STA 314: Statistical Methods for Machine Learning I

Lecture 8 - Principle Components Analysis

Chris J. Maddison

University of Toronto

Research opportunity

I am looking for undergraduates to study the dynamics of large language model training.

Please fill out this form: `https://docs.google.com/forms/d/e/1FAIpQLSf6xRSgxIiKOXx7X9ovqP73B4dh9PlbcEobXJd8-vDIbMmKxg/viewform?usp=dialog`.

Recall

- We showed that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ shifted by $\boldsymbol{\mu}$ and "scaled" by $\boldsymbol{\Sigma}^{\frac{1}{2}}$.
- So, how can you think of "scaling" space by the square root of a matrix?
- Recall, for a PSD matrix $\boldsymbol{\Sigma}$, its spectral decomposition is

$$\boldsymbol{\Sigma} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^T$$

- Since $\mathbf{Q}$ is orthonormal, we have $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$, and that:

$$\boldsymbol{\Sigma}^{\frac{1}{2}} = \mathbf{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{Q}^T$$

Matrix Square Roots & the Multivariate Gaussian

- We want to understand what it means to scale space by $\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{x}$.
- Multiplying a vector $\mathbf{x}$ by $\mathbf{Q}^\top\mathbf{x}$ is the same as projecting $\mathbf{x}$ onto the columns of $\mathbf{Q}$, so this is like rotating spaces so that the basis of $\mathbf{Q}$ becomes the standard basis.
- Since $\boldsymbol{\Lambda}$ is diagonal, it is easy to calculate

$$\boldsymbol{\Lambda}^{\frac{1}{2}} = \begin{pmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_D} \end{pmatrix}$$

  and multiplying by is the same as scaling the (current) standard basis by $\sqrt{\lambda_i}$.
- Multiplying by $\mathbf{Q}$ rotates the standard basis back into the basis of $\mathbf{Q}$.

Matrix Square Roots & the Multivariate Gaussian

- To to summarize, you can think of scaling space by $\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{x}$ as the effect of expanding space along the eigenvectors of $\boldsymbol{\Sigma}^{\frac{1}{2}}$ by their corresponding eigenvalues.
- So multivariate "scaling" has both magnitude and direction.

Back to PCA

- Back to principal component analysis (PCA)
- Dimensionality reduction: map data to a lower dimensional space
- PCA is a linear model. It's useful for understanding lots of other algorithms.
- PCA is about finding linear structure in data.

Recall: Multivariate Parameters

- Setup: given a iid dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \subset \mathbb{R}^D$.
- N instances/observations/examples

$$\mathbf{X} = \begin{bmatrix} [\mathbf{x}^{(1)}]^\top \\ [\mathbf{x}^{(2)}]^\top \\ \vdots \\ [\mathbf{x}^{(N)}]^\top \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_D^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \cdots & x_D^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(N)} & x_2^{(N)} & \cdots & x_D^{(N)} \end{bmatrix}$$

Recall: Mean and Covariance Estimators

- We can estimate mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ using these sample approximations:

$$\text{Sample mean:} \quad \hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^{(i)}$$

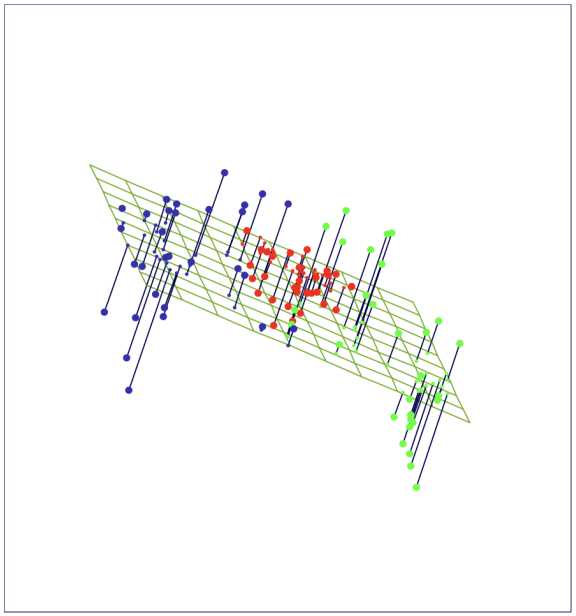
Sample covariance:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top$$

- $\hat{\boldsymbol{\mu}}$ quantifies where your data is located in space **(shift)**
- $\hat{\boldsymbol{\Sigma}}$ quantifies the shape of spread of your data points **(scale)**

Goal: Low dimensional representation

- In practice, even though data is very high dimensional, its important features can be accurately captured in a low dimensional subspace.



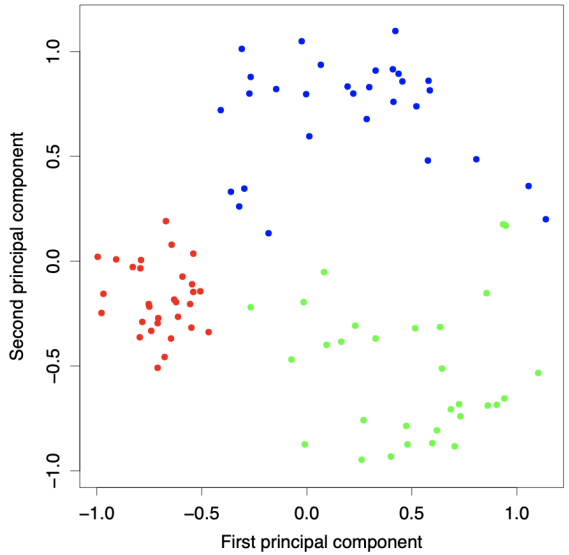


Image credit: Elements of Statistical Learning

- Find a low dimensional representation of your data.
  - Computational benefits
  - Interpretability, visualization
  - Generalization

Goal: Low dimensional representation

More specifically, our goal:

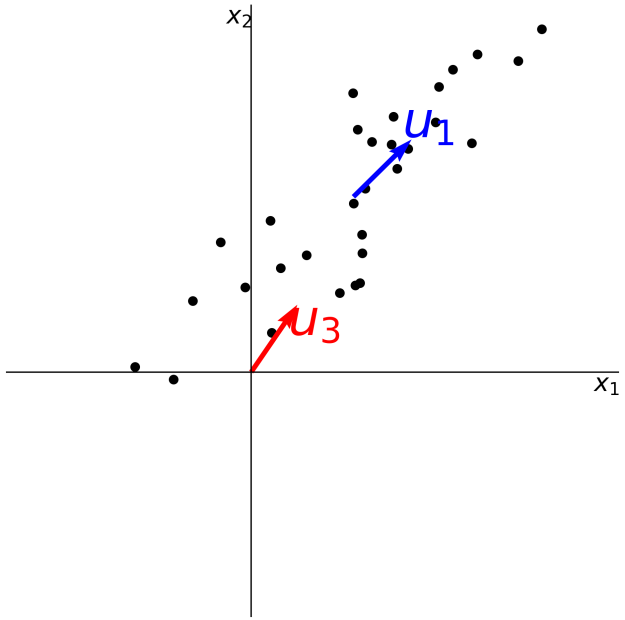
- Given a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \subset \mathbb{R}^D$
- Find a K-dimensional subspace $\mathcal{S} \subset \mathbb{R}^D$ such that $\mathbf{x}^{(n)} - \hat{\boldsymbol{\mu}}$ is "well-represented" by its projection onto $\mathcal{S}$

PCA step-by-step:

- Center data
- Project onto $\mathcal{S}$
  - Coordinates in $\mathcal{S}$ give us a low dimensional representation.
- Add back mean.
  - This gives us a low dimensional reconstruction of the data to visualize our approximation.

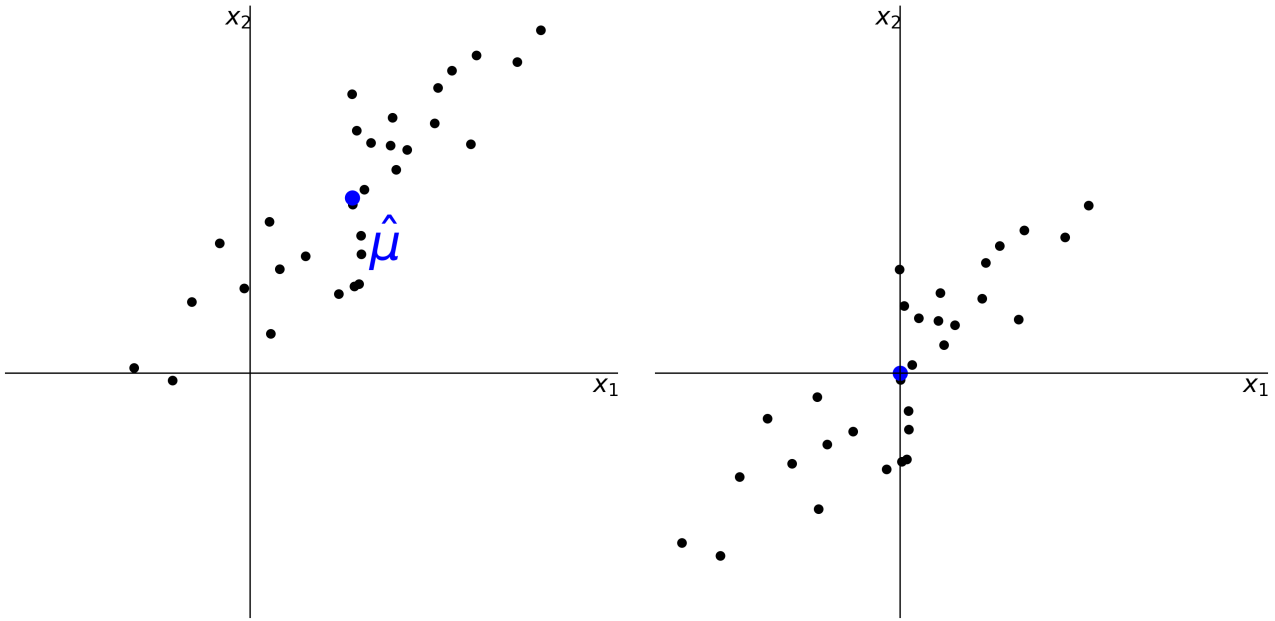
Now, let's see what this looks like in 2D.

We are looking for directions



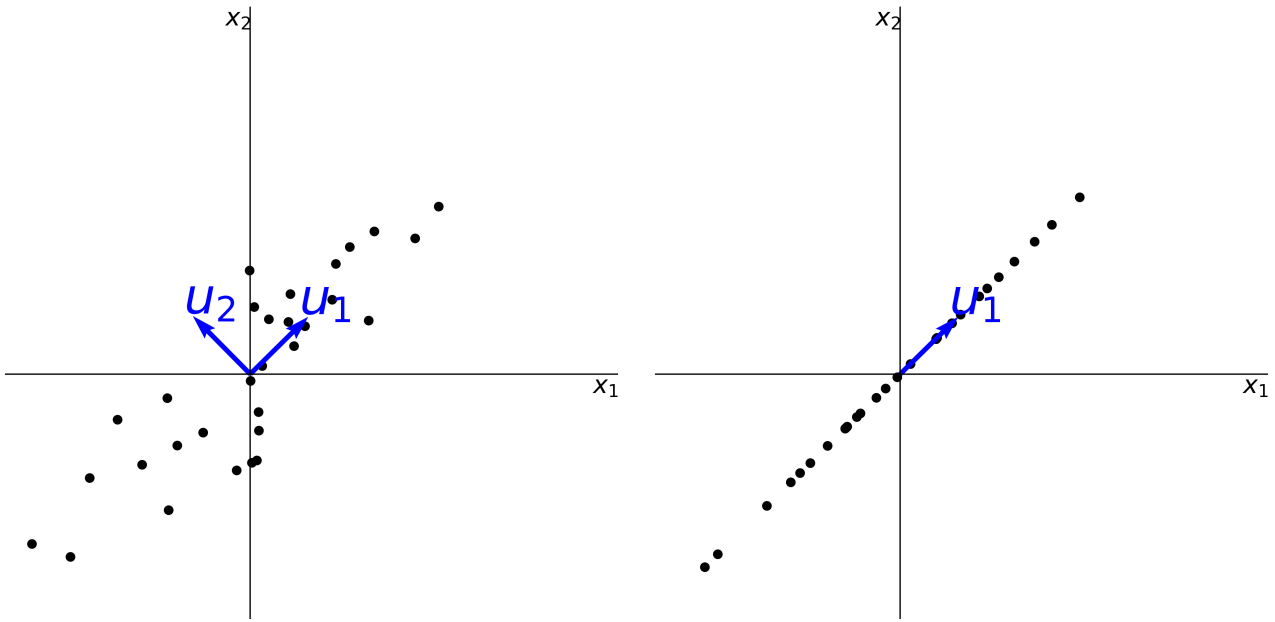
For example, in a 2-dimensional problem, we are looking for the unit vector $\boldsymbol{u}_1$ along which the data is well represented. We don't want location of data to influence our calculations, i.e., we are **not** interested in $\mathbf{u}_3$.

First step: Center data



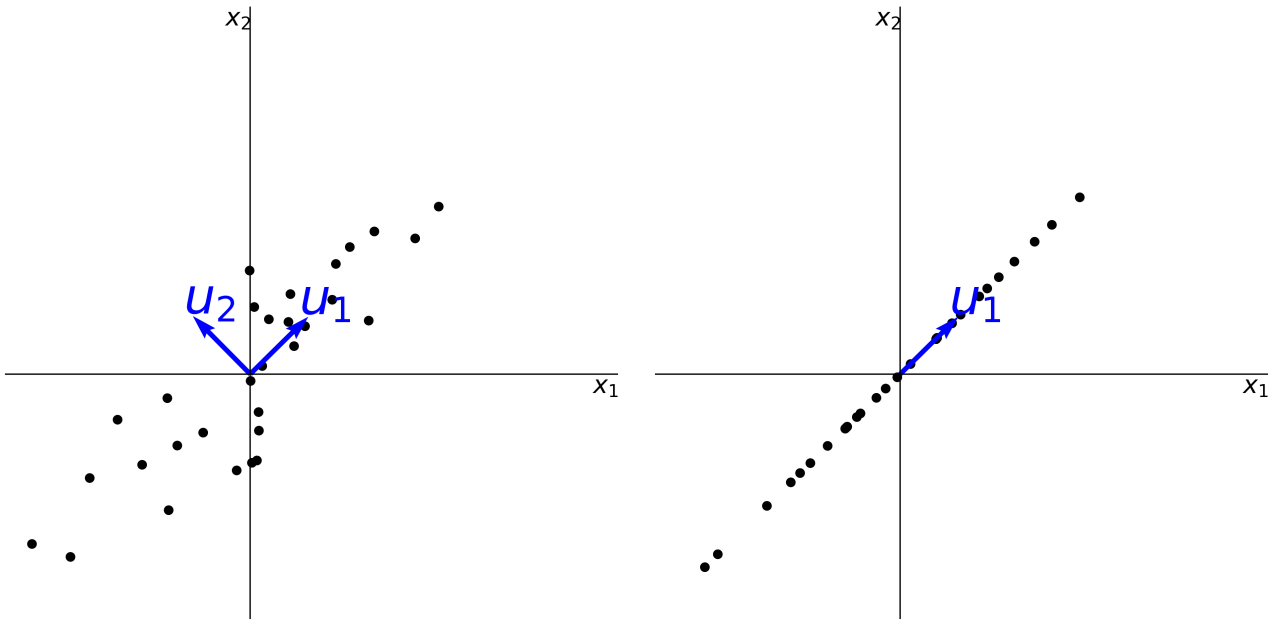
- We are only interested in finding the direction of highest variance. Directions pass through the origin.
- So, we need to center our data since we don't want location of data to influence our calculation of direction.

Second step: Project onto subspace spanned by $\mathbf{u}_1$



- Let $\mathbf{u}_1$ be the direction of highest variance (we will discuss how to find) and $\mathbf{u}_2$ be an orthogonal direction to $\mathbf{u}_1$.
- We want to reduce dimensionality of the data by projecting onto $\mathbf{u}_1$. This is just a multivariate "scale" by 0 in the pruned directions. You already know how to do this!

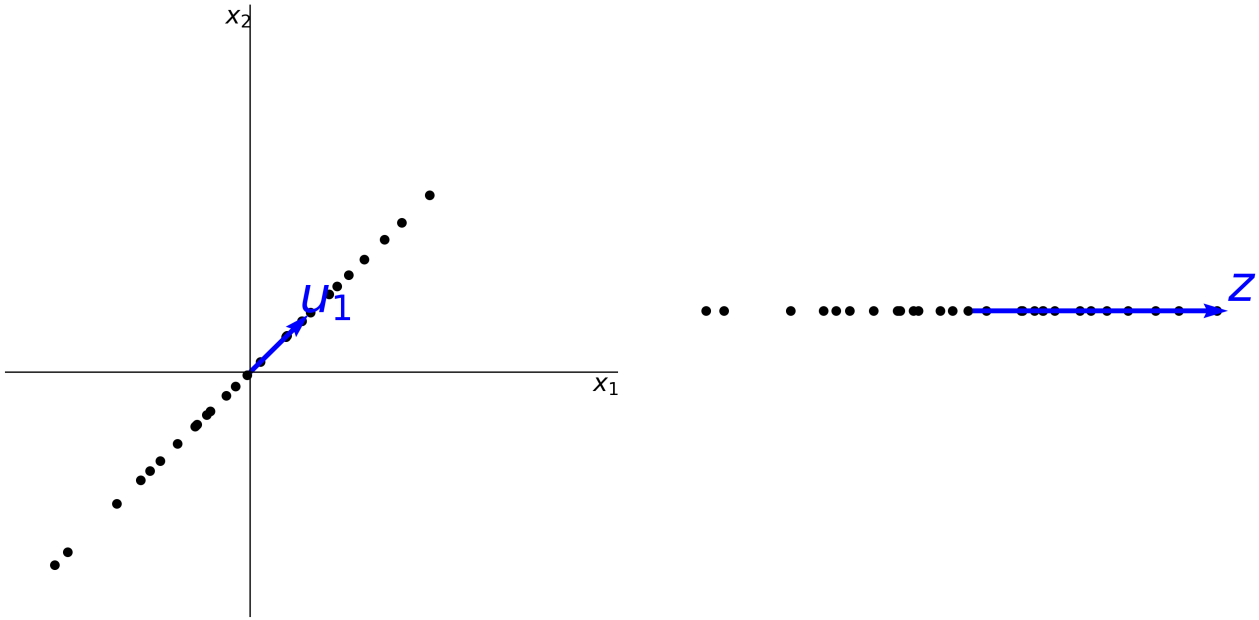
Second step: Project onto subspace spanned by $\mathbf{u}_1$



Assuming you know unit vectors $\mathbf{u}_1, \mathbf{u}_2$, use positive semi-definite matrix:

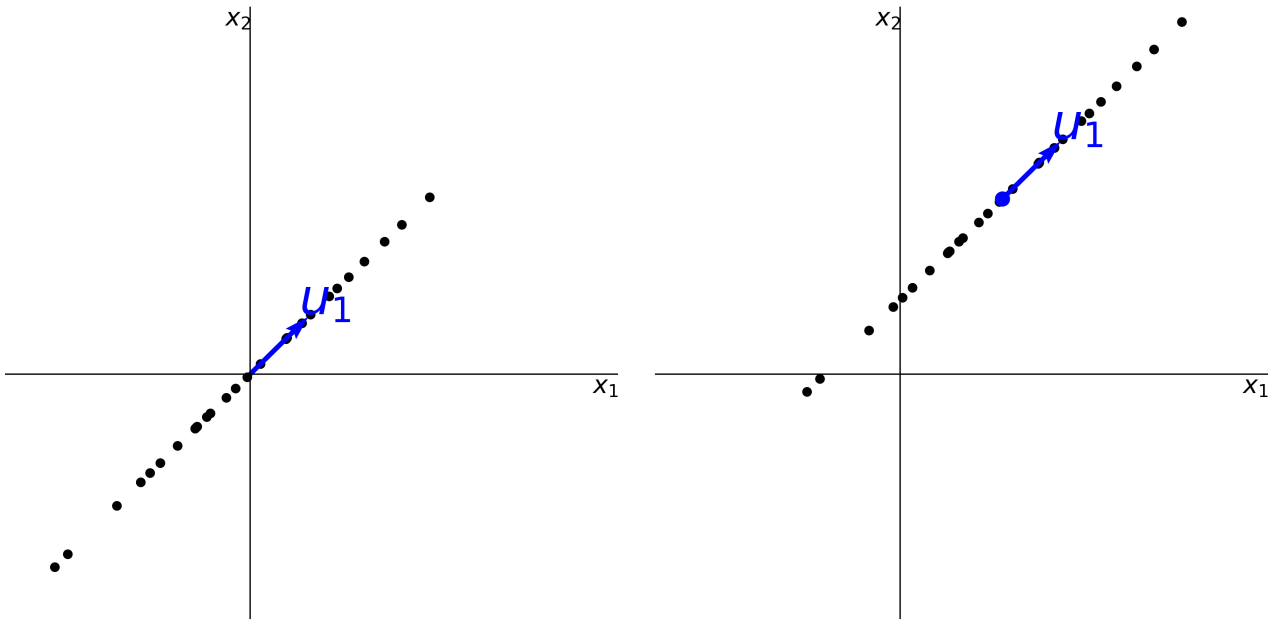
$$\text{Proj}_{\mathbf{u}_1} = \mathbf{Q}\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}\mathbf{Q}^\top = \mathbf{u}_1\mathbf{u}_1^\top \quad \text{where} \quad \mathbf{Q} = \begin{pmatrix} | & | \\ \mathbf{u}_1 & \mathbf{u}_2 \\ | & | \end{pmatrix}$$

Second step: Project onto subspace spanned by $\mathbf{u}_1$



- Coordinates $\mathbf{z} = \mathbf{u}_1^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$ along the direction $\mathbf{u}_1$ centered at $\hat{\boldsymbol{\mu}}$ are lower dimensional representations of $\mathbf{x}$.
- Projection $\mathbf{u}_1\mathbf{u}_1^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$ is the projection onto $\mathbf{u}_1$ centered at $\hat{\boldsymbol{\mu}}$.

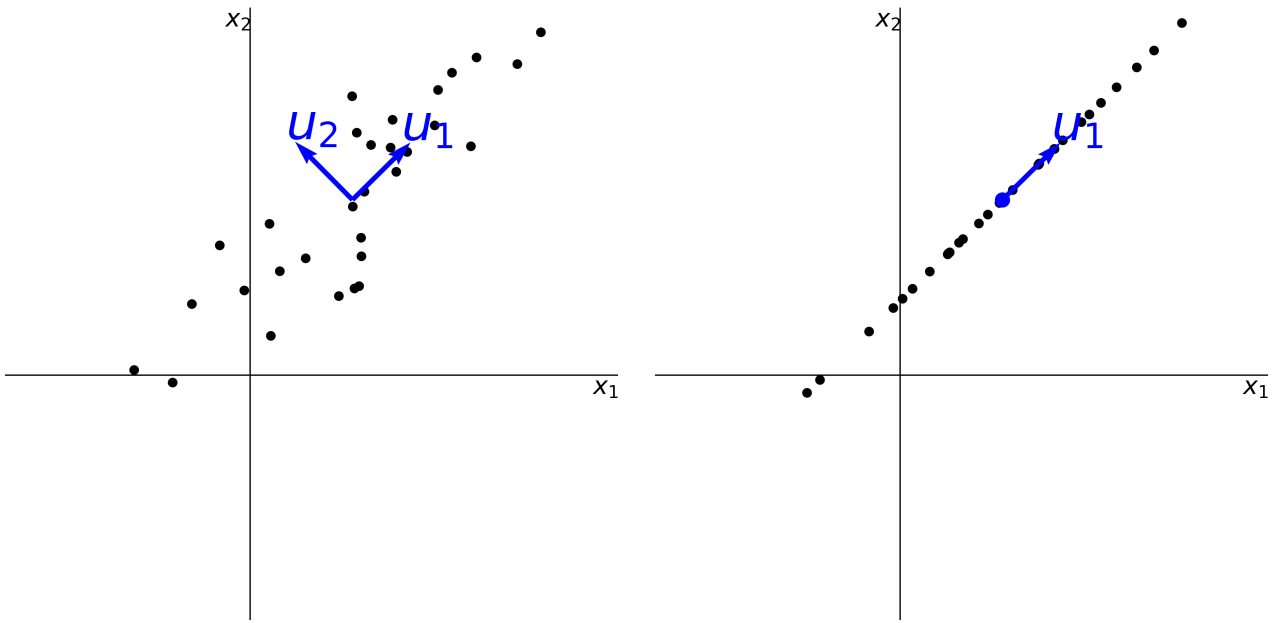
Third step: Add back mean



To get a low dimensional reconstruction for $\mathbf{x}$:

1. Subtract mean: $\mathbf{x} - \hat{\boldsymbol{\mu}}$
2. Project on $\mathcal{S}$: $\mathbf{u}_1\mathbf{u}_1^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$.
3. Add back mean to get low dimensional reconstruction: $\tilde{\mathbf{x}} = \hat{\boldsymbol{\mu}} + \mathbf{u}_1\mathbf{u}_1^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$

Third step: Add back mean



And that's it! We've done Principal Components Analysis (PCA)!

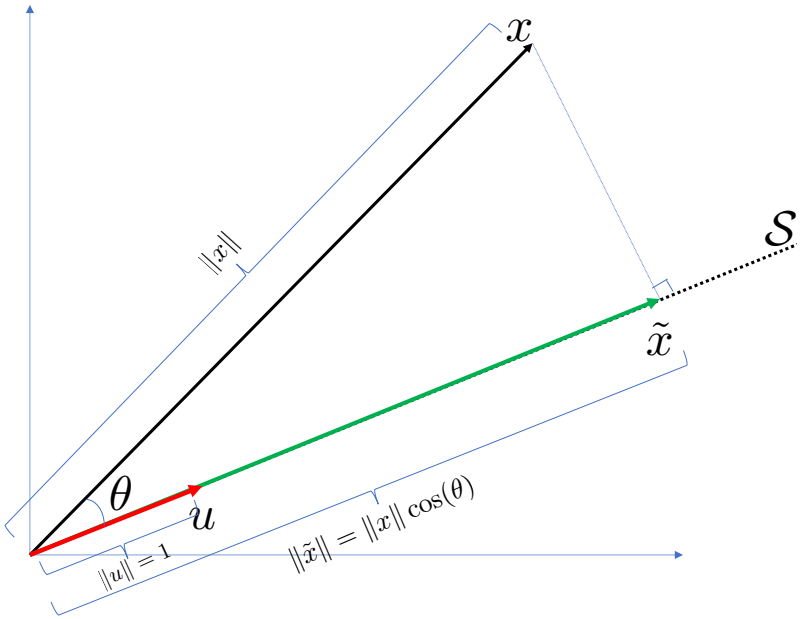
Goal: find a low dimensional representation $\mathbf{z}$ of data $\mathbf{x}$ (or alternatively, a low dimensional reconstruction $\tilde{\mathbf{x}}$ of $\mathbf{x}$).

Outline:

- Review projection onto a K dimensional subspace $\mathcal{S}$.
- Review projection onto a K dimensional affine space.
- Selecting the *best* affine space onto which to project.
- Internal coordinates in that affine space centered at $\hat{\mu}$ give us our low dimensional representation $\mathbf{z}$.
- The projection of $\mathbf{x}$ onto the affine space is our low dimensional reconstruction $\tilde{\mathbf{x}}$.

Euclidean projection

Projection onto a 1-D subspace



- Subspace $\mathcal{S}$ is the line along the unit vector $\mathbf{u}$
  - $\{\mathbf{u}\}$ is a basis for $\mathcal{S}$: any point in $\mathcal{S}$ can be written as $z\mathbf{u}$ for some z.

- Projection of $\mathbf{x}$ on $\mathbf{u}$ is the closest point to $\mathbf{x}$ on $\mathbf{u}$ denoted by $\text{Proj}_{\mathbf{u}}(\mathbf{x})$

Euclidean projection

Projection onto a 1-D subspace

- To derive $\text{Proj}_{\mathbf{u}}(\mathbf{x})$, let's solve the minimization problem directly.

$$\min_z {}^1/_2 \|z\mathbf{u} - \mathbf{x}\|^2$$

- The gradient of this objective is (assuming $\mathbf{u}$ is a unit vector)

$$\frac{\partial {}^1/_2 \|z\mathbf{u} - \mathbf{x}\|^2}{\partial z} = (z\mathbf{u} - \mathbf{x})^\top \mathbf{u} = z - \mathbf{x}^\top \mathbf{u}$$

- Solving for 0 gives us $z = \mathbf{x}^\top \mathbf{u}$ and our formula :

$$\text{Proj}_{\mathbf{u}}(\mathbf{x}) = (\mathbf{x}^\top \mathbf{u})\mathbf{u}$$

Projection onto subspaces

- How to project onto a K-dimensional subspace?
  - **Idea:** choose an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_K\}$ for $\mathcal{S}$ (i.e. all unit vectors and orthogonal to each other)
  - Project onto each unit vector individually (as in previous slide), and sum together the projections.
- Mathematically, the projection is given as:

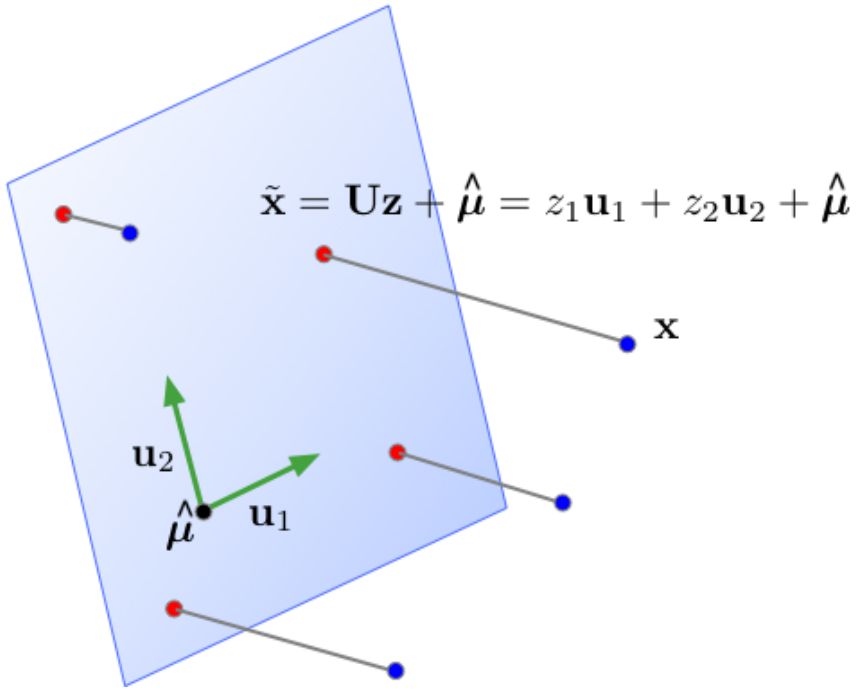
$$\text{Proj}_{\mathcal{S}}(\mathbf{x}) = \sum_{i=1}^{K} z_i \mathbf{u}_i \quad \text{where} \quad z_i = \mathbf{x}^\top \mathbf{u}_i.$$

- In vector form:

$$\text{Proj}_{\mathcal{S}}(\mathbf{x}) = \mathbf{U}\mathbf{z} \quad \text{where} \quad \mathbf{z} = \mathbf{U}^\top \mathbf{x} \text{ and } \mathbf{U} = \begin{pmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_K \\ | & & | \end{pmatrix}$$

Projection onto an affine space

- So far, we assumed the subspace passes through $\mathbf{0}$.
- In mathematical terminology, the "subspaces" we want to project onto are really affine spaces, and can have an arbitrary origin $\hat{\boldsymbol{\mu}}$.



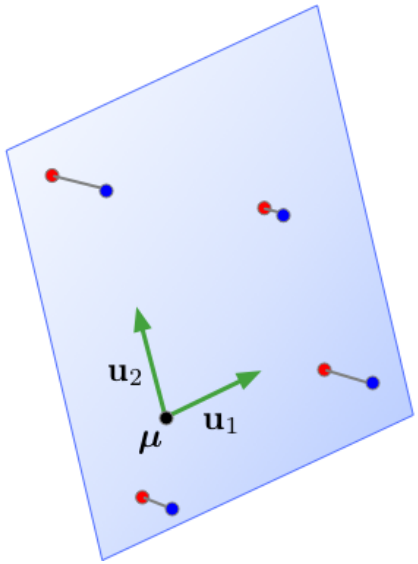
$$\mathbf{z} = \mathbf{U}^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$$

$$\begin{aligned}\tilde{\mathbf{x}} &= \mathbf{U}\mathbf{z} + \hat{\boldsymbol{\mu}} \\ &= \mathbf{U}\mathbf{U}^\top(\mathbf{x} - \hat{\boldsymbol{\mu}}) + \hat{\boldsymbol{\mu}} \\ &= \text{Proj}_{\mathbf{U}}(\mathbf{x} - \hat{\boldsymbol{\mu}}) + \hat{\boldsymbol{\mu}}\end{aligned}$$

- In machine learning, $\tilde{\mathbf{x}}$ is also called the reconstruction of $\mathbf{x}$.
- $\mathbf{z}$ is its representation, or code.

Projection onto an affine space

- If we have a K-dimensional subspace in a D-dimensional input space, then $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{z} \in \mathbb{R}^K$.
- If the data points $\mathbf{x}$ all lie close to their reconstructions, then we can approximate distances, etc. in terms of these same operations on the code vectors $\mathbf{z}$.
- If $K \ll D$, then it's much cheaper to work with $\mathbf{z}$ than $\mathbf{x}$.
- A mapping to a space that's easier to manipulate or visualize is called a representation, and learning such a mapping is representation learning.
- Mapping data to a low-dimensional space is called dimensionality reduction.



How to learn the subspace?

- How to choose a good subspace $\mathcal{S}$?
  - Origin $\hat{\boldsymbol{\mu}}$ is the empirical mean of the data
  - Need to choose a $D \times K$ matrix $\mathbf{U}$ with orthonormal columns.
- Two criteria:
  - Minimize the reconstruction error:

$$\min_{\mathbf{U}} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2$$

  - Maximize the variance of reconstructions: Find a subspace where data has the most variability.

$$\max_{\mathbf{U}} \frac{1}{N} \sum_{i} \|\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}}\|^2$$

    - Note: The data and its reconstruction have the same means (exercise)!

Learning a Subspace

- These two criteria are equivalent! I.e., we'll show

$$\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)}-\tilde{\mathbf{x}}^{(i)}\|^2 = \text{const} - \frac{1}{N}\sum_{i}\|\tilde{\mathbf{x}}^{(i)}-\hat{\boldsymbol{\mu}}\|^2$$

- Recall $\tilde{\mathbf{x}}^{(i)} = \hat{\boldsymbol{\mu}} + \mathbf{U}\mathbf{z}^{(i)}$ and $\mathbf{z}^{(i)} = \mathbf{U}^\top(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})$.

Learning a Subspace

- **Warmup Observation**: Because the columns of $\mathbf{U}$ are orthogonal, $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$, so

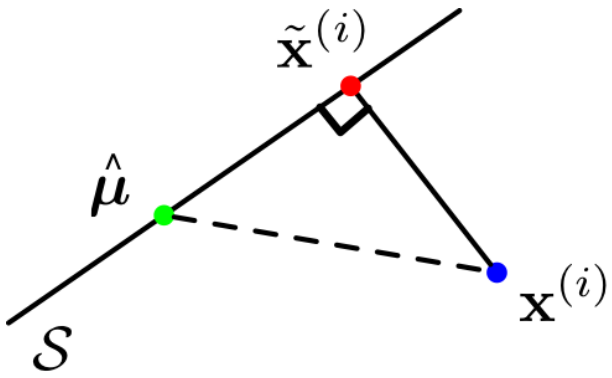
$$\|\tilde{\mathbf{x}} - \hat{\boldsymbol{\mu}}\|^2 = \|\mathbf{U}\mathbf{z}\|^2 = \mathbf{z}^\top \mathbf{U}^\top \mathbf{U}\mathbf{z} = \mathbf{z}^\top \mathbf{z} = \|\mathbf{z}\|^2.$$

  $\Longrightarrow$ norm of centered reconstruction is equal to norm of representation. (If you draw it, this is obvious).

  - Variance of reconstructions is equal to variance of code vectors: $\frac{1}{N}\sum_i \|\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}}\|^2 = \frac{1}{N}\sum_i \|\mathbf{z}^{(i)}\|^2$ (exercise $\frac{1}{N}\sum_i \mathbf{z}^{(i)} = 0$)

Pythagorean Theorem

- **Key Observation**: orthogonality of $\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}}$ and $\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}$
  (Two vectors $\mathbf{a}, \mathbf{b}$ are orthogonal $\iff \mathbf{a}^\top \mathbf{b} = 0$)
- Recall $\tilde{\mathbf{x}}^{(i)} = \hat{\boldsymbol{\mu}} + \mathbf{U}\mathbf{U}^\top(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})$.



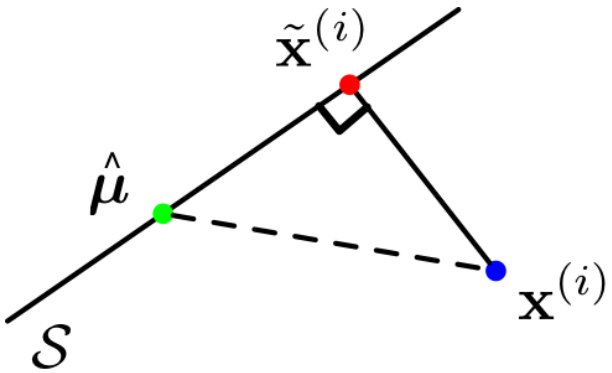
$$
\begin{aligned}
&(\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}})^\top (\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}) \\
&= (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top \mathbf{U}\mathbf{U}^\top (\hat{\boldsymbol{\mu}} - \mathbf{x}^{(i)} + \mathbf{U}\mathbf{U}^\top (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})) \\
&= (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top \mathbf{U}\mathbf{U}^\top (\hat{\boldsymbol{\mu}} - \mathbf{x}^{(i)}) + (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top \mathbf{U}\mathbf{U}^\top (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}) \\
&= 0
\end{aligned}
$$

Pythagorean Theorem

The Pythagorean Theorem tells us:

$$\|\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}}\|^2 + \|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2 = \|\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\|^2 \qquad \text{for each } i$$

By averaging over data and from observation 2, we obtain



$$\underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}}\|^2}_{\text{projected variance}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)} - \tilde{\mathbf{x}}^{(i)}\|^2}_{\text{reconstruction error}}$$

$$= \underbrace{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}\|^2}_{\text{constant}}$$

Therefore,

projected variance = constant − reconstruction error

Maximizing the variance is equivalent to minimizing the reconstruction error!

Principal Component Analysis

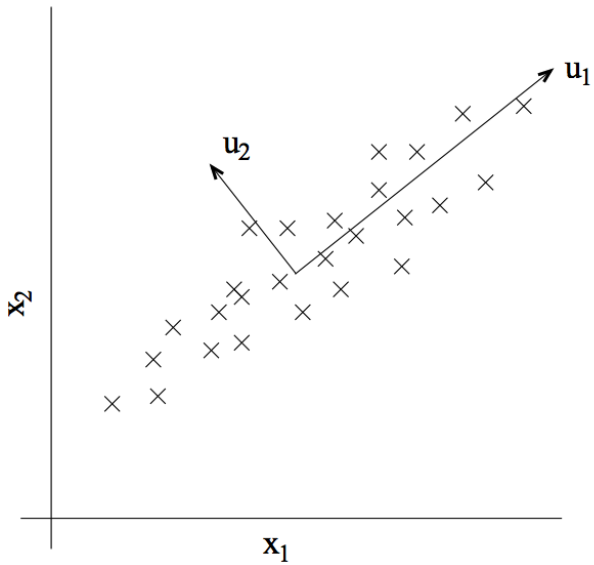
Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called principal component analysis (PCA).

- Consider the empirical covariance matrix:

$$\hat{\mathbf{\Sigma}} = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top$$

- Recall: $\hat{\mathbf{\Sigma}}$ is symmetric and positive semidefinite.

- The optimal PCA subspace is spanned by the top K eigenvectors of $\hat{\mathbf{\Sigma}}$.
  - More precisely, choose the first K of any orthonormal eigenbasis for $\hat{\mathbf{\Sigma}}$.
  - We'll show this for $K = 1$.
- These eigenvectors are called principal components, analogous to the principal axes of an ellipse.



Supplement: Deriving PCA

- For $K = 1$, we are fitting a unit vector $\mathbf{u}$, and the code is a scalar $z^{(i)} = \mathbf{u}^\top(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})$. Let's maximize the projected variance. From our warmup observation, we have

$$\frac{1}{N}\sum_i \|\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}}\|^2 = \frac{1}{N}\sum_i [z^{(i)}]^2 = \frac{1}{N}\sum_i (\mathbf{u}^\top(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}))^2$$

$$= \frac{1}{N}\sum_{i=1}^N \mathbf{u}^\top(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top\mathbf{u} \qquad (\mathbf{a}^\top\mathbf{b})^2 = \mathbf{a}^\top\mathbf{b}\mathbf{b}^\top\mathbf{a}$$

$$= \mathbf{u}^\top\left[\frac{1}{N}\sum_{i=1}^N (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})(\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top\right]\mathbf{u}$$

$$\propto \mathbf{u}^\top\hat{\boldsymbol{\Sigma}}\mathbf{u}$$

$$= \mathbf{u}^\top\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top\mathbf{u} \qquad \text{Spectral Decomposition } \hat{\boldsymbol{\Sigma}} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$$

$$= \mathbf{a}^\top\boldsymbol{\Lambda}\mathbf{a} \qquad \text{for } \mathbf{a} = \mathbf{Q}^\top\mathbf{u}$$

$$= \sum_{j=1}^D \lambda_j a_j^2$$

Supplement: Deriving PCA

- Maximize $\mathbf{a}^\top \mathbf{\Lambda a} = \sum_{j=1}^{D} \lambda_j a_j^2$ for $\mathbf{a} = \mathbf{Q}^\top \mathbf{u}$.
  - This is a change-of-basis to the eigenbasis of $\hat{\mathbf{\Sigma}}$.
- Assume the λ_i are in sorted order, $\lambda_1 \geq \lambda_2, \geq \ldots$
- Observation: since $\mathbf{u}$ is a unit vector, then by unitarity, $\mathbf{a}$ is also a unit vector: $\mathbf{a}^\top \mathbf{a} = \mathbf{u}^\top \mathbf{Q}\mathbf{Q}^\top \mathbf{u} = \mathbf{u}^\top \mathbf{u}$, i.e., $\sum_j a_j^2 = 1$.
- By inspection, set $a_1 = \pm 1$ and $a_j = 0$ for $j \neq 1$.
- Hence, $\mathbf{u} = \mathbf{Q}\mathbf{a} = \mathbf{q}_1$ (the top eigenvector).

- A similar argument shows that the kth principal component is the kth eigenvector of $\hat{\mathbf{\Sigma}}$.

Decorrelation

- Interesting fact: the dimensions of $\mathbf{z}$ are decorrelated. For now, let Cov denote the empirical covariance.

$$
\begin{aligned}
\text{Cov}(\mathbf{z}) &= \text{Cov}(\mathbf{U}^\top(\mathbf{x}-\boldsymbol{\mu})) \\
&= \mathbf{U}^\top \text{Cov}(\mathbf{x})\mathbf{U} \\
&= \mathbf{U}^\top \hat{\boldsymbol{\Sigma}}\mathbf{U} \\
&= \mathbf{U}^\top \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top\mathbf{U} && \triangleright \text{ spectral decomposition} \\
&= \begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}\boldsymbol{\Lambda}\begin{pmatrix}\mathbf{I} \\ \mathbf{0}\end{pmatrix} && \triangleright \text{ by orthogonality} \\
&= \text{top left } K \times K \text{ block of } \boldsymbol{\Lambda}
\end{aligned}
$$

- If the covariance matrix is diagonal, this means the features are uncorrelated.

Recap

Recap:

- Dimensionality reduction aims to find a low-dimensional representation of the data.
- PCA projects the data onto a subspace which maximizes the projected variance, or equivalently, minimizes the reconstruction error.
- The optimal subspace is given by the top eigenvectors of the empirical covariance matrix.
- PCA gives a set of decorrelated features.

Applying PCA to faces

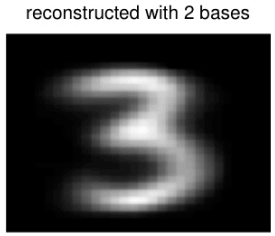
- Consider running PCA on 2429 19x19 grayscale images (CBCL data)
- Can get good reconstructions with only 3 components

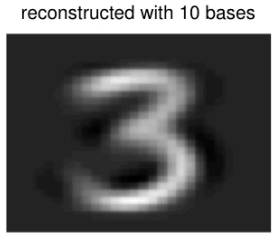
- PCA for pre-processing: can apply classifier to latent representation
  - Original data is 361 dimensional
  - For face recognition PCA with 3 components obtains 79% accuracy on face/non-face discrimination on test data vs. 76.8% for a Gaussian mixture model (GMM) with 84 states. (We'll cover GMMs later in the course.)
- Can also be good for visualization

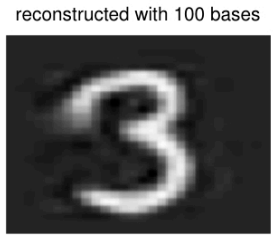
Applying PCA to faces: Learned basis

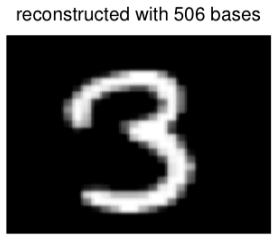
Principal components of face images ("eigenfaces")

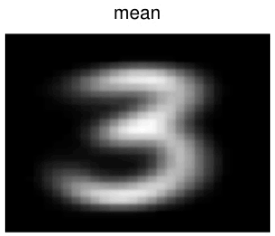
Applying PCA to digits

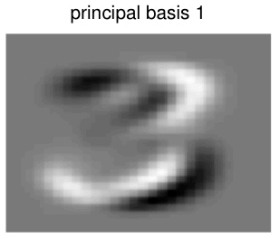


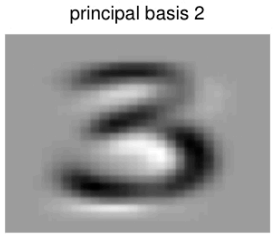


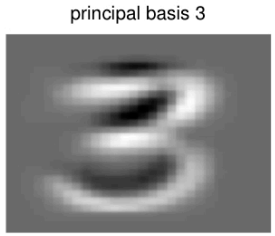












Next

One more interpretation of PCA, which has an interesting generalization: **Matrix factorization**.