Computational Intelligence Lab

Cristian Perez Jensen

April 9, 2024

Contents

1 Dimensionality reduction 1
1.1 Linear autoencoders 1
1.2 Projection 2
1.3 Principal component analysis 6
1.4 Learning algorithms 6
1.5 Non-linear autoencoders 7

List of symbols

$\doteq$	Equality by definition
$\approx$	Approximate equality
$\propto$	Proportional to
$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$i:j$	Set of natural numbers between i and j. *I.e.*, $\{i, i+1, \ldots, j\}$
$f: A \to B$	Function f that maps elements of set A to elements of set B
$\mathbb{1}\{\text{predicate}\}$	Indicator function (1 if predicate is true, otherwise 0)
$\boldsymbol{v} \in \mathbb{R}^n$	n-dimensional vector
$\boldsymbol{M} \in \mathbb{R}^{m \times n}$	$m \times n$ matrix
$\mathbf{T} \in \mathbb{R}^{d_1 \times \cdots \times d_n}$	Tensor
$\boldsymbol{M}^\top$	Transpose of matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	Inverse of matrix $\boldsymbol{M}$
$\det(\boldsymbol{M})$	Determinant of $\boldsymbol{M}$
$\frac{\mathrm{d}}{\mathrm{d}x} f(x)$	Ordinary derivative of $f(x)$ w.r.t. x at point $x \in \mathbb{R}$
$\frac{\partial}{\partial x} f(\boldsymbol{x})$	Partial derivative of $f(\boldsymbol{x})$ w.r.t. x at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^n$	Gradient of $f: \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\nabla}^2_{\boldsymbol{x}} f(\boldsymbol{x}) \in \mathbb{R}^{n \times n}$	Hessian of $f: \mathbb{R}^n \to \mathbb{R}$ at point $\boldsymbol{x} \in \mathbb{R}^n$
$\boldsymbol{\theta} \in \Theta$	Parametrization of a model, where Θ is a compact subset of $\mathbb{R}^K$
$\mathcal{X}$	Input space
$\mathcal{Y}$	Output space
$\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$	Labeled training data

1 Dimensionality reduction

The motivation behind dimensionality reduction is to find a low-dimensional representation of high-dimensional data.[1] This can be split into two goals: (1) compressing the data, while preserving as much as possible of the relevant information, and (2) interpreting the data in low dimensionality is easier than high dimensionality.

[1] Often, the original raw representation is high-dimensional and redundant, *e.g.*, images, audio, time series.

Dimensionality reduction is often performed by an autoencoder, which typically has a bottleneck and aims to predict its input; see Figure 1. In general, we have an encoder F and a decoder G,

$$F : \mathbb{R}^n \to \mathbb{R}^m, \quad G : \mathbb{R}^m \to \mathbb{R}^n. \qquad m \ll n.$$

The reconstruction function is then the following function,

$$G \circ F : \mathbb{R}^n \to \mathbb{R}^n,$$

which is ideally the identity function.

1.1 Linear autoencoders

To build a nice theory, we will only consider a single layer linear autoencoder, which means that we have the following functions,

$$F : \boldsymbol{x} \mapsto \boldsymbol{z} = \boldsymbol{W}\boldsymbol{x}, \quad \boldsymbol{W} \in \mathbb{R}^{m \times n}.$$
$$G : \boldsymbol{z} \mapsto \hat{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{z}, \quad \boldsymbol{V} \in \mathbb{R}^{n \times m}.$$

The objective to minimize of the linear encoder is the following,

$$\mathcal{R}(\boldsymbol{W}, \boldsymbol{V}) = \mathcal{R}(\boldsymbol{P} \doteq \boldsymbol{V}\boldsymbol{W}) \doteq \mathbb{E}\left[\frac{1}{2}\|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}\|^2\right].$$

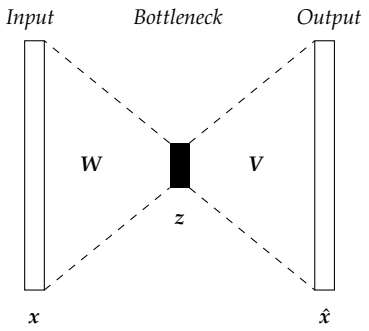

Figure 1. Diagram of a single layer linear autoencoder.

Corollary. For centered data, *i.e.* $E[\boldsymbol{x}] = 0$, optimal affine maps degenerate to linear ones.

Proof. Proof by contradiction. Let $\boldsymbol{a} \neq \boldsymbol{0}$, then

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{x} - (\boldsymbol{P}\boldsymbol{x} + \boldsymbol{a})\|^2\right] &= \mathbb{E}[\langle \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x} - \boldsymbol{a}, \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x} - \boldsymbol{a}\rangle] \\
&= \mathbb{E}[\langle \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}, \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}\rangle + 2\langle \boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}, -\boldsymbol{a}\rangle + \langle -\boldsymbol{a}, -\boldsymbol{a}\rangle] \\
&= \mathbb{E}\left[\|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}\|^2\right] - 2\langle \underbrace{\mathbb{E}[\boldsymbol{x}] - \boldsymbol{P}\mathbb{E}[\boldsymbol{x}]}_{=\boldsymbol{0}}, \boldsymbol{a}\rangle + \underbrace{\|\boldsymbol{a}\|^2}_{>0} \\
&> \mathbb{E}\left[\|\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}\|^2\right].
\end{aligned}$$

Thus, the risk is strictly worse if $\boldsymbol{a} \neq \boldsymbol{0}$. ∎

Thus, we will assume that the data is centered, which makes the analysis easier, since we do not need to consider the affine case.

Note that while the optimal linear reconstruction map $\boldsymbol{P}$ is unique, its parametrization $\boldsymbol{W}, \boldsymbol{V}$ is not unique, since for any invertible matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$, we can construct an optimal parametrization,

$$\boldsymbol{VW} = \boldsymbol{VIW} = \boldsymbol{V}(\boldsymbol{AA}^{-1})\boldsymbol{W} = (\boldsymbol{VA})(\boldsymbol{A}^{-1}\boldsymbol{W}),$$

with $\boldsymbol{A}^{-1}\boldsymbol{W}, \boldsymbol{VA}$. The weight matrices are non-identifiable.

Since $\boldsymbol{P}$ cannot be any $n \times n$ matrix, we want to know how the composition of $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times m}$ characterizes the matrix $\boldsymbol{P}$ and which constraints they impose. The answer to this is that the weight matrices impose a rank constraint on $\boldsymbol{P}$,

$$\operatorname{rank}(\boldsymbol{P}) = \min\{\operatorname{rank}(\boldsymbol{W}), \operatorname{rank}(\boldsymbol{W})\} \leq \min\{m, n\} = m.$$

Thus, when optimizing for $\boldsymbol{P}$, we are constrained to matrices with rank less or equal to m.

1.2 *Projection*

The rank constraint and linearity of $\boldsymbol{P}$ means that the image (column space) of $\boldsymbol{P}$ is a linear subspace $\mathcal{U} \subseteq \mathbb{R}^n$ of dimension at most m. We will break the solution to our problem into two parts: (1) finding the optimal subspace $\mathcal{U}$, and (2) finding the optimal mapping to that subspace.[2]

[2] We do not search for the weight matrices $\boldsymbol{W}, \boldsymbol{V}$, since they are not unique, but $\boldsymbol{P}$ is unique.

Finding the optimal mapping to a subspace. We will first focus on (2); given subspace $\mathcal{U}$, we need to determine the optimal linear map $\boldsymbol{P}^\star$, such that

$$\boldsymbol{P}^\star = \operatorname*{argmin}_{\boldsymbol{P}} \|\boldsymbol{x} - \boldsymbol{Px}\|^2, \quad \operatorname{col}(\boldsymbol{P}) = \mathcal{U}.$$

Definition 1 (Orthogonal projection)**.** A linear transformation $P : \mathcal{V} \to \mathcal{V}$ is called an orthogonal projection onto $\mathcal{U}$ if $\forall \boldsymbol{x} \in \mathcal{V}$:

1. *Projection*: $P(\boldsymbol{x}) \in \mathcal{U}$;
2. *Orthogonality*: $\operatorname{null}(P) \perp \operatorname{col}(P)$, which is equivalent to the following holding, $\langle P(\boldsymbol{x}), \boldsymbol{y} \rangle = \langle \boldsymbol{x}, P(\boldsymbol{y}) \rangle$ (self-adjointness);
3. *Idempotency*: $P(P(\boldsymbol{x})) = P(\boldsymbol{x})$.

Definition 2. The orthogonal projection to a linear subspace $\mathcal{U} \subseteq \mathbb{R}^n$ is defined as

$$\Pi_{\mathcal{U}} : \mathbb{R}^n \to \mathcal{U}, \quad \Pi_{\mathcal{U}}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{x} - \boldsymbol{x}'\|.$$

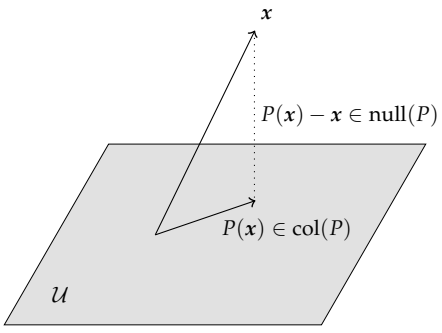

Figure 2. Orthogonal projection of $\boldsymbol{x}$ onto subspace plane $\mathcal{U}$.

Proof. We need to show that the definition of $\Pi_{\mathcal{U}}$ indeed is an orthogonal projection by showing that it adheres to the properties of Definition 1.

1. *Projection*: This is true by definition of the values that the argmin are allowed to take on;
2. *Idempotency*: For all $\boldsymbol{u} \in \mathcal{U}$, $\Pi_{\mathcal{U}}(\boldsymbol{u}) = \operatorname{argmin}_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{u} - \boldsymbol{x}'\| = \boldsymbol{u}$. Thus, $\Pi_{\mathcal{U}} = \Pi_{\mathcal{U}} \circ \Pi_{\mathcal{U}}$;
3. *Orthogonality*: We need to show that $\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x} \in \mathcal{U}^{\perp}$. Decompose it into $\boldsymbol{u} \in \mathcal{U}$ and $\boldsymbol{u}^{\perp} \in \mathcal{U}^{\perp}$ by $\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x} = \boldsymbol{u} + \boldsymbol{u}^{\perp}$. Then, we only need to show that $\boldsymbol{u} = \boldsymbol{0}$.

   Proof by contradiction. Let $\boldsymbol{u} \neq \boldsymbol{0}$, then

$$\begin{aligned}\|\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}\|^2 &= \underbrace{\|\boldsymbol{u}\|^2}_{>0} + \|\boldsymbol{u}^{\perp}\|^2 + 2\underbrace{\langle \boldsymbol{u}, \boldsymbol{u}^{\perp} \rangle}_{=0} \\ &> \|\boldsymbol{u}^{\perp}\|^2 \\ &= \|\underbrace{\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{u}}_{\in \mathcal{U}} - \boldsymbol{x}\|^2,\end{aligned}$$

   which contradicts with

$$\|\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}\| = \min_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{x}' - \boldsymbol{x}\| \leq \|\tilde{\boldsymbol{u}} - \boldsymbol{x}\|, \quad \forall \tilde{\boldsymbol{u}} \in \mathcal{U}.$$

   Hence, $\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x} \in \mathcal{U}^{\perp}$ and is unique;
4. *Linearity*: We need to show homogeneity,

$$\begin{aligned}\Pi_{\mathcal{U}}(\alpha \boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{x}'' \in \mathcal{U}} \|\alpha \boldsymbol{x} - \boldsymbol{x}''\| \\ &= \operatorname*{argmin}_{\alpha \boldsymbol{x}' \in \mathcal{U}} \|\alpha \boldsymbol{x} - \alpha \boldsymbol{x}'\| && \boldsymbol{x}' = \tfrac{1}{a}\boldsymbol{x}''. \\ &= \alpha \operatorname*{argmin}_{\boldsymbol{x}' \in \mathcal{U}} |\alpha| \|\boldsymbol{x} - \boldsymbol{x}'\| && \operatorname{argmin}_{\lambda x} f(\boldsymbol{x}) = \lambda \operatorname{argmin}_{x} f(\boldsymbol{x}). \\ &= \alpha \operatorname*{argmin}_{\boldsymbol{x}' \in \mathcal{U}} \|\boldsymbol{x} - \boldsymbol{x}'\| \\ &= \alpha \Pi_{\mathcal{U}}(\boldsymbol{x}).\end{aligned}$$

   And, we need to show additivity,

$$\begin{aligned}\Pi_{\mathcal{U}}(\boldsymbol{x} + \boldsymbol{y}) &= \operatorname*{argmin}_{\boldsymbol{z} \in \mathcal{U}} \|\boldsymbol{z} - (\boldsymbol{x} + \boldsymbol{y})\| \\ &= \operatorname*{argmin}_{\Pi_{\mathcal{U}}(\boldsymbol{x}) + \boldsymbol{y}' \in \mathcal{U}} \|\Pi_{\mathcal{U}}(\boldsymbol{x}) + \boldsymbol{y}' - \boldsymbol{x} - \boldsymbol{y}\| && \boldsymbol{z} \doteq \Pi_{\mathcal{U}}(\boldsymbol{x}) + \boldsymbol{y}' \text{ for some } \boldsymbol{y}' \in \mathcal{U}. \\ &= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \operatorname*{argmin}_{\boldsymbol{y}' \in \mathcal{U}} \|(\boldsymbol{y}' - \boldsymbol{y}) + (\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x})\|^2 && \operatorname{argmin}_{x+\lambda} f(\boldsymbol{x}) = \lambda + \operatorname{argmin}_{x} f(\boldsymbol{x}). \\ &= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \operatorname*{argmin}_{\boldsymbol{y}' \in \mathcal{U}} \|\boldsymbol{y}' - \boldsymbol{y}\|^2 + \|\Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}\|^2 + 2\langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' - \boldsymbol{y} \rangle \\ &= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \operatorname*{argmin}_{\boldsymbol{y}' \in \mathcal{U}} \|\boldsymbol{y}' - \boldsymbol{y}\|^2 + 2\underbrace{\langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' \rangle}_{\langle \mathcal{U}^{\perp}, \mathcal{U} \rangle = 0} && \langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' - \boldsymbol{y} \rangle = \langle \Pi_{\mathcal{U}}(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y}' \rangle - \boldsymbol{y}. \\ &= \Pi_{\mathcal{U}}(\boldsymbol{x}) + \Pi_{\mathcal{U}}(\boldsymbol{y}).\end{aligned}$$

■

So, we know that $\Pi_{\mathcal{U}}$ is an orthogonal projection. Now, we want to find the matrix $\boldsymbol{P}$ representing that linear transformation.

Given an orthonormal basis $\boldsymbol{U}$ of $\mathcal{U}$, we can compute the optimal projection matrix,

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top.$$

Note that in this case $\boldsymbol{W}$ and $\boldsymbol{V}$ share parameters, and $\boldsymbol{U}\boldsymbol{U}^\top$ is the optimal weight matrix if we enforce parameter sharing via $\boldsymbol{V} = \boldsymbol{W}^\top$.

Proof. The image of the projection matrix is $\mathcal{U}$,

$$\boldsymbol{P}\boldsymbol{x} = \left(\sum_{i=1}^{m} \boldsymbol{u}_i \boldsymbol{u}_i^\top\right)\boldsymbol{x} = \sum_{i=1}^{m} \boldsymbol{u}_i \boldsymbol{u}_i^\top \boldsymbol{x} = \sum_{i=1}^{m} \langle \boldsymbol{u}_i, \boldsymbol{x}\rangle \boldsymbol{u}_i.$$

Self-adjointness,

$$\boldsymbol{P}^\top = (\boldsymbol{U}\boldsymbol{U}^\top)^\top = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{P}.$$

Idempotency,

$$\boldsymbol{P}\boldsymbol{P} = \boldsymbol{U}\underbrace{\boldsymbol{U}^\top\boldsymbol{U}}_{\boldsymbol{I}_m}\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{P}.$$

Orthogonality, TODO ■

In general, we do not have an orthonormal basis for $\mathcal{U}$. In a non-orthonormal basis $\boldsymbol{V}$ for $\mathcal{U}$, we can recover the projection matrix,

$$\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^+, \quad \boldsymbol{V}^+ \doteq \left(\boldsymbol{V}^\top\boldsymbol{V}\right)^{-1}\boldsymbol{V}^\top.$$

Note that $\boldsymbol{V}^+$ is the left Moore-Penrose pseudo-inverse of $\boldsymbol{V}$.

Proof. $\boldsymbol{P}$ is the projection matrix of $\mathcal{U}$,

$$\boldsymbol{P}\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^+\boldsymbol{V} = \boldsymbol{V}\left(\boldsymbol{V}^\top\boldsymbol{V}\right)^{-1}\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{V}.$$

Together with the rank constraint, this yields $\boldsymbol{P}\boldsymbol{u}^\perp = \boldsymbol{0}$ for all $\boldsymbol{u}^\perp \in \mathcal{U}^\perp$.

Self-adjointness,

$$\boldsymbol{P}^\top = \left(\boldsymbol{V}\left(\boldsymbol{V}^\top\boldsymbol{V}\right)^{-1}\boldsymbol{V}^\top\right)^\top = \boldsymbol{V}\left(\boldsymbol{V}^\top\boldsymbol{V}\right)^{-1}\boldsymbol{V}^\top = \boldsymbol{P}.$$

Idempotency,

$$\boldsymbol{P}\boldsymbol{P} = \boldsymbol{V}\left(\boldsymbol{V}^\top\boldsymbol{V}\right)^{-1}\boldsymbol{V}^\top\boldsymbol{V}\left(\boldsymbol{V}^\top\boldsymbol{V}\right)^{-1}\boldsymbol{V}^\top = \boldsymbol{V}\boldsymbol{V}^+ = \boldsymbol{P}.$$

■

Finding the optimal subspace. Now we need to find out which subspace of dimension m or less is optimal to project onto. First, we need to rewrite the objective function to find a new interpretation,

$$\begin{aligned}
\mathcal{R}(\boldsymbol{P}) &= \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{x}-\boldsymbol{P}\boldsymbol{x}\|^2\right] \\
&= \frac{1}{2}\mathbb{E}[\langle \boldsymbol{x},\boldsymbol{x}\rangle + 2\langle \boldsymbol{x},-\boldsymbol{P}\boldsymbol{x}\rangle + \langle -\boldsymbol{P}\boldsymbol{x},-\boldsymbol{P}\boldsymbol{x}\rangle] \\
&= \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{x}\|^2\right] + \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] - \mathbb{E}[\langle \boldsymbol{x},\boldsymbol{P}\boldsymbol{x}\rangle] \\
&= \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{x}\|^2\right] + \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] - \mathbb{E}[\langle \boldsymbol{x},\boldsymbol{P}^2\boldsymbol{x}\rangle] && \text{Idempotency of projection matrices.} \\
&= \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{x}\|^2\right] + \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] - \mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] && \langle \boldsymbol{x},\boldsymbol{P}^2\boldsymbol{x}\rangle = \langle \boldsymbol{P}\boldsymbol{x},\boldsymbol{P}\boldsymbol{x}\rangle. \\
&= \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{x}\|^2\right] - \frac{1}{2}\mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right].
\end{aligned}$$

Because our data is centered, we know the following,

$$\begin{aligned}
\mathrm{Var}[\boldsymbol{x}] &= \mathbb{E}\left[\|\boldsymbol{x}\|^2\right] - \|\underbrace{\mathbb{E}[\boldsymbol{x}]}_{=\boldsymbol{0}}\|^2 \\
&= \mathbb{E}\left[\|\boldsymbol{x}\|^2\right]. \\
\mathrm{Var}[\boldsymbol{P}\boldsymbol{x}] &= \mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] - \|\mathbb{E}[\boldsymbol{P}\boldsymbol{x}]\|^2 \\
&= \mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] - \|\boldsymbol{P}\underbrace{\mathbb{E}[\boldsymbol{x}]}_{=\boldsymbol{0}}\|^2 \\
&= \mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right].
\end{aligned}$$

$$\mathcal{R}(\boldsymbol{P}) = \frac{1}{2}(\mathrm{Var}[\boldsymbol{x}] - \mathrm{Var}[\boldsymbol{P}\boldsymbol{x}]) \propto -\frac{1}{2}\mathrm{Var}[\boldsymbol{P}\boldsymbol{x}].$$

Hence, minimizing $\mathcal{R}(\boldsymbol{P})$ is equivalent to maximizing the variance $\mathrm{Var}[\boldsymbol{P}\boldsymbol{x}]$.

We can further simplify this expression to find a sufficient statistic for the objective function,

$$\begin{aligned}
-\frac{1}{2}\mathrm{Var}[\boldsymbol{P}\boldsymbol{x}] &= -\frac{1}{2}\mathbb{E}\left[\|\boldsymbol{P}\boldsymbol{x}\|^2\right] \\
&= -\frac{1}{2}\mathbb{E}[\langle \boldsymbol{x},\boldsymbol{P}\boldsymbol{x}\rangle] && \|\boldsymbol{P}\boldsymbol{x}\|^2 = \langle \boldsymbol{P}\boldsymbol{x},\boldsymbol{P}\boldsymbol{x}\rangle = \langle \boldsymbol{x},\boldsymbol{P}^2\boldsymbol{x}\rangle = \langle \boldsymbol{x},\boldsymbol{P}\boldsymbol{x}\rangle. \\
&= -\frac{1}{2}\mathbb{E}\left[\mathrm{tr}\left(\boldsymbol{x}^\top\boldsymbol{P}\boldsymbol{x}\right)\right] \\
&= -\frac{1}{2}\mathrm{tr}\left(\mathbb{E}\left[\boldsymbol{P}\boldsymbol{x}\boldsymbol{x}^\top\right]\right) && \text{Cyclic property of trace.} \\
&= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{P}\mathbb{E}\left[\boldsymbol{x}\boldsymbol{x}^\top\right]\right).
\end{aligned}$$

$\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is a sufficient statistic for $\mathcal{R}(\boldsymbol{P})$.

The optimal projection is fully determined by the covariance matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$, together with $\mathbb{E}[\boldsymbol{x}]$ for centering.

1.3 Principal component analysis

Theorem 3 (Spectral theorem)**.** Any symmetric and positive semidefinite matrix $\boldsymbol{\Sigma}$ can be non-negatively diagonalized with an orthogonal matrix,

$$\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top, \quad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_n),$$

where $\lambda \geq \cdots \geq \lambda_n \geq 0$ and $\boldsymbol{Q}$ is orthogonal.

Remark. $\boldsymbol{Q}$ is composed of ordered eigenvectors of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Lambda}$ is composed of ordered eigenvalues of $\boldsymbol{\Sigma}$.

Theorem 4 (PCA theorem)**.** The variance maximizing projection matrix $\boldsymbol{P}$ for a covariance matrix $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ as in the spectral theorem is given by

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top, \quad \boldsymbol{U} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{I}_m \\ \boldsymbol{0}\end{bmatrix}.$$

Proof.

$$\begin{aligned}
\operatorname{Var}[\boldsymbol{P}\boldsymbol{x}] &= \operatorname{tr}\Big(\boldsymbol{P}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]\Big) \\
&= \operatorname{tr}\Big(\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top\Big) && \boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top,\ \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top. \\
&= \operatorname{tr}\Big(\Big(\boldsymbol{Q}^\top\boldsymbol{U}\Big)\Big(\boldsymbol{Q}^\top\boldsymbol{U}\Big)^\top\boldsymbol{\Lambda}\Big). && \text{Cyclic property.}
\end{aligned}$$

This term is maximized by $\boldsymbol{Q}^\top\boldsymbol{U} = \begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{0}\end{bmatrix}^\top$. ■

1.4 Learning algorithms

Eigenvalue decomposition of the (symmetric) sample covariance matrix has $\mathcal{O}(n^3)$ complexity. Furthermore, the complexity of computing $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ is $\mathcal{O}(Nn^2)$.[3] This is quite costly, thus we need to search for algorithms that have lower runtime complexity.

[3] Typically, $N \gg n$.

Power method. The power method is a recursive algorithm for computing principal eigenvectors. It initializes a vector at random $\boldsymbol{v}^{(0)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then, it iteratively improves this guess,

$$\boldsymbol{v}^{(t+1)} = \frac{\boldsymbol{A}\boldsymbol{v}^{(t)}}{\|\boldsymbol{A}\boldsymbol{v}^{(t)}\|}.$$

The computational complexity of this algorithm is $\mathcal{O}(Tn^2)$.

Lemma 5. Let $\boldsymbol{u}_1$ be the unique principal eigenvector of a diagonalizable matrix $\boldsymbol{A}$ with eigenvalues $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n \geq 0$. If $\langle \boldsymbol{v}_0, \boldsymbol{u}_1 \rangle \neq 0$, then

$$\lim_{t \to \infty} \boldsymbol{v}^{(t)} = \boldsymbol{u}_1.$$

Proof. We can decompose vectors as a linear combination of eigenvectors, $\boldsymbol{v}^{(0)} = \sum_{i=1}^{n} \alpha_i \boldsymbol{u}_i$. Then,

$$\begin{aligned} \boldsymbol{v}^{(k)} &\propto \boldsymbol{A}^k \boldsymbol{v}^{(0)} \\ &= \sum_{i=1}^{n} \alpha_i \lambda_i^k \boldsymbol{u}_i && \boldsymbol{A}^k \boldsymbol{v}^{(0)} = \textstyle\sum_{i=1}^{n} \alpha_i \boldsymbol{A}^k \boldsymbol{u}_i. \\ &\propto \alpha_1 \boldsymbol{u}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1} \right)^k \boldsymbol{u}_i. && \text{Divide by } \lambda_1. \end{aligned}$$

$\lambda_i / \lambda_1 < 1$ for $i > 1$, thus the sum goes to o and $\boldsymbol{v}^{(k)} \to \boldsymbol{u}_1$. ■

We can use this algorithm to also compute the next principal eigenvectors by factoring out $\boldsymbol{u}_1$ and then doing the algorithm again to recover $\boldsymbol{u}_2$, and continue doing that until we have the m principal eigenvectors.

Thus, the total complexity of finding the m principal eigenvectors is $\mathcal{O}(Tmn^2)$. However, this does not get rid of the $\mathcal{O}(Nn^2)$ complexity for computing the sample covariance matrix.

Gradient descent. By treating the autoencoder as a neural network, we can use deep learning techniques, such as gradient descent. Gradient descent iteratively updates the weights by

$$\boldsymbol{P}^{(t+1)} = \boldsymbol{P}^{(t)} - \eta \nabla_{\boldsymbol{P}} \mathcal{R}(\boldsymbol{P}).$$

The gradient is computed by $(\boldsymbol{P} - \boldsymbol{I})\boldsymbol{x}\boldsymbol{x}^\top$. The problem with this is that we cannot constrain $\boldsymbol{P}$ to be a projection. Thus, we actually need to update $\boldsymbol{V}$ and $\boldsymbol{W}$. Thus, by the chain rule for matrix derivatives,

$$\begin{aligned} \nabla_{\boldsymbol{W}} \mathcal{R}(\boldsymbol{W}, \boldsymbol{V}) &= (\boldsymbol{P} - \boldsymbol{I})\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{W}^\top \\ \nabla_{\boldsymbol{V}} \mathcal{R}(\boldsymbol{W}, \boldsymbol{V}) &= \boldsymbol{V}^\top (\boldsymbol{P} - \boldsymbol{I})\boldsymbol{x}\boldsymbol{x}^\top. \end{aligned}$$

The complexity for T iterations is then $\mathcal{O}(T(m+k)n^2)$, where k is the batch size.

1.5 Non-linear autoencoders

We can get much better performance by considering non-linearities, and we can easily use gradient descent to work with non-linear architectures.