EE378B Inference, Estimation, and Information Processing

Non-negative Matrix Factorization

Andrea Montanari *Lecture 15-16 - Due on 3/15/2021*

Homework should be submitted via Gradescope, by Monday afternoon (unless Monday is a holiday): the code will be communicated by an announcement on Canvas.

For getting credit for the class, you are required to present solutions of some of these homeworks during the first 15 minutes of class starting on 1/20. Please, sign up for (at least) one slot, and be sure that your explanation lasts 15 minutes (or less). For these presentations, you are free to choose whatever format you prefer (slides, typed notes, handwriting, ...).

This week, the presentations will be:

- Monday: Questions (a), (b)
- Wednesday: Questions (c), (d).

This homework is about dimensionality reduction using non-negative matrix factorization. Part of this homework will use the MNIST test dataset in the file mnist_test.csv that you can find at the following url

http://web.stanford.edu/class/ee378b/homework/mnist_test.csv

The data consist of $n = 10,000$ images. An image is a 28×28 gray-level array. Each image is stored as a line of the file mnist_test.csv, in the format

label, pix(1,1), pix(1,2), pix(1,3), ...,

where label is a label corresponding to the digit represented by the image, and pix(s,t) is the pixel intensity at row s, column t (an integer in the range $\{0, 1, \dots, 255\}$).

Hereafter, we will denote the i-th image by the vector $\boldsymbol{x}_i \in \mathbb{R}^d$, $d = 28^2 = 784$.

We will compare two approaches:

Principal component analysis (PCA). Given data $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, this computes the empirical covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ via

$$\boldsymbol{\Sigma} \equiv \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}\,, \tag{1}$$

and let $\hat{\boldsymbol{U}}_r \in \mathbb{R}^{d\times r}$ be the matrix whose columns are given by the first r eigenvector of $\boldsymbol{\Sigma}$ (corresponding to the largest eigenvalues).

Non-negative matrix factorization (NMF). Represent the data by a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, whose i-th row is the vector $\boldsymbol{x}_i$. We suggest to rescale the rows so that $\boldsymbol{X}\mathbf{1} = \mathbf{1}$ and implement the original algorithm in (Seung, Lee, 1999) which attempts at maximizing the following cost over $\boldsymbol{W} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{H} \in \mathbb{R}^{r\times d}$

$$F(\boldsymbol{W}, \boldsymbol{H}) = \sum_{i=1}^{n}\sum_{j=1}^{d} \left\{ \boldsymbol{X}_{ij} \log(\boldsymbol{W}\boldsymbol{H})_{ij} - (\boldsymbol{W}\boldsymbol{H})_{ij} \right\}. \tag{2}$$

The algorithm proceeds according to the following iteration

$$\tilde{W}_{ij}^t = W_{ij}^t \sum_k \frac{X_{ik}}{(W^t H^t)_{ik}} H_{jk}^t \,, \tag{3}$$

$$W_{ij}^{t+1} = \frac{\tilde{W}_{ij}^t}{\sum_k \tilde{W}_{kj}^t} \,, \tag{4}$$

$$\tilde{H}_{ij}^{t+1} = H_{ij}^t \sum_k H_{ki}^{t+1} \frac{X_{kj}}{(W^{t+1} H^t)_{kj}} \,, \tag{5}$$

$$H_{ij}^{t+1} = \frac{\tilde{H}_{ij}^{t+1}}{\sum_k \tilde{H}_{ik}^{t+1}} \tag{6}$$

We denote by $\hat{\boldsymbol{H}}_r$ the output of this iteration. We leave it up to you to decide on a reasonable initialization and convergence criterion, as long as you describe your choices.

(a) We first consider the behavior of the two algorithms in a synthetic data model. For $d = 200$ generate $\boldsymbol{\theta} \in \{0,1\}^d$ with half of its entries equal to 1, and half equal to 0, and $\boldsymbol{w} \in \mathbb{R}^n$, with i.i.d. entries $w_i \sim \mathsf{Unif}([0,1])$. Then for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,d\}$, generate

$$X_{ij} \sim \sqrt{d} \cdot \text{Bernoulli}\Big(\frac{1}{\sqrt{d}} w_i \theta_j\Big) \,, \tag{7}$$

with $(X_{ij})_{i\le n, j\le d}$ conditionally independent given $\boldsymbol{\theta}, \boldsymbol{w}$. The vectors $\boldsymbol{x}_i$ are the rows of matrix $\boldsymbol{X} = (X_{ij})_{i\le n, j\le d}$.

Use PCA with $r = 1$ to get an estimate $\hat{\boldsymbol{U}}_1 = \hat{\boldsymbol{u}}_1 \in \mathbb{R}^d$. We evaluate this approach by estimating the similarity

$$Q_n = \mathbb{E}\Big\{ \frac{|\langle \hat{\boldsymbol{u}}_1, \boldsymbol{\theta}\rangle|}{\|\hat{\boldsymbol{u}}_1\|_2 \|\boldsymbol{\theta}\|_2} \Big\} \,. \tag{8}$$

Repeat this experiment for $n \in \{200, 400, 800, 1600, 3200\}$, and estimate Q_n at each value of n by averaging over 10 realization of the matrix $\boldsymbol{X}$.

(b) Repeat the same experiment as in point (a), but using NMF instead of PCA. Compare the results with the ones of PCA.

(c) We now switch to the MNIST data. In this case, as mentioned above, $d = 784 = 28^2$ and $n = 10,000$. Set $r = 6$, anf compute $\hat{\boldsymbol{U}}_r$. Plot the r principal components (columns of $\hat{\boldsymbol{U}}$) as 28×28 pixels images. (Scale them as to make them visible!)

(d) Repeat the analysis at the previous point using NMF. Plot the resulting archetypes $(\boldsymbol{h}_i)_{i\le r}$ (the rows of $\hat{\boldsymbol{H}}_r$) as 28×28 pixels images. (Scale them as to make them visible!) Compare the results with the ones of PCA.