Chapter 12
Latent linear models

12.1 Factor analysis

One problem with mixture models is that they only use a single latent variable to generate the observations. In particular, each observation can only come from one of K prototypes. One can think of a mixture model as using K hidden binary variables, representing a one-hot encoding of the cluster identity. But because these variables are mutually exclusive, the model is still limited in its representational power.

An alternative is to use a vector of real-valued latent variables,$\boldsymbol{z}_i \in \mathbb{R}^L$. The simplest prior to use is a Gaussian (we will consider other choices later):

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \tag{12.1}$$

If the observations are also continuous, so $\boldsymbol{x}_i \in \mathbb{R}^D$, we may use a Gaussian for the likelihood. Just as in linear regression, we will assume the mean is a linear function of the (hidden) inputs, thus yielding

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}) \tag{12.2}$$

where $\boldsymbol{W}$ is a $D \times L$ matrix, known as the **factor loading matrix**, and $\boldsymbol{\Psi}$ is a $D \times D$ covariance matrix. We take $\boldsymbol{\Psi}$ to be diagonal, since the whole point of the model is to force $\boldsymbol{z}_i$ to explain the correlation, rather than baking it in to the observations covariance. This overall model is called **factor analysis** or **FA**. The special case in which $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$ is called **probabilistic principal components analysis** or **PPCA**. The reason for this name will become apparent later.

12.1.1 FA is a low rank parameterization of an MVN

FA can be thought of as a way of specifying a joint density model on $\boldsymbol{x}$ using a small number of parameters. To see this, note that from Equation 4.39, the induced marginal distribution $p(\boldsymbol{x}_i|\boldsymbol{\theta})$ is a Gaussian:

$$\begin{aligned} p(\boldsymbol{x}_i|\boldsymbol{\theta}) &= \int \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi})\mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)\mathrm{d}\boldsymbol{z}_i \\ &= \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{\mu}_0 + \boldsymbol{\mu}, \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_0\boldsymbol{W}) \end{aligned} \tag{12.3}$$

From this, we see that we can set $\boldsymbol{\mu}_0 = 0$ without loss of generality, since we can always absorb $\boldsymbol{W}\boldsymbol{\mu}_0$ into $\boldsymbol{\mu}$. Similarly, we can set $\boldsymbol{\Sigma}_0 = \boldsymbol{I}$ without loss of generality, because we can always emulate a correlated prior by using defining a new weight matrix, $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{\Sigma}_0^{-\frac{1}{2}}$. So we can rewrite Equation 12.6 and 12.2 as:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{12.4}$$

$$p(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}, \boldsymbol{\Psi}) \tag{12.5}$$

We thus see that FA approximates the covariance matrix of the visible vector using a low-rank decomposition:

$$\boldsymbol{C} \triangleq \mathrm{cov}[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\Psi} \tag{12.6}$$

This only uses $O(LD)$ parameters, which allows a flexible compromise between a full covariance Gaussian, with $O(D^2)$ parameters, and a diagonal covariance, with $O(D)$ parameters. Note that if we did not restrict $\boldsymbol{\Psi}$ to be diagonal, we could trivially set $\boldsymbol{\Psi}$ to a full covariance matrix; then we could set $\boldsymbol{W} = 0$, in which case the latent factors would not be required.

12.1.2 Inference of the latent factors

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{12.7}$$

$$\boldsymbol{\Sigma}_i \triangleq (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W})^{-1} \tag{12.8}$$

$$= (\boldsymbol{I} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W})^{-1} \tag{12.9}$$

$$\boldsymbol{\mu}_i \triangleq \boldsymbol{\Sigma}_i[\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0] \tag{12.10}$$

$$= \boldsymbol{\Sigma}_i\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}) \tag{12.11}$$

Note that in the FA model, $\boldsymbol{\Sigma}_i$ is actually independent of i, so we can denote it by $\boldsymbol{\Sigma}$. Computing this matrix takes $O(L^3 + L^2D)$ time, and computing each $\boldsymbol{\mu}_i = \mathbb{E}[\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta}]$ takes $O(L^2 + LD)$ time. The $\boldsymbol{\mu}_i$ are sometimes called the **latent scores**, or **latent factors**.

12.1.3 Unidentifiability

Just like with mixture models, FA is also unidentifiable. To see this, suppose $\boldsymbol{R}$ is an arbitrary orthogonal rotation matrix, satisfying $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{I}$. Let us define $\tilde{\boldsymbol{W}} = \boldsymbol{W}\boldsymbol{R}$, then the likelihood function of this modified matrix is the same as for the unmodified matrix, since $\boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^T\boldsymbol{W}^T + \boldsymbol{\Psi} = \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\Psi}$. Geometrically, multiplying $\boldsymbol{W}$ by an orthogonal matrix is like rotating $\boldsymbol{z}$ before generating $\boldsymbol{x}$.

To ensure a unique solution, we need to remove $L(L-1)/2$ degrees of freedom, since that is the number of orthonormal matrices of size $L \times L$.[22] In total, the FA model has $D + LD - L(L-1)/2$ free parameters (excluding the mean), where the first term arises from $\boldsymbol{\Psi}$. Obviously we require this to be less than or equal to $D(D+1)/2$, which is the number of parameters in an unconstrained (but symmetric) covariance matrix. This gives us an upper bound on L, as follows:

$$L_{\max} = \lfloor D + 0.5(1 - \sqrt{1+8D}) \rfloor \tag{12.12}$$

For example, $D = 6$ implies $L \leq 3$. But we usually never choose this upper bound, since it would result in overfitting (see discussion in Section 12.3 on how to choose L).

Unfortunately, even if we set $L < L_{\max}$, we still cannot uniquely identify the parameters, since the rotational ambiguity still exists. Non-identifiability does not affect the predictive performance of the model. However, it does affect the loading matrix, and hence the interpretation of the latent factors. Since factor analysis is often used to uncover structure in the data, this problem needs to be addressed. Here are some commonly used solutions:

- **Forcing $\boldsymbol{W}$ to be orthonormal** Perhaps the cleanest solution to the identifiability problem is to force $\boldsymbol{W}$ to be orthonormal, and to order the columns by decreasing variance of the corresponding latent factors. This is the approach adopted by PCA, which we will discuss in Section 12.2. The result is not necessarily more interpretable, but at least it is unique.
- **Forcing $\boldsymbol{W}$ to be lower triangular** One way to achieve identifiability, which is popular in the Bayesian community (e.g., (Lopes and West 2004)), is to ensure that the first visible feature is only generated by the first latent factor, the second visible feature is only generated by the first two latent factors, and so on. For example, if $L = 3$ and $D = 4$, the correspond factor loading matrix is given by

$$\boldsymbol{W} = \begin{pmatrix} w_{11} & 0 & 0 \\ w_{21} & w_{22} & 0 \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{32} & w_{43} \end{pmatrix}$$

  We also require that $w_{jj} > 0$ for $j = 1 : L$. The total number of parameters in this constrained matrix is $D + DL - L(L-1)/2$, which is equal to the number of uniquely identifiable parameters. The disadvantage of this method is that the first L visible variables, known as the **founder variables**, affect the interpretation of the latent factors, and so must be chosen carefully.
- **Sparsity promoting priors on the weights** Instead of pre-specifying which entries in $\boldsymbol{W}$ are zero, we can encourage the entries to be zero, using ℓ_1 regularization (Zou et al. 2006), ARD (Bishop 1999; Archambeau and Bach 2008), or spike-and-slab priors (Rattray et al. 2009). This is called sparse factor analysis. This does not necessarily ensure a unique MAP estimate, but it does encourage interpretable solutions. See Section 13.8 TODO.
- **Choosing an informative rotation matrix** There are a variety of heuristic methods that try to find rotation matrices $\boldsymbol{R}$ which can be used to modify $\boldsymbol{W}$ (and hence the latent factors) so as to try to increase the interpretability, typically by encouraging them to be (approximately) sparse. One popular method is known as **varimax** (Kaiser 1958).
- **Use of non-Gaussian priors for the latent factors** In Section 12.6, we will dicuss how replacing $p(\boldsymbol{z}_i)$ with a non-Gaussian distribution can enable us to sometimes uniquely identify $\boldsymbol{W}$ as well as the latent factors. This technique is known as ICA.

12.1.4 Mixtures of factor analysers

The FA model assumes that the data lives on a low dimensional linear manifold. In reality, most data is better modeled by some form of low dimensional *curved* manifold. We can approximate a curved manifold by a piecewise linear manifold. This suggests the following model: let the k'th linear subspace of dimensionality L_k be represented by $\boldsymbol{W}_k$, for $k = 1 : K$. Suppose we have a latent indicator $qi \in \{1, \cdots, K\}$ specifying which subspace we should use to generate the data. We then sample $\boldsymbol{z}_i$ from a Gaussian prior and pass it through the $\boldsymbol{W}_k$ matrix (where $k = q_i$), and add noise. More precisely, the model is as follows:

$$p(q_i|\boldsymbol{\theta}) = \text{Cat}(q_i\boldsymbol{\pi}) \tag{12.13}$$

$$p(\boldsymbol{z}_i|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{12.14}$$

$$p(\boldsymbol{x}_i|q_i = k, \boldsymbol{z}_i, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{W}\boldsymbol{z}_i + \boldsymbol{\mu}_k, \boldsymbol{\Psi}) \tag{12.15}$$

[22] To see this, note that there are $L-1$ free parameters in $\boldsymbol{R}$ in the first column (since the column vector must be normalized to unit length), there are $L-2$ free parameters in the second column (which must be orthogonal to the first), and so on.

This is called a **mixture of factor analysers**(MFA) (Hinton et al. 1997).

Another way to think about this model is as a low-rank version of a mixture of Gaussians. In particular, this model needs $O(KLD)$ parameters instead of the $O(KD^2)$ parameters needed for a mixture of full covariance Gaussians. This can reduce overfitting. In fact, MFA is a good generic density model for high-dimensional real-valued data.

12.1.5 EM for factor analysis models

Below we state the results without proof. The derivation can be found in (Ghahramani and Hinton 1996a). To obtain the results for a single factor analyser, just set $r_{ic} = 1$ and $c = 1$ in the equations below. In Section 12.2.4 we will see a further simplification of these equations that arises when fitting a PPCA model, where the results will turn out to have a particularly simple and elegant interpretation.

In the E-step, we compute the posterior responsibility of cluster k for data point i using

$$r_{ik} \triangleq p(q_i = k|\boldsymbol{x}_i, \boldsymbol{\theta}) \propto \pi_k \mathcal{N}(\boldsymbol{x}_i|\boldsymbol{\mu}_k, \boldsymbol{W}_k \boldsymbol{W}_k^T \boldsymbol{\Psi}_k) \quad (12.16)$$

The conditional posterior for $\boldsymbol{z}_i$ is given by

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i, q_i = k, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (12.17)$$

$$\boldsymbol{\Sigma}_{ik} \triangleq (\boldsymbol{I} + \boldsymbol{W}_k^T \boldsymbol{\Psi}_k^{-1} \boldsymbol{W})_k^{-1} \quad (12.18)$$

$$\boldsymbol{\mu}_{ik} \triangleq \boldsymbol{\Sigma}_{ik} \boldsymbol{W}_k^T \boldsymbol{\Psi}_k^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) \quad (12.19)$$

In the M step, it is easiest to estimate $\boldsymbol{\mu}_k$ and $\boldsymbol{W}_k$ at the same time, by defining $\tilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k, \boldsymbol{\mu}_k)$, $\tilde{\boldsymbol{z}} = (\boldsymbol{z}, 1)$, also, define

$$\tilde{\boldsymbol{W}}_k = (\boldsymbol{W}_k, \boldsymbol{\mu}_k) \quad (12.20)$$

$$\tilde{\boldsymbol{z}} = (\boldsymbol{z}, 1) \quad (12.21)$$

$$\boldsymbol{b}_{ik} \triangleq \mathbb{E}[\tilde{\boldsymbol{z}}|\boldsymbol{x}_i, q_i = k] = \mathbb{E}[(\boldsymbol{\mu}_{ik}; 1)] \quad (12.22)$$

$$\boldsymbol{C}_{ik} \triangleq \mathbb{E}[\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}^T|\boldsymbol{x}_i, q_i = k] \quad (12.23)$$

$$= \begin{pmatrix} \mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T|\boldsymbol{x}_i, q_i = k] & \mathbb{E}[\boldsymbol{z}|\boldsymbol{x}_i, q_i = k] \\ \mathbb{E}[\boldsymbol{z}|\boldsymbol{x}_i, q_i = k]^T & 1 \end{pmatrix} \quad (12.24)$$

Then the M step is as follows:

$$\hat{\pi}_k = \frac{1}{N} \sum_{i=1}^N r_{ik} \quad (12.25)$$

$$\hat{\tilde{\boldsymbol{W}}}_k = \left(\sum_{i=1}^N r_{ik} \boldsymbol{x}_i \boldsymbol{b}_{ik}^T \right) \left(\sum_{i=1}^N r_{ik} \boldsymbol{x}_i \boldsymbol{C}_{ik}^T \right)^{-1} \quad (12.26)$$

$$\hat{\boldsymbol{\Psi}} = \frac{1}{N} \text{diag} \left[\sum_{i=1}^N r_{ik} (\boldsymbol{x}_i - \hat{\tilde{\boldsymbol{W}}}_{ik} \boldsymbol{b}_{ik}) \boldsymbol{x}_i^T \right] \quad (12.27)$$

Note that these updates are for vanilla EM. A much faster version of this algorithm, based on ECM, is described in (Zhao and Yu 2008).

12.1.6 Fitting FA models with missing data

In many applications, such as collaborative filtering, we have missing data. One virtue of the EM approach to fitting an FA/PPCA model is that it is easy to extend to this case. However, overfitting can be a problem if there is a lot of missing data. Consequently it is important to perform MAP estimation or to use Bayesian inference. See e.g., (Ilin and Raiko 2010) for details.

12.2 Principal components analysis (PCA)

Consider the FA model where we constrain $\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}$, and $\boldsymbol{W}$ to be orthonormal. It can be shown (Tipping and Bishop 1999) that, as $\sigma^2 \to 0$, this model reduces to classical (nonprobabilistic) **principal components analysis**(PCA), also known as the Karhunen Loeve transform. The version where $\sigma^2 > 0$ is known as **probabilistic PCA**(**PPCA**) (Tipping and Bishop 1999), orsensible PCA(Roweis 1997).

12.2.1 Classical PCA

12.2.1.1 Statement of the theorem

The synthesis viewof classical PCA is summarized in the forllowing theorem.

Theorem 12.1. *Suppose we want to find an orthogonal set of L linear basis vectors $\boldsymbol{w}_j \in \mathbb{R}^D$, and the corresponding scores $\boldsymbol{z}_i \in \mathbb{R}^L$, such that we minimize the average* ***reconstruction error***

$$J(\boldsymbol{W}, \boldsymbol{Z}) = \frac{1}{N} \sum_{i=1}^N \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 \quad (12.28)$$

where $\hat{\boldsymbol{x}}_i = \boldsymbol{W}\boldsymbol{z}_i$, subject to the constraint that $\boldsymbol{W}$ is orthonormal. Equivalently, we can write this objective as follows

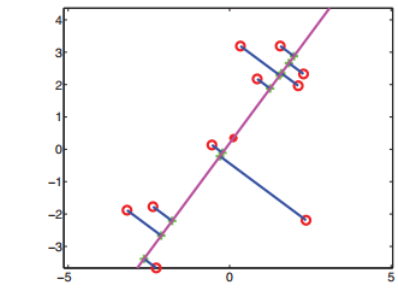
$$J(\boldsymbol{W}, \boldsymbol{Z}) = \frac{1}{N} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z}^T\|^2 \quad (12.29)$$

where $\boldsymbol{Z}$ is an $N \times L$ matrix with the $\boldsymbol{z}_i$ in its rows, and $\|A\|_F$ is the ***Frobenius norm*** *of matrix $\boldsymbol{A}$, defined by*

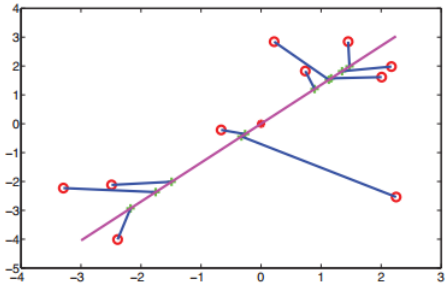
$$\|A\|_F \triangleq \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} a_{ij}^2} = \sqrt{\text{tr}(\boldsymbol{A}^T\boldsymbol{A})} \tag{12.30}$$

The optimal solution is obtained by setting $\hat{\boldsymbol{W}} = \boldsymbol{V}_L$, where $\boldsymbol{V}_L$ contains the L eigenvectors with largest eigenvalues of the empirical covariance matrix, $\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T$. (We assume the $\boldsymbol{x}_i$ have zero mean, for notational simplicity.) Furthermore, the optimal low-dimensional encoding of the data is given by $\hat{\boldsymbol{z}}_i = \boldsymbol{W}^T\boldsymbol{x}_i$, which is an orthogonal projection of the data onto the column space spanned by the eigenvectors.

An example of this is shown in Figure 12.1(a) for $D = 2$ and $L = 1$. The diagonal line is the vector $\boldsymbol{w}_1$; this is called the first principal component or principal direction. The data points $\boldsymbol{x}_i \in \mathbb{R}^2$ are orthogonally projected onto this line to get $\boldsymbol{z}_i \in \mathbb{R}$. This is the best 1-dimensional approximation to the data. (We will discuss Figure 12.1(b) later.)



(a)



(b)

Fig. 12.1: An illustration of PCA and PPCA where $D = 2$ and $L = 1$. Circles are the original data points, crosses are the reconstructions. The red star is the data mean. (a) PCA. The points are orthogonally projected onto the line. (b) PPCA. The projection is no longer orthogonal: the reconstructions are shrunk towards the data mean (red star).

The principal directions are the ones along which the data shows maximal variance. This means that PCA can be misled by directions in which the variance is high merely because of the measurement scale. It is therefore standard practice to standardize the data first, or equivalently, to work with correlation matrices instead of covariance matrices.

12.2.1.2 Proof *

See Section 12.2.2 of MLAPP.

12.2.2 Singular value decomposition (SVD)

We have defined the solution to PCA in terms of eigenvectors of the covariance matrix. However, there is another way to obtain the solution, based on the **singular value decomposition**, or **SVD**. This basically generalizes the notion of eigenvectors from square matrices to any kind of matrix.

Theorem 12.2. *(SVD). Any matrix can be decomposed as follows*

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\boldsymbol{U}}_{N\times N}\underbrace{\boldsymbol{\Sigma}}_{N\times D}\underbrace{\boldsymbol{V}^T}_{D\times D} \tag{12.31}$$

where $\boldsymbol{U}$ is an $N \times N$ matrix whose columns are orthornormal(so $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$), $\boldsymbol{V}$ is $D \times D$ matrix whose rows and columns are orthonormal (so $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}_D$), and $\boldsymbol{\Sigma}$ is a $N \times D$ matrix containing the $r = \min(N,D)$ singular values $\sigma_i \geq 0$ on the main diagonal, with 0s filling the rest of the matrix.

This shows how to decompose the matrix X into the product of three matrices: $\boldsymbol{V}$ describes an orthonormal basis in the domain, and $\boldsymbol{U}$ describes an orthonormal basis in the co-domain, and $\boldsymbol{\Sigma}$ describes how much the vectors in $\boldsymbol{V}$ are stretched to give the vectors in $\boldsymbol{U}$.

Since there are at most D singular values (assuming $N > D$), the last ND columns of $\boldsymbol{U}$ are irrelevant, since they will be multiplied by 0. The **economy sized SVD**, or **thin SVD**, avoids computing these unnecessary elements. Let us denote this decomposition by $\hat{\boldsymbol{U}}\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{V}}^T$.If $N > D$, we have

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\hat{\boldsymbol{U}}}_{N\times D}\underbrace{\hat{\boldsymbol{\Sigma}}}_{D\times D}\underbrace{\hat{\boldsymbol{V}}^T}_{D\times D} \tag{12.32}$$

as in Figure 12.2(a). If $N < D$, we have

$$\underbrace{\boldsymbol{X}}_{N\times D} = \underbrace{\hat{\boldsymbol{U}}}_{N\times N}\underbrace{\hat{\boldsymbol{\Sigma}}}_{N\times N}\underbrace{\hat{\boldsymbol{V}}^T}_{N\times D} \tag{12.33}$$

Computing the economy-sized SVD takes $O(ND\min(N,D))$ time (Golub and van Loan 1996, p254).

The connection between eigenvectors and singular vectors is the following:

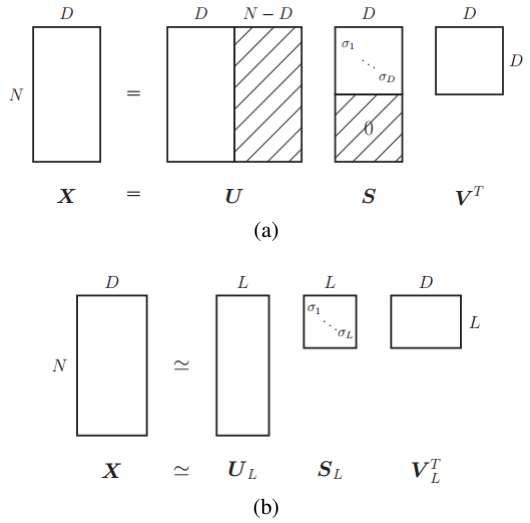


Fig. 12.2: (a) SVD decomposition of non-square matrices $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The shaded parts of $\boldsymbol{\Sigma}$, and all the off-diagonal terms, are zero. The shaded entries in $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$ are not computed in the economy-sized version, since they are not needed. (b) Truncated SVD approximation of rank L.

$$\boldsymbol{U} = \text{evec}(\boldsymbol{X}\boldsymbol{X}^T) \tag{12.34}$$
$$\boldsymbol{V} = \text{evec}(\boldsymbol{X}^T\boldsymbol{X}) \tag{12.35}$$
$$\boldsymbol{\Sigma}^2 = \text{eval}(\boldsymbol{X}\boldsymbol{X}^T) = \text{eval}(\boldsymbol{X}^T\boldsymbol{X}) \tag{12.36}$$

For the proof please read Section 12.2.3 of MLAPP.

Since the eigenvectors are unaffected by linear scaling of a matrix, we see that the right singular vectors of $\boldsymbol{X}$ are equal to the eigenvectors of the empirical covariance $\hat{\boldsymbol{\Sigma}}$. Furthermore, the eigenvalues of $\hat{\boldsymbol{\Sigma}}$ are a scaled version of the squared singular values.

However, the connection between PCA and SVD goes deeper. From Equation 12.31, we can represent a rank r matrix as follows:

$$\boldsymbol{X} = \sigma_1 \begin{pmatrix} | \\ \boldsymbol{u}_1 \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1 & - \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} | \\ \boldsymbol{u}_r \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_r^T & - \end{pmatrix}$$

If the singular values die off quickly, we can produce a rank L approximation to the matrix as follows:

$$\begin{aligned} \boldsymbol{X} &\approx \sigma_1 \begin{pmatrix} | \\ \boldsymbol{u}_1 \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1 & - \end{pmatrix} + \cdots + \sigma_r \begin{pmatrix} | \\ \boldsymbol{u}_L \\ | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_L^T & - \end{pmatrix} \\ &= \boldsymbol{U}_{:,1:L}\boldsymbol{\Sigma}_{1:L,1:L}\boldsymbol{V}_{:,1:L}^T \end{aligned} \tag{12.37}$$

This is called a **truncated SVD** (see Figure 12.2(b)).

One can show that the error in this approximation is given by

$$\|\boldsymbol{X} - \boldsymbol{X}_L\|_F \approx \sigma_L \tag{12.38}$$

Furthermore, one can show that the SVD offers the best rank L approximation to a matrix (best in the sense of minimizing the above Frobenius norm).

Let us connect this back to PCA. Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be a truncated SVD of $\boldsymbol{X}$. We know that $\hat{\boldsymbol{W}} = \boldsymbol{V}$, and that $\hat{\boldsymbol{Z}} = \boldsymbol{X}\hat{\boldsymbol{W}}$, so

$$\hat{\boldsymbol{Z}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma} \tag{12.39}$$

Furthermore, the optimal reconstruction is given by $\hat{\boldsymbol{X}} = \boldsymbol{Z}\hat{\boldsymbol{W}}$,so we find

$$\hat{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{12.40}$$

This is precisely the same as a truncated SVD approximation! This is another illustration of the fact that PCA is the best low rank approximation to the data.

12.2.3 Probabilistic PCA

Theorem 12.3. *((**Tipping and Bishop 1999**)). Consider a factor analysis model in which $\boldsymbol{\Psi} = \sigma^2\boldsymbol{I}$ and $\boldsymbol{W}$ is orthogonal. The observed data log likelihood is given by*

$$\begin{aligned} \log p(\boldsymbol{X}|\boldsymbol{W}, \sigma^2\boldsymbol{I}) &= -\frac{N}{2}\ln|\boldsymbol{C}| - \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{x}_i^T\boldsymbol{C}^{-1}\boldsymbol{x}_i \\ &= -\frac{N}{2}\ln|\boldsymbol{C}| + \text{tr}(\boldsymbol{C}^{-1}\boldsymbol{\Sigma}) \end{aligned} \tag{12.41}$$

where $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$ and $\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T$. (We are assuming centred data, for notational simplicity.) The maxima of the log-likelihood are given by

$$\hat{\boldsymbol{W}} = \boldsymbol{V}(\boldsymbol{\Lambda} - \sigma^2\boldsymbol{I})^{\frac{1}{2}}\boldsymbol{R} \tag{12.42}$$

where $\boldsymbol{R}$ is an arbitrary $L \times L$ orthogonal matrix, $\boldsymbol{V}$ is the $D \times L$ matrix whose columns are the first L eigenvectors of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Lambda}$ is the corresponding diagonal matrix of eigenvalues. Without loss of generality, we can set $\boldsymbol{R} = \boldsymbol{I}$. Furthermore, the MLE of the noise variance is given by

$$\hat{\sigma}^2 = \frac{1}{D-L}\sum_{j=L+1}^{D}\lambda_j \tag{12.43}$$

which is the average variance associated with the discarded dimensions.

Thus, as $\sigma^2 \to 0$, we have $\hat{\boldsymbol{W}} \to \boldsymbol{V}$, as in classical PCA. What about $\hat{\boldsymbol{Z}}$? It is easy to see that the posterior over the latent factors is given by

$$p(\boldsymbol{z}_i|\boldsymbol{x}_i,\hat{\boldsymbol{\theta}}) = \mathcal{N}(\boldsymbol{z}_i|\hat{\boldsymbol{F}}^{-1}\hat{\boldsymbol{W}}^T\boldsymbol{x}_i, \sigma^2\hat{\boldsymbol{F}}^{-1}) \quad (12.44)$$

$$\hat{\boldsymbol{F}} \triangleq \hat{\boldsymbol{W}}^T\hat{\boldsymbol{W}} + \sigma^2\boldsymbol{I} \quad (12.45)$$

(Do not confuse $\boldsymbol{F} = \boldsymbol{W}^T\boldsymbol{W} + \sigma^2\boldsymbol{I}$ with $\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I}$.) Hence, as $\sigma^2 \to 0$, we find $\hat{\boldsymbol{W}} \to \boldsymbol{V}$, $\hat{\boldsymbol{F}} \to \boldsymbol{I}$ and $\boldsymbol{z}_i \to \boldsymbol{V}^T\boldsymbol{x}_i$. Thus the posterior mean is obtained by an orthogonal projection of the data onto the column space of $\boldsymbol{V}$, as in classical PCA.

Note, however, that if $\sigma^2 \to 0$, the posterior mean is not an orthogonal projection, since it is shrunk somewhat towards the prior mean, as illustrated in Figure 12.1(b). This sounds like an undesirable property, but it means that the reconstructions will be closer to the overall data mean, $\hat{\boldsymbol{\mu}} = \bar{\boldsymbol{x}}$.

12.2.4 EM algorithm for PCA

Although the usual way to fit a PCA model uses eigenvector methods, or the SVD, we can also use EM, which will turn out to have some advantages that we discuss below. EM for PCA relies on the probabilistic formulation of PCA. However the algorithm continues to work in the zero noise limit, $\sigma^2 = 0$, as shown by (Roweis 1997).

Let $\tilde{\boldsymbol{Z}}$ be a $L \times N$ matrix storing the posterior means (low-dimensional representations) along its columns. Similarly, let $\tilde{\boldsymbol{X}} = \boldsymbol{X}^T$ store the original data along its columns. From Equation 12.44, when $\sigma^2 = 0$, we have

$$\tilde{\boldsymbol{Z}} = (\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\tilde{\boldsymbol{X}} \quad (12.46)$$

This constitutes the E step. Notice that this is just an orthogonal projection of the data.

From Equation 12.26, the M step is given by

$$\hat{\boldsymbol{W}} = \left(\sum_{i=1}^N \boldsymbol{x}_i\mathbb{E}[\boldsymbol{z}_i]^T\right)\left(\sum_{i=1}^N \mathbb{E}[\boldsymbol{z}_i]\mathbb{E}[\boldsymbol{z}_i]^T\right)^{-1} \quad (12.47)$$

where we exploited the fact that $\boldsymbol{\Sigma} = \text{cov}[\boldsymbol{z}_i|\boldsymbol{x}_i,\boldsymbol{\theta}] = \boldsymbol{0}$ when $\sigma^2 = 0$.

(Tipping and Bishop 1999) showed that the only stable fixed point of the EM algorithm is the globally optimal solution. That is, the EM algorithm converges to a solution where $\boldsymbol{W}$ spans the same linear subspace as that defined by the first L eigenvectors. However, if we want $\boldsymbol{W}$ to be orthogonal, and to contain the eigenvectors in descending order of eigenvalue, we have to orthogonalize the resulting matrix (which can be done quite cheaply). Alternatively, we can modify EM to give the principal basis directly (Ahn and Oh 2003).

This algorithm has a simple physical analogy in the case $D = 2$ and $L = 1$(Roweis 1997). Consider some points in $\mathbb{R}^2$ attached by springs to a rigid rod, whose orientation is defined by a vector $\boldsymbol{w}$. Let $\boldsymbol{z}_i$ be the location where the i'th spring attaches to the rod. See Figure 12.11 of MLAPP for an illustration.

Apart from this pleasing intuitive interpretation, EM for PCA has the following advantages over eigenvector methods:

- EM can be faster. In particular, assuming $N, D \gg L$, the dominant cost of EM is the projection operation in the E step, so the overall time is $O(TLND)$, where T is the number of iterations. This is much faster than the $O(\min(ND^2, DN^2))$ time required by straightforward eigenvector methods, although more sophisticated eigenvector methods, such as the Lanczos algorithm, have running times comparable to EM.
- EM can be implemented in an online fashion, i.e., we can update our estimate of $\boldsymbol{W}$ as the data streams in.
- EM can handle missing data in a simple way (see Section 12.1.6).
- EM can be extended to handle mixtures of PPCA/ FA models.
- EM can be modified to variational EM or to variational Bayes EM to fit more complex models.

12.3 Choosing the number of latent dimensions

In Section 11.5, we discussed how to choose the number of components K in a mixture model. In this section, we discuss how to choose the number of latent dimensions L in a FA/PCA model.

12.3.1 Model selection for FA/PPCA

TODO

12.3.2 Model selection for PCA

TODO

12.4 PCA for categorical data

In this section, we consider extending the factor analysis model to the case where the observed data is categorical rather than real-valued. That is, the data has the form

$y_{ij} \in \{1,...,C\}$, where $j = 1 : R$ is the number of observed response variables. We assume each y_{ij} is generated from a latent variable $\boldsymbol{z}_i \in \mathbb{R}^L$, with a Gaussian prior, which is passed through the softmax function as follows:

$$p(\boldsymbol{z}_i) = \mathcal{N}(\boldsymbol{z}_i|\boldsymbol{0}, \boldsymbol{I}) \tag{12.48}$$

$$p(\boldsymbol{y}_i|\boldsymbol{z}_i, \boldsymbol{\theta}) = \prod_{j=1}^{R} \text{Cat}(y_{ir}|\mathcal{S}(\boldsymbol{W}_r^T \boldsymbol{z}_i + \boldsymbol{w}_{0r})) \tag{12.49}$$

where $\boldsymbol{W}_r \in \mathbb{R}^L$ is the factor loading matrix for response j, and $\boldsymbol{W}_{0r} \in \mathbb{R}^M$ is the offset term for response r, and $\boldsymbol{\theta} = (\boldsymbol{W}_r, \boldsymbol{W}_{0r})_{r=1}^R$. (We need an explicit offset term, since clamping one element of $\boldsymbol{z}_i$ to 1 can cause problems when computing the posterior covariance.) As in factor analysis, we have defined the prior mean to be $\boldsymbol{\mu}_0 = \boldsymbol{0}$ and the prior covariance $\boldsymbol{V}_0 = \boldsymbol{I}$, since we can capture non-zero mean by changing $\boldsymbol{w}_{0j}$ and non-identity covariance by changing $\boldsymbol{W}_r$. We will call this categorical PCA. See Chapter 27 TODO for a discussion of related models.

In (Khan et al. 2010), we show that this model outperforms finite mixture models on the task of imputing missing entries in design matrices consisting of real and categorical data. This is useful for analysing social science survey data, which often has missing data and variables of mixed type.

12.5 PCA for paired and multi-view data

12.5.1 Supervised PCA (latent factor regression)

12.5.2 Discriminative supervised PCA

12.5.3 Canonical correlation analysis

12.6 Independent Component Analysis (ICA)

Let $\boldsymbol{x}_t \in \mathbb{R}^D$ be the observed signal at the sensors at time t, and $\boldsymbol{z}_t \in \mathbb{R}^L$ be the vector of source signals. We assume that

$$\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{z}_t + \boldsymbol{\epsilon}_t \tag{12.50}$$

where $\boldsymbol{W}$ is an $D \times L$ matrix, and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$. In this section, we treat each time point as an independent observation, i.e., we do not model temporal correlation (so we could replace the t index with i, but we stick with t to be consistent with much of the ICA literature). The goal is to infer the source signals, $p(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{\theta})$. In this context, $\boldsymbol{W}$ is called the **mixing matrix**. If $L = D$ (number of sources = number of sensors), it will be a square matrix. Often we will assume the noise level, $|\boldsymbol{\Psi}|$, is zero, for simplicity.

So far, the model is identical to factor analysis. However, we will use a different prior for $p(\boldsymbol{z}_t)$. In PCA, we assume each source is independent, and has a Gaussian distribution. We will now relax this Gaussian assumption and let the source distributions be any *non-Gaussian* distribution

$$p(\boldsymbol{z}_t) = \prod_{j=1}^{L} p_j(z_{tj}) \tag{12.51}$$

Without loss of generality, we can constrain the variance of the source distributions to be 1, because any other variance can be modelled by scaling the rows of $\boldsymbol{W}$ appropriately. The resulting model is known as **independent component analysis** or **ICA**.

The reason the Gaussian distribution is disallowed as a source prior in ICA is that it does not permit unique recovery of the sources. This is because the PCA likelihood is invariant to any orthogonal transformation of the sources $\boldsymbol{z}_t$ and mixing matrix $\boldsymbol{W}$. PCA can recover the best linear subspace in which the signals lie, but cannot uniquely recover the signals themselves.

ICA requires that $\boldsymbol{W}$ is square and hence invertible. In the non-square case (e.g., where we have more sources than sensors), we cannot uniquely recover the true signal, but we can compute the posterior $p(\boldsymbol{z}_t|\boldsymbol{x}_t, \hat{\boldsymbol{W}})$, which represents our beliefs about the source. In both cases, we need to estimate Was well as the source distributions p_j. We discuss how to do this below.

12.6.1 Maximum likelihood estimation

In this section, we discuss ways to estimate square mixing matrices $\boldsymbol{W}$ for the noise-free ICA model. As usual, we will assume that the observations have been centered; hence we can also assume $\boldsymbol{z}$ is zero-mean. In addition, we assume the observations have been whitened, which can be done with PCA.

If the data is centered and whitened, we have $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \boldsymbol{I}$. But in the noise free case, we also have

$$\text{cov}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \boldsymbol{W}\mathbb{E}[\boldsymbol{z}\boldsymbol{z}^T]\boldsymbol{W}^T \tag{12.52}$$

Hence we see that $\boldsymbol{W}$ must be orthogonal. This reduces the number of parameters we have to estimate from D^2 to $D(D-1)/2$. It will also simplify the math and the algorithms.

Let $\boldsymbol{V} = \boldsymbol{W}^{-1}$; these are often called the recognition weights, as opposed to $\boldsymbol{W}$, which are the generative weights.

Since $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z}$, we have, from Equation 2.46,

$$p_x(\boldsymbol{W}\boldsymbol{z}_t) = p_z(\boldsymbol{z}_t)|\det(\boldsymbol{W}^{-1})| \\ = p_z(\boldsymbol{V}\boldsymbol{x}_t)|\det(\boldsymbol{V})| \quad (12.53)$$

Hence we can write the log-likelihood, assuming T iid samples, as follows:

$$\frac{1}{T}\log p(\mathcal{D}|\boldsymbol{V}) = \log|\det(\boldsymbol{V})| + \frac{1}{T}\sum_{j=1}^{L}\sum_{t=1}^{T}\log p_j(\boldsymbol{v}_j^T\boldsymbol{x}_t)$$

where $\boldsymbol{v}_j$ is the j'th row of $\boldsymbol{V}$. Since we are constraining $\boldsymbol{V}$ to be orthogonal, the first term is a constant, so we can drop it. We can also replace the average over the data with an expectation operator to get the following objective

$$\text{NLL}(\boldsymbol{V}) = \sum_{j=1}^{L}\mathbb{E}[G_j(\boldsymbol{z}_j)] \quad (12.54)$$

where $\boldsymbol{z}_j = \boldsymbol{v}_j^T\boldsymbol{x}$ and $G_j(\boldsymbol{z}) \triangleq -\log p_j(\boldsymbol{z})$. We want to minimize this subject to the constraint that the rows of $\boldsymbol{V}$ are orthogonal. We also want them to be unit norm, since this ensures that the variance of the factors is unity (since, with whitened data, $\mathbb{E}[\boldsymbol{v}_j^T\boldsymbol{x}] = \|\boldsymbol{v}_j\|^2$, which is necessary to fix the scale of the weights. In otherwords, $\boldsymbol{V}$ should be an orthonormal matrix.

It is straightforward to derive a gradient descent algorithm to fit this model; however, it is rather slow. One can also derive a faster algorithm that follows the natural gradient; see e.g., (MacKay 2003, ch 34) for details. A popular alternative is to use an approximate Newton method, which we discuss in Section 12.6.2. Another approach is to use EM, which we discuss in Section 12.6.3.

12.6.2 The FastICA algorithm

12.6.3 Using EM

12.6.4 Other estimation principles *