Prepoznavanje lica pomoću tenzorskog SVD-a

Marin Šako

***Abstract*— Tenzorski SVD je jako močan matematički alat, koji se usoješno koristi u mnogim problemima analize višedimenzionalnih podataka. U ovom radu ću ja pokazati kako se tenzorski SVD može iskoristiti za prepoznavanje lica.**

I. UVOD

U ovom seminaru radim algoritam za prepoznavanje lica pomoću tenzorskog SVD-a te uspoređujem njegove performanse, sa performansama konvolucijske neuralne mreže (CNN) koju sam implementirao i trenirao/testirao na istoj bazi podataka. Detaljima CNN se neću baviti u ovom radu. Tenzorski SVD se sastoji u tome da dekompoiramo N-dimenzionalni tenzor u manji N-dimenzionalni tenzor i N ortogonalnih matrica, analogno matričnom SVD-u. Time se razni utjecaji, razne "komponente", lica razdvoje u tih N matrica, te se na taj način mogu prepozanti osobe u različitim uvjetima, u raznim pozama, osvjeljenima i slično. Takav algoritam se ne može ostvariti pomoću PCA i matričnog SVD-a.
U drugom poglavlju ću govoriti općenito o tenzorskoj linearnoj algebri i nekim tenzorskim konceptima, te implementaciji HOSVD u nekom programskom jeziku.
U trećem poglavlju ću objasniti strukturu moje baze podataka, i strukturi baze podataka općenito.
U četvrtom poglavlju je detaljno opisan algoritam za klasifikaciju nepoznate slike.
U petom poglavlju su rezultati algoritma, a u šestom diskusija i objašnjenje dobivenih rezultata.

II. TENZORSKI SVD

Tenzorski SVD, još poznat i pod imenom HOSVD, je metoda dekompozicije tenzora takva da vrijedi: Ako je $A \in \mathbb{R}^{m\times n\times l}$ tenzor, tada za A vrijedi:

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \tag{1}$$

gdje je $S \in \mathbb{R}^{nl\times n\times l}$ tenzor istih dimezija kao A, a $U^{(1)} \in \mathbb{R}^{m\times m}$, $U^{(2)} \in \mathbb{R}^{n\times n}$, $U^{(3)} \in \mathbb{R}^{l\times l}$ su unitarne matrice.

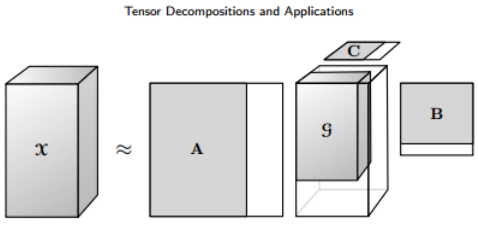

Fig. 4.2: Truncated Tucker decomposition of a three-way array

Za svaki član tenzora A vrijedi:

$$A_{ijk} = \sum_{p=1}^{l}\sum_{q=1}^{m}\sum_{s=1}^{n} u^{(1)}_{ip} u^{(2)}_{jk} u^{(3)}_{ks} S_{pqs} \tag{2}$$

A. *Aproksimacija tenzora*

Tenzor $A \in \mathbb{R}^{m\times n\times l}$ se može aproksimirati sumom matrica pomnoženih sa singularnim vektorom:

$$A = \sum_{i=1}^{n} A_i \times_3 u^{(3)}_i, \quad A_i = S(:,:,i) \times_1 U^{(1)} \times_2 U^{(2)}, \tag{3}$$

gdje su $u^{(3)}_i$ vektori stupci od $U^{(3)}$.

B. *Razmotavanje (unfold) tenzora*

Razmotavanje tenzora je operacija koja tenzor pretvara u matricu, tako da se vektori tenzora naslažu na određen način, ovisno o tome po kojoj dimeziji se razmotava.
Razmotavanje po dimenziji k učini k-tu dimenziju tenzora prvom dimezijom matrice, tj:

1. Vektori stupci od A postaju vektori stupci $A_{(1)}$.
2. Vektori retci od A postaju vektori stupci $A_{(2)}$.
3. Vektori po dimenziji 3 od A postaju vektori stupci $A_{(3)}$.

Gdje je $A_{(k)} = unfold(A, k)$

C. *Produkt tenzora i matrice*

Za tenzor $A \in \mathbb{R}^{m\times n\times l}$ i matricu $U \in \mathbb{R}^{k\times l}$ definiramo:
produkt po prvoj dimeziji

$$(A \times_1 U)_{ijk} = \sum_{z=1}^{m} u_{iz} a_{zjk} \tag{4}$$

produkt po drugoj dimeziji

$$(A \times_2 U)_{ijk} = \sum_{z=1}^{n} u_{jz} a_{izk} \tag{5}$$

i produkt po trećoj dimeziji

$$(A \times_3 U)_{ijk} = \sum_{z=1}^{l} u_{kz} a_{ijz} \tag{6}$$

Produkt po prvoj dimenziji je ekivalentan množenju svakog stupca tenzora A matricom U, slično produkt po drugoj dimenziji je ekvivalentan množenju svakog retka A matricom U.
Produkt po trećoj dimenziji je analogan.
Primjetite kako je produkt po drugoj dimenziji, matrice M sa v jedank množenju sa V^T s desna, neovisno o tome je li

V matrica ili vektor.

$$M \times_2 V = AV^T \tag{7}$$

Generalizacija na više dimenzije je jednostavna.

D. *Implementacija*

HOSVD za tenzor dimenzije 3 se može jednostavno implementirati na slijedeći način.

```
def HOSVD(A):
  U1,s1,v = svd(unfold(A,1))
  U2,s2,v = svd(unfold(A,2))
  U3,s3,v = svd(unfold(A,3))

  S = tmul(tmul(tmul(A,U1.T,1), U2.T,2), U3.T,3)
  return S, U1, U2, U3, s1, s2, s3
```

Gdje je svd(M) funkcija koja računa matrični SVD, unfold(M,k) "razmota" tenzor po dimenziji k i tmul(A,M,k) računa produkt matrice i tenzora po dimenziji k.

III. BAZA PODATAKA

Kao bazu podataka sam koristio slike od AT&T[1] Laboratories Cambridge.
Baza podataka se sastoji od crno-bijelih slika četrdeset osoba u deset različitih poza i izraza lica.
Prvo sliku treba pretvoriti u crno-bijelu ako već nije takva, može se raditi i sa slikama u boji ali je algoritam tada nešto složeniji. Ja se sa slikama u boji u ovom radu neću baviti.
Nakon toga se svaka slika treba pretvoriti u vektor, te se takve vektorizirane slike slažu kao stupci u matricu. Time ćemo dobiti matricu dimenzija $broj\ izraza\ lica \times duljina\ vektora$.
$duljina\ vektora$ iznosi $vertikalna\ rezolucija \times horizontalna\ rezolucja$.
Postupak ponovimo za sve osobe koje želimo dodati u bazu podataka.
Na kraju dobijemo tenzor trećeg ranka $T \in \mathbb{R}^{n_p \times n_r \times n_e}$, gdje su n_p, n_r i n_e broj osoba, duljina vektorizirane slike i broj poza/izraza lica.

IV. PREPOZNAVANJE LICA

Prije nego što krenemo u detaljniji opis algoritma, definirajmo prvo par pojmova koje ćemo koristiti u algoritmu.
Unutrašnji produkt tenzora $A \in \mathbb{R}^{m \times n \times l}$ i $B \in \mathbb{R}^{m \times n \times l}$ definiramo kao:

$$\langle A, B \rangle = \sum_{i,j,k} a_{ijk} b_{ijk} \tag{8}$$

Frobeniusovu norma je tenzora A,

$$\|A\|_F = \langle A, A \rangle^{\frac{1}{2}} = \left(\sum_{i,j,k} a_{ijk}^2 \right)^{\frac{1}{2}} \tag{9}$$

[1]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

A. *Dekompozicija tenzora* T

Neka imamo tenzor T definiran u poglavlju III.
Računanjem tenzorske dekompozicije dobijemo,

$$T = S \times_p P \times_r F \times_e E \tag{10}$$

Gdje su $S \in \mathbb{R}^{n_p \times n_p n_e \times e}$, tzv. "core" tenzor, $P \in \mathbb{R}^{n_p \times n_p}$ unitarna matrica čiji stupci sadrže informacije o kojoj se osobi radi, $F \in \mathbb{R}^{n_r \times n_p n_e}$, unitarna matrica koja sadrži informacije o samim pixelima originalnih slika i $E \in \mathbb{R}^{n_e \times n_e}$, unitarna matrica čiji stupci sadrže informacije o kojoj pozi/izrazu lica se radi.
n_r je duljina vektorizirane slike i $n_r \gg n_p n_e$.
Primjetite da koristim oznake $\times_p, \times_r, \times_e$ umjesto $\times_1, \times_2, \times_3$, to je zbog toga što nije unaprijed bitno po kojoj dimenziji treba množiti tenzor sa matricama. To je određeno načinom slaganja slika u tenzor T, npr ako matrice, (ili tenzore), koji sadrže osobe slažemo po prvoj dimenziji, onda će $\times_p$ zapravo biti $\times_1$.
Sada jednostavno možemo dobiti sliku određene osobe iz dekompozicije tako da fikisramo vektore retke u matricama E i P.

$$A(p_0, :, e_0) = D \times_e e_{e_0} \times_p p_{p_0}; \quad D = S \times_r F, \tag{11}$$

gdje je e_{e_0} neki vektor redak iz matrice E, a p_{p_0} neki vektor redak iz matrice P.

B. *Algoritam*

Prvo zapišimo HOSVD na slijedeći način:

$$A = C \times_p P, \quad C = S \times_i F \times_e E. \tag{12}$$

Za određeni izraz lica imamo:

$$A(:, :, e) = C(:, :, e) \times_p P. \tag{13}$$

Tenzori $A(:, :, e)$ i $C(:, :, e)$ su matrice i ozančimo ih sa A_e i C_e.
Sada imamo:

$$A_e = C_e P^T, \quad e = 1, 2, ..., n_e. \tag{14}$$

Ortogonalna matrica P, se pojavljuje u svih n_e relacija, tako da jednadžbu (14) možemo zapisati i u obliku:

$$a_p^{(e)} = C_e p_p \tag{15}$$

Jednadžbe (14) i (15) možemo interpretirati ovako:

Stupac p od A_e sadrži sliku osobe p sa izrazom lica e. Stupci C_e su vektori baze za izraz lica e, a u retku p matrice P, (h_p), se nalaze koordinate osobe p u bazi C_e, p_p sadrži koordinate slike osobe p u svim bazama izraza lica C_e.

Pretpostavimo da je $z \in \mathbb{R}^{n_r}$ slika nepoznate osobe koju želimo klasificirati.
Očito, ako je z slika osobe p u izrazu lica e, tada su koordinate z u toj bazi jednake koordinatama p_p.
Sliku z možemo klasificirati tako da računamo njene koordinate za sve baze C_e i provjeravamo za svaki izraz lica, jesu li koordinate z blizu koordiantama p_p, tj.redcima matrice P.

Koordinate z u bazi C_e možemo naći rješavajući problem najmanjih kvadrata:

$$\min_{\alpha_e} \|C_e\alpha_e - z\|_2. \tag{16}$$

Ovaj algoritam zahtijeva puno posla, za svaku sliku trebamo riješiti n_e problema najmanjih kvadrata.
Definiramo matricu $B_e \in \mathbb{R}^{n_p n_e \times n_p}$, kao $B_e = (S \times_e E)(:, :, e)$. Tada iz (12) slijedi:

$$C_e = FB_e. \tag{17}$$

Povećajmo matricu F, tako da postane kvadratna i ortogonalna,

$$\hat{F} = (FF^{\perp}), \tag{18}$$

tako da vrijedi,

$$\hat{F}^T\hat{F} = I. \tag{19}$$

Sada ubacimo $\hat{F}^T$ u (16), to možemo jer smo $\hat{F}$ definirali da bude ortogonalna, samim time je i $\hat{F}^T$ ortogonalna, a ortogonalne matrice ne mijenjaju normu. Sada imamo:

$$\begin{aligned} \|C_e\alpha_e - z\|_2^2 &= \|\hat{F}^T(FB_e\alpha_e - z)\|_2^2 = \\ &= \left\| \begin{pmatrix} B_e\alpha_e - F^Tz \\ -(F)^{\perp}z \end{pmatrix} \right\|_2^2 = \\ &= \|B_e\alpha_e - F^Tz\|_2^2 + \|(F^{\perp})^Tz|_2^2. \end{aligned} \tag{20}$$

Drugi član očito ne ovisi o osbama, niti izrazima lica, tako da ga možemo u potpunosti zanemariti za potrebe klasifikacije. Sada se algoritam svodi na rješavanje n_e problema najmanjih kvadrata,

$$\min_{\alpha_e} \|B_e\alpha_e - F^Tz\|_2. \tag{21}$$

Matrica B_e ima dimezije $n_e n_p \times n_p$, za razliku od matrice C_e koja ima dimezije $n_r \times n_p$, sjetimo se $n_r \gg n_e n_p$, tako da (21) zahtjeva puno manje posla za riješiti nego (16).
Nadalje, algoritam možemo još optimizirati tako da prvo izračunamo QR dekompoziciju za svaku matricu B_e. Tada se algoritam konačno svodi na rješavanje

$$R_e\alpha_e = Q_e^TF^Tz, \quad \forall e \in n_e. \tag{22}$$

Rezimirajmo konačni algoritam.

- Izračunati HOSVD $A = S \times_r F \times_e E \times_p P$
- Izračunati $B = S \times_e E$
- Izračunati QR dekompoziciju $\forall B_e \in B$, $B_e = Q_eR_e$
- Izračunati $\hat{z} = F^Tz$, za testnu sliku z
- $\forall e \in \{1, ..., n_e\}$
  - Riješiti $R_e\alpha_e = Q_e^T\hat{z}$
  - $\forall p \in \{1, ..., n_p\}$
    - Ako $\|\alpha_e - h_p\|_2 < tol$; sliku z klasificiramo kao sliku osobe p

V. REZULTATI

Točnost algoritma varira između 87% i 95% ovisno o omjeru testnog seta podataka i seta podataka na kojem se računao HOSVD.
Za podjelu 30% - 70% točnost je 91%. Vrijeme računanja HOSVD-a je nešto ispod dvije sekunde, a vrijeme potrebno za testiranje je zanemarivo.
Ja sam imao prilično oskudnu bazu podataka, zbog toga su vremena izvršavanja tako kratka, a i točnost bi se zasigurno povećala sa povećanjem baze podataka.
Točnost konvolucijske neuralne mreže sa dva konvolucijska sloja, jedan sa 32 i drugi sa 64 filtera, te sa tri *fully connected hidden* sloja, sa 200, te 100 neurona, koristeći "relu" aktivacijsku funkciju, na istoj bazi podataka sa istom 30% - 70% podjelom, je 95%, nakon 15 epoha treniranja. Vrijeme treniranja je oko 88 sekundi.

VI. ZAKLJUČAK

U rezultatima vidimo da je HOSVD nešto lošiji od neuralne mreže po točnosti, ali je zato mnogo brži.
To je očekivano, jer neuralnu mrežu se može gotovo uvijek istrenirati da dobijemo željene rezultate, povećavajući kompleksnost mreže uvođenjem dodatnih slojeva, ako nam vrijeme potrebno za treniranje nije bitno, što nije realistična pretpostavka, pogotovo jer vrijeme treniranja jako brzo raste s kompleksnošću neuralne mreže.
Ovo je ujedno i nedostatak neuralnih mreža u odnosu na HOSVD, jako su komputacijski zahtjevne, puno zahtjevnije od HOSVD-a. Za svaki neuron u svakom sloju za svaku sliku treba numerički računati gradijenete i taj cijeli psotupak se ponavlja N puta, gdje je N broj epoha. Ovaj postupak je jako zahtjevan. Neuralna mreža također zahtjeva puno više memorije iz istog razloga.
Još jedan nedostatak neuralne mreže je to što performanse jako ovise o parametrima koje programer mora postaviti, i još da stvar bude gora takvih parametara ima 5-10, uz problematiku odabira optimalne arhitekture mreže, što također drastično utjeće na performanse mreže, dok je za HOSVD potrebno postaviti samo jedan parametar i nema odabira konstrukcije modela, jer je model uvijek isti.
Prednost nuralne mreže je veća fleksibilnost i prilagodljivost, dok se za HOSVD gotovo da nema što prilagoditi niti optimizirati.

REFERENCES

[1] Lars Elden; Matrix Methods in Data Mining and Pattern Recognition; *Society for Industrial and Applied Mathematics*, 2007, Philadelphia, USA

[2] Debals O., Bousse M., Vervliet N. and De Lathauwer L. ;Numerical optimization algorithms for tensor-based face recognition; http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[3] Samaria F., Harter A.; Parameterisation of a Stochastic Model for Human Face Identification; *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*; Dec 1994. Sarasota FL, USA

[4] Hao N., Kilmer M. E., Braman K. and Hoover R. C.; Facial Recognition Using Tensor-Tensor Decompositions; *Society for Industrial and Applied Mathematics*, Feb 2013; DOI: 10.1137/110842570

[5] Vasilescu M.A.O., Terzopoulos D.; Multilinear Projection for Appearance-Based Recognition in the Tensor Framework, *Proc. Eleventh IEEE International Conf. on Computer Vision (ICCV'07)*, Rio de Janeiro, Brazil, October, 2007, 1–8.