PREPOZNAVANJE RUKOM PISANIH ZNAMENAKA POMOĆU HOSVD

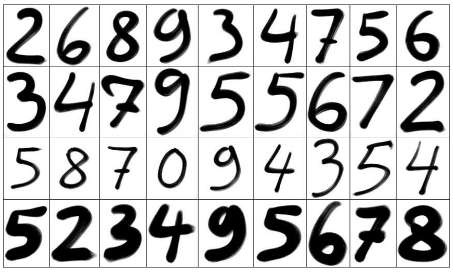
Vedran Ciganović, Eduard Kalčićek,
Mihael Končić & Valentino Novak

Svibanj 2018.

SADRŽAJ

UVOD

Raspoznavanje rukom pisanih znamenaka $(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)$ čest je i standardan problem u području prepoznavanja obrazaca (engl. *pattern recognition*). Dodjeljivanje nepoznatog objekta jednoj od 10 unaprijed definiranih klasa vrlo je težak problem za riješiti. Pogledajmo slijedeću sliku:



Slika 1: Rukom pisane znamenke

Ljudi bez većeg truda raspoznaju različite znamenke. Jednostavnost prepoznavanja je zapravo vrlo varljiva. Mozak podsvjesno obrađuje golemu količinu podataka, te na temelju "iskustva" zaključuje o kojoj je znamenci riječ. Kada bismo algoritamski htjeli riješiti dani problem, vidimo da se susrećemo s mnoštvom netrivijalnih problema, kao što su npr. prepoznavanje znamenaka različitih veličina i oblika, varijacije u rukopisu …

U ovom seminaru implementirat ćemo algoritam koji se bazira na HOSVD dekompoziciji tenzora. Algoritam koristi HOSVD dekompoziciju kako bi odredili skup matrica koje razapinju dominantni potprostor za svaku klasu znamenaka. Tada pomoću danih matrica baze pokušavamo opisati nepoznate znamenke. Algoritam bi trebao prepoznavati znamenke s greškom manjom od 6%.

Koristit ćemo rukom pisane znamenke iz baze podataka Američke pošte kako bi testirali algoritam. Znamenke su reprezentirane pomoću 16×16 crno-bijelih slika skeniranih s kuverta.

TEORIJSKA PODLOGA

Neformalno govoreći, tenzor reda N je matematički objekt sa N indeksa odnosno stupnja slobode. Vektori i matrice možemo klasificirati kao tenzori prvog i drugog reda, respektivno. U ovom seminaru proučavati ćemo slučaj kada je $N = 3$.

Neka je $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$, gdje su $I, J, K \in \mathbb{N}$. Vektorski prostor $\mathbb{R}^{I \times J \times K}$ je dimenzije IJK. Analogno možemo generalizirati definiciju na N dimenzija.

Definirati ćemo mod-1 niti $3-$tenzora $\mathcal{A}$ kao vektore stupce $\mathcal{A}(:, j, k)$. Analogno definiramo ostale niti. Slično definiramo rez tenzora $\mathcal{A}$ kao pod-tenzore (matrice) tako što fiksiramo indeks u jednom modu, na primjer $\mathcal{A}(:,:,k)$.

Na prostoru $\mathbb{R}^{I\times J\times K}$ definiramo skalarni produkt $\langle \mathcal{A}, \mathcal{B} \rangle$ tenzora $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I\times J\times K}$

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} a_{ijk} b_{ijk}.$$

Kažemo da su tenzori $\mathcal{A}$ i $\mathcal{B}$ ortogonalni ako je njihov skalarni produkt jednak nula

$$\langle \mathcal{A}, \mathcal{B} \rangle = 0.$$

Norma tenzora $\mathcal{A}$ definirana je kao

$$\|A\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}.$$

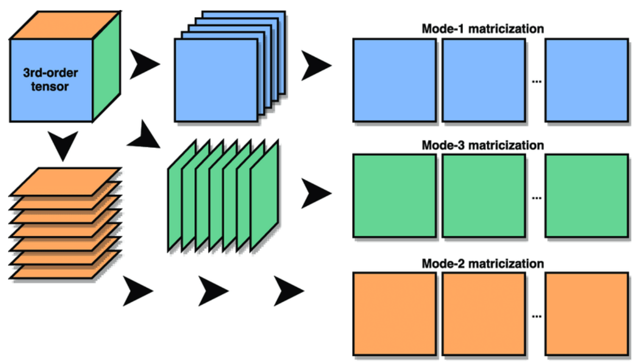
Ponekad je poželjno tenzor transformirati u matricu. Mod$-n$ transformacija tenzora $\mathcal{K}$ je operacija koja mod$-n$ vlakna tenzora $\mathcal{K}$ slaže u stupce matrice koju ćemo označiti $\mathbf{K}_{(n)}$. Koristit ćemo konvenciju iz članka [3]. Transformacija 3$-$tenzora $\mathcal{A} \in \mathbb{R}^{I\times J\times K}$ tada je jedinstveno određena izborom pozicije u matrici na koju ćemo staviti dani element tenzora $\mathcal{A}$

$$\mathbf{A}_{(1)} : a_{ijk} = a^{(1)}_{iv}, \quad v = j + (k-1)K, \quad \mathbf{A}_{(1)} \in \mathbb{R}^{I\times JK},$$

$$\mathbf{A}_{(2)} : a_{ijk} = a^{(2)}_{jv}, \quad v = k + (i-1)I, \quad \mathbf{A}_{(1)} \in \mathbb{R}^{J\times IK},$$

$$\mathbf{A}_{(3)} : a_{ijk} = a^{(3)}_{kv}, \quad v = i + (j-1)J, \quad \mathbf{A}_{(1)} \in \mathbb{R}^{K\times IJ}.$$

Možemo primijetiti da su stupci matrice $\mathbf{A}_{(n)}$ mod$-n$ niti tenzora $\mathcal{A}$.



Slika 2: Transformacija tenzora u matrice

Ne postoji jedinstven način generaliziranja koncepta ranga matrice na rang tenzora. Jedna mogućnost jest definirati $n-$rang tenzora $\mathcal{A}$ kao dimenziju potprostora razapetog $n-$mod vektorima, odnosno:

$$\mathrm{rang}_n(\mathcal{A}) = \mathrm{rang}(\mathbf{A}_{(n)}),$$

gdje je $\mathbf{A}_{(n)}$ matrica dobivena $n-$mod transformacijom tenzora $\mathcal{A}$, te "rang" bez indeksa označava rang matrice. Jasno je da različiti $n-$rangovi tenzora nisu jednaki kao što je to slučaj kod matrica.

Definicija 1 (Množenje tenzora i matrice u modu$-n$)**.** Neka je $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ te $F \in \mathbb{R}^{J_n \times I_n}$. Množenje tenzora $\mathcal{A}$ i matrice F u modu$-n$, u oznaci $\mathcal{A} \times_n F$, definiramo kao

$$(\mathcal{A} \times_n F)(i_1, \ldots, i_{n-1}, j_n, j_{n+1}, \ldots, i_N) = \sum_{i_n=1}^{I_n} \mathcal{A}(i_1, \ldots, i_N) F(j_n, i_n).$$

Sljedeći teorem govori o generalizaciji SVD nad tenzorima:

Teorem 1 (HOSVD - *Higher Order* SVD)**.** *Tenzor* $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ *možemo zapisati kao produkt*

$$\mathcal{A} = \mathcal{S} \times_1 U \times_2 V \times_3 W,$$

gdje su $U \in \mathbb{R}^{I \times I}, V \in \mathbb{R}^{J \times J}, W \in \mathbb{R}^{K \times K}$ *ortogonalne matrice, a* $\mathcal{S}$ *tenzor istih dimenzija kao i* $\mathcal{A}$ *koji ima svojstva:*

1. *Potpuna ortogonalnost: Svaka dva odsječka istog tipa međusobno su okomiti:*
$$\langle \mathcal{S}(i,:,:),\ \mathcal{S}(j,:,:) \rangle = 0, \quad i \neq j,$$
$$\langle \mathcal{S}(:,i,:),\ \mathcal{S}(:,j,:) \rangle = 0, \quad i \neq j,$$
$$\langle \mathcal{S}(:,:,i),\ \mathcal{S}(:,:,j) \rangle = 0, \quad i \neq j,$$

2. *Singularne vrijednosti u modu*-1 *su uređene:*
$$\|\mathcal{S}(1,:,:)\| \geqslant \|\mathcal{S}(2,:,:)\| \geqslant \cdots \geqslant 0.$$

PRVI ALGORITAM

HOSVD od tenzora $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ možemo računati promatrajući lijeve ortogonalne matrice U, V i W, koje su lijeve singularne matrice od A_n:

1. Računamo SVD:
$$A_{(1)} = U S^{(1)} (V^{(1)})^T,$$
$$A_{(2)} = V S^{(2)} (V^{(2)})^T,$$
$$A_{(3)} = W S^{(3)} (V^{(3)})^T.$$
2. Računamo tenzor $\mathcal{S}$: $\mathcal{S} = \mathcal{A} \times_1 U^T \times_2 V^T \times_3 W^T$.

Proizvoljna matrica F može se zapisati kao suma matrica ranga jedan:

$$F = U \Sigma V^T = \sum_{i=1}^{n} \sigma_i u_i v_i^T.$$

S tenzorima trećeg reda to izgleda ovako:

$$\mathcal{A} = \sum_{\nu=1}^{K} A_\nu \times_3 w_\nu,$$

gdje je $A_\nu = \mathcal{S}(:,:,\nu) \times_1 U \times_2 V$. Zbog svojstva (2) iz teorema 1, svi A_ν su ortogonalni pa ih možemo interpretirati kao skup linearno nezavisnih matrica:

$$\begin{aligned}
\langle A_\nu, A_\mu \rangle &= \mathrm{tr}(A_\nu^T A_\mu) \\
&= \mathrm{tr}((U\mathcal{S}(:,:,\nu)V^T)^T (U\mathcal{S}(:,:,\mu)V^T)) \\
&= \mathrm{tr}(V\mathcal{S}(:,:,\nu)^T \mathcal{S}(:,:,\mu)V^T) \\
&= \langle \mathcal{S}(:,:,\nu),\ \mathcal{S}(:,:,\mu) \rangle \\
&= 0
\end{aligned} \tag{1}$$

Za algoritam ulazne podatke preuzeli smo s izvora [1]. Svaka je znamenka prikazana kao crno-bijela slika piksela, što predstavlja 16×16 matricu. Uzeli smo 64 slike za svaku znamenku. Znamenku možemo shvatiti kao vektor iz prostora $\mathbb{R}^{16\times16}$. Znamenke seta za učenje čine 10 "dobro" odvojenih klastera. U suprotnom, algoritam ne bi dobro radio. Dominantni vektori u svakom klasteru razapinju potprostor od $\mathbb{R}^{256}$. U algoritmu konstruiramo skup ortogonalnih matrica (vektora) za svaku klasu (tj. znamenku od 0 do 9) koji razapinje dominantni dio odgovarajućeg potprostora. Za svaku klasu uzeli smo 64 matrice. Tada određujemo koja od baza najbolje opisuje danu znamenku. Zapravo određujemo grešku aproksimacije za svaku od 10 baza.

Promotrimo problem minimizacije:

$$\min_{\alpha_\nu^\mu} \|D - \sum_{\nu=1}^{k} \alpha_\nu^\mu A_\nu^\mu\|, \tag{2}$$

gdje su α_ν^μ nepoznati skalari. To je zapravo problem najmanjih kvadrata kojeg je lako riješiti pošto su matrice A_ν^μ ortogonalne, za fiksni μ. Rješenje je dano s

$$\alpha_\nu^\mu = \langle D,\ A_\nu^\mu \rangle, \quad \nu = 1, 2, \ldots, k.$$

Tu formulu ubacujemo kao rješenje u jednadžbu 2 te korištenjem ortogonalnosti matrica dobivamo sljedeći izraz za kvadrat norme:

$$\begin{aligned}
R(\mu) &= \|D - \sum_{\nu=1}^{k} \alpha_\nu^\mu A_\nu^\mu\|^2 \\
&= \langle D - \sum_{\nu=1}^{k} \alpha_\nu^\mu A_\nu^\mu,\ D - \sum_{\nu=1}^{k} \alpha_\nu^\mu A_\nu^\mu \rangle \\
&= \langle D, D \rangle - \sum_{\nu=1}^{k} \langle D, A_\nu^\mu \rangle^2 \\
&= 1 - \sum_{\nu=1}^{k} \langle D, A_\nu^\mu \rangle^2
\end{aligned} \tag{3}$$

Matricu D pridružujemo klasi za koju je $R(\mu)$ najmanji.

Pseudokod

Algoritam: klasifikacija pomoću HOSVD

1. Faza učenja
   a) Spremanje slika istih znamenaka u tenzore.
   b) Računanje HOSVD za tenzore.
   c) Računanje i spremanje normaliziranih baznih matrica $(A_\nu^\mu)_{\nu=1}^k, \ \mu = 0, 1, \ldots, 9$.
2. Faza testiranja
   a) Normaliziranje nepoznate znamenke.
   b) Računanje $R(\mu), \mu = 0, 1, \ldots, 9$.
   c) Određivanje $\mu_{min} := \mathrm{argmin}_\mu R(\mu)$.
   d) Klasificiranje D kao μ_{min}.

Time dolazimo do algoritma pisanog u programskom jeziku MATLAB:

Kôd u MATLAB-u

```
[A]=tensor(azip, dzip);

D=imread('004.jpg'); %Primjer ulazne datoteke
D=double(D(:,:,1));
D=D/norm(D,'fro');

R=zeros(10,1);
Av=zeros(16,16,64,10);
H=zeros(16,16,64);
for i=1:10
   [S,U1,U2,U3] = HOSVD(A(:,:,:,i));
   [~,~,n]=size(S);
   H1=0;
   for j=1:n
      Av(:,:,j,i)=tmul(tmul(S(:,:,j),U1,1),U2,2);
      if (norm(Av(:,:,j,i),'fro')~=0)
         Av(:,:,j,i)=Av(:,:,j,i)/norm(Av(:,:,j,i),'fro');
      end
         H(:,:,j)=D(:,:).*Av(:,:,j,i);
      s=0;
      for k=1:16
         for m=1:16
            s=s+H(k,m,j);
         end
      end
      H1=H1+s*s;
   end
```

```
   R(i)=1-H1;
end
[x,ix]=min(R);
ix-1
```

```
function [A]=tensor(azip, dzip)
   A=zeros(16,16,64,10);
   for i=0:9
      itmp=find(dzip==i);
      [~,d]=size(itmp);
      for j=1:64
         a1=squeeze(azip(:,itmp(j)));
         a1=reshape(a1,16,16)';
         A(:,:,j,i+1)=a1;
      end
   end
end
```

Funkcija za HOSVD tenzora:

```
function [S,U1,U2,U3] = HOSVD(T)
   [U1,~,~] = svd(unfold(T,1));
   [U2,~,~] = svd(unfold(T,2));
   [U3,~,~] = svd(unfold(T,3));

   S = tmul(tmul(tmul(T,U1',1),U2',2),U3',3);
end
```

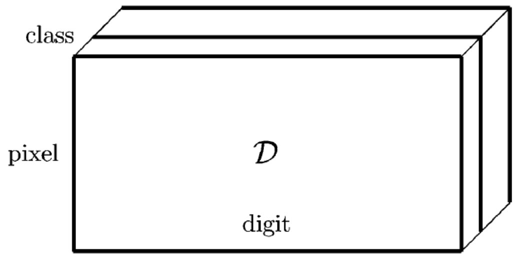
Funkcija `tmul()` za množenje tenzora i matrice u modu$-n$:

```
function [T1] = tmul(T, M, d)
   [l, m, n]=size(T);
   [j,k]=size(M);
   if d==1
      T1=fold(M*unfold(T, 1), 1, n);
   end
   if d==2
      T1=fold(M*unfold(T, 2), 2, n);
   end
   if d==3
      T1=fold(M*unfold(T, 3), 3, 16);
   end
end
```

DRUGI ALGORITAM

U drugom algoritmu test znamenke iz različitih klasa spojiti ćemo u jedan zajednički prostor, a nepoznatu znamenku projicirat ćemo samo jednom. Kada bismo imali po jedan prostor za svaku klasu, tada bi nepoznatu znamenku morali projicirati na svaki od tih prostora. Algoritam bi tada zahtjevao veći broj operacija i više memorije.

Prvo, dakle, trebamo kreirati tenzor koji sadrži sve znamenke iz seta za učenje. Te znamenke (matrice) preoblikujemo u vektore $\mathbb{R}^{256}$. Posložimo ih tako da sve znamenke u kriški pripadaju samo jednoj klasi (vidi sliku 3).



Slika 3: Tenzor sa svim znamenkama

Taj tenzor nazvat ćemo D. Sada računamo njegov HOSVD

$$\mathcal{D} = \mathcal{S} \times_1 U \times_2 V \times_3 W \approx \mathcal{F} \times_1 U_p \times_2 V_q,$$

gdje su $U_p = U(:, 1 : p)$, $V_q = V(:, 1 : q)$ i $\mathcal{F} = \mathcal{S}(1 : p, 1 : q, :) \times_3 W$. Takvim aproksimacijama reducirali smo reprezentaciju pojedinih znamenaka sa $\mathbb{R}^{256}$ na $\mathbb{R}^p$, a broj znamenaka u svakoj klasi na q. Reducirani tenzor $\mathcal{F}$ možemo izračunati kao

$$\mathcal{F} = \mathcal{D} \times_1 U_p^T \times_2 V_q^T.$$

Sada iz gornje relacije dobivamo

$$\mathcal{D}_p = \mathcal{D} \times_1 U_p^T = \mathcal{F} \times_2 V_q.$$

Svaki vektor stupac od $\mathcal{D}_p$ predstavlja neku znamenku, a različite kriške od $\mathcal{F}$ sadrže q vektora baze za svaku pojedinu klasu. Bazu za proizvoljnu znamenku μ označit ćemo sa $F^\mu = \mathcal{F}(:, :, \mu)$. Sada za svaku klasu računamo SVD

$$F^\mu = (B^\mu (B^\mu)^\perp)\Sigma^\mu (Q^\mu)^T, \quad \mu = 0, 1, \ldots, 9.$$

Stupci matrice B^μ razapinju dominantan potprostor od F^μ. Računamo $d_p = U_p^T d$ i rješavamo problem najmanjih kvadrata koji ima niže dimenzije od početnih

$$\min_{x^\mu} \|d_p - B^\mu x^\mu\|.$$

Budući da su vektori u matrici B^μ ortogonalni, rješenje je dano s

$$\hat{x}^\mu = (B^\mu)^T d_p.$$

Uvrštavanjem $\hat{x}^\mu$ u gornji problem dobivamo

$$R(\mu) = \|d_p - B^\mu (B^\mu)^T d_p\|.$$

Pseudokod

1. Faza učenja
   a) Spremanje vektoriziranih slika u tenzor $\mathcal{D}$.
   b) Računanje HOSVD tenzora $\mathcal{D}$.
   c) Računanje reduciranog tenzora $\mathcal{F}$.
   d) Računanje i spremanje matrice baze B^{μ} za svaku klasu.
2. Faza testiranja
   a) Računanje niže dimenzionalne reprezentacije $d_p = U_p^T d$ nepoznate znamenke.
   b) Računanje reziduala $R(\mu)$.
   c) Klasificiranje d kao μ_{min}.

LITERATURA

[1] Lars Eldén. Classification of handwritten digits, 2006.

[2] Berkant Savas i Lars Eldén. Handwritten digit classification using higher order singular value decomposition. 2005.

[3] Brett W. Bader i Tamara G. Kolda. Matlab tensor classes for fast algorithm prototyping, 2004.