EECS 189 Introduction to Machine Learning
Fall 2020

Note 10

In machine learning, the data we have are often very high-dimensional. In fact, when we introduced the idea of features (like polynomial features), these made the dimensionality of the data even higher. The kernel trick was something that let us partially deal with this by working with vectors only as long as there are training samples.

However, there are a number of reasons why we might want to work with a lower-dimensional representation:

- Visualization (if we can get it down to 2 or 3 dimensions), e.g. for exploratory data analysis
- Reduce computational load
- Reduce variance in estimation — regularize the problem

So, how can we reduce the dimensionality of data? There are obvious ways — just keeping a subset of features. But which features? In general, that presumably depends on what you are trying to do. What could you do if you didn't know what you were trying to predict with those features? This corresponds to unsupervised dimensionality reduction. There are a couple of intuitive choices. First, just pick some features at random to keep. This is appealing for its symmetry, but it makes you wonder if we could do better by actually looking at the data before deciding which features to keep.

Consequently, another thing that you could do is to just keep the few features that have the most variability — which you could measure by the variance of that feature. But what if two of the most variable features were actually very correlated to each other? Should we really be including both of them? Maybe we should focus on "fresh" variability somehow. To do this, maybe it would be helpful to allow ourselves to synthesize linear combinations of features and keep some of these synthesized features.

1 Principal Component Analysis

Principal Component Analysis (PCA) is exactly such an unsupervised dimensionality reduction technique. Given a matrix of data points, it finds one or more orthogonal directions that capture the largest amount of variance in the data. Intuitively, the directions with less variance contain less information and may be discarded without introducing too much error. One of the practical motivations for taking this kind of unsupervised approach to dimensionality reduction is that labeled training data might be hard or expensive to get, but unlabeled training data (i.e. no y just $\mathbf{x}$) might be more easily available. PCA is able to extract meaningful directions from such unlabeled data.

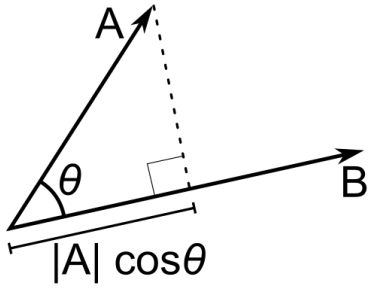
Not coincidentally, PCA turns out to be intimately connected to the ideas of Total Least Squares.

1.1 Projection

Let us first review the meaning of scalar projection of one vector onto another. If $\mathbf{v} \in \mathbb{R}^d$ is a unit vector, i.e. $\|\mathbf{v}\| = 1$, then the scalar projection of another vector $\mathbf{x} \in \mathbb{R}^d$ onto $\mathbf{v}$ is given by $\mathbf{x}^\top \mathbf{v}$. This quantity tells us roughly how much of the projected vector $\mathbf{x}$ lies along the direction given direction $\mathbf{v}$. Why does this expression make sense? Recall the slightly more general formula which holds for vectors of any length:

$$\mathbf{x}^\top \mathbf{v} = \|\mathbf{x}\|\|\mathbf{v}\| \cos\theta$$

where θ is the angle between the vectors. In this case, since $\|\mathbf{v}\| = 1$, the expression simplifies to $\mathbf{x}^\top \mathbf{v} = \|\mathbf{x}\| \cos\theta$. But since cosine gives the ratio of the adjacent side (the projection we want to find) to the hypotenuse ($\|\mathbf{x}\|$), this is exactly what we want:



One approach to dimensionality reduction by using projections is to choose projections at random — sample from an iid Gaussian and then normalize the vector to get our $\mathbf{v}$. This creates a degree of fairness across the individual features since an iid Gaussian is uniform over directions in d-dimensional space. As you will see in homework, this approach to dimensionality reduction actually has many interesting properties. By construction, however, it does not look at any data itself and thus is unable to prioritize important vs unimportant feature directions.

1.2 The first principal component

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be our matrix of data, where each row is a d-dimensional datapoint. These are to be thought of as i.i.d. samples from some random vector $\mathbf{x}$.

We will assume that the data points have mean zero; if this is not the case, we can make it so by subtracting the average of all the rows, $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i$, from each row. The motivation for this is that we want to find directions of high variance within the data, and variance is defined relative to the mean of the data. If we did not zero-center the data, the directions found would be heavily influenced by where the data lie relative to the origin, rather than where they lie relative to the other data, which is more useful.

Since $\mathbf{X}$ is zero-mean, the sample variance of the datapoints' projections onto a unit vector $\mathbf{v}$ is given by

$$\frac{1}{n}\sum_{i=1}^n (\mathbf{x}_i^\top \mathbf{v})^2 = \frac{1}{n}\|\mathbf{X}\mathbf{v}\|^2 = \frac{1}{n}\mathbf{v}^\top \mathbf{X}^\top \mathbf{X}\mathbf{v}$$

where $\mathbf{v}$ is constrained to have unit norm.[1]

With this motivation, we define the **first loading vector** $\mathbf{v}_1$ as the solution to the constrained optimization problem

$$\max_{\mathbf{v}} \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} \quad \text{subject to} \quad \mathbf{v}^\top \mathbf{v} = 1$$

Note that we have discarded the positive constant factor $1/n$ which does not affect the optimal value of $\mathbf{v}$.

To reduce this constrained optimization problem to an unconstrained one, we write down its Lagrangian:

$$\mathcal{L}(\mathbf{v}) = \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} - \lambda(\mathbf{v}^\top \mathbf{v} - 1)$$

First-order necessary conditions for optima imply that

$$\mathbf{0} = \nabla \mathcal{L}(\mathbf{v}_1) = 2\mathbf{X}^\top \mathbf{X} \mathbf{v}_1 - 2\lambda \mathbf{v}_1$$

Hence $\mathbf{X}^\top \mathbf{X} \mathbf{v}_1 = \lambda \mathbf{v}_1$, i.e. $\mathbf{v}_1$ is an eigenvector of $\mathbf{X}^\top \mathbf{X}$ with eigenvalue λ. Since we constrain $\mathbf{v}_1{}^\top \mathbf{v}_1 = 1$, the value of the objective is precisely

$$\mathbf{v}_1{}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v}_1 = \mathbf{v}_1{}^\top (\lambda \mathbf{v}_1) = \lambda \mathbf{v}_1{}^\top \mathbf{v}_1 = \lambda$$

so the optimal value is $\lambda = \lambda_{\max}(\mathbf{X}^\top \mathbf{X})$, which is achieved when $\mathbf{v}_1$ is a unit eigenvector of $\mathbf{X}^\top \mathbf{X}$ corresponding to its largest eigenvalue.

1.3 Finding more principal components

We have seen how to find the first loading vector, which is the unit vector that maximizes the variance of the projected data points. However, in most applications, we want to find more than one direction. We want the subsequent directions found to also be directions of high variance, but they ought to be orthogonal to the existing directions in order to minimize redundancy in the information captured. Thus we define the kth loading vector $\mathbf{v}_k$ as the solution to the constrained optimization problem

$$\max_{\mathbf{v}} \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} \quad \text{subject to} \quad \mathbf{v}^\top \mathbf{v} = 1$$
$$\mathbf{v}^\top \mathbf{v}_i = 0, \quad i = 1, \ldots, k-1$$

We claim that $\mathbf{v}_k$ is a unit eigenvector of $\mathbf{X}^\top \mathbf{X}$ corresponding to its kth largest eigenvalue.

[1] To make sense of the sample variance, recall that for any random variable Z,

$$\mathrm{Var}(Z) = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$$

so if $\mathbb{E}[Z] = 0$ then $\mathrm{Var}(Z) = \mathbb{E}[Z^2]$. In practice we will not have the true random variable Z, but rather i.i.d. observations $z_1, \ldots, z_n$ of Z. The expected value can then be approximated by a sample average, i.e.

$$\mathbb{E}[Z^2] \approx \frac{1}{n} \sum_{i=1} z_i^2$$

which is justified by the law of large numbers, which states that (under mild conditions) the sample average converges to the expected value as $n \to \infty$. In our case the random variable Z is the principal component $\mathbf{v}^\top \mathbf{x}$, and the i.i.d. observations are the projections of our datapoints, i.e. $z_i = \mathbf{v}^\top \mathbf{x}_i$.

Proof. By induction on k. We have already shown that the claim is true for the base case $k = 1$ (where there are no orthogonality constraints). Now assume that it is true for the first k loading vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$, and consider the problem of finding $\mathbf{v}_{k+1}$.

By the inductive hypothesis, we know that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are orthonormal eigenvectors of $\mathbf{X}^\top\mathbf{X}$. Denote the ith largest eigenvalue of $\mathbf{X}^\top\mathbf{X}$ by λ_i, noting that $\mathbf{X}^\top\mathbf{X}\mathbf{v}_i = \lambda_i\mathbf{v}_i$.

The Lagrangian of the objective function is

$$\mathcal{L}(\mathbf{v}) = \mathbf{v}^\top\mathbf{X}^\top\mathbf{X}\mathbf{v} - \lambda(\mathbf{v}^\top\mathbf{v} - 1) + \sum_{i=1}^{k} \eta_i \mathbf{v}^\top\mathbf{v}_i$$

First-order necessary conditions for optima imply that

$$\mathbf{0} = \nabla\mathcal{L}(\mathbf{v}_{k+1}) = 2\mathbf{X}^\top\mathbf{X}\mathbf{v}_{k+1} - 2\lambda\mathbf{v}_{k+1} + \sum_{i=1}^{k} \eta_i \mathbf{v}_i$$

This implies that, if $\mathbf{v}_{k+1}$ is orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_k$ (as we constrain it to be), then

$$\begin{aligned}
0 &= \mathbf{v}_j{}^\top\mathbf{0} \\
&= 2\mathbf{v}_j{}^\top\mathbf{X}^\top\mathbf{X}\mathbf{v}_{k+1} - 2\lambda \underbrace{\mathbf{v}_j{}^\top\mathbf{v}_{k+1}}_{0} + \sum_{i=1}^{k} \eta_i \underbrace{\mathbf{v}_j{}^\top\mathbf{v}_i}_{\delta_{ij}} \\
&= 2(\mathbf{X}^\top\mathbf{X}\mathbf{v}_j)^\top\mathbf{v}_{k+1} + \eta_j \\
&= 2(\lambda_j\mathbf{v}_j)^\top\mathbf{v}_{k+1} + \eta_j \\
&= 2\lambda_j \underbrace{\mathbf{v}_j{}^\top\mathbf{v}_{k+1}}_{0} + \eta_j \\
&= \eta_j
\end{aligned}$$

for all $j = 1, \ldots, k$.

Plugging these values back into the optimality equation above, we see that $\mathbf{v}_{k+1}$ must satisfy $\mathbf{X}^\top\mathbf{X}\mathbf{v}_{k+1} = \lambda\mathbf{v}_{k+1}$, i.e. $\mathbf{v}_{k+1}$ is an eigenvector of $\mathbf{X}^\top\mathbf{X}$ with eigenvalue λ. As before, the value of the objective function is then λ. To maximize, we want the largest eigenvalue, but we must respect the constraints that $\mathbf{v}_{k+1}$ is orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_k$. Clearly if $\mathbf{v}_{k+1}$ is equal to any of these eigenvectors (up to sign), then one of these constraints will not be satisfied. Thus to maximize the expression, $\mathbf{v}_{k+1}$ should be a unit eigenvector of $\mathbf{X}^\top\mathbf{X}$ corresponding to its $(k+1)$st largest eigenvalue. By the spectral theorem, we can always choose this vector in such a way that it is orthogonal to $\mathbf{v}_1, \ldots, \mathbf{v}_k$, so we are done. □

We have shown that the loading vectors are orthonormal eigenvectors of $\mathbf{X}^\top\mathbf{X}$. In other words, they are right-singular vectors of $\mathbf{X}$, so they can all be found simultaneously by computing the SVD of $\mathbf{X}$.

1.4 Projecting onto the PCA coordinate system

Once we have computed the loading vectors, we can use them as a new coordinate system. The kth principal component of a datapoint $\mathbf{x}_i \in \mathbb{R}^d$ is defined as the scalar projection of $\mathbf{x}_i$ onto the kth

loading vector $\mathbf{v}_k$, i.e. $\mathbf{x}_i^\top \mathbf{v}_k$. We can compute all the principal components of all the datapoints at once using a matrix-matrix multiplication:

$$\mathbf{Z}_k = \mathbf{X}\mathbf{V}_k$$

where $\mathbf{V}_k \in \mathbb{R}^{d \times k}$ is a matrix whose columns are the first k loading vectors $\mathbf{v}_1, \ldots, \mathbf{v}_k$.

Below we plot the result of such a projection in the case $d = k = 2$:

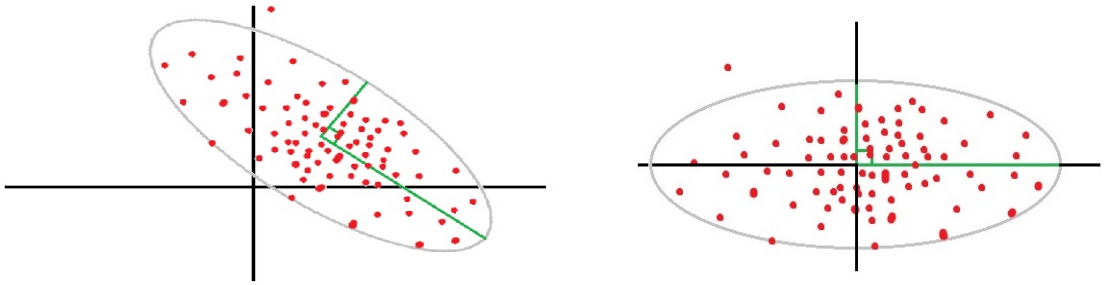


Figure 1: Left:data points; Right: PCA projection of data points

Observe that the data are uncorrelated in the projected space. Also note that this example does not show the power of PCA since we have not reduced the dimensionality of the data at all – the plot is merely to show the PCA coordinate transformation.

Once we've computed the principal components, we can approximately reconstruct the original points by

$$\tilde{\mathbf{X}}_k = \mathbf{Z}_k \mathbf{V}_k^\top = \mathbf{X}\mathbf{V}_k\mathbf{V}_k^\top$$

The rows of $\tilde{\mathbf{X}}_k$ are the projections of the original rows of $\mathbf{X}$ onto the subspace spanned by the loading vectors.

1.5 Other derivations of PCA

We have given the most common derivation of PCA above, but it turns out that there are other ways to solve the optimization problem, or to arrive at the same formulation. These give us helpful additional perspectives on what PCA is doing.

1.6 Changing coordinates

In PCA we want to find the unit length $\mathbf{v}$ that maximizes $\mathbf{v}^\top \mathbf{X}^\top \mathbf{X}\mathbf{v}$. It turns out that there is a result, sometimes referred to as the **variational characterization of eigenvalues**, that tells us which vectors $\mathbf{v}$ achieve this. The key idea in the proof is a length-preserving change of coordinates.

Theorem. *Let $\mathbf{A} \in \mathbb{R}^{d \times d}$ be symmetric. Then for any $\mathbf{v} \in \mathbb{R}^d$ satisfying $\|\mathbf{v}\|_2 = 1$,*

$$\lambda_{\min}(\mathbf{A}) \leq \mathbf{v}^\top \mathbf{A}\mathbf{v} \leq \lambda_{\max}(\mathbf{A})$$

where for both bounds, equality holds if and only if $\mathbf{v}$ is a corresponding eigenvector.

Proof. We show only the max case because the argument for the min case is entirely analogous.

Since $\mathbf{A}$ is symmetric, we can decompose it as $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q} \in \mathbb{R}^{d\times d}$ is orthogonal and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$ contains the eigenvalues of $\mathbf{A}$. For any $\mathbf{v}$ satisfying $\|\mathbf{v}\|_2 = 1$, define $\mathbf{z} = \mathbf{Q}^\top\mathbf{v}$, noting that the relationship between $\mathbf{v}$ and $\mathbf{z}$ is one-to-one because $\mathbf{Q}$ is invertible and that $\|\mathbf{z}\|_2 = 1$ because $\mathbf{Q}$ is orthogonal. Hence

$$\max_{\|\mathbf{v}\|_2=1} \mathbf{v}^\top\mathbf{A}\mathbf{v} = \max_{\|\mathbf{z}\|_2=1} \mathbf{z}^\top\mathbf{\Lambda}\mathbf{z} = \max_{\|\mathbf{z}\|_2^2=1} \sum_{i=1}^{d} \lambda_i z_i^2$$

We note that

$$\sum_{i=1}^{d} \lambda_i z_i^2 \leq \sum_{i=1}^{d} \lambda_{\max}(\mathbf{A}) z_i^2 = \lambda_{\max}(\mathbf{A}) \sum_{i=1}^{d} z_i^2$$

so the constraint $\|\mathbf{z}\|_2^2 = \sum_{i=1}^{d} z_i^2 = 1$ implies

$$\sum_{i=1}^{d} \lambda_i z_i^2 \leq \lambda_{\max}(\mathbf{A})$$

Defining $I = \{i : \lambda_i = \lambda_{\max}(\mathbf{A})\}$, the index set of the largest eigenvalue, we see that the bound is achieved with equality if and only if $\sum_{i\in I} z_i^2 = 1$ and $z_j = 0$ for $j \notin I$. Suppose $\mathbf{z}^*$ satisfies this condition. Then writing $\mathbf{q}_1, \ldots, \mathbf{q}_d$ for the columns of $\mathbf{Q}$, we have

$$\mathbf{v}^* = \mathbf{Q}\mathbf{z}^* = \sum_{i=1}^{d} z_i^* \mathbf{q}_i = \sum_{i\in I} z_i^* \mathbf{q}_i$$

Recall that $\mathbf{q}_1, \ldots, \mathbf{q}_d$ are eigenvectors of $\mathbf{A}$ and form an orthonormal basis for $\mathbb{R}^d$. Therefore by construction, the set $\{\mathbf{q}_i : i \in I\}$ forms an orthonormal basis for the eigenspace of $\lambda_{\max}(\mathbf{A})$. Hence $\mathbf{v}^*$, which is a linear combination of these, lies in that eigenspace and thus is an eigenvector of $\mathbf{A}$ corresponding to $\lambda_{\max}(\mathbf{A})$.

Conversely, suppose $\mathbf{v} \in \mathbb{R}^d$ is unit-length but not an eigenvector corresponding to $\lambda_{\max}(\mathbf{A})$. The vectors $\mathbf{q}_1, \ldots, \mathbf{q}_d$ are still a basis for $\mathbb{R}^d$, so we have a unique expansion

$$\mathbf{v} = z_1\mathbf{q}_1 + \cdots + z_d\mathbf{q}_d$$

Since $\mathbf{v}$ does not lie in the eigenspace of $\lambda_{\max}(\mathbf{A})$, one of the components z_j must be nonzero for an index $j \notin I$, so equality does not hold in the bound above. □

With this result established, we see that the vector we seek (which maximizes $\mathbf{v}^\top\mathbf{X}^\top\mathbf{X}\mathbf{v}$) must be an eigenvector corresponding to $\lambda_{\max}(\mathbf{X}^\top\mathbf{X})$. This is the same solution we derived via the Lagrangian formulation above.

1.7 Minimizing reconstruction error

Recall that ordinary least squares minimizes the vertical distance between the fitted line and the data points. We show that PCA can be interpreted as minimizing the perpendicular distance between the data points and the subspace onto which we are projecting them.

The orthogonal projection of a vector $\mathbf{x}$ onto the subspace spanned by a unit vector $\mathbf{v}$ equals $\mathbf{v}$ scaled by the scalar projection of $\mathbf{x}$ onto $\mathbf{v}$:

$$P_{\mathbf{v}}\mathbf{x} = (\mathbf{x}^\top \mathbf{v})\mathbf{v}$$

Suppose we want to minimize the total reconstruction error:

$$\sum_{i=1}^{n} \|\mathbf{x}_i - P_{\mathbf{v}}\mathbf{x}_i\|^2$$

For any $\mathbf{x} \in \mathbb{R}^d$, we know $\mathbf{x} - P_{\mathbf{v}}\mathbf{x} \perp P_{\mathbf{v}}\mathbf{x}$, so the Pythagorean Theorem tells us that

$$\|\mathbf{x} - P_{\mathbf{v}}\mathbf{x}\|^2 + \|P_{\mathbf{v}}\mathbf{x}\|^2 = \|\mathbf{x}\|^2$$

Thus

$$\begin{aligned}
\sum_{i=1}^{n} \|\mathbf{x}_i - P_{\mathbf{v}}\mathbf{x}_i\|^2 &= \sum_{i=1}^{n} \left(\|\mathbf{x}_i\|^2 - \|P_{\mathbf{v}}\mathbf{x}_i\|^2 \right) \\
&= \sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - \sum_{i=1}^{n} \|(\mathbf{x}_i^\top \mathbf{v})\mathbf{v}\|^2 \\
&= \sum_{i=1}^{n} \|\mathbf{x}_i\|^2 - \sum_{i=1}^{n} (\mathbf{x}_i^\top \mathbf{v})^2
\end{aligned}$$

Then since the first term $\sum_{i=1}^{n} \|\mathbf{x}_i\|^2$ is constant with respect to $\mathbf{v}$, minimizing reconstruction error is equivalent to maximizing $\sum_{i=1}^{n} (\mathbf{x}_i^\top \mathbf{v})^2$, which is (up to an irrelevant positive constant factor $1/n$) the projected variance.

Another way to write this interpretation is that the reconstructed matrix $\tilde{\mathbf{X}}_k$ is the best rank-k approximation to $\mathbf{X}$ in the Frobenius norm. To see this, first note that (writing $\mathbf{X} = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$)

$$\tilde{\mathbf{X}}_k = \mathbf{X}\mathbf{V}_k\mathbf{V}_k^\top = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top \mathbf{V}_k \mathbf{V}_k^\top$$

By orthonormality, the product $\mathbf{v}_i^\top \mathbf{V}_k$ results in a k-dimensional row vector with 1 in the ith place and 0 everywhere else, i.e. $\mathbf{e}_i^\top$, as long as $i \leq k$. In this case,

$$\mathbf{v}_i^\top \mathbf{V}_k \mathbf{V}_k^\top = \mathbf{e}_i^\top \mathbf{V}_k^\top = (\mathbf{V}_k \mathbf{e}_i)^\top = \mathbf{v}_i^\top$$

If $i > k$, $\mathbf{v}_i^\top \mathbf{V}_k = \mathbf{0}^\top$, so the term disappears. Therefore we see that

$$\tilde{\mathbf{X}}_k = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top \mathbf{V}_k \mathbf{V}_k^\top = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

which is the best rank-k approximation to $\mathbf{X}$ by the Eckart-Young theorem.

1.8 Probabilistic PCA

We have seen probabilistic motivations or derivations of many of the methods discussed so far in this class. In a similar vein, **probabilistic PCA** (PPCA) is a **generative** model for PCA. Here we make the following assumptions about how the data were generated: for each datapoint i, there is a k-dimensional **latent variable**

$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

which we cannot observe, and the actual d-dimensional observation is distributed conditionally on this latent variable as

$$\mathbf{x}_i | \mathbf{z}_i \sim \mathcal{N}(\mathbf{\Lambda} \mathbf{z}_i + \boldsymbol{\mu}, \mathbf{\Psi})$$

Here $\mathbf{\Lambda} \in \mathbb{R}^{d \times k}$ and $\boldsymbol{\mu} \in \mathbb{R}^d$ are parameters to be estimated. Since $\mathbf{z}_i$ is Gaussian and $\mathbf{x}_i | \mathbf{z}_i$ is Gaussian, $\begin{bmatrix} \mathbf{x}_i \\ \mathbf{z}_i \end{bmatrix}$ is Gaussian, so its marginal $\mathbf{x}_i$ is Gaussian. In particular, by integrating out the latent variable

$$p(\mathbf{x}_i) = \int_{\mathbf{z}} p(\mathbf{x}_i, \mathbf{z}) \, \mathrm{d}\mathbf{z} = \int_{\mathbf{z}} p(\mathbf{x}_i | \mathbf{z}) p(\mathbf{z}) \, \mathrm{d}\mathbf{z}$$

one can show that

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Lambda}\mathbf{\Lambda}^\top + \mathbf{\Psi})$$

It is common to assume $\mathbf{\Psi} = \sigma^2 \mathbf{I}$. In this case, if we let $\sigma^2 \to 0$, we recover the original PCA solution in the sense that the columspace of $\hat{\mathbf{\Lambda}}_{\text{MLE}}$ approaches the PCA subspace (i.e. the columnspace of $\mathbf{V}_k$). [2]

[2] See Tipping and Bishop's original paper for derivations and more information.