Lecture 14: Unsupervised Learning: Dimensionality Reduction and Advanced Topics

Readings: ISL (Ch. 12), PRML[1] (Ch. 12), ESL (Ch. 14); code

Topics: Principal component analysis (PCA), variance maximization and reconstruction viewpoints, eigendecomposition and singular value decomposition, explained variance, probabilistic PCA, kernel PCA, autoencoders and variational autoencoders, advanced topics

1 Principal Component Analysis (PCA)

Given data $\{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^d$, we want to find a transformation from $\mathbb{R}^d$ to $\mathbb{R}^m$ with $m \leq d$ (often much smaller) to obtain a reduced representation of the data that retains its key information.

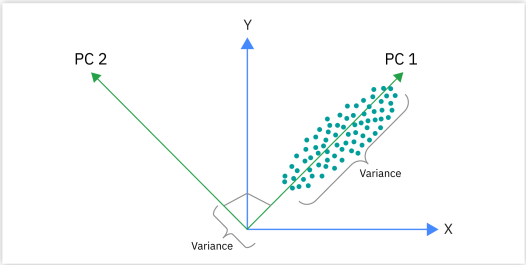
We focus on dimensionality reduction methods today. PCA refers to the process by which **principal components** are computed, and the subsequent use of these PCs in understanding the data.

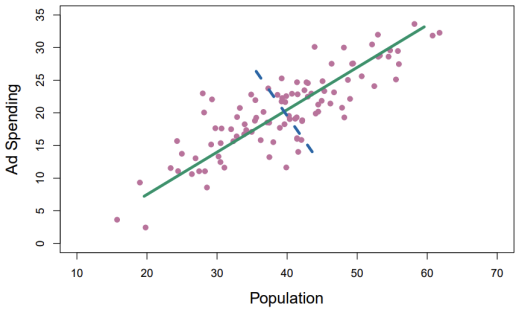
Goal of PCA: Find a low-dimensional representation of the data that captures as much information as possible using a **linear** transformation.

PCA is useful not only for visualization, but also for dimensionality reduction, compression, denoising, and preprocessing.

1.1 Intuition and Visualization

Let us think about how to summarize a two-dimensional dataset using a one-dimensional representation.





Intuition 1: PCA identifies directions in feature space along which the data exhibits the greatest variability. The first PC corresponds to the direction of maximum variance, the second

[1] In addition to ISL and ESL, we use Bishop's PRML as our reference for unsupervised learning, which provides more in-depth treatment of core methods with a unified probabilistic framework. ESL Ch. 14 covers a broader range of topics but with less depth on the fundamentals we emphasize. Also, note that Ch. 16 in a modern version of PRML covers similar material; see `https://www.bishopbook.com/`.

captures the next largest variance subject to orthogonality to the first, and so on.

Intuition 2: PCA can also be viewed as finding a low-dimensional subspace that best reconstructs the data. The first PC defines the one-dimensional subspace minimizing reconstruction error, and subsequent PCs reduce the remaining reconstruction error while remaining orthogonal to the previous ones.

1.2 Linear Algebra Review

Mathematically, PCA corresponds to an eigendecomposition of the sample covariance matrix. Before deriving this, let us recall some linear algebra facts.

For a square matrix $A \in \mathbb{R}^{d\times d}$:

1. **Eigenvector and eigenvalue:** A pair (u, λ) such that $u \neq 0$ and
$$Au = \lambda u.$$
2. **Diagonalizable:** A is diagonalizable if there exists a diagonal matrix Λ and invertible matrix P such that
$$A = P\Lambda P^{-1}.$$
3. **Symmetric:** A is symmetric if $A^\top = A$.
4. **Orthogonal:** A is orthogonal if $A^\top A = AA^\top = I$.

Theorem 1.1 (Spectral Theorem)**.** *A matrix $A \in \mathbb{R}^{d\times d}$ is **normal** ($A^\top A = AA^\top$) if and only if there exists an orthogonal matrix P such that*

$$P^{-1}AP = D,$$

where D is diagonal with the eigenvalues of A on its diagonal, and the columns of P are corresponding eigenvectors of A.

In particular, if $A = A^\top$ is real symmetric, then

$$A = U\Lambda U^\top,$$

where U is orthogonal and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ contains real eigenvalues.

1.3 Positive (Semi-)Definite Matrices

Positive semidefinite (PSD): A symmetric matrix A is PSD if

$$x^\top Ax \geq 0 \qquad \text{for all } x \in \mathbb{R}^d.$$

Equivalently, A is PSD if and only if all eigenvalues satisfy $\lambda_i \geq 0$.

Positive definite (PD): A symmetric matrix A is PD if

$$x^\top Ax > 0 \qquad \text{for all } x \neq 0.$$

Equivalently, A is PD if and only if all eigenvalues satisfy $\lambda_i > 0$.

For a PSD matrix with eigendecomposition $A = U\Lambda U^\top$, we arrange eigenvalues in decreasing order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0.$$

This ordering is crucial in PCA because we keep the eigenvectors corresponding to the largest eigenvalues.

1.4 Sample Covariance Matrix

Given data $\{x_i\}_{i=1}^N$, where $x_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d$, define the sample mean in each coordinate:

$$\bar{x}_j = \frac{1}{N}\sum_{i=1}^N x_{ij}.$$

Let $\bar{x} \in \mathbb{R}^d$ denote the vector of sample means.

Centering assumption: Without loss of generality, we assume $\bar{x} = 0$. This is achieved by replacing $x_i \leftarrow x_i - \bar{x}$ for all i.

For centered data, the sample covariance matrix is

$$S = \frac{1}{N}\sum_{i=1}^N x_i x_i^\top = \frac{1}{N}\mathbf{X}^\top\mathbf{X},$$

where $\mathbf{X} \in \mathbb{R}^{N\times d}$ is the data matrix whose i-th row is $x_i^\top$.

Elementwise,

$$S_{jk} = \frac{1}{N}\sum_{i=1}^N x_{ij}x_{ik}.$$

Since S is symmetric and PSD, it admits an orthogonal eigendecomposition.

1.5 Two Approaches to Dimensionality Reduction

How should we project high-dimensional data onto a lower-dimensional subspace?

There are two natural criteria:

1. **Maximize variance:** Choose the projection direction that preserves the most variation in the data.
2. **Minimize reconstruction error:** Choose the projection that minimizes the distance between the original and reconstructed data.

These two criteria are equivalent. Both lead to the same solution via eigenvectors of the covariance matrix S.

1.6 Two Formulations of PCA

Formulation 1: Maximum Variance.

Find a unit vector $u_1 \in \mathbb{R}^d$ such that the projected data has maximum variance:

$$u_1 = \arg\max_{u:\|u\|=1} u^\top S u.$$

For m dimensions, we find orthonormal vectors $u_1, \ldots, u_m$ sequentially.

Formulation 2: Minimum Reconstruction Error.

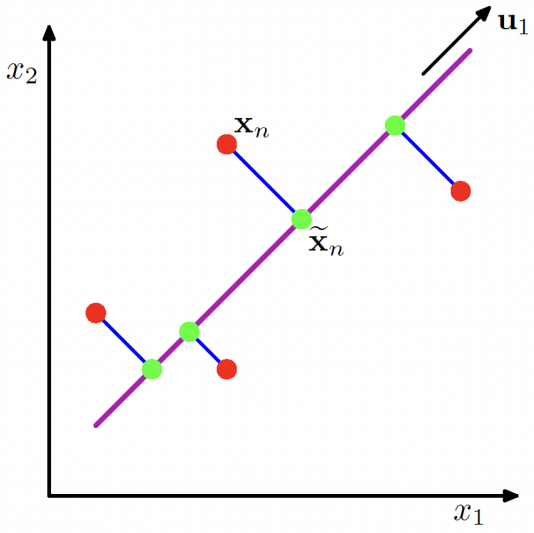
Find an m-dimensional subspace minimizing average squared distance from the data:

$$\min_{\{u_j\}_{j=1}^m} \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - \sum_{j=1}^{m} (u_j^\top x_i) u_j \right\|^2,$$

subject to orthonormality of $\{u_j\}$.

Proposition 1.2. *The two formulations yield the same solution: $u_1, \ldots, u_m$ are the top m eigenvectors of S, corresponding to the largest eigenvalues.*

Principal component analysis seeks a space of lower dimensionality, known as the principal subspace and denoted by the magenta line, such that the orthogonal projection of the data points (red dots) onto this subspace maximizes the variance of the projected points (green dots). An alternative definition of PCA is based on minimizing the sum-of-squares of the projection errors, indicated by the blue lines.



1.7 Derivation: Maximum Variance Formulation

Let us start with the case $m = 1$. We seek a linear transformation from $\mathbb{R}^d$ to $\mathbb{R}$:

$$z_i = u^\top x_i, \qquad u \in \mathbb{R}^d, \quad \|u\| = 1.$$

The goal is to choose u to maximize the sample variance of $\{z_i\}_{i=1}^N$.

Since the data are centered,

$$\bar{z} = u^\top \bar{x} = 0.$$

Therefore,

$$\begin{aligned} \mathrm{Var}(\{z_i\}) &= \frac{1}{N} \sum_{i=1}^{N} z_i^2 && (1) \\ &= \frac{1}{N} \sum_{i=1}^{N} (u^\top x_i)^2 && (2) \\ &= \frac{1}{N} \sum_{i=1}^{N} u^\top x_i x_i^\top u && (3) \\ &= u^\top S u. && (4) \end{aligned}$$

Hence we must solve

$$u_1 = \arg\max_{u:\|u\|=1} u^\top S u.$$

Using Lagrange multipliers, define

$$\mathcal{L}(u, \lambda) = u^\top S u + \lambda(1 - u^\top u).$$

The first-order condition gives

$$2Su - 2\lambda u = 0 \quad \Longrightarrow \quad Su = \lambda u.$$

Thus u must be an eigenvector of S. Moreover, if $Su = \lambda u$ and $\|u\| = 1$, then

$$u^\top S u = \lambda.$$

So the variance is maximized by choosing the eigenvector corresponding to the largest eigenvalue λ_1.

For higher dimensions, we repeat this procedure subject to orthogonality constraints, obtaining the eigenvectors corresponding to $\lambda_1, \ldots, \lambda_m$.

1.8 Derivation: Reconstruction Error Formulation

Consider an orthonormal basis $\{u_j\}_{j=1}^d$ for $\mathbb{R}^d$. Then each data point can be expanded as

$$x_i = \sum_{j=1}^{d} (u_j^\top x_i) u_j.$$

If we project onto the subspace spanned by the first m basis vectors, we obtain

$$\hat{x}_i^{(m)} = \sum_{j=1}^{m} (u_j^\top x_i) u_j.$$

The reconstruction error is

$$\frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x}_i^{(m)}\|^2.$$

By orthogonality, minimizing this error is equivalent to maximizing the variance captured by the first m directions. Therefore, the minimizing subspace is spanned by the top m eigenvectors of S.

This proves the equivalence of the two formulations.

1.9 PCA Solution via Eigendecomposition

Suppose

$$S = U \Lambda U^\top,$$

where $U = [u_1 | \cdots | u_d]$ and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$.

Then:

- The j-th principal component direction is u_j.

- The variance along the j-th principal component is λ_j.
- Collect the first m principal directions in

$$U_m = [u_1 | \cdots | u_m] \in \mathbb{R}^{d \times m}.$$

- The principal component score matrix is

$$\mathbf{Z}_m = \mathbf{X} U_m \in \mathbb{R}^{N \times m}.$$

Interpretation:

- u_1 points in the direction of greatest variance.
- u_2 points in the direction of greatest remaining variance, orthogonal to u_1.
- Eigenvectors with small eigenvalues often correspond to low-variance or noise directions.

Reconstruction:

$$\mathbf{X}_m = \mathbf{Z}_m U_m^\top = \mathbf{X} U_m U_m^\top.$$

For an individual point,

$$\hat{x}_i^{(m)} = \sum_{j=1}^{m} (u_j^\top x_i) u_j = U_m U_m^\top x_i.$$

1.10 PCA Algorithm

Principal Component Analysis (PCA)

1. **Input:** Data $\{x_i\}_{i=1}^N$, target dimension $m < d$.
2. Form the data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$.
3. **Center the data:** Compute

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i,$$

   and replace $x_i \leftarrow x_i - \bar{x}$ for all i.
4. Compute the sample covariance

$$S = \frac{1}{N} \mathbf{X}^\top \mathbf{X}.$$

5. Compute the eigendecomposition

$$S = U \Lambda U^\top.$$

6. Extract the top m eigenvectors:

$$U_m = [u_1 | \cdots | u_m].$$

7. Compute the PC scores

$$\mathbf{Z}_m = \mathbf{X} U_m.$$

8. **Return:** PC scores $\mathbf{Z}_m$, loadings U_m, and eigenvalues $\lambda_1, \ldots, \lambda_m$.

Computational complexity:

- Full eigendecomposition of S: $O(d^3)$.
- For $m \ll d$: iterative methods can compute the leading eigenvectors more efficiently.
- If $N \ll d$, it can be cheaper to work with $\mathbf{X}\mathbf{X}^\top$ instead.
- In practice, PCA is usually computed via SVD of $\mathbf{X}$.

1.11 PCA via Eigendecomposition vs SVD

Let the centered data matrix have singular value decomposition

$$\mathbf{X} = UDV^\top,$$

where $U \in \mathbb{R}^{N\times N}$, $V \in \mathbb{R}^{d\times d}$, and D is diagonal (rectangular) with singular values $d_1 \geq d_2 \geq \cdots \geq 0$.

Then

$$S = \frac{1}{N}\mathbf{X}^\top\mathbf{X} = \frac{1}{N}VD^\top DV^\top.$$

So:

- the PCA directions are the columns of V,
- the PCA eigenvalues are

$$\lambda_j = \frac{d_j^2}{N}.$$

Practice: Modern PCA implementations compute the SVD of $\mathbf{X}$ directly because it is numerically more stable than forming and diagonalizing S.

1.12 Explained Variance

Definition 1.3. The **proportion of variance explained (PVE)** by the k-th principal component is

$$\mathrm{PVE}_k = \frac{\lambda_k}{\sum_{j=1}^{d}\lambda_j}.$$

This quantifies how much of the total variance is captured by component k.

Proposition 1.4. *For centered data,*

$$\sum_{k=1}^{m} PVE_k = 1 - \frac{RSS}{TSS},$$

where

$$TSS = \sum_{i=1}^{N}\|x_i\|^2, \qquad RSS = \sum_{i=1}^{N}\|x_i - \hat{x}_i^{(m)}\|^2.$$

Proof. See Exercise 14.1. □

This is analogous to the R^2 statistic in regression: it measures the fraction of total variance captured by the PCA subspace.

1.13 Why Scaling the Variables Matters

Because it is undesirable for principal components to depend on arbitrary measurement units, we often scale each variable to have standard deviation one before performing PCA.

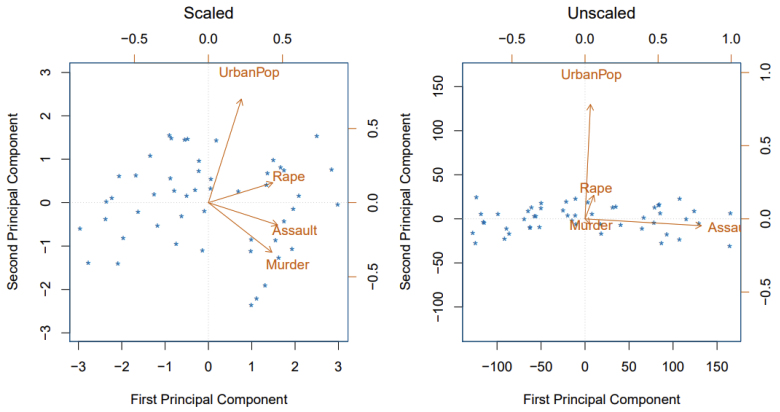


FIGURE 12.4. *Two principal component biplots for the* `USArrests` *data.* Left: *the same as Figure 12.1, with the variables scaled to have unit standard deviations.* Right: *principal components using unscaled data.* `Assault` *has by far the largest loading on the first principal component because it has the highest variance among the four variables. In general, scaling the variables to have standard deviation one is recommended.*

If variables are measured in comparable units and scale carries meaning, one may choose not to standardize.

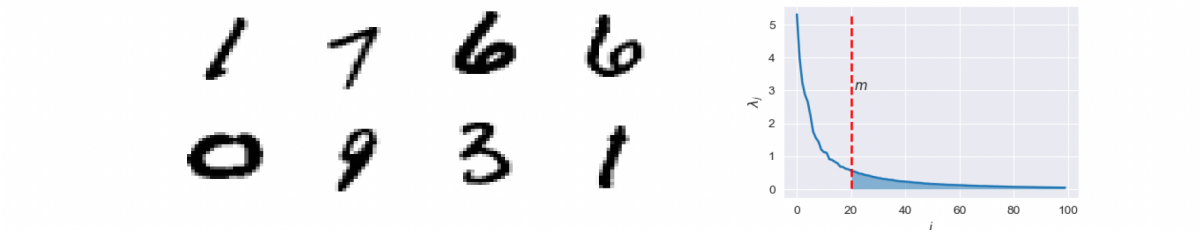
1.14 Choosing the Number of Components

How should we choose m?

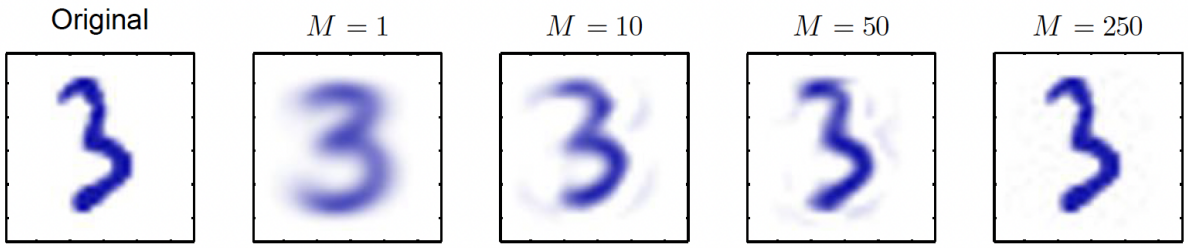
1. **Scree plot:** Plot $\lambda_1, \lambda_2, \ldots$ and look for an elbow.
2. **Cumulative PVE:**
$$\text{PVE}(m) = \frac{\sum_{j=1}^{m} \lambda_j}{\sum_{j=1}^{d} \lambda_j}.$$
Common thresholds are 80% or 90%.
3. **Cross-validation:** If PCA is used as preprocessing for supervised learning, then m can be treated as a tuning parameter.

1.15 PCA on MNIST

Let us consider the MNIST dataset of hand-written digits. Performing a PCA on the dataset ($d = 784$) we obtain a eigenvalue distribution as shown in the following figure. Notice the decay of the eigenvalues of the sample covariance matrix (right plot). We can choose m to be sufficiently large so that the projection error $\sum_{j=m+1}^{d} \lambda_j$ (shaded area) is small enough.



PCA reconstructions obtained by retaining the first M principal components for various values of M:



1.16 Applications of PCA

- Data visualization
- Preprocessing and decorrelation
- Compression
- Denoising
- Feature extraction
- Principal components regression
- Imputation and matrix completion

Remark 1.5. PCA identifies orthogonal directions of maximum variance. A related method, **Independent Component Analysis (ICA)**, seeks statistically independent rather than merely uncorrelated components.

1.17 PCA as Whitening

Given $S = U\Lambda U^\top$, define the whitening transform

$$x_i' = \Lambda^{-1/2} U^\top x_i, \qquad \text{equivalently} \qquad \mathbf{X}' = \mathbf{X} U \Lambda^{-1/2}.$$

For full whitening, we assume $S \succ 0$. If S is rank-deficient, use the pseudoinverse:

$$\Lambda^\dagger = \operatorname{diag}(\lambda_1^\dagger, \ldots, \lambda_d^\dagger), \qquad \lambda_i^\dagger = \begin{cases} \lambda_i^{-1/2}, & \lambda_i > 0, \\ 0, & \lambda_i = 0. \end{cases}$$

Then the whitened data satisfy

$$\frac{1}{N}\sum_i x_i' = 0, \qquad \frac{1}{N}\sum_i x_i'(x_i')^\top = I.$$

Whitening removes correlations and standardizes variances, but it does not imply independence.

1.18 Ridge Regression in PC Space

Let $\mathbf{X} = U\Sigma V^\top$ be the SVD of $\mathbf{X}$. Then ridge fitted values are

$$\begin{aligned} \mathbf{X}\hat{\beta}_{\text{ridge}} &= \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda I)^{-1}\mathbf{X}^\top y \\ &= U\Sigma^2(\Sigma^2 + \lambda I)^{-1}U^\top y \\ &= \sum_{j=1}^{\min(N,d)} \frac{\sigma_j^2}{\sigma_j^2 + \lambda} u_j u_j^\top y. \end{aligned}$$

Compare this to least squares:

$$\mathbf{X}\hat{\beta}_{LS} = UU^\top y = \sum_{j=1}^{\min(N,d)} u_j u_j^\top y.$$

Thus ridge performs **non-uniform shrinkage**:

- directions with small singular values are shrunk more,
- directions with large singular values are shrunk less.

2 Probabilistic PCA (PPCA)

2.1 Motivation

Question: Can we give principal components a probabilistic interpretation?

Benefits of a probabilistic formulation:

- handling missing data naturally,
- quantifying uncertainty in latent representations,
- likelihood-based model comparison,
- Bayesian extensions,
- EM-based fitting.

Key idea: Model data as generated from a low-dimensional latent variable via a linear transformation plus Gaussian noise.

2.2 Model Definition

Definition 2.1 (Probabilistic PCA)**.** The PPCA model assumes

$$z \sim \mathcal{N}(0, I_m),$$
$$x \mid z \sim \mathcal{N}(Wz + \mu, \sigma^2 I_d),$$

where

- $z \in \mathbb{R}^m$ is a latent variable,
- $W \in \mathbb{R}^{d \times m}$,
- $\mu \in \mathbb{R}^d$,
- $\sigma^2 > 0$ is isotropic noise variance.

In PPCA the latent variable is continuous and Gaussian. This differs from mixture models, where latent variables are discrete cluster labels.

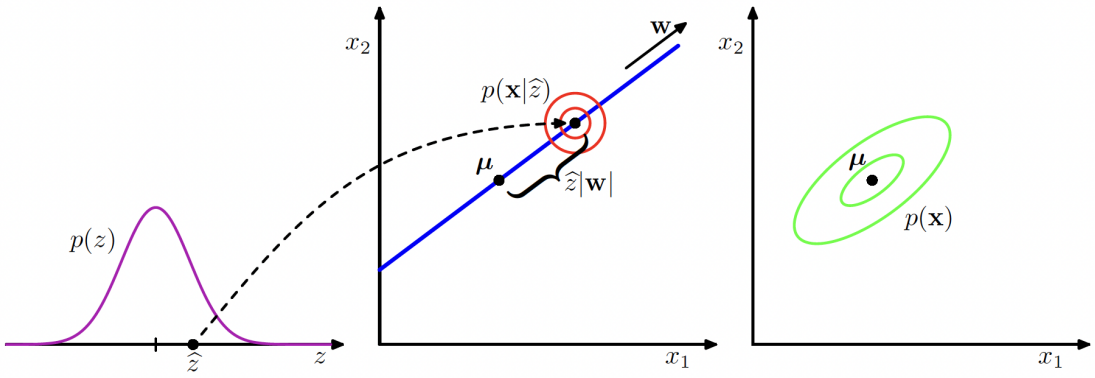


Figure 16.7 An illustration of the generative view of a probabilistic PCA model for a two-dimensional data space and a one-dimensional latent space. An observed data point $\mathbf{x}$ is generated by first drawing a value $\widehat{z}$ for the latent variable from its prior distribution $p(z)$ and then drawing a value for $\mathbf{x}$ from an isotropic Gaussian distribution (illustrated by the red circles) having mean $\mathbf{w}\widehat{z} + \boldsymbol{\mu}$ and covariance $\sigma^2\mathbf{I}$. The green ellipses show the density contours for the marginal distribution $p(\mathbf{x})$.

2.3 Marginal Distribution

Write

$$x = \mu + Wz + \mathrm{Var}epsilon, \qquad \mathrm{Var}epsilon \sim \mathcal{N}(0, \sigma^2 I_d), \qquad \mathrm{Var}epsilon \perp z.$$

Then

$$\mathbb{E}[x] = \mu, \qquad \mathrm{Cov}(x) = W\mathbb{E}[zz^\top]W^\top + \mathrm{Cov}(\mathrm{Var}epsilon) = WW^\top + \sigma^2 I_d.$$

Hence the marginal distribution is

$$p(x) = \mathcal{N}(\mu, C), \qquad C = WW^\top + \sigma^2 I_d.$$

PPCA can be viewed as a constrained Gaussian model: instead of learning a full covariance matrix, it models covariance as a low-rank structure plus isotropic noise.

2.4 Posterior Distribution

We now derive $p(z \mid x)$. Using Bayes' rule,

$$\begin{aligned}\log p(z \mid x) &= \log p(z) + \log p(x \mid z) + \text{const} \\ &= -\frac{1}{2} z^\top z - \frac{1}{2\sigma^2} \|x - \mu - Wz\|^2 + \text{const.}\end{aligned}$$

Expanding the quadratic term,

$$\|x - \mu - Wz\|^2 = (x-\mu)^\top (x-\mu) - 2z^\top W^\top (x-\mu) + z^\top W^\top W z.$$

Collecting terms in z,

$$\log p(z \mid x) = -\frac{1}{2} z^\top \left(I_m + \frac{1}{\sigma^2} W^\top W \right) z + \frac{1}{\sigma^2} z^\top W^\top (x-\mu) + \text{const.}$$

Define

$$M = W^\top W + \sigma^2 I_m.$$

Then completing the square yields

$$p(z \mid x) = \mathcal{N}\big(M^{-1} W^\top (x-\mu),\ \sigma^2 M^{-1}\big).$$

2.5 Connection to Bayesian Linear Regression

The PPCA model

$$z \sim \mathcal{N}(0, I_m), \qquad x \mid z \sim \mathcal{N}(Wz + \mu, \sigma^2 I_d)$$

can be viewed as a Bayesian linear regression model in the latent variable z.

In Bayesian linear regression,

$$y = A\theta + \text{Var}epsilon, \qquad \theta \sim \mathcal{N}(0, I), \qquad \text{Var}epsilon \sim \mathcal{N}(0, \sigma^2 I),$$

and the posterior is

$$p(\theta \mid y) = \mathcal{N}\Big((A^\top A + \sigma^2 I)^{-1} A^\top y,\ \sigma^2 (A^\top A + \sigma^2 I)^{-1} \Big).$$

The PPCA posterior has the same algebraic form, with

$$A \leftrightarrow W, \qquad y \leftrightarrow x - \mu, \qquad \theta \leftrightarrow z.$$

> PPCA is the "dual" of Bayesian linear regression: in regression we infer parameters given observations, while in PPCA we infer latent coordinates given observations.

2.6 MLE for PPCA

The PPCA parameters can be estimated by maximum likelihood. One can derive closed-form MLEs:

$$W_{\text{MLE}} = U_m (\Lambda_m - \sigma^2 I_m)^{1/2} R, \qquad \mu_{\text{MLE}} = \bar{x}, \qquad \sigma^2_{\text{MLE}} = \frac{1}{d-m} \sum_{j=m+1}^{d} \lambda_j,$$

where

- U_m contains the top m eigenvectors of S,
- $\Lambda_m = \text{diag}(\lambda_1, \ldots, \lambda_m)$,
- R is an arbitrary orthogonal $m \times m$ rotation matrix.

Why is R arbitrary? Because the marginal covariance depends only on $WW^\top$, not on the particular basis of the latent space.

2.7 Posterior Mean and PPCA Scores

Once the parameters are estimated, the posterior mean of z given x is

$$\mathbb{E}[z \mid x] = M_{\text{MLE}}^{-1} W_{\text{MLE}}^\top (x - \bar{x}),$$

where

$$M_{\text{MLE}} = W_{\text{MLE}}^\top W_{\text{MLE}} + \sigma_{\text{MLE}}^2 I_m.$$

This is the **PPCA score** of x.

Relationship to standard PCA:

- PCA score:
$$z_{\text{PCA}} = U_m^\top (x - \bar{x}).$$
- PPCA score:
$$z_{\text{PPCA}} = M_{\text{MLE}}^{-1} W_{\text{MLE}}^\top (x - \bar{x}).$$

The coordinates differ because PPCA includes scaling, noise, and rotational freedom, but the subspace is the same.

2.8 Noiseless Limit

If we treat σ^2 as a small parameter and let $\sigma^2 \to 0$, then

$$W_{\text{MLE}} \to U_m \Lambda_m^{1/2} R.$$

In this limit, the PPCA model recovers standard PCA. The reconstruction operator becomes

$$\hat{x}_{\text{proj}} = \bar{x} + W_{\text{MLE}} (W_{\text{MLE}}^\top W_{\text{MLE}})^{-1} W_{\text{MLE}}^\top (x - \bar{x}),$$

which simplifies to

$$\hat{x}_{\text{proj}} = \bar{x} + U_m U_m^\top (x - \bar{x}).$$

So PPCA preserves the geometry of PCA while adding a probabilistic interpretation.

3 Kernel PCA

3.1 Motivation

So far, our dimensionality reduction methods have been linear:

- PCA: deterministic, variance-maximizing
- PPCA: probabilistic, linear-Gaussian

What if the data lie on a nonlinear manifold?

One approach is **Kernel PCA**: apply PCA after a nonlinear feature mapping.

3.2 Kernel PCA

Let $\phi : \mathbb{R}^d \to \mathbb{R}^M$ be a feature map, possibly with M very large or infinite.

Kernel PCA

1. Choose a feature map ϕ, usually implicitly through a kernel.
2. Form the design matrix Φ with rows $\phi(x_i)^\top$.
3. Center Φ in feature space.
4. Compute the covariance matrix
$$S_\phi = \frac{1}{N}\Phi^\top\Phi.$$
5. Perform PCA in feature space.

The kernel trick avoids explicit computation of ϕ. Use a kernel

$$k(x, x') = \phi(x)^\top\phi(x').$$

Common examples:

- Polynomial kernel:
$$k(x, x') = (1 + x^\top x')^p.$$
- Gaussian RBF kernel:
$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right).$$

Kernel PCA is a nonlinear extension of PCA that can capture curved structure in the data.

4 Autoencoders (AEs)

4.1 From PCA to Autoencoders

Recall that in PCA, the principal component scores are

$$\mathbf{Z} = \mathbf{X}U.$$

If we retain only the first m components, then

- **Encoding:**
$$\mathbf{Z}_m = \mathbf{X}U_m \in \mathbb{R}^{N\times m},$$
- **Decoding:**
$$\mathbf{X}_m = \mathbf{Z}_m U_m^\top \in \mathbb{R}^{N\times d}.$$

This is a linear encoder-decoder model.

The low-dimensional representation $\mathbf{Z}_m$ is called the **latent representation** or **latent code**.

4.2 Encoding and Decoding Maps in PCA

For PCA:

- **Encoder**
$$T_{\text{enc}}(x) = U_m^\top x \in \mathbb{R}^m,$$
- **Decoder**
$$T_{\text{dec}}(z) = U_m z \in \mathbb{R}^d.$$

Thus
$$x \xrightarrow{T_{\text{enc}}} z = U_m^\top x \xrightarrow{T_{\text{dec}}} \hat{x} = U_m U_m^\top x.$$

⚠ These are linear maps. Can we generalize them to nonlinear ones?

4.3 Autoencoders

Definition 4.1 (Autoencoder)**.** An autoencoder consists of two parameterized maps:

- **Encoder:**
$$T_{\text{enc}}(x;\theta) : \mathbb{R}^d \to \mathbb{R}^m,$$
- **Decoder:**
$$T_{\text{dec}}(z;\phi) : \mathbb{R}^m \to \mathbb{R}^d.$$

Typically $m < d$, so the latent representation is lower-dimensional than the input.

The training objective is to minimize reconstruction error:
$$\min_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} \|x_i - T_{\text{dec}}(T_{\text{enc}}(x_i;\theta);\phi)\|^2 .$$

Training is usually done using SGD and backpropagation.

4.4 Properties and Applications

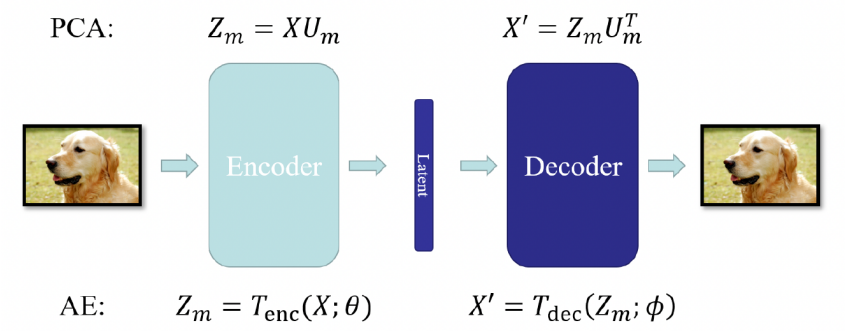
Advantages over PCA:

- can capture nonlinear structure,
- more flexible representations,
- can use deep architectures.

Applications:

- visualization,
- feature learning,
- denoising,
- compression,
- anomaly detection,
- pretraining.

Remark 4.2. A linear autoencoder recovers the same subspace as PCA, though not necessarily the same orthogonal basis.



💡 Both PCA and autoencoders can be viewed as compression-decompression algorithms using latent representations.

4.5 Variational Autoencoders (VAEs)

Ordinary autoencoders are deterministic. Variational autoencoders make the latent space probabilistic.

The generative model is

$$p_\phi(x, z) = p_\phi(x \mid z)p(z),$$

where typically

$$p(z) = \mathcal{N}(0, I).$$

The encoder outputs an approximate posterior

$$q_\theta(z \mid x) = \mathcal{N}(z; \mu_\theta(x), \Sigma_\theta(x)).$$

Training maximizes the **evidence lower bound (ELBO)** on $\log p_\phi(x)$.

💡 VAEs are a nonlinear probabilistic extension of PPCA and autoencoders.

5 Summary

5.1 Dimensionality Reduction Methods

Method	Type	Prob.?	Comments
PCA	Linear	No	Orthogonal projection, maximizes variance
PPCA	Linear	Yes	Linear-Gaussian latent variable model
ICA	Linear	Yes	Independent non-Gaussian sources
Kernel PCA	Nonlinear	No	PCA in feature space via kernels
AE	Nonlinear	No	Deterministic encoder-decoder
VAE	Nonlinear	Yes	Probabilistic nonlinear latent model

5.2 Broader Perspective

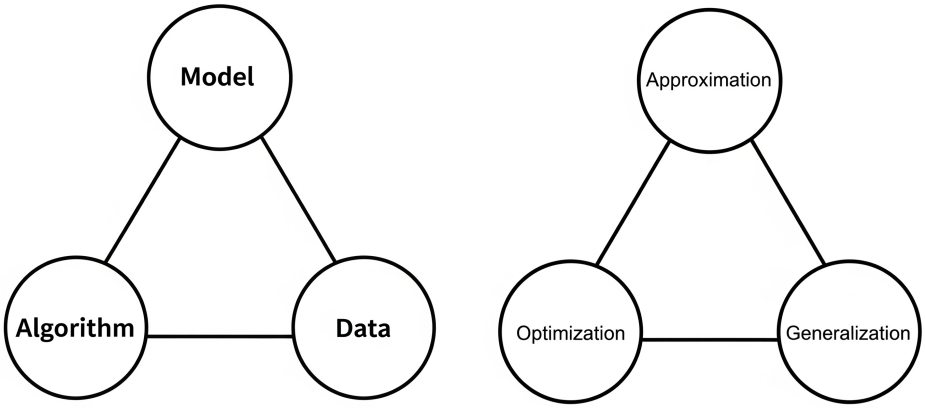
- **Clustering and mixture models:** K-means (hard) and GMMs (soft, probabilistic).
- **Dimensionality reduction:** PCA → PPCA → autoencoders → VAEs.

💡 Unsupervised learning is exploratory. Latent variables help explain structure and variation in observed data.

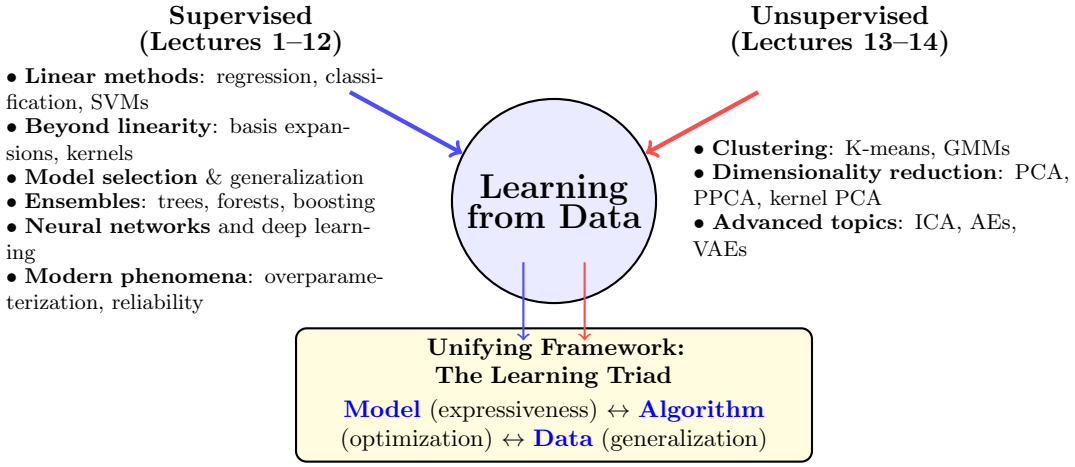
5.3 The Full Circle

We began the course with the broad distinction between supervised and unsupervised learning:

Supervised Learning	**Unsupervised Learning**
Regression	Clustering
Classification	Dimensionality reduction
Function approximation	Distribution learning



Since then, we have enriched this picture with a wide range of methods and principles.



All machine learning methods, supervised or unsupervised, involve balancing model complexity, algorithmic efficiency, and generalization.

6 Exercises

1. Proportion of Variance Explained by Principal Components

Let $\{x_i\}_{i=1}^N$ be centered data with covariance

$$S = \frac{1}{N}\sum_i x_i x_i^\top.$$

Suppose

$$S = U\Lambda U^\top,$$

where $U = [u_1, \dots, u_d]$ and $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_d)$ with

$$\lambda_1 \geq \cdots \geq \lambda_d \geq 0.$$

(a) Show that the total variance equals $\operatorname{tr}(S) = \sum_{j=1}^d \lambda_j$, and that the variance captured by projecting onto the first m PCs is $\sum_{j=1}^m \lambda_j$.

(b) Define

$$\mathrm{PVE}(m) = \frac{\sum_{j=1}^m \lambda_j}{\sum_{j=1}^d \lambda_j}.$$

Show that $\mathrm{PVE}(m)$ is invariant under orthogonal changes of coordinates.

(c) Let $U_m = [u_1, \dots, u_m]$ and $\hat{x}_i^{(m)} = U_m U_m^\top x_i$. Show that

$$\mathrm{PVE}(m) = 1 - \frac{\sum_{i=1}^N \|x_i - \hat{x}_i^{(m)}\|^2}{\sum_{i=1}^N \|x_i\|^2}.$$

Interpret this formula.

2. Inductive Proof for PCA Subspace

Show by induction that the m-dimensional subspace maximizing projected variance is spanned by the m eigenvectors corresponding to the largest eigenvalues of S.

(a) Base case $m = 1$: show that the maximizing direction satisfies $Su_1 = \lambda_1 u_1$.

(b) Inductive step: assume the claim for m, and introduce u_{m+1} orthogonal to the previous directions.

(c) Form the Lagrangian and derive the first-order condition.

(d) Show that the maximizing u_{m+1} corresponds to λ_{m+1}.

3. General Latent Prior $\Rightarrow$ Identical Marginal $p(x)$

Consider PPCA with

$$p(x \mid z) = \mathcal{N}(x \mid Wz + \mu, \sigma^2 I),$$

but replace the prior $z \sim \mathcal{N}(0, I)$ by $z \sim \mathcal{N}(m, \Sigma_z)$, where $\Sigma_z \succ 0$.

(a) Derive the marginal over x and show that it is Gaussian.

(b) Let $\Sigma_z = LL^\top$ and define $z' = L^{-1}(z - m)$. Rewrite the model in standard PPCA form.

(c) Conclude that the family of marginals $p(x)$ is unchanged up to reparameterization.

4. Least-Squares Reconstruction under PPCA

In PPCA, $x \mid z \sim \mathcal{N}(Wz + \mu, \sigma^2 I)$ with $z \sim \mathcal{N}(0, I)$.

(a) Derive the $z^\star$ minimizing $\|x - \mu - Wz\|^2$.

(b) Substitute $z^\star$ to express the reconstructed point $\hat{x}$.

(c) Using the MLE form

$$W = U_m(\Lambda_m - \sigma^2 I)^{1/2} R,$$

show that the reconstruction operator reduces to $U_m U_m^\top$. Hence PPCA reconstruction coincides with PCA projection[a].

[a] **This is beautiful!** Probabilistic structure does not complicate the geometry – the rotation R and noise variance σ^2 in W_{MLE} elegantly cancel, leaving pure projection $U_m U_m^T$. PPCA *enriches* PCA (adding uncertainty quantification and likelihood-based inference) while preserving its geometric essence. Complexity simplifies when we ask the right question.

5. Experimental

(a) **Two approaches to compute PVE.** Solve Exercise 12.6.8 in ISL.

(b) **PCA on MNIST.** Work through: `https://colab.research.google.com/github/uu-sml/course-apml-public/blob/master/exercises/Session11.ipynb`

(c) **PPCA on MNIST.** Work through: `https://colab.research.google.com/github/uu-sml/course-apml-public/blob/master/lab/PPCA.ipynb`

7 Appendix: Advanced Topics

7.1 Independent Component Analysis (ICA)

Motivation: PCA finds uncorrelated components. ICA goes further and seeks statistically independent components.

- Observed data:
$$x = As,$$
where $x \in \mathbb{R}^d$,
- $s \in \mathbb{R}^d$ are latent independent sources,
- $A \in \mathbb{R}^{d \times d}$ is an unknown invertible mixing matrix.

The goal is to recover an unmixing matrix

$$W = A^{-1}$$

so that

$$\hat{s} = Wx$$

recovers the underlying sources.

Example: the cocktail party problem, where multiple microphones record mixtures of multiple speakers.

Difference from PCA:

- PCA: orthogonal transformation, uncorrelated components.
- ICA: general linear transformation, independent components.

7.2 Visualization and Representation Methods

Other useful unsupervised learning tools include:

- **t-SNE, UMAP:** nonlinear embedding methods for visualization,
- **PCA, MDS:** classical geometry-preserving methods,
- **NMF:** nonnegative matrix factorization for interpretable parts-based representations,
- **SOMs, Isomap, LLE:** neural and geometric manifold learning methods.

These methods help us visualize, compress, and interpret high-dimensional data.

7.3 Generative Modeling

Generative models learn data distributions and can synthesize new samples.

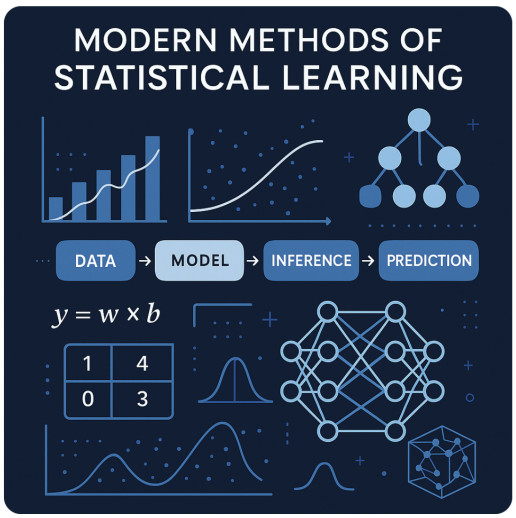
Examples:

- **VAEs:** probabilistic autoencoders,
- **GANs:** adversarial training for data generation,
- **Normalizing flows:** invertible neural density models,
- **Diffusion models:** learn to reverse a noising process.

Applications include image, text, audio, and scientific data generation.

Generate a visually engaging illustration for a master's-level course titled "Modern Methods of Statistical Learning."

Image created



Thank you for being part of this course.
Wishing you success in applying what you've learned!