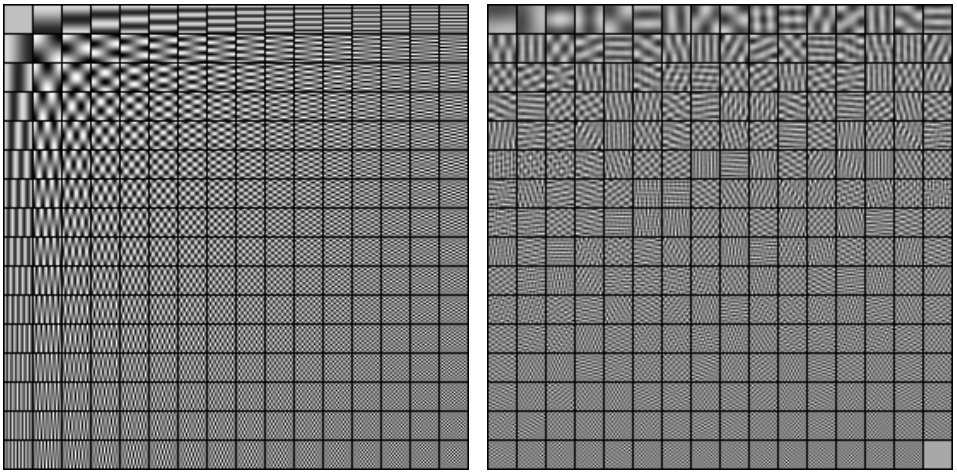
Multivariate Statistics – non-negative matrix factorisation and sparse dictionary learning

The PCA decomposition is by construction optimal solution to

$$\text{argmin}_{A\in\mathbb{R}^{n\times q}, H\in\mathbb{R}^{q\times p}} \|X - AH\|_2^2 \quad \text{under constraint } : HH^t = \mathbf{1}_{q\times q}. \tag{1}$$

1. Entries in A are often called **scores**. Each of the n rows of A can be thought of as containing the "activity-coefficients" of the basis vectors in the "dictionary" H.
2. Each of the q rows of the **"dictionary"** H contains a basis vector (which in the PCA solution is just the corresponding eigenvector $v_1, \ldots, v_q$ of X^tX) and this is sometimes called an atom.[1]

With PCA we get the best low-rank approximation of X, where best is measured in a Euclidean metric. PCA learns both the score matrix A and the "dictionary" H. Incidentally, if the PCA is trained on small patches of natural images, the PCA objective (1) leads to very similar dictionaries as a real-valued Fourier or DCT fixed basis, see Figure 1 from Mairal, Bach and Ponce (2014). Before adapting the objective in (1) to different applications, a quick comparison with fixed dictionaries.



(a) DCT Dictionary. (b) Principal components.

Figure 1: *The left contains in each panel from top left to bottom a row of H if the dictionary is a discrete cosine transform. The right shows the same for the PCA solution of H if trained on 400000 natural images. Now each row of H is a panel and corresponds to the eigenvector of the empirical covariance matrix of the pixel intensities across all these natural images. The two dictionaries H turn out to be quite similar.*

[1] We can in general also use 'basis' instead of 'dictionary' and 'basis vector' instead of 'atom'. However, the terms 'dictionary' and 'atom' are slightly more general. A dictionary can, for example, be overcomplete if it contains p' atoms in a p-dimensional space and $p' > p$.

Comparison with fixed basis/dictionary. A quick comparison with Fourier or wavelet analysis and similar harmonic analysis tools. In Fourier analysis (or its real-valued equivalent discrete cosine transform DCT) and wavelet analysis, the matrix H is fixed and known and is generally of the same dimension p as the original data (so $q = p$). Each row of H contains for DCT/Fourier basis a sine-wave or cosine-wave at a given frequency (for 1d-signals) or a wavelet (for images) [2] etc. We then search for the coefficients in the new basis given by the rows of H by solving

$$\hat{A} \;=\; \text{argmin}_{A\in\mathbb{R}^{n\times q}} \|X - AH\|_2^2. \tag{2}$$

Since H is usually of dimension $p \times p$ (and in particular the rows of H form an orthogonal basis of $\mathbb{R}^p$ for both DCT and wavelet transforms in that $H^tH = \mathbf{1}_{p\times p}$), we have that the objective is 0 for the solution since $X = \hat{A}H$ for the optimal solution. We can then write the i-th data sample (which can be an image, a time-series or gene expressions for one patient etc) as a linear superposition of the basis vectors (the rows of H)

$$X_{i\cdot} = \sum_{k=1}^{p} \hat{A}_{ik} H_{k\cdot}, i \in \{1, \ldots, n\}.$$

If q is large, perhaps even $q = p$ for a wavelet basis or or $q > p$ for an overcomplete dictionary, often just the largest coefficients of A are kept for compression and/or de-noising. Either hard- or soft-thresholding are popular, where the latter can be formulated as the solution to the convex ℓ_1-penalized optimisation problem:

$$\hat{A}^\lambda = \text{argmin}_{A\in\mathbb{R}^{n\times q}} \frac{1}{2}\|X - AH\|_2^2 + \lambda\|A\|_1. \tag{3}$$

It will be shown further below that the solution $\hat{A}^\lambda$ is for orthogonal dictionaries ($HH^t = \mathbf{1}_{q\times q}$) simply a soft-thresholded version of $\hat{A} = \hat{A}^0$ in that for all $1 \le i \le n$ and $1 \le k \le q$,

$$\hat{A}^\lambda_{ik} = \begin{cases} \hat{A}^0_{ik} - \lambda & \text{if } \hat{A}^0_{ik} > \lambda \\ 0 & \text{if } -\lambda \le \hat{A}^0_{ik} \le \lambda \\ \hat{A}^0_{ik} + \lambda & \text{if } \hat{A}^0_{ik} < -\lambda \end{cases}.$$

By soft-thresholding the coefficients, we only keep the large coefficients. This leads to:

(i) higher **compression** as only a few non-zero entries of A need to be stored (for example this strategy is used for the JPEG format for image compression).

(ii) **de-noising** as noise in an image will be removed by increasing the value of λ

(iii) and also to a **loss of signal**.

The value of λ thus needs to balance the loss of signal with the noise reduction effect, analogous to the number q of PCA components that are kept.

Learning a dictionary. While we can work with a fixed basis H like a Fourier or wavelet basis, we can also learn a basis (or dictionary for an overcomplete basis, that is if $q > p$) that is optimally adapted to the task.

[2] In image analysis, each image corresponds to one row in X

(i) **Non-negative matrix factorisation (NNMF)** for non-negative data X:

$$\text{argmin}_{A\in\mathbb{R}^{n\times q},H\in\mathbb{R}^{q\times p}} \|X-AH\|_2^2 \quad \text{under constraint } A\geq 0 \text{ and } H\geq 0, \tag{4}$$

where the non-negativity constraint is understood elementwise in the matrix. The NMF formulation is useful for non-negative data $X\geq 0$ (say pixel intensities in images or count data in general) and induces sparsity in that many entries of A and H will be zero in the solution. Note that the objective in (1) for the original PCA is zero as soon as $q\geq p$ because then $X=AH$. This is not necessarily the case for NMF and sparse dictionary learning where even values of $q>p$ can be useful. If $q>p$, the basis H is overcomplete since the solutions to

$$\text{argmin}_{A\in\mathbb{R}^{n\times q}} \|X-AH\|_2^2.$$

are not unique any longer and H is then often called a dictionary. The rows of H are again sometimes called atoms and they together define the "learned" dictionary H.

(ii) **Sparse dictionary learning / sparse PCA**

$$\text{argmin}_{A\in\mathbb{R}^{n\times q},H\in\mathbb{R}^{q\times p}} \|X-AH\|_2^2 \quad \text{under constraint } \|A\|_1\leq\tau_A \text{ and } \|H\|_1\leq\tau_H. \tag{5}$$

This can be equivalently formulated (using Lagrange multipliers) for suitable Langrange multpliers $\lambda_A,\lambda_H\geq 0$ as

$$\text{argmin}_{A\in\mathbb{R}^{n\times q},H\in\mathbb{R}^{q\times p}} \|X-AH\|_2^2+\lambda_A\|A\|_1+\lambda_H\|H\|_1, \tag{6}$$

where $\|A\|_1$ is the sum of the absolute values of all entries in A. Two special cases are[3]:

(a) If $\lambda_A=0$, the penalty is only on the dictionary which implies that the learned atoms (the rows of H) will be sparse with many zero entries. This is often called **sparse PCA**.

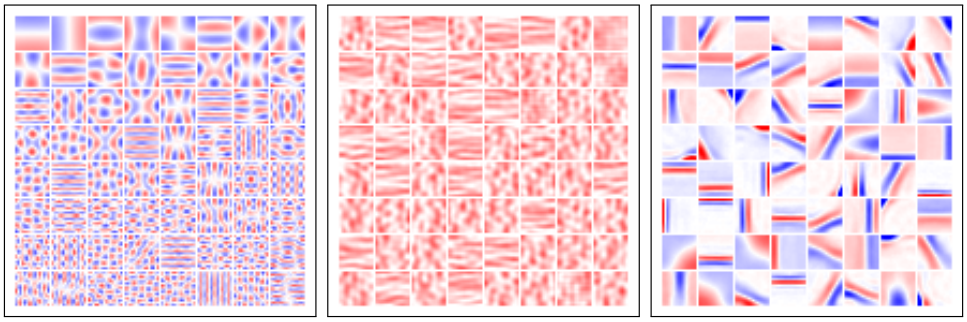
(b) If $\lambda_H=0$, then there is only a penalty on the activity coefficients (or scores) in A and the dictionary is chosen such that the activities can be as sparse as possible. This is typically called **sparse dictionary learning**. For natural images, the learnt dictionaries are often very similar to Gabor filters which again resemble the receptive fields of neurons in the early visual cortex. Note that in general also a fixed wavelet basis achieves very sparse activities A for natural images and this is the reason why denoising with soft-thresholding of wavelet coefficients works in practice. The JPEG compression algorithm also relies on this fact by keeping only the large coefficients in A for a wavelet dictionary.

Two examples for dictionaries H if optimised via NMF or sparse dictionary learning are shown in Figures 2 and 3 for a face dataset and natural images from Mairal et al (2014).

Optimisation. The optimal PCA solution in (1) can be computed exactly via the eigenvalues decomposition or singular value decomposition, as discussed. Optimising the objectives (4) and (5) is harder since the objective functions are not jointly convex in A and H. Let us take a look at (5) again (the situation is analogous for NMF):

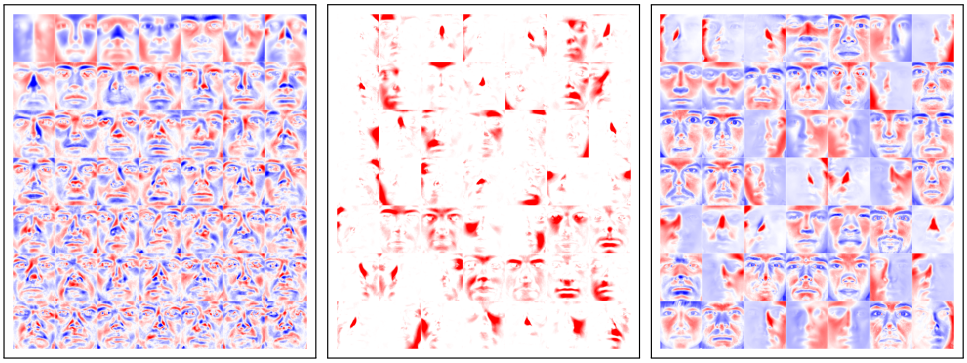
$$(\hat{A},\hat{H})=\text{argmin}_{A\in\mathbb{R}^{n\times q},H\in\mathbb{R}^{q\times p}} \|X-AH\|_2^2+\lambda_A\|A\|_1+\lambda_H\|H\|_1. \tag{7}$$

[3]In both of these special cases, we need to put a constraint on the size of A in the first case and H in the second case, for example by bounding $\|A\|_2^2$ (resp. $\|H\|_2^2$) or by bounding each column of A (resp. each row of H) to have norm less than or equal to 1 (or any other fixed constant).



(a) PCA (b) NNMF (c) DL

Figure 2: *The dictionaries H if trained on natural images via (a) PCA, (b) NMF and (c) sparse dictionary learning. Each panel corresponds to one row in H. From Mairal et al (2010)*



(a) PCA (b) NNMF (c) DL

Figure 3: *The same as in Figure 2 if trained on face data. From Mairal et al (2010)*

If we fix A and optimise for H (or vice versa) then we get convex optimisation problems:

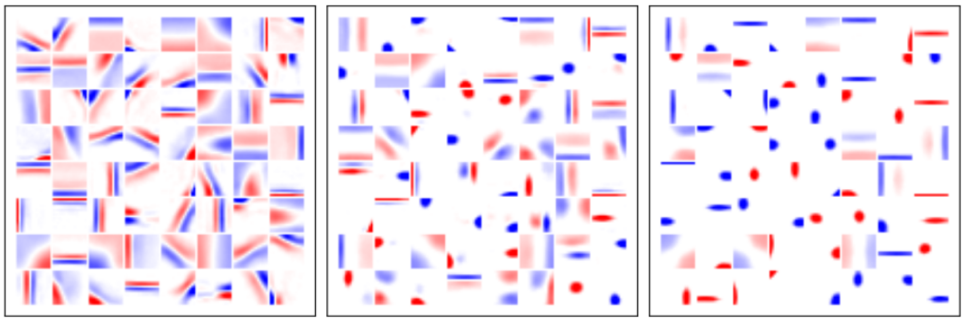
$$\text{fix } H: \quad \hat{A} = \text{argmin}_{A \in \mathbb{R}^{n \times q}} \|X - AH\|_2^2 + \lambda_A \|A\|_1 \tag{8}$$

$$\text{fix } A: \quad \hat{H} = \text{argmin}_{H \in \mathbb{R}^{q \times p}} \|X - AH\|_2^2 + \lambda_H \|H\|_1. \tag{9}$$

We can iterate (8) and (9) until some convergence criterion is reached. Since the overall objective is not jointly convex in (A, H), there is no guarantee that the global optimum will be attained and multiple starts from different starting values are usually performed (and the best solution kept in the end).

The optimisations (8) and (9) are standard ℓ_1-penalized regressions (Lasso). Take the first optimization (8) as an example. It can be solved row-wise for $\hat{A}$ as the objective decomposes over the rows $A_{i\cdot}$ for $i = 1, \ldots, n$ of A as

$$\|X - AH\|_2^2 + \lambda_A \|A\|_1 = \sum_{i=1}^{n} \Big[\|X_{i\cdot}^t - H^t A_{i\cdot}^t\|_2^2 + \lambda_A \|A_{i\cdot}\|_1 \Big],$$



(d) SPCA, $\tau = 70\%$ (e) SPCA, $\tau = 30\%$ (f) SPCA, $\tau = 10\%$

Figure 4: *Sparse PCA solutions on natural images. From Mairal et al (2010)*

where the transpose operation is performed after indexing so that for example $X^t_{i\cdot}$ is the i-th row of X written as a column-vector. The important point is that the objective **decomposes over the rows of** A, that is the objective can be written as a sum and the i-th summand depends on A through the i-th row $A_{i\cdot}$ only. Hence the rows of A can be optimised separately. Defining for a given $i \in \{1, \ldots, n\}$,

$$Y := X^t_{i\cdot} \in \mathbb{R}^p, \quad Z := H^t \in \mathbb{R}^{p\times q} \quad \text{and } b := A^t_{i\cdot} \in \mathbb{R}^q,$$

we get the solution of the transpose of row i, which we denote by q-dimensional column vector column-vector $b = \hat{A}^t_{i\cdot}$,as

$$\text{argmin}_b \|Y - Zb\|_2^2 + \lambda_A \|b\|_1, \tag{10}$$

which is a standard sparse regression problem and can be solved for example with `glmnet` or `lars` in R.

The second optimisation (9) **decomposes column-wise over** H and can be solved analogously. Note that if Z is orthogonal ($Z^tZ = 1_{q\times q}$) –as for example the case if we take for H a fixed wavelet or Fourier basis– then the solutions to (10) are especially simple.

Let $b^{(0)}$ be the solution in the absence of a penalty:

$$b^{(0)} := \text{argmin}_b \frac{1}{2}\|Y - Zb\|_2 = (Z^tZ)^{-1}Z^tY.$$

Note that this solution is only well defined if $q \leq p$ as otherwise Z^tZ will have at most rank $\min\{q, p\} < p$ and will hence not be full-rank. If $q \leq p$ and Z is orthogonal, the solution with penalty is given by soft-thresholding:

$$b := \text{argmin}_b \frac{1}{2}\|Y - Zb\|_2^2 + \lambda\|b\|_1 = [b^{(0)}]_\lambda,$$

where

$$[x]_\lambda = \begin{cases} x - \lambda & \text{if } x > \lambda \\ 0 & \text{if } -\lambda \leq x \leq \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases}.$$

Note that the general solution is well defined even for $q > p$. However, is not a soft-thresholded version of a least-squares estimator any longer, since the least-squares estimator does not exist.

Soft-thresholding the wavelet coefficients A and using $[A]_\lambda$ (for a fixed wavelet basis H) is thus equivalent to solving (8), where the amount of shrinkage depends on the penalty parameter $\lambda = \lambda_A$. Higher values of λ lead to more zero coefficients in the score matrix and thus to higher compression if saving the coefficients, for example. Higher values of λ also lead to more de-noising but also to a loss of signal in the image. The same tradeoff (higher compression and less noise but also loss of some signal) applies to the choice of the penalty parameters in general.

Figure 5: *An example of compressed sensing. The image on the left is missing many pixels. If expanded in a wavelet basis, we can search for the sparsest vector that fits the observed pixels and then use this coefficient vector to complete the image on the unseen pixels as done in the image on the right. (The intermediate images can be ignored here).*

Compressed sensing. Assume we have fixed a dictionary H (for example a wavelet basis for images) such that the coefficient matrix A is usually sparse. Assume we observe just a few entries of a new vector $Y \in \mathbb{R}^p$ (for example observe just a few pixels in a new image – the vector Y can be thought of asa row in the larger matrix X), such that for a set $S \subseteq \{1, \ldots, p\}$ we observe Y_S and do not observe the complement Y_{S^c}. Then we can ask for the sparsest set of coefficients that match the observed pixels Y_S as well as possible by just repeating (10) on the observed part of the vector:

$$\hat{b} = \text{argmin}_{b \in \mathbb{R}^q} \ \|Y_S - Z_{S\cdot} b\|_2 + \lambda_A \|b\|_1, \tag{11}$$

This allows to "complete the observations" by extending to the unobserved part as

$$\hat{Y} = Z\hat{b}.$$

If we make the penalty λ_A very small in (12), then we will converge for λ_A to the so-called basis pursuit solution

$$\hat{b} = \text{argmin}_{b \in \mathbb{R}^q} \ \|b\|_1 \ \text{ such that } \ Y_S = Z_{S\cdot} b, \tag{12}$$

that is we look for the sparsest vector b such that the observed points are matched exactly. Note that in general $|S| < q$ and a perfect match $Y_S \equiv Z_{S\cdot} b$ can be obtained for multiple solutions b. We seek here the sparsest among all these solutions, which is obtained as the limit of (12) for $\lambda_A \to 0$. In this case the solution $\hat{Y}$ will match the observed part exactly, that is $\hat{Y}_S = Y_S$. The reconstruction of the entire vector Y will in general succeed if there is a sparse approximation b such that $Y \approx Zb$. There are also some design conditions on Z, which are usually satisfied for an

orthogonal matrix but some non-orthogonality is also possible [4]

The reconstruction of Y from incomplete observations is sometimes called **compressed sensing**. Note that a crucial element for the success is that Y (or a row of $X_{i\cdot}$) can be represented by a sparse combination of the rows of the dictionary H, that is by a sparse row of $A_{i\cdot}$, where a row is called sparse here again if it has a low ℓ_1-norm, that is if $\|A_{i\cdot}\|_1$ is small, or if it has a low ℓ_0-quasinorm, that is if $\|A_{i\cdot}\|_0 = \sum_{j=1}^{p} 1\{A_{ij} \neq 0\}$, the number of non-zero entries in this row, is small. This is a further motivation for trying to find a dictionary H in which A is sparse (besides the data compression and potential noise reduction effect).

[4] the so-called sparse eigenvalues of the sensing matrix –here $Z_{S\cdot}$– need to be bounded away from 0, which means that we need $\min_{b':\|b'\|_2=1,\|b'\|_0\leq|S|} \|Z_{S\cdot}b\|_2 > \delta$ for some constant $\delta > 0$ or similar conditions. A setting where these conditions hold, with high probability, is if we observe ΦZ where Φ is a random projection into a lower-dimensional space, for example by choosing the entries in Φ iid $\mathcal{N}(0,1)$, which corresponds to observing random linear combinations of the pixels.