

## ELEC 548 Dimensionality Reduction

*Dimensionality reduction* is a very useful tool for visualizing and understanding data that are in a high dimensional space. If we assume that most high dimensional data actually exist primarily in a lower dimensional space, the extra dimensions are a form of noise. Thus, many other machine learning problems, such as clustering, would be more robustly performed in the lower dimensional space. Dimensionality reduction tools allow us to estimate the connection between higher dimensional data and lower dimensional representations. A good reference for this topic is Chapter 12 of Bishop's *Pattern Recognition and Machine Learning*.

### A. Principal Components Analysis

Data set:  $\mathbf{x}_n \in \mathbb{R}^D, n = 1, \dots, N$

Goal: Project data into a space with dimensionality  $M < D$  while maximizing the variance of the projected data.

*Intuition: Why do we want to maximize the variance?* Imagine the corner case that in one dimension of our data  $i$ , there is no variability at all (the  $\mathbf{x}_n^{(i)}$  are equal for all  $n$ ). Then, that dimension is not particularly useful and could be safely ignored as we look for interesting features. But in the new  $D - 1$ -dimensional data, the variance would be the same as in the original data set (i.e., maximized for  $M = D - 1$ ).

Principal Components Analysis starts with the **sample covariance matrix,  $\mathbf{S}$** .

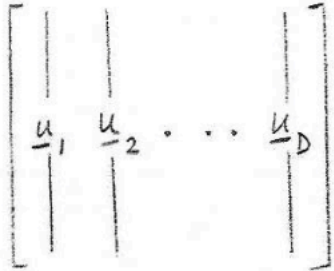
$$\mathbf{S} \equiv \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T, \text{ where } \boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$$

#### A.1 Diagonalization / "Eigendecomposition"

Any covariance matrix (symmetric, positive semidefinite) can be expressed as

$$\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T,$$

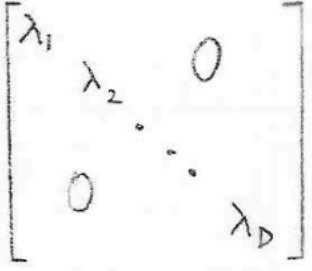
where the columns of  $\mathbf{U}$  are orthonormal and  $\boldsymbol{\Lambda}$  is diagonal.



$U$   
( $D \times D$  matrix)

$u_i$  is  $i^{\text{th}}$  eigenvector

$$u_i^T u_j = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{else} \end{cases}$$



$\Lambda$   
( $D \times D$  matrix)

$\lambda_i$  is  $i^{\text{th}}$  eigenvalue

$$\lambda_1 > \lambda_2 > \dots > \lambda_D \geq 0$$

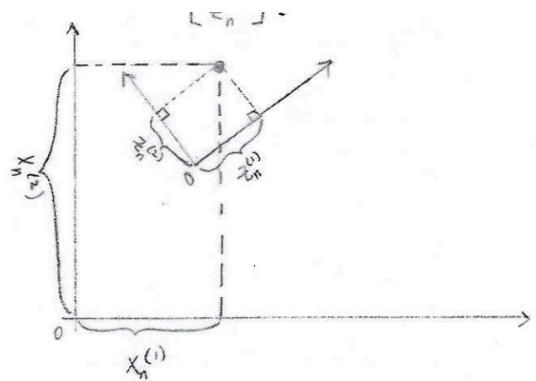
Note that  $u_1, \dots, u_D$  form an orthonormal basis for  $\mathbb{R}^D$ . In other words, any point in  $\mathbb{R}^D$  can be expressed as a linear combination of the  $u_1, \dots, u_D$ .

### A.2 Principal Component directions

- The first PC direction,  $u_1$ , captures the greatest data variance.
- The second PC direction ( $u_2$ ) captures the second greatest data variance and is orthogonal to  $u_1$ .
- And so on...

Another way of saying that the PC directions are a basis is that they define a new set of coordinate axes. This

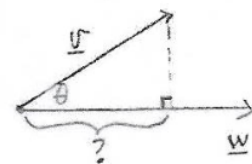
means that a data point  $x_n \in \mathbb{R}^2 = \begin{bmatrix} x_n^{(1)} \\ x_n^{(2)} \end{bmatrix}$  can equivalently be described in the new coordinate system as  $z_n = \begin{bmatrix} z_n^{(1)} \\ z_n^{(2)} \end{bmatrix}$ .



How do we relate  $x_n$  and  $z_n$ ?

In general, if we have two vectors,  $v$  and  $w$ , the projection of  $v$  onto  $w$  is

$$\|v\| \cos \theta = \frac{\|v\| \|w\| \cos \theta}{\|w\|} = \frac{v^T w}{\|w\|} \quad (1)$$

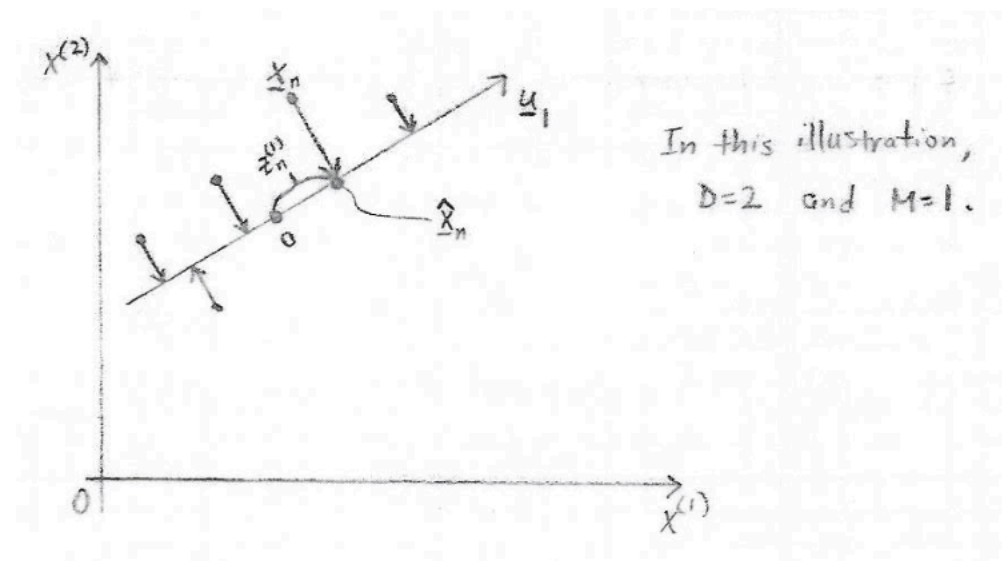


### A.3 Projecting into PCA dimensions

In order to project from our high-dimensional data space ( $\mathbf{x} \in \mathbb{R}^D$ ) into our lower dimensional PC space ( $\mathbf{z} \in \mathbb{R}^M$ )

$$z^{(i)} = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{u}_i, i = 1, \dots, M \quad (2)$$

The coordinates in PC space are also called “PC scores”. Intuitively, this means that we center the high-dimensional data, and then project onto the axes defined by the  $\mathbf{u}_i$ 's (remember that  $\|\mathbf{u}_i\| = 1$ , so the denominator in (1) is 1). This is illustrated for  $D = 2$  and  $M = 1$  below.



If we define  $\mathbf{U}_M$  as the first  $M$  columns of the eigenvector matrix  $\mathbf{U}$  above (i.e.,  $\mathbf{U}_M = [\mathbf{u}_1 \mathbf{u}_2, \dots, \mathbf{u}_M]$ ), then we can write the projection in vector form

$$\mathbf{z} = \mathbf{U}_M^T (\mathbf{x} - \boldsymbol{\mu}).$$

### A.4 “Back-projecting” reduced dimensional data

In low-dimensional coordinates, the projection of  $\mathbf{x}_n$  is  $\mathbf{z}_n$ . What does  $\mathbf{z}_n$  look like back in the original coordinate system?

$$\hat{\mathbf{x}}_n = \sum_{i=1}^M z_n^{(i)} \mathbf{u}_i + \boldsymbol{\mu} = \mathbf{U}_M \mathbf{z} \quad (3)$$

Because we’ve initially projected  $\mathbf{x}$  into a low-dimensional space, we call estimating  $\hat{\mathbf{x}}_n$  “projecting back into the high-dimensional space.”

*Thought question:*

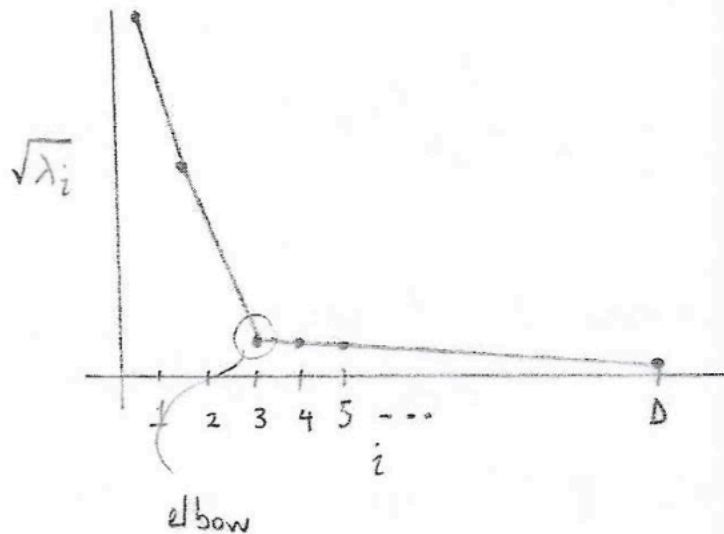
$$\text{What is } \sum_{i=1}^D z_n^{(i)} \mathbf{u}_i + \boldsymbol{\mu}?$$

*Answer:* It’s just  $\mathbf{x}_n$ !

### A.5 How to choose $M$ ?

So when we're doing PCA, how do we choose how many eigenvectors to include in our lower dimensional representation; what value of  $M < D$  should we pick?

Eigendecompositions always sort the eigenvectors by eigenvalue. So, if we plot the eigenvalue "spectrum" of  $\mathbf{S}$ , we can look for an "elbow".



The typical instruction is to choose  $M$  to be the number of eigenvalues above the elbow. So in the example above, one would choose  $M = 2$ . In many cases, one does not actually see a clear-cut elbow. This is one of the motivations of a probabilistic model-based approach to dimensionality reduction, such as probabilistic PCA (P-PCA) (which we'll introduce in the next section), where one can use cross-validated likelihoods to determine  $M$ .

With PCA, it can be shown that the fraction (percentage) of variance in the data which is explained by the first  $M$  eigenvectors (principal components) can be found from the eigenvalues

$$\text{Fraction of variance explained by } M < D \text{ eigenvectors} = \frac{\sum_{i=1}^M \lambda_i}{\sum_{i=1}^D \lambda_i}.$$

**Note:** We described PCA in terms of maximizing the variance of the projected data. Equivalently, we could have formulated PCA in terms of minimizing the projection error.

### A.6 Summary of PCA

1. Data set:  $\mathbf{x}_b \in \mathbb{R}^D, n = 1, \dots, N$
2. Find the sample covariance,  $\mathbf{S}$ , and mean,  $\boldsymbol{\mu}$ :

$$\mathbf{S} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T$$

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}$$

### 3. Diagonalize S

$$\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$$

where  $\mathbf{\Lambda}$  is a diagonal matrix with ordered eigenvalues on the diagonal, and  $\mathbf{U}$  contains the eigenvectors

$$\begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_D \\ | & | & & | \end{bmatrix}$$

### 4. Choose M: Choose the number of reduced dimensions $M < D$ . Define

$$\mathbf{U}_M = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_M \\ | & | & & | \end{bmatrix}$$

### 5. PC directions are the columns of $\mathbf{U}_M$ .

### 6. PC scores:

$$\mathbf{z}_n = \mathbf{U}_M^T(\mathbf{x}_n - \boldsymbol{\mu}), \quad \mathbf{z}_n \in \mathbb{R}^M$$

The “PC score” for a data point  $\mathbf{x}_n$  is its low-dimensional projection,  $\mathbf{z}_n$ .

### 7. Back-projection: The low-dimensional point can be projected back into the data space:

$$\begin{aligned} \hat{\mathbf{x}}_n &= \mathbf{U}_M \mathbf{z}_n + \boldsymbol{\mu} \\ &= \mathbf{U}_M \mathbf{U}_M^T (\mathbf{x}_n - \boldsymbol{\mu}) + \boldsymbol{\mu} \end{aligned}$$

**Note:** The PC directions are only unique up to a sign difference. In other words, the  $i^{\text{th}}$  PC direction can be  $\mathbf{u}_i$  or  $-\mathbf{u}_i$ . This will determine the sign of the  $i^{\text{th}}$  PC score (i.e.,  $z^{(i)}$ ).

## B. Probabilistic PCA

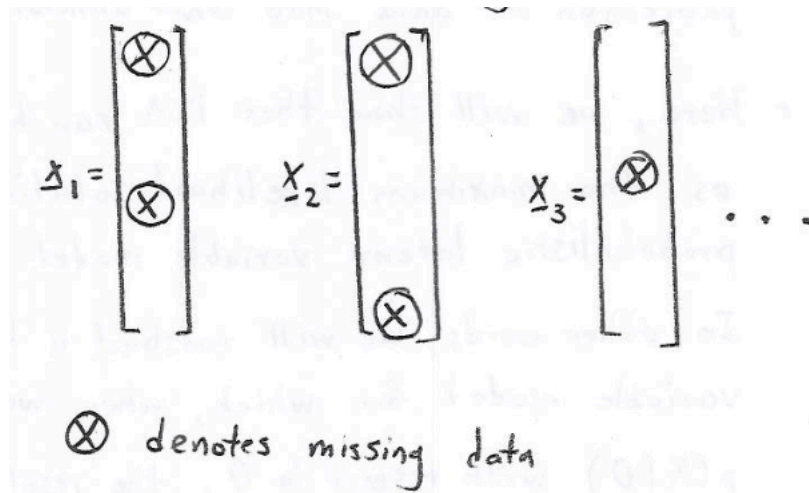
In the last section we described traditional principal components analysis as formulated as a linear projection into an orthogonal lower dimensional space which maximizes variance (or minimizes projection error). Next we will describe a latent variable model for dimensionality reduction. The maximum likelihood solution for this P-PCA model will end up yielding a nearly identical solution as PCA but with some key advantages.

### B.1 Advantages of P-PCA over conventional PCA

By virtue of being a probabilistic model, Probabilistic PCA has advantages over traditional PCA.

- P-PCA assigns probabilities to data, so we can compare different models – particularly with different values of the reduced number of dimensions,  $M$  – using cross-validated data likelihoods.
- P-PCA has an explicit noise model, so it can more effectively remove “noise” (i.e., variability not explained by variation in the low-dimensional space).

- If the data dimensionality,  $D$ , is large, calculation of the eigenvectors of the covariance matrix can be an  $O(D^3)$  operation. The EM algorithm solution for P-PCA gives a nice (and more simple) way to compute just the first  $M$  eigenvectors.
- Because it is a probabilistic model, with P-PCA one can learn the low-dimensional space when there is missing data. There is no principled way to modify the PCA algorithm in this condition.



- Because P-PCA is a probabilistic model, we can easily propose extensions like mixtures of P-PCA models (which would be analogous to mixtures of Gaussians).
- Like all probabilistic latent variable models, P-PCA is generative, so one can generate synthetic samples to examine.

## B.2 Generative model for P-PCA

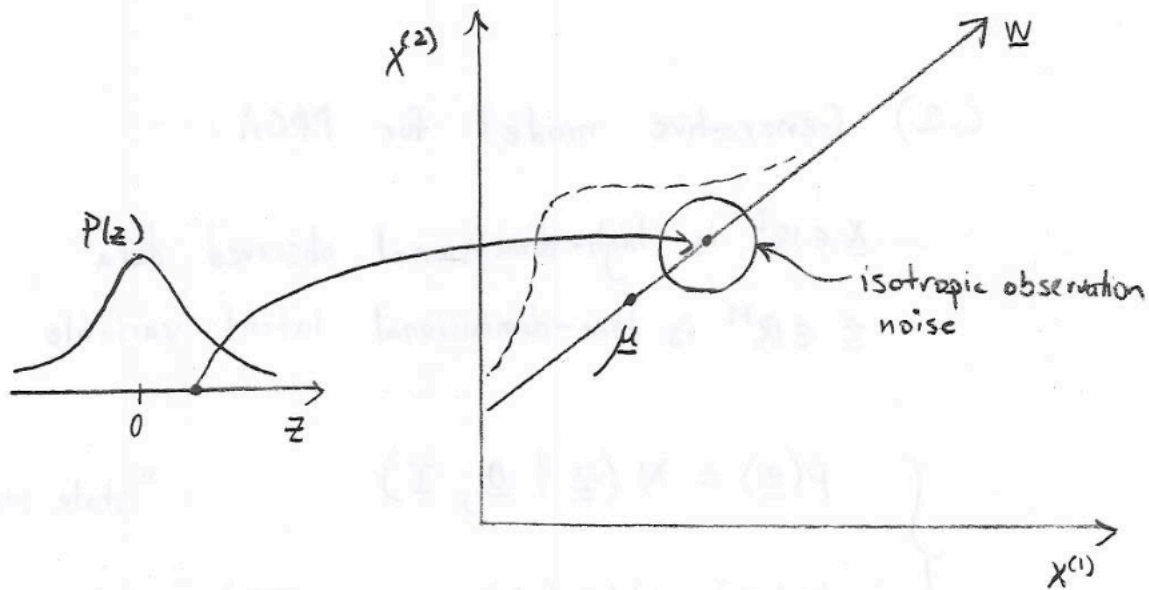
$\mathbf{x} \in \mathbb{R}^D$  is high dimensional observed data

$\mathbf{z} \in \mathbb{R}^M$  is a lower dimensional latent variable

$$\begin{aligned}
 \Pr(\mathbf{z}) &= \mathcal{N}(\mathbf{0}, \mathbf{I}) && \text{"state model"} \\
 \Pr(\mathbf{x} | \mathbf{z}) &= \mathcal{N}(\underbrace{\mathbf{W}\mathbf{z}}_{D \times M \text{ matrix}} + \underbrace{\boldsymbol{\mu}}_{\text{"observation noise"}}, \underbrace{\sigma^2 \mathbf{I}}_{\text{"observation noise"}}) && \text{"observation model"}
 \end{aligned} \tag{4}$$

*Looking ahead:* If we fit this model to data  $\mathbf{x}_1, \dots, \mathbf{x}_N$  – that is if we find the model parameters  $\theta = \{\mathbf{W}, \boldsymbol{\mu}, \sigma^2\}$  which maximize the data likelihood – the columns of  $\mathbf{W}$  will span the same space as the PCA principal components (i.e., the columns of  $\mathbf{U}_M$ ). In the limit of  $\sigma^2 \rightarrow 0$ , the low-dimensional projections of P-PCA approach those of PCA.

Here is a picture of the P-PCA model where  $M = 1$  and  $D = 2$ :



**B.3 P-PCA is a linear Gaussian model**

Since the latent variable model and the observation model have Gaussian variability and the observations ( $\mathbf{x}$ ) are linearly related to the latent variables ( $\mathbf{z}$ ) P-PCA falls into a very convenient model category – the “linear Gaussian model”. For these models all the marginal, conditional, and joint distributions are Gaussian!

- $\Pr(\mathbf{x}), \Pr(\mathbf{z})$  : Marginal distributions
- $\Pr(\mathbf{x} | \mathbf{z}), \Pr(\mathbf{z} | \mathbf{x})$  : Conditional distributions
- $\Pr(\mathbf{x}, \mathbf{z})$  : Joint distribution

Since these distributions are all Gaussian, we can specify them completely by their means and covariances. Our model specifies  $\Pr(\mathbf{z})$  and  $\Pr(\mathbf{x} | \mathbf{z})$ , so lets find the parameters of the other distributions.

(i).  $\Pr(\mathbf{x}, \mathbf{z})$

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{x} \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} E(\mathbf{z}) \\ E(\mathbf{x}) \end{bmatrix}, \begin{bmatrix} \text{Cov}(\mathbf{z}) & E(\mathbf{z}\mathbf{x}^T) - E(\mathbf{z})E(\mathbf{x})^T \\ E(\mathbf{x}\mathbf{z}^T) - E(\mathbf{x})E(\mathbf{z})^T & \text{Cov}(\mathbf{x}) \end{bmatrix} \right)$$

$$E(\mathbf{z}) = 0 \qquad \text{From model specification}$$

An equivalent way of describing our observations is

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \tag{5}$$

Note that since we have originally specified this as the conditional distribution of  $\mathbf{x}$  given  $\mathbf{z}$ , this implies that  $\boldsymbol{\epsilon}$  is independent of  $\mathbf{z}$ . Taking the expectation over both  $\mathbf{z}$  and  $\boldsymbol{\epsilon}$  (since we’re interested in the *joint* distribution), we find

$$E(\mathbf{x}) = E(\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon})$$

$$\begin{aligned}
 &= \mathbf{W} \mathbf{E}(\mathbf{z}) + \boldsymbol{\mu} + \mathbf{E}(\boldsymbol{\epsilon}) \\
 &= \boldsymbol{\mu}
 \end{aligned}$$

$$\text{Cov}(\mathbf{z}) = \mathbf{I}$$

From model specification

$$\begin{aligned}
 \text{Cov}(\mathbf{x}) &= \mathbf{E}(\mathbf{x}\mathbf{x}^T) - \mathbf{E}(\mathbf{x})\mathbf{E}(\mathbf{x})^T \\
 &= \mathbf{E}((\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon})(\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon})^T) - \boldsymbol{\mu}\boldsymbol{\mu}^T \\
 &= \mathbf{E}(\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T + \mathbf{W}\mathbf{z}\boldsymbol{\mu}^T + \mathbf{W}\mathbf{z}\boldsymbol{\epsilon}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\mu}\boldsymbol{\epsilon}^T + \boldsymbol{\epsilon}\mathbf{z}^T\mathbf{W}^T + \boldsymbol{\epsilon}\boldsymbol{\mu}^T + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) - \boldsymbol{\mu}\boldsymbol{\mu}^T \\
 &\quad \text{where we have cancelled all products of constants and zero mean variables} \\
 &= \mathbf{E}(\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T + \mathbf{W}\mathbf{z}\boldsymbol{\epsilon}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\epsilon}\mathbf{z}^T\mathbf{W}^T + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) - \boldsymbol{\mu}\boldsymbol{\mu}^T \quad \text{Because } \boldsymbol{\epsilon} \text{ and } \mathbf{z} \text{ are independent.} \\
 &= \mathbf{W}\mathbf{E}(\mathbf{z}\mathbf{z}^T)\mathbf{W}^T + \mathbf{E}(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) \\
 &= \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} \quad \text{From the definitions}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{E}(\mathbf{x}\mathbf{z}^T) - \mathbf{E}(\mathbf{x})\mathbf{E}(\mathbf{z})^T &= \mathbf{E}((\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon})\mathbf{z}^T) - \mathbf{0} \\
 &= \mathbf{W}\mathbf{E}(\mathbf{z}\mathbf{z}^T) + \boldsymbol{\mu}\mathbf{E}(\mathbf{z}^T) + \mathbf{E}(\boldsymbol{\epsilon}\mathbf{z}^T) \\
 &= \mathbf{W}
 \end{aligned}$$

So, plugging in, we have the joint distribution

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{x} \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\mu} \end{bmatrix}, \begin{bmatrix} \mathbf{I} & \mathbf{W}^T \\ \mathbf{W} & \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I} \end{bmatrix} \right) \quad (6)$$

(ii).  $\Pr(\mathbf{x})$

We can read the marginal distribution directly from (6).

$$\Pr(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \sigma^2\mathbf{I}) \quad (7)$$

(iii).  $\Pr(\mathbf{z} | \mathbf{x})$

In order to find the other marginal distribution, we could write down the equations and use Bayes rule and a bunch of linear algebra to simplify. Easier is to make use of a useful factoid (see *PRML* Section 2.3.1):



### Conditioning for multivariate Gaussian random variables

If  $\mathbf{x} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{bmatrix} \right)$  then  $\Pr(\mathbf{x}_a | \mathbf{x}_b)$  is Gaussian with mean

$$E(\mathbf{x}_a | \mathbf{x}_b) = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} (\mathbf{x}_b - \boldsymbol{\mu}_b)$$

and covariance

$$\text{Cov}(\mathbf{x}_a | \mathbf{x}_b) = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} \boldsymbol{\Sigma}_{ba}$$

Let's try to make intuitive sense of these equations. Basically, once we observe  $\mathbf{x}_b$ , we should modify our estimate of  $\mathbf{x}_a$  (the  $\boldsymbol{\Sigma}_{ab}$  term), but weighted by how noisy  $\mathbf{x}_b$  is (the  $\boldsymbol{\Sigma}_{bb}^{-1}$  term). And once we make that new estimate, the covariance decreases – we believe that we know a little more about  $\mathbf{x}_a$  than we did before. The amount of that decrease doesn't depend on the actual measurement of  $\mathbf{x}_b$ , just on its covariance and the transformation from one space to the other (the  $\boldsymbol{\Sigma}_{ab} \boldsymbol{\Sigma}_{bb}^{-1} \boldsymbol{\Sigma}_{ba}$  term).

Plugging in from (6), we have

$$\begin{aligned} E(\mathbf{z} | \mathbf{x}) &= \mathbf{0} + \mathbf{W}^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \\ \text{Cov}(\mathbf{z} | \mathbf{x}) &= \mathbf{I} - \mathbf{W}^T \mathbf{C}^{-1} \mathbf{W}, \end{aligned}$$

where  $\mathbf{C} = \mathbf{W} \mathbf{W}^T + \sigma^2 \mathbf{I}$ .

Thus,

$$\mathbf{z} | \mathbf{x} \sim \mathcal{N}(\mathbf{W}^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}), \mathbf{I} - \mathbf{W}^T \mathbf{C}^{-1} \mathbf{W}) \quad (8)$$

### B.4 EM Algorithm for P-PCA

**Goal:** Maximize  $\log \Pr(\{\mathbf{x}\} | \theta)$  where  $\theta = \{\mathbf{W}, \boldsymbol{\mu}, \sigma^2\}$ .

The EM algorithm will find an exact solution for the mean,  $\boldsymbol{\mu}$ , and iteratively optimize values of  $\mathbf{W}$  and  $\sigma^2$  such that the sample covariance is

$$\mathbf{S} \approx \mathbf{W} \mathbf{W}^T + \sigma^2 \mathbf{I}.$$

This makes sense if we look back at (7).

The maximum likelihood solution for the mean is just the sample mean, which we're not going to show.

#### E-Step:

Find  $\Pr(\mathbf{z}_n | \mathbf{x}_n)$  for each data point using (8).

#### M-Step:

$$\log \Pr(\mathbf{x}_N, \mathbf{z}_N) = \sum_{n=1}^N \log \Pr(\mathbf{x}_n, \mathbf{z}_n)$$

$$\begin{aligned}
&= \sum_{n=1}^N (\log \Pr(\mathbf{x}_n | \mathbf{z}_n) + \log \Pr(\mathbf{z}_n)) \\
&= \sum_{n=1}^N \left( -\frac{D}{2} \log(2\pi) - \frac{1}{2} \log |\sigma^2 \mathbf{I}| - \frac{1}{2} (\mathbf{x}_n - \mathbf{W}\mathbf{z}_n - \boldsymbol{\mu})^\top (\sigma^2 \mathbf{I})^{-1} (\mathbf{x}_n - \mathbf{W}\mathbf{z}_n - \boldsymbol{\mu}) \right. \\
&\quad \left. - \frac{M}{2} \log(2\pi) - \frac{M}{2} \log |\mathbf{I}| - \frac{1}{2} \mathbf{z}_n^\top \mathbf{z}_n \right)
\end{aligned}$$

$$\begin{aligned}
Q &= \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\log \Pr(\mathbf{x}_N, \mathbf{z}_N)) \\
&= \sum_{n=1}^N \left( -\frac{D}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \mathbb{E}_{\mathbf{z}|\mathbf{x}} \left( (\mathbf{x}_n - \boldsymbol{\mu} - \mathbf{W}\mathbf{z}_n)^\top (\mathbf{x}_n - \boldsymbol{\mu} - \mathbf{W}\mathbf{z}_n) \right) \right. \\
&\quad \left. - \frac{M}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n^\top \mathbf{z}_n) \right) \\
&= \sum_{n=1}^N \left( -\frac{D}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left( (\mathbf{x}_n - \boldsymbol{\mu})^\top (\mathbf{x}_n - \boldsymbol{\mu}) - \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n^\top \mathbf{W}^\top (\mathbf{x}_n - \boldsymbol{\mu})) \right) \right. \\
&\quad \left. - \mathbb{E}_{\mathbf{z}|\mathbf{x}} ((\mathbf{x}_n - \boldsymbol{\mu})^\top \mathbf{W} \mathbf{z}_n) + \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n^\top \mathbf{W}^\top \mathbf{W} \mathbf{z}_n) \right) \\
&\quad \left. - \frac{M}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n^\top \mathbf{z}_n) \right) \\
&= \sum_{n=1}^N \left( -\frac{D}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \left( (\mathbf{x}_n - \boldsymbol{\mu})^\top (\mathbf{x}_n - \boldsymbol{\mu}) - \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n)^\top \mathbf{W}^\top (\mathbf{x}_n - \boldsymbol{\mu}) \right) \right. \\
&\quad \left. - (\mathbf{x}_n - \boldsymbol{\mu})^\top \mathbf{W} \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n) + \text{Tr}(\mathbf{W}^\top \mathbf{W} \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n \mathbf{z}_n^\top)) \right) \\
&\quad \left. - \frac{M}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n^\top \mathbf{z}_n) \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial Q}{\partial \mathbf{W}} &= \sum_{n=1}^N -\frac{1}{2\sigma^2} \left( -(\mathbf{x}_n - \boldsymbol{\mu}) \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n)^\top - (\mathbf{x}_n - \boldsymbol{\mu}) \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n)^\top + 2\mathbf{W} \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n \mathbf{z}_n^\top) \right) = 0 \\
&\Rightarrow \mathbf{W} \left( \sum_{n=1}^N \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n \mathbf{z}_n^\top) \right) = \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}) \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n)^\top
\end{aligned}$$

$$\mathbf{W}_{new} = \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}) \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n)^\top \right) \left( \sum_{n=1}^N \mathbb{E}_{\mathbf{z}|\mathbf{x}} (\mathbf{z}_n \mathbf{z}_n^\top) \right)^{-1} \quad (9)$$

$$\begin{aligned}
\frac{\partial Q}{\partial \sigma^2} &= \sum_{n=1}^N -\frac{D}{2} \frac{1}{\sigma^2} + \frac{1}{2} \frac{1}{(\sigma^2)^2} (\cdot) = 0 \\
-ND + \frac{1}{\sigma^2} \sum_{n=1}^N (\cdot) &= 0
\end{aligned}$$

$$\begin{aligned} \Rightarrow \sigma^2 &= \frac{1}{ND} \sum_{n=1}^N (\cdot) \\ \Rightarrow \sigma^2 &= \frac{1}{ND} \sum_{n=1}^N \left( (\mathbf{x}_n - \boldsymbol{\mu})^\top (\mathbf{x}_n - \boldsymbol{\mu}) - \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n)^\top \mathbf{W}^\top (\mathbf{x}_n - \boldsymbol{\mu}) \right. \\ &\quad \left. - (\mathbf{x}_n - \boldsymbol{\mu})^\top \mathbf{W} \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n) + \text{Tr}(\mathbf{W}^\top \mathbf{W} \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top)) \right) \end{aligned}$$

This is a complicated expression, but looking back, we remember that

$$\sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}) \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n)^\top = \mathbf{W}_{new} \left( \sum_{n=1}^N \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top) \right),$$

and note that all the individual expressions are scalars and thus can be wrapped in a trace operation.

Plugging this in, we have

$$\begin{aligned} \sigma^2 &= \frac{1}{ND} \left( \text{Tr} \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^\top (\mathbf{x}_n - \boldsymbol{\mu}) \right) - \text{Tr} \left( \sum_{n=1}^N \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n)^\top \mathbf{W}^\top (\mathbf{x}_n - \boldsymbol{\mu}) \right) \right. \\ &\quad \left. - \text{Tr} \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^\top \mathbf{W} \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n) \right) + \text{Tr} \left( \sum_{n=1}^N \mathbf{W}^\top \mathbf{W} \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top) \right) \right) \\ &= \frac{1}{ND} \left( \text{Tr} \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^\top (\mathbf{x}_n - \boldsymbol{\mu}) \right) - \text{Tr} \left( \mathbf{W}^\top \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}) \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n)^\top \right) \right. \\ &\quad \left. - \text{Tr} \left( \mathbf{W} \sum_{n=1}^N \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n) (\mathbf{x}_n - \boldsymbol{\mu})^\top \right) + \text{Tr} \left( \sum_{n=1}^N \mathbf{W}^\top \mathbf{W} \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top) \right) \right) \\ &= \frac{1}{ND} \left( \text{Tr} \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^\top (\mathbf{x}_n - \boldsymbol{\mu}) \right) - \text{Tr} \left( \mathbf{W}^\top \mathbf{W}_{new} \left( \sum_{n=1}^N \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top) \right) \right) \right. \\ &\quad \left. - \text{Tr} \left( \mathbf{W} \sum_{n=1}^N \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n) (\mathbf{x}_n - \boldsymbol{\mu})^\top \right) + \text{Tr} \left( \sum_{n=1}^N \mathbf{W}^\top \mathbf{W} \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top) \right) \right) \end{aligned}$$

Making sure that all of our traces are over the same dimension of matrices we can combine to get our final result:

$$\sigma_{new}^2 = \frac{1}{ND} \text{Tr} \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^\top - \mathbf{W}_{new} \sum_{n=1}^N \mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n) (\mathbf{x}_n - \boldsymbol{\mu})^\top \right) \quad (10)$$

**Implementation Aside:** A common error in implementing the EM algorithm is for students to confuse  $\mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top)$  for the covariance in (8). Rather, it is the expected outer product,

$$\mathbf{E}_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n \mathbf{z}_n^\top) = \text{Cov}(\mathbf{z}_n | \mathbf{x}_n) + \mathbf{E}(\mathbf{z}_n | \mathbf{x}_n) \mathbf{E}(\mathbf{z}_n | \mathbf{x}_n)^\top$$

## B.5 Relating P-PCA to PCA

### PC Directions

The columns of  $\mathbf{W}$  for P-PCA span the same space as that spanned by the columns of  $\mathbf{U}_M$  from PCA. The difference between them is that the columns of  $\mathbf{U}_M$  are orthonormal and ordered based on the amount of variance explained. Neither of these aspects will in general be true for  $\mathbf{W}$ .

However, with some linear algebra one can obtain  $\mathbf{U}_M$  from  $\mathbf{W}$ . Specifically, we can calculate the singular value decomposition (SVD) of  $\mathbf{W}$ . The SVD is a generalization of the eigendecomposition for non-square matrices.

$$\mathbf{W} = \begin{bmatrix} | & | & \dots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_M \\ | & | & \dots & | \end{bmatrix} \begin{bmatrix} d_1 & & & 0 \\ & d_2 & & \\ 0 & & \ddots & \\ & & & d_M \end{bmatrix} \begin{bmatrix} | & | & \dots & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_M \\ | & | & \dots & | \end{bmatrix}^T$$

$\tilde{\mathbf{U}} \qquad \tilde{\mathbf{D}} \qquad \tilde{\mathbf{V}}^T$   
 $(D \times M) \quad (M \times M) \quad (M \times M)$   
 columns of  $\tilde{\mathbf{U}}$  orthonormal      diagonal matrix      columns of  $\tilde{\mathbf{V}}$  orthonormal  
 $d_1 \geq d_2 \geq \dots \geq d_M \geq 0$   
 "singular values"

The matrix  $\tilde{\mathbf{U}}$  calculated from  $\mathbf{W}$  in P-PCA will be identical (in the limit of EM convergence) to  $\mathbf{U}_M$  for PCA.

### Low-dimensional projections

In P-PCA, the low-dimensional projection corresponding to  $\mathbf{W}$  is  $E(\mathbf{z}_n | \mathbf{x}_n) = \mathbf{W}^T \mathbf{C}^{-1}(\mathbf{x}_n - \boldsymbol{\mu})$  from (8). The same point can be back-projected to the original state by

$$\begin{aligned} \hat{\mathbf{x}}_n &= \mathbf{W} E(\mathbf{z}_n | \mathbf{x}_n) + \boldsymbol{\mu} \\ &= \underbrace{\tilde{\mathbf{U}} \tilde{\mathbf{D}} \tilde{\mathbf{V}}^T E(\mathbf{z}_n | \mathbf{x}_n)}_{\text{call this } \tilde{\mathbf{z}}_n} + \boldsymbol{\mu} \end{aligned}$$

There are several important reasons why  $\tilde{\mathbf{z}}_n$  is easier to interpret than  $E(\mathbf{z}_n | \mathbf{x}_n)$ . All of the reasons stem from the fact that the columns of  $\tilde{\mathbf{U}}$  are orthonormal and ordered while those of  $\mathbf{W}$  are not.

1.  $\tilde{\mathbf{z}}_n$  has the same units as  $\mathbf{x}_n$  (as in PCA)
2. The dimensions of  $\tilde{\mathbf{z}}_n$  are ordered (as in PCA), whereas  $E(\mathbf{z}_n \mid \mathbf{x}_n)$  can be arbitrarily rotated and scaled in the latent space.
3.  $\tilde{\mathbf{z}}_n$  can be easily compared to the low-dimensional PCA projection.

How does the low-dimensional projection for P-PCA ( $\tilde{\mathbf{z}}_n$ ) compare with that of PCA ( $\mathbf{U}_M(\mathbf{x}_n - \boldsymbol{\mu})$ )? Without proof,

$$\tilde{\mathbf{z}}_n = \begin{bmatrix} \frac{\lambda_1 - \sigma^2}{\lambda_1} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_M - \sigma^2}{\lambda_M} \end{bmatrix} \mathbf{U}_M(\mathbf{x}_n - \boldsymbol{\mu}) \quad (11)$$

Unpacking, this means that the low-dimensional projection for P-PCA shrinks the PCA low-dimensional projection to the origin in  $\mathbf{z}$ -space, because  $0 \leq \frac{\lambda_i - \sigma^2}{\lambda_i} \leq 1$ . Furthermore, as  $\sigma^2 \rightarrow 0$ , the P-PCA projections converge to the PCA projections.

### Intuition for P-PCA vs PCA

Each of these models is trying to explain the variability of  $\mathbf{x}$  away from its mean  $\boldsymbol{\mu}$ . P-PCA can explain this variability as a combination of variation in low-dimensional space – the latent variable  $\mathbf{z}$  – and observation noise,  $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ . But how much of each?

As  $\sigma^2$  increases (more observation noise), the proportion of the variability attributed to observation noise increases, and the proportion attributed to the latent space decreases, shrinking  $\mathbf{z}$  to its mean (which is zero, corresponding to  $\boldsymbol{\mu}$  in the data space). PCA is the opposite limit; as  $\sigma^2$  decreases to zero, there is no observation noise, and all variability of  $\mathbf{x}$  from  $\boldsymbol{\mu}$  must be explained by the latent variable space. Thus, *P-PCA is more effective at denoising data than PCA.*

### B.6 Redux - Advantages of P-PCA over PCA

- The dimensionality,  $M$ , of the latent space for P-PCA can be selected using cross-validated likelihoods, where  $\Pr(\{\mathbf{x}\}_N)$  is given in (7).
- P-PCA defines a constrained Gaussian in (7), with  $\text{Cov}(\mathbf{x}) = \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I}$ , which is a useful compromise between a Gaussian with diagonal covariance (too constraining in many situations) and a Gaussian with a full covariance matrix (underconstrained in the scenarios where we would want to employ dimensionality reduction).

## C. Factor Analysis (FA)

### C.1 Motivation

P-PCA assumes that the observation noise is “isotropic” – the same in all observation dimensions. This makes sense if each dimension is a similar measurement, but if they are different (i.e., height and weight), we would like each dimension of  $\mathbf{x}$  to have a different level of observation noise.

The only difference between FA and P-PCA is that instead of modeling the covariance of the observation noise as  $\sigma^2 \mathbf{I}$  (in (5)), the observation noise covariance is a diagonal matrix  $\boldsymbol{\Psi}$ . So, for FA

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^T + \boldsymbol{\Psi}), \quad (12)$$

where the similarity to (7) is clear.

We can define an EM algorithm that is nearly identical to P-PCA. When solving for the M-step, after replacing all the instances of  $\sigma^2 \mathbf{I}$  with  $\Psi$ , the update for  $\mathbf{W}$  is unchanged and the update for the covariance becomes

$$\Psi_{new} = \frac{1}{N} \text{diag} \left( \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})(\mathbf{x}_n - \boldsymbol{\mu})^T - \mathbf{W}_{new} \sum_{n=1}^N E_{\mathbf{z}|\mathbf{x}}(\mathbf{z}_n)(\mathbf{x}_n - \boldsymbol{\mu})^T \right) \quad (13)$$

where the  $\text{diag}()$  operator zeros the off-diagonal elements. For the E-step, everything is the same except the marginal covariance of  $\mathbf{x}$  ( $\mathbf{C}$  in (8)) is given by (12).

### (i). Comparing FA and P-PCA

Because of the non-isotropic observation noise, FA will identify different latent dimensions within the data space than P-PCA/PCA. As with P-PCA, we will need to orthonormalize the columns of  $\mathbf{W}$  for interpretability.

- PCA/P-PCA is invariant to rotations in the data space, whereas FA is not. This is because in FA, the observation noise is associated with each axis independently.
- FA is invariant to component-wise rescaling of the data, whereas P-PCA/PCA is not. An example of component-wise rescaling would be  $\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} \rightarrow \begin{bmatrix} 3x^{(1)} \\ \frac{1}{2}x^{(2)} \end{bmatrix}$ . The reason is that the observation noise is assumed to be the same for each data dimension in P-PCA/PCA.

## Appendix

### Useful matrix derivatives

$$\frac{\partial}{\partial \mathbf{X}} \mathbf{a}^T \mathbf{X} \mathbf{b}^T = \mathbf{a} \mathbf{b}^T \quad (14)$$

$$\frac{\partial}{\partial \mathbf{X}} \mathbf{a}^T \mathbf{X}^T \mathbf{b}^T = \mathbf{b} \mathbf{a}^T \quad (15)$$

$$\frac{\partial}{\partial \mathbf{X}} \text{Tr}(\mathbf{X} \mathbf{X}^T \mathbf{A}) = \mathbf{X}(\mathbf{A} + \mathbf{A}^T) \quad (16)$$

### Matrix inversion lemma

Inverting  $\mathbf{C} = \mathbf{W} \mathbf{W}^T + \sigma^2 \mathbf{I}$  directly in (8) can be quite a costly  $O(D^3)$  operation. Instead, one can use the matrix inversion lemma

$$\mathbf{C}^{-1} = \sigma^{-2} \mathbf{I} - \sigma^{-2} \mathbf{W} \underbrace{(\sigma^2 \mathbf{I} + \mathbf{W}^T \mathbf{W})^{-1}}_{M \times M} \mathbf{W}^T.$$

Assuming  $M \ll D$ , the inverse is now a much easier  $O(M^3)$  operation. There's an equivalent trick for FA, which is slightly more complex because of the non-constant diagonal.