2 - \overset{(4uv-2a)^2}{\cancel{16u^2v^2 = -12u^2v^2 < 0}} = 4\left[-3(uv)^2 + 4a(uv) - a^2\right] < 0 \text{ for } uv \notin \left[\frac{a}{3}, a\right]$$

Sylvester's criterion: a symmetric matrix is positive definite if and only if all leading principal minors have positive determinant.

In our case: $|2v^2| = 2v^2 > 0$ (as a 1×1 matrix) but $|\nabla^2 L'(u,v)| < 0$ (whole matrix)

$\Rightarrow L(u,v)$ is not convex.

② $g(U) = L(U,V)$ is convex (think: ridge regression)

③ Update rule for u_i.

$$\frac{\partial L(U,V)}{\partial u_i} = -2 \sum_{j:(i,j)\in I} (a_{ij} - u_i^T v_j)\, v_j + 2\lambda u_i \overset{!}{=} 0$$

$$\Rightarrow \sum_{j:(i,j)\in I} a_{ij} v_j = \sum_{j:(i,j)\in I} (u_i^T v_j)\, v_j + \lambda u_i =$$

$$= \sum v_j (v_j^T u_i) + \lambda u_i =$$

$$= \left(\sum v_j v_j^T + \lambda I_K\right) u_i$$

Can easily be used to embed new users and movies.

$$\Rightarrow u_i = \underbrace{\Big(\sum_{j:(i,j)\in I} v_j v_j^T + \lambda I_K\Big)^{-1}}_{m_i k^2 + \underset{\text{inverse}}{\boxed{k^3}}} \underbrace{\sum_{j:(i,j)\in I} a_{ij} v_j}_{m_i k} + \underset{③\ \text{product}}{\boxed{k^2}} \Rightarrow O(m_i k^2 + k^3)$$

④ Complexity:

⑤ The update rule looks like

$$u_i = \left(V_{\{j\}}^T V_{\{j\}}\right)^{-1} V_{\{j\}}^T a_{i\{j\}} \qquad V_{\{j\}} = \begin{bmatrix} — v_{j(1)} — \\ — v_{j(2)} — \\ \vdots \\ — v_{j(m_i)} — \end{bmatrix} \in \mathbb{R}^{m_i \times K}$$

if $\lambda = 0$, which is the solution to OLS.

Interpretation: • given the representation of the items $\{v_j\}$, compute the representation of the users independently, based on their "ratings"!

- the "ratings" $\{a_{ij}\}_j$ are orthogonally projected on the column space of $V_{\{j\}}$.
- the user representation u_i is given by the "indices" in the orthogonal projection.

(Recall $\hat{\beta}_{LS} = (X^TX)^{-1}X^TY$, $X\hat{\beta}(Y) = \mathrm{Id}_X Y$, where $\mathrm{Id}_X = X(X^TX)^{-1}X^T$ is the orthogonal projector on the column space of X.)

→ How to use the representations?

$$a_{pq} = u_p^T v_q \quad \text{for } (p,q) \notin \mathcal{I}$$

Bonus: we have representations of both "movies" and "users" in the same space, so we can compute similarities between movies, distances between users and movies etc.

Next time

$$\mathrm{rank}(X) = \dim(\mathrm{span}(X)) = \#\{\sigma_i > 0\}|$$

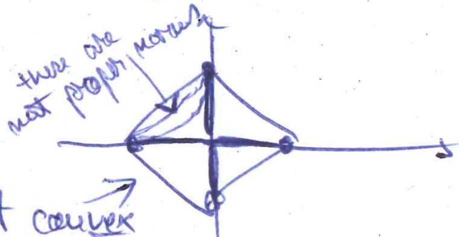
↑ should remind of the "0-norm"

$$\left| \begin{array}{l} \min\limits_X \ \mathrm{rank}(X) \\ \text{s.t. } \|A - X\|_{\mathcal{I}} = 0 \end{array} \right.$$

↳ gets relaxed to:

$$\left| \begin{array}{l} \min \ \|X\|_* \\ \text{s.t. } \|A - X\|_{\mathcal{I}} = 0 \end{array} \right. \quad \leftarrow \text{nuclear norm} \quad \|X\|_* = \sum_i \sigma_i(X)$$

↖ should remind of the L1-norm



Problem ①

$$\left|\begin{array}{l} \text{maximize } \langle x, Ax\rangle = x^T A x =: f(x) \\ \text{s.t. } \underbrace{\|x\|^2 - 1}_{g(x)} = 0 \end{array}\right.$$ with A - an $m\times m$ symmetric matrix

① $v_1 = \underset{\|x\|^2-1=0}{\arg\max}\ x^T A x$ and $\boxed{f(v_1) = \lambda_1}$

Prove $Av_1 = \lambda_1 v_1$.

Lagrange multiplier theory: $\exists \lambda \in \mathbb{R}$ such that $\nabla(f - \lambda g)(v_1) = 0$

We have: $\nabla g(x) = 2x$

$\nabla f(x) = 2Ax$

Thus: $2Av_1 = 2\lambda v_1 \Rightarrow Av_1 = \lambda v_1 \mid v_1^T(\cdot) \Rightarrow v_1^T A v_1 = \lambda v_1^T v_1 \Rightarrow$

$\Rightarrow f(v_1) = \lambda$ (we also know $\boxed{f(v_1) = \lambda_1}$)

Hence: $Av_1 = \lambda_1 v_1$.

② $v_2 = \underset{\left\{\begin{array}{l} g(x) := \|x\|^2 - 1 = 0 \\ h(x) := x^T v_1 = 0 \end{array}\right.}{\arg\max} f(x)$ and $\boxed{f(v_2) = \lambda_2}$

S^{m-2} (the $(m-2)$-dimensional unit sphere)

Prove: $Av_2 = \lambda_2 v_2$

As before: $\exists \lambda, \mu \in \mathbb{R}$ s.t. $\nabla(f - \lambda g - \mu h)(v_2) = 0$

$\nabla h(x) = v_1$

$\nabla g(x) = 2x$

$\nabla f(x) = 2Ax$

So we have: $2Av_2 = 2\lambda v_2 + \mu v_1 \mid v_1^T(\cdot) \Rightarrow$

$\Rightarrow 2 v_1^T A v_2 = 2\lambda \underbrace{v_1^T v_2}_{0} + \mu \underbrace{v_1^T v_1}_{1}$

$\overset{x^Ty = y^Tx}{\Leftrightarrow} 2 v_2^T A^T v_1 = \mu$

$\overset{A=A^T}{\Leftrightarrow} 2 v_2^T A v_1 = \mu$

$\Leftrightarrow 2 v_2^T v_1 = \mu$

$\Leftrightarrow 0 = \mu.$

$\Rightarrow Av_2 = \lambda v_2 \mid \cdot v_2^T(\cdot)$

$\Rightarrow f(v_2) = \lambda$ (we also have $f(v_2) = \lambda_2$)

$\Rightarrow Av_2 = \lambda_2 v_2$

(3) $v_3 = \arg\max_{\substack{g(x)=\|x\|^2-1=0 \\ h(x)=x^T v_1=0 \\ l(x)=x^T v_2=0}} f(x)$ and $f(v_3)=\lambda_3$

Prove: $\boxed{Av_3 = \lambda_3 v_3}$

Introduce a new Lagrange multiplier:

$$\nabla(f-\lambda g-\mu h-\vartheta l)(v_3)=0$$

which yields

$$2Av_3 = 2\lambda v_3 + \mu v_1 + \vartheta v_2$$

→ multiply by v_1 ⇒ $\mu=0$

→ multiply by v_2 ⇒ $\vartheta=0$

⇒ $Av_3 = \lambda v_3 \mid v_3^T(\cdot)$

⇒ $f(v_3)=\lambda \overset{\boxed{f(v_3)=\lambda_3}}{\Longrightarrow} Av_3=\lambda_3 v_3.$

NOTE: By construction: $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

(4) Iterate the above procedure:

$$\langle v_j, Av_m\rangle = \langle Av_j, v_m\rangle = \lambda_j \langle v_j, v_m\rangle = 0, \quad \forall j=1,\dots,m-1$$

So $Av_m = \lambda_m v_m$.

The vectors $\{v_1,\dots,v_m\}$ form an orthonormal basis

and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$.

(5) In PCA, this is applied on the ~~[illegible]~~ empirical covariance matrix $\frac{1}{m}\sum_{i=1}^{m} x_i x_i^T = \Sigma$