Handwritten digit classification using higher order singular value decomposition

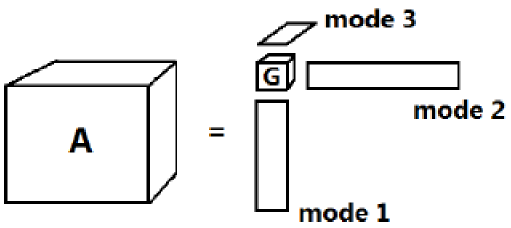
Course: Introduction to data mining

Faculty of Science – Department of Mathematics

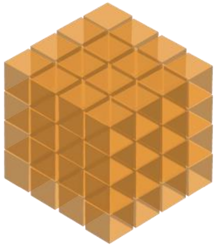
22.2.2021.

Tin Kranželić

Contents



Tensors



- Familiarity with basic definitions and operations with tensors is assumed (scalar product, norm, tensor unfolding, tensor-matrix multiplication, etc.)

- We will primarily work with third order tensors, so from now on, we fix $N = 3$

HOSVD

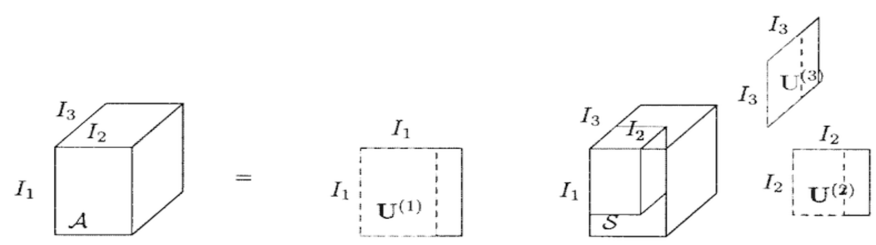
- Generalization of SVD ($F = U\Sigma V^T$) to tensors
- Let $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Then, A can be written as a product:

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

With the following properties:

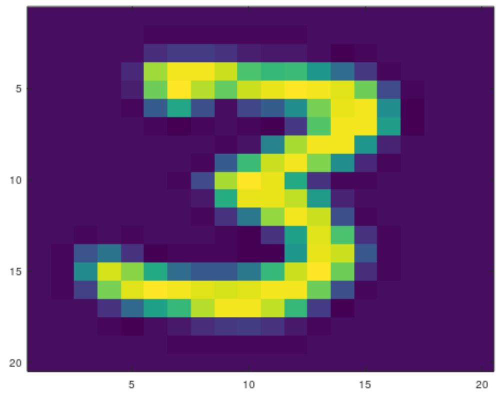
1) $U^{(1)} \in \mathbb{R}^{I_1 \times I_1}, U^{(2)} \in \mathbb{R}^{I_2 \times I_2}, U^{(3)} \in \mathbb{R}^{I_3 \times I_3}$ are orthogonal matrices
2) S is a tensor of the same dimensions as A and fulfils the following two properties:

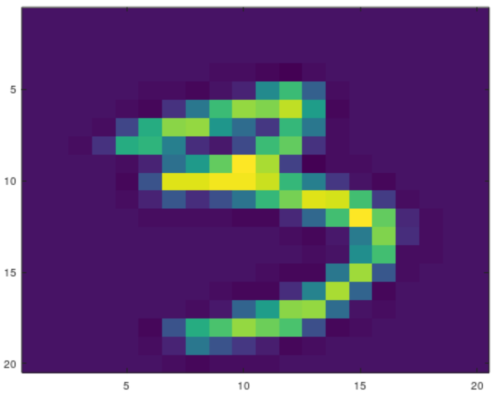
- Any two slices in the same mode are orthogonal
- Norms of slices in every mode are in descending order



Our task

- Dataset of handwritten digits ($20x20$ images)
- Example:



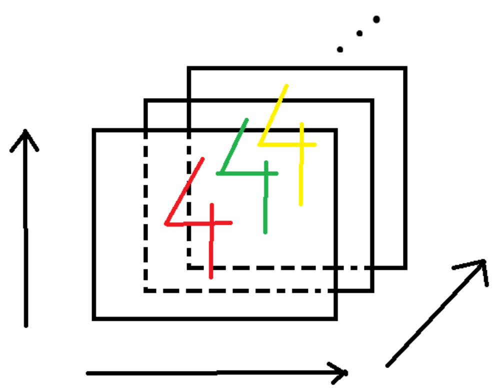


- Construct two algorithms based on HOSVD
- Classify unknown digit as one of the classes $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Algorithm 1

- Idea: for each of the classes calculate a set of basis matrices and determine which one describes a given unknown digit in the best way
- Key steps in training phase:

1. We sort digits from the training dataset into tensors with digits from the same class
2. Calculate HOSVD of all 10 tensors
3. Calculate normalized basis matrices



$$A^{four} = S^{four} \times_1 U^{four} \times_2 V^{four} \times_3 W^{four} \qquad (1)$$

1.

2.

Normalized basis matrices

- We can observe orthogonal basis matrices as the most dominant matrices which span the subspace for the given class, where the subspaces for each class will be well separated

- From the equation (1) follows $A^{four} = \sum_{v=1}^{K} B_v^{four} \times_3 w_v^{four}$, where K is the number of images with digit 4, and $B_v^{four} = S^{four}(:,:,v) \times_1 U^{four} \times_2 V^{four}$

- B_v^{four} - **basis matrix** for class 4
- Orthogonality: $\left\langle B_v^{four}, B_\mu^{four} \right\rangle = 0$, za $v \neq \mu$

- For each of the classes $\{0,1,2,\ldots,9\}$ we calculate a set of k most dominant orthogonal basis matrices (then we normalize them)

Classification

- Now, how do we classify a normalized unknown digit D?
- $(\forall\ \mu \in \{0,1,2,\ldots,9\})\ \min_{\alpha_v^\mu} \left\| D - \sum_{v=1}^{k} \alpha_v^\mu B_v^\mu \right\|$
- Least squares problem: $\hat{\alpha}_v^\mu = \langle D, B_v^\mu \rangle,\ v = 1, \ldots, k$
- Final solution: $\boxed{\arg\min_{\mu} R(\mu)}$, where: $R(\mu) = 1 - \sum_{v=1}^{k} \langle D, B_v^\mu \rangle^2$

Algorithm 2

- Idea: compress the training dataset and calculate basis matrices (vectors) for each class

- Key steps in training phase:

1. Sort the dataset into tensor T of dimensions 400 x (number of digits in the smallest class ≈ 400) x 10 $\longrightarrow$ digits are vector columns, i.e. Mode-1 fibers, every slice from T contains digits of the same class

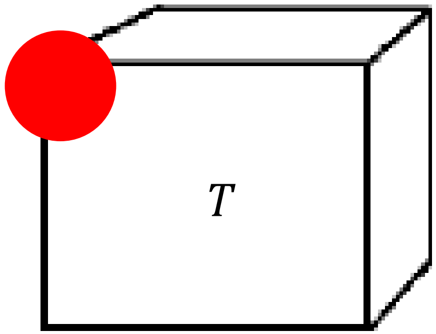
2. Calculate HOSVD of tensor T

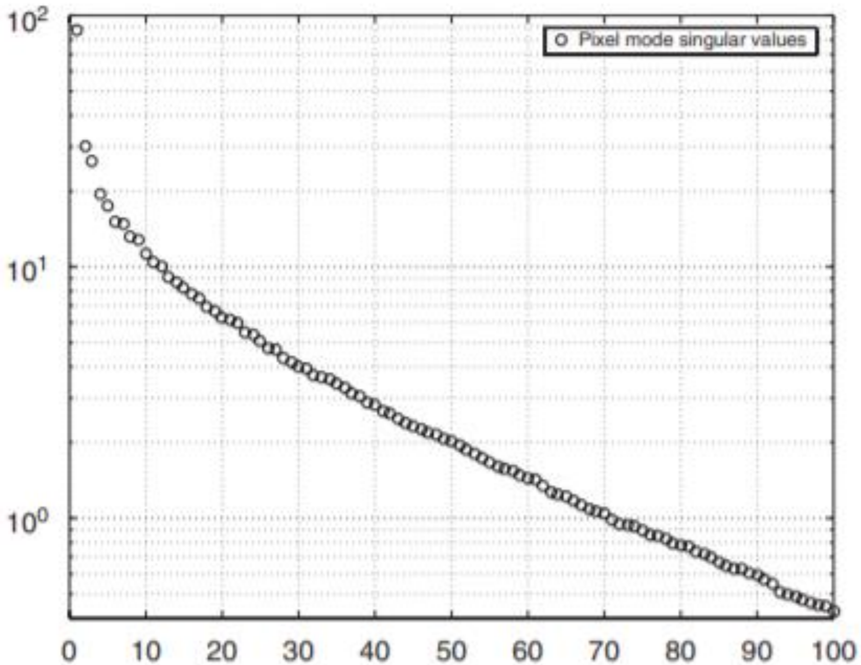
3. Calculate reduced representation of the training dataset, tensor F

4. Calculate basis matrices B^{μ}, $\forall\, \mu \in \{0,1,2,\dots,9\}$

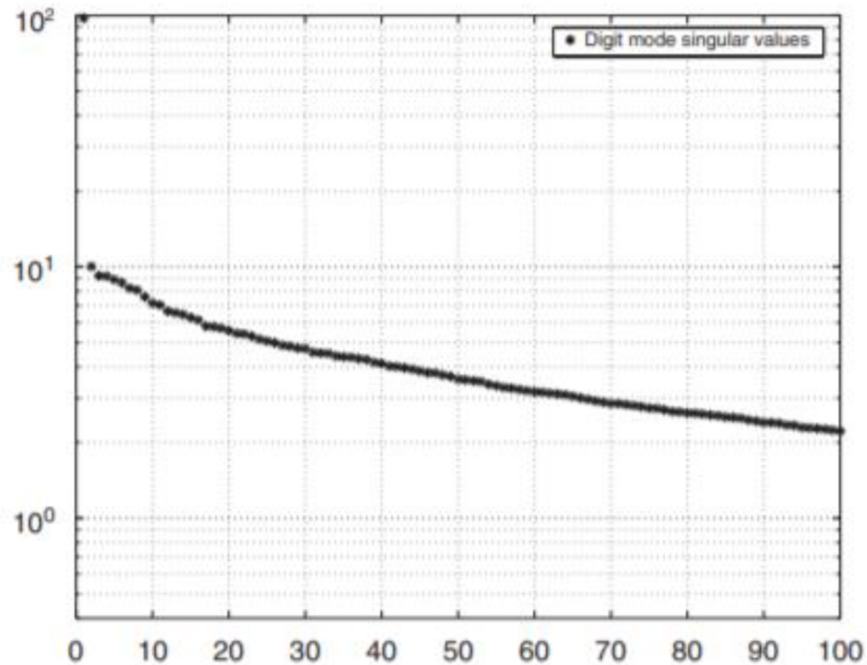
Tensor F

- $HOSVD \quad \rightarrow \quad T = S \times_1 U \times_2 V \times_3 W \quad \approx \quad F \times_1 U_p \times_2 V_q$, where $U_p = U(:,1:p)$, $V_q = V(:,1:q)$, $\quad F = S(1:p,1:q,:) \times_3 W$
- By doing this, we reduced the representation of the individual digits from $\mathbb{R}^{400}$ to $\mathbb{R}^{p}$ and the number of digits in each class to q
- $p,\ q \ll 400$, but how do we choose them?





Mode-1 singular values



Mode-2 singular values

Basis matrices B^{μ}

- $F^{\mu} := F(:,:,\mu)$

- SVD of F^{μ} $\Longrightarrow$ $F^{\mu} = \left[B^{\mu}(B^{\mu})^{\perp}\right] \Sigma^{\mu} (Q^{\mu})^{T}, \quad \mu = 0,1,\dots,9$

- We take k most significant left singular vectors

- $B^{\mu} \in \mathbb{R}^{p\times k}$, columns ortonormal

Classification

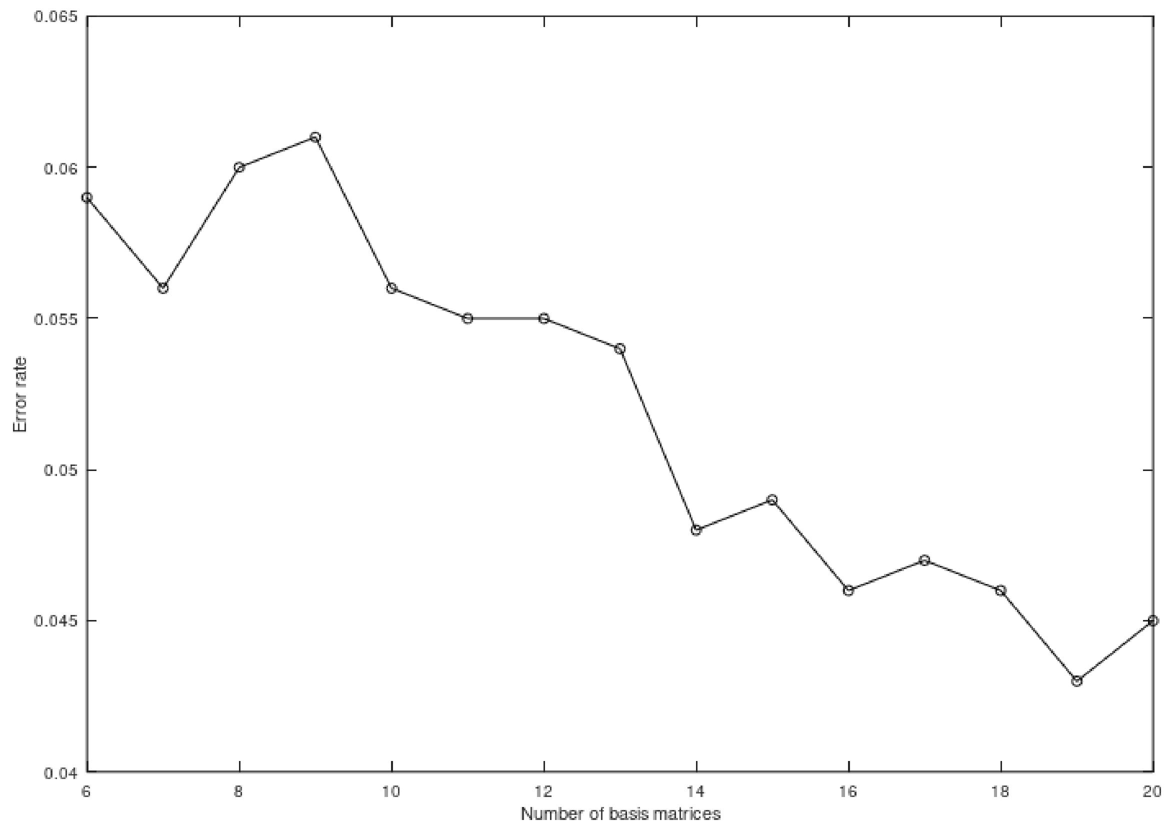
- Similarly to algorithm 1, we wonder how to classify unknown digit $d \in \mathbb{R}^{400}$?
- We calculate low-dimensional representation $d_p = U_p^T d$
- $\min_{x^\mu} \|d_p - B^\mu x^\mu\| \Rightarrow$ least squares problem: $\hat{x}^\mu = (B^\mu)^T d_p$
- Final solution: $\boxed{\arg\min_{\mu} R(\mu)}$, where $R(\mu) = \|d_p - B^\mu (B^\mu)^T d_p\|$

Results and discussion

- Description of the dataset we used: 500 digits in each class
- We have randomly shuffled them and split into train and test datasets
- We got the following distribution of digits:

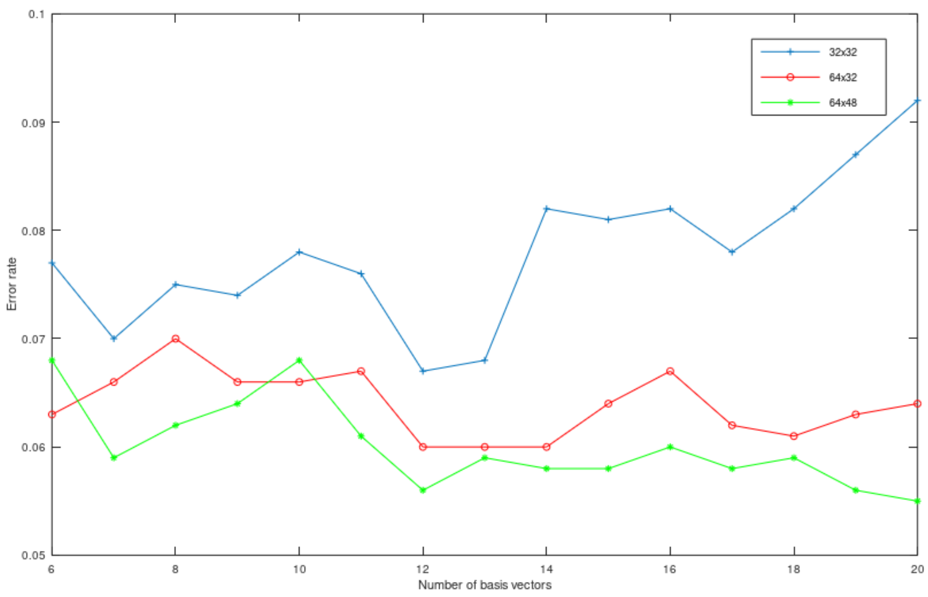
	0	1	2	3	4	5	6	7	8	9	Total
train	386	399	399	400	384	421	394	418	395	404	4000
test	114	101	101	100	116	79	106	82	105	96	1000

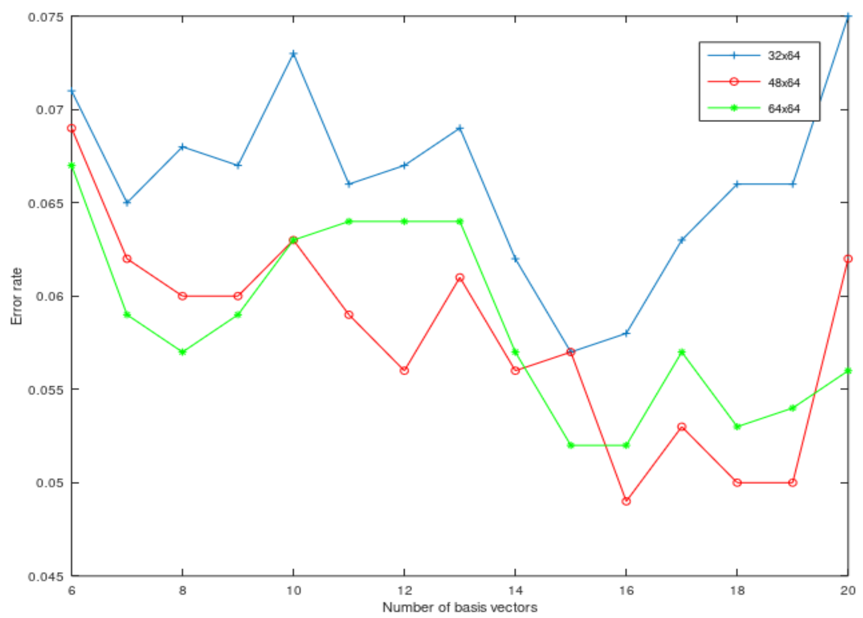
Performance of algorithm 1 for different choices of k



We can see that the highest precision was achieved for $k = 19$ and it was almost 96%

Performance of algorithm 2 for different choices of p, q and k

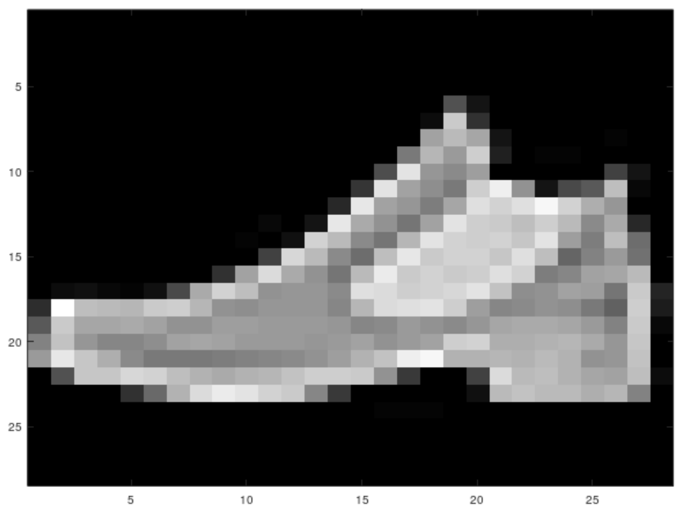




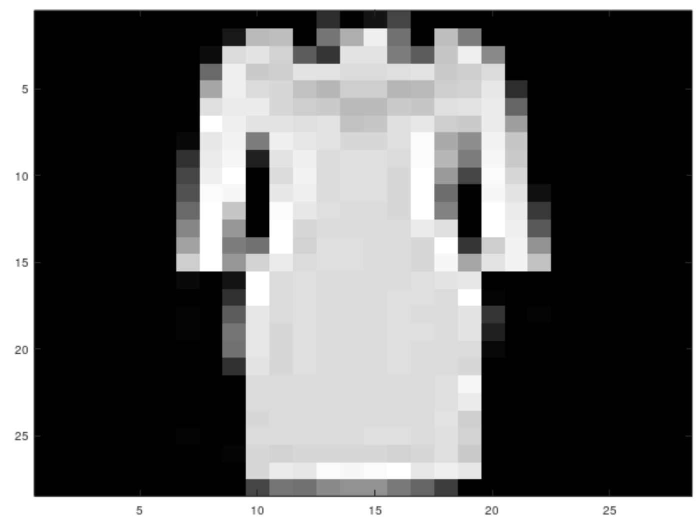
We can see that the highest precision achieved for algorithm 2 was around 95%

Another classification example

- Fashion items
- 10 classes - {"T-shirt/top"; "Trouser"; "Pullover"; "Dress"; "Coat"; "Sandal"; "Shirt"; "Sneaker"; "Bag"; "Ankle boot"}



Ankle boot



Dress

Algorithm 2 achieved precision of about 75%

Conclusion

- We saw how to efficiently solve the digit classification problem using HOSVD

- Algorithm 2, even after great compression of the dataset, achieves high precision in comparison with algorithm 1, while faster and more efficient

List of functions

- sem2.m – main program
- unfold() – returns the unfolded tensor in given mode
- fold() – returns the folded tensor of given matrix in given mode, opposite function of unfold
- tm_multiply() – returns the tensor which is the product of given tensor-matrix multiplication in given mode
- frobnorm() – return the frobenius norm of the given matrix
- getbasismat() – returns tensor which contains k basis matrices
- HOSVD() – calculates HOSVD of the given tensor
- samedigittensor() – returns tensor which contains digits from the same class
- digitclassify() – return classification of the given digit for algorithm 1
- digitclassify2() – return classification of the given digit for algorithm 2
- Item_classifier.m – code for classification of fashion items

References

- [1] Handwritten digit classification using higher order singular value decomposition Berkant Savas, Lars Eldén Department of Mathematics, Linköping University
- [2] Zlatko Drmač: Lectures from the Course: Introduction to data mining, Zagreb 2021.
- [3] Data for classification of fashion items was downloaded from: https://www.kaggle.com/zalando-research/fashionmnist