STA 314: Statistical Methods for Machine Learning I

Lecture 8 - Principle Components Analysis

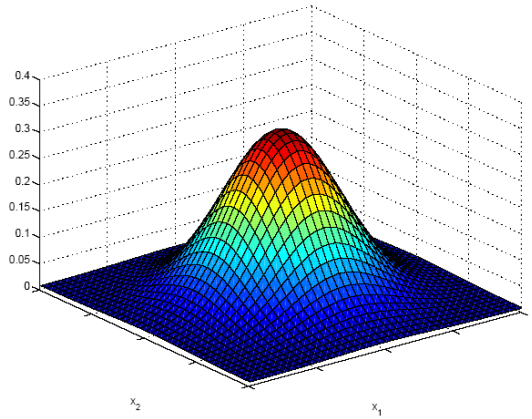
Chris J. Maddison

University of Toronto

Multivariate Gaussian Model

- $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, a Gaussian (or normal) distribution defined as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$



- To understand the shape of the density, we will study now the standard normal $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is transformed to produce $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
  - Last week I mentioned that the multivariate Gaussian requires understanding multivariate scaling by positive definite matrices.
  - I didn't do a great job of explaining this, so I'm going to try again.

Recall some definitions (details optional)

First, recall:

- **Definition.** Symmetric matrix A is positive semidefinite if $\mathbf{x}^\top \mathbf{A}\mathbf{x} \geq 0$ for all non-zero $\mathbf{x}$. It is positive definite if $\mathbf{x}^\top \mathbf{A}\mathbf{x} > 0$ for all non-zero $\mathbf{x}$.
  - Any positive definite matrix is positive semidefinite.
  - Positive definite matrices have positive eigenvalues, and positive semidefinite matrices have non-negative eigenvalues.
  - For any matrix $\mathbf{X}$, $\mathbf{X}^\top \mathbf{X}$ and $\mathbf{X}\mathbf{X}^\top$ are positive semidefinite.
- **Theorem** (*Unique Positive Square Root*)**.** Let $\mathbf{A}$ be a positive semidefinite real matrix. Then there is a unique positive semidefinite matrix $\mathbf{B}$ such that $\mathbf{A} = \mathbf{B}^\top \mathbf{B} = \mathbf{B}\mathbf{B}$. We call $\mathbf{A}^{\frac{1}{2}} \triangleq \mathbf{B}$ the positive square root of $\mathbf{A}$.
- **Theorem** (*Spectral Theorem*)**.** If $\mathbf{A} \in \mathbb{R}^{d \times d}$ is symmetric, then
  1. $\mathbb{R}^D$ has an orthonormal basis consisting of the eigenvectors of $\mathbf{A}$.
  2. There exists orthonormal matrix $\mathbf{Q}$ and diagonal matrix $\mathbf{\Lambda}$ such that $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$. This is called the spectral decomposition of $\mathbf{A}$.
     - The columns of $\mathbf{Q}$ are (unit) eigenvectors of $\mathbf{A}$.

Matrix Square Roots & the Multivariate Gaussian

- Suppose $\mathbf{x}$ is a standard Gaussian in D dimensions with density

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \exp\left[-\frac{\|\mathbf{x}\|_2^2}{2}\right]$$

- Transform $\mathbf{x}$ to $\mathbf{y} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{x}$. Then by change of variables

$$\begin{aligned} p(\mathbf{y}) &= \frac{1}{(2\pi)^{D/2}} \exp\left[-\frac{\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\mathbf{y}-\boldsymbol{\mu})\|_2^2}{2}\right] \left|\boldsymbol{\Sigma}^{-\frac{1}{2}}\right| \\ &= \frac{1}{(2\pi)^{D/2}\,|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \end{aligned}$$

  - Be careful, this derivative use many facts about determinants, inverses, and square roots that one would have to verify.

Matrix Square Roots & the Multivariate Gaussian

- So $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ shifted by $\boldsymbol{\mu}$ and "scaled" by $\boldsymbol{\Sigma}^{\frac{1}{2}}$.
- How can you think of "scaling" space by the square root of a matrix? For a PSD matrix $\boldsymbol{\Sigma}$, find it's spectral decomposition:

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

- 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

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

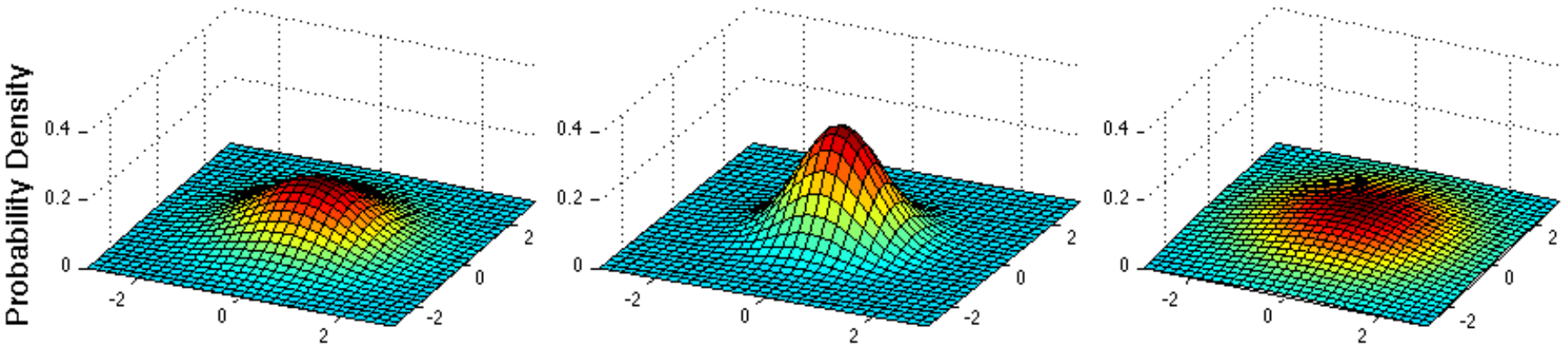
  and multiplying by is the same as scaling the (current) standard basis by $\sqrt{\lambda_i}$.
- Multiplying by $\mathbf{Q}$ rotates the standard basis back into the basis of $\mathbf{Q}$.

Matrix Square Roots & the Multivariate Gaussian

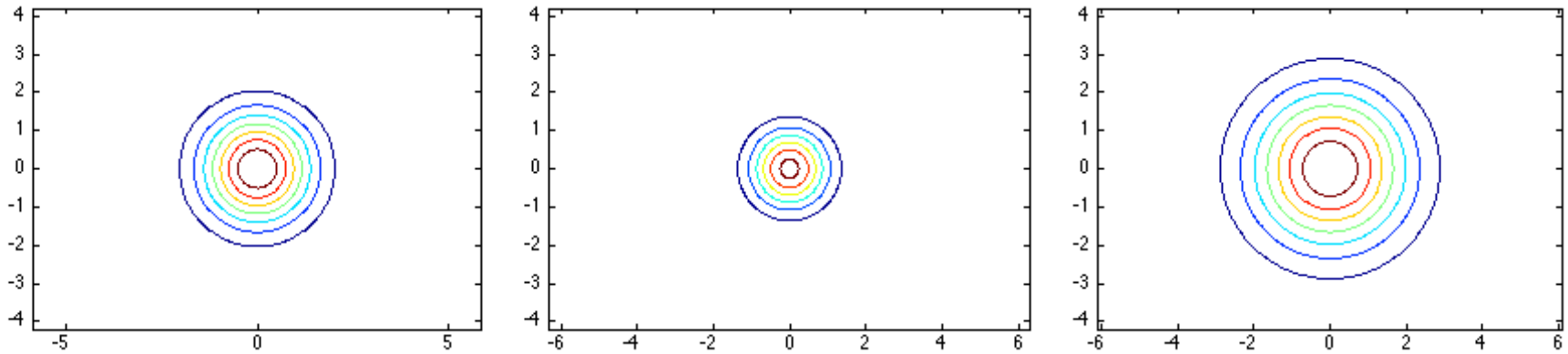
- To to summarize, you can think of scaling space by $\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{x}$ as the effect of rotating the standard basis into the eigenvectors of $\boldsymbol{\Sigma}^{\frac{1}{2}}$ and scaling space along those orthogonal directions.
- So multivariate "scaling" has both magnitude and direction.

Bivariate Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = 0.5 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = 2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



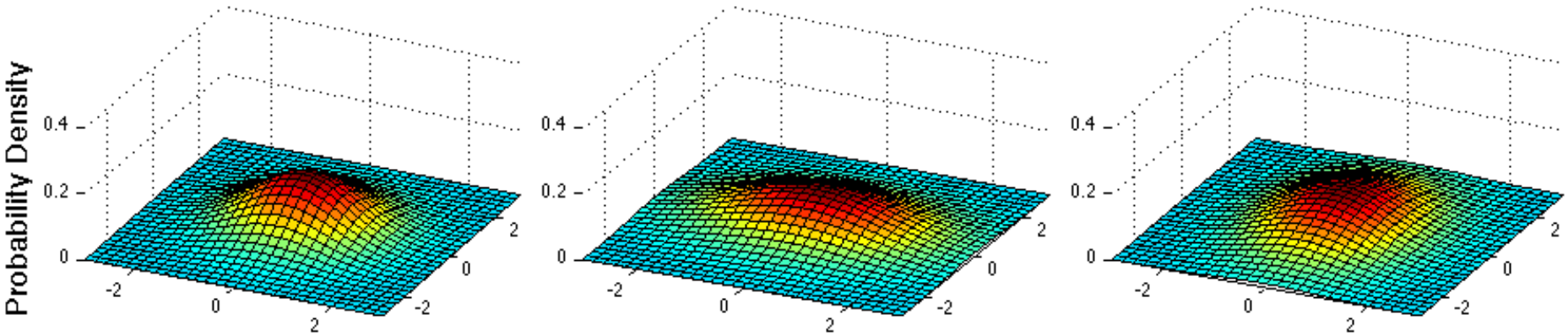
Probability density function



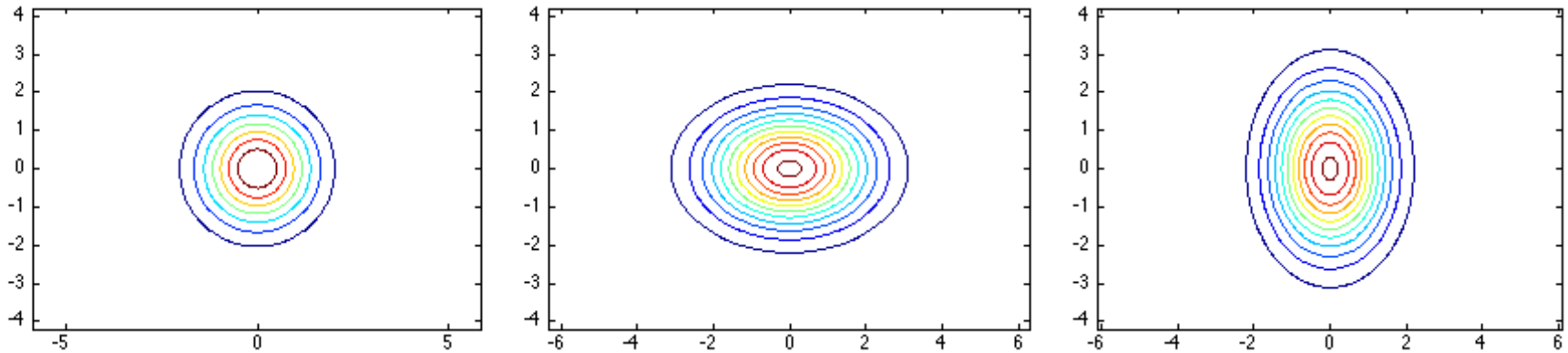
Contour plot of the pdf

Bivariate Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$



Probability density function



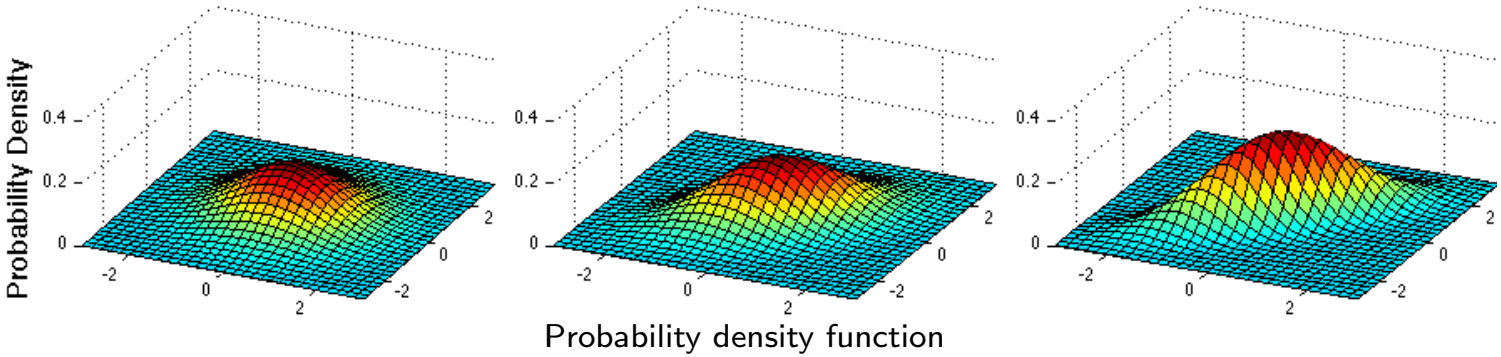
Contour plot of the pdf

Bivariate Gaussian

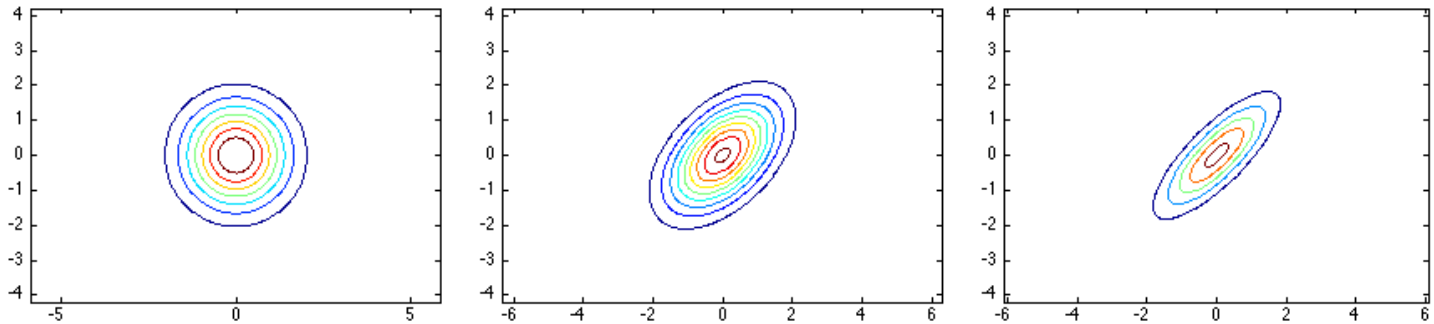
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}$$

$$= \mathbf{Q}_1 \begin{pmatrix} 1.5 & 0. \\ 0. & 0.5 \end{pmatrix} \mathbf{Q}_1^\top \qquad = \mathbf{Q}_2 \begin{pmatrix} 1.8 & 0. \\ 0. & 0.2 \end{pmatrix} \mathbf{Q}_2^\top$$

Test your intuition: Does $Q_1 = Q_2$?



Probability density function



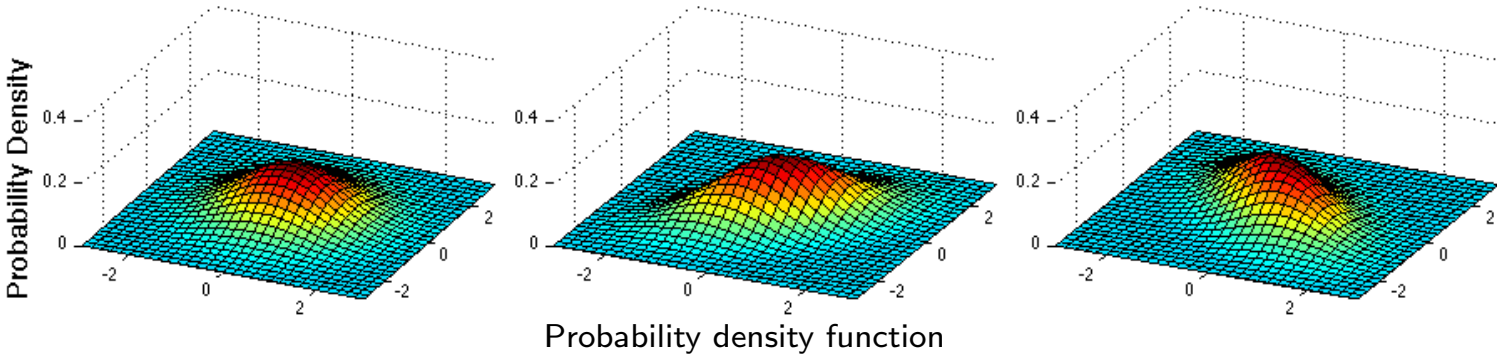
Contour plot of the pdf

Bivariate Gaussian

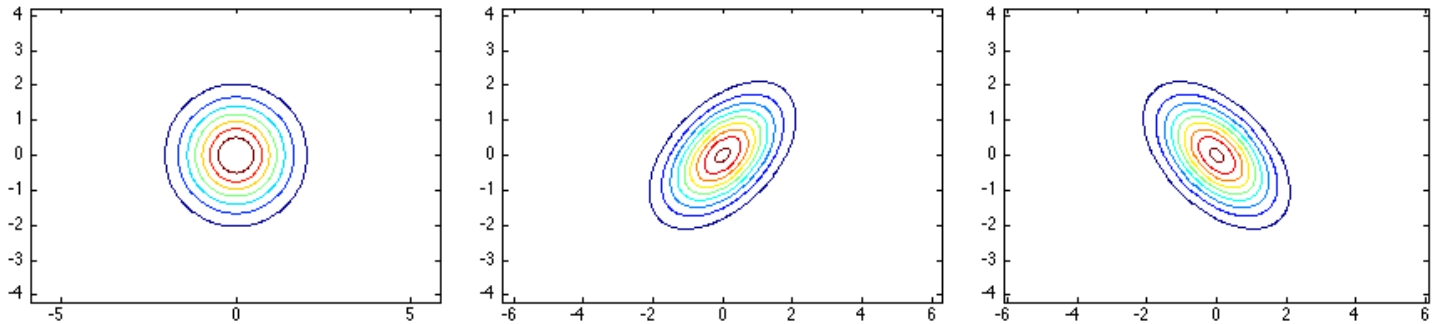
$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & -0.5 \\ -0.5 & 1 \end{pmatrix}$$

$$= \mathbf{Q}_1 \begin{pmatrix} 1.5 & 0. \\ 0. & 0.5 \end{pmatrix} \mathbf{Q}_1^\top \qquad = \mathbf{Q}_2 \begin{pmatrix} \lambda_1 & 0. \\ 0. & \lambda_2 \end{pmatrix} \mathbf{Q}_2^\top$$

Test your intuition: Does $Q_1 = Q_2$? What are λ_1 and λ_2?



Probability density function



Contour plot of the pdf

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

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

Mean and Covariance Estimators

- We can estimate mean $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ under the multivariate Gaussian model using these sample approximations:

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

Sample covariance:

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

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

Low dimensional representation

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

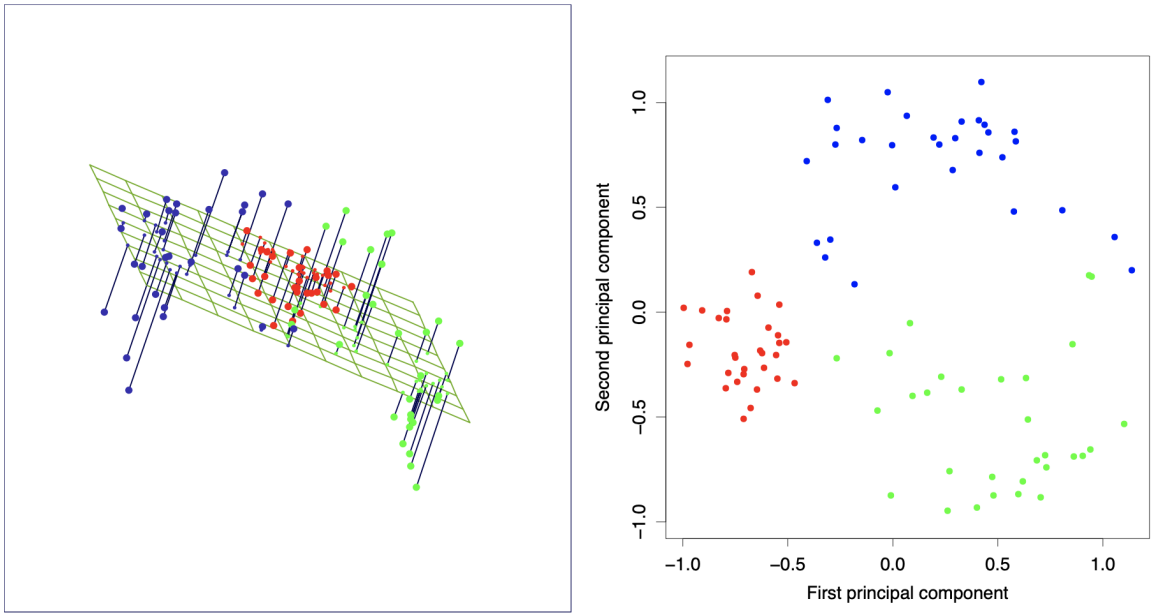


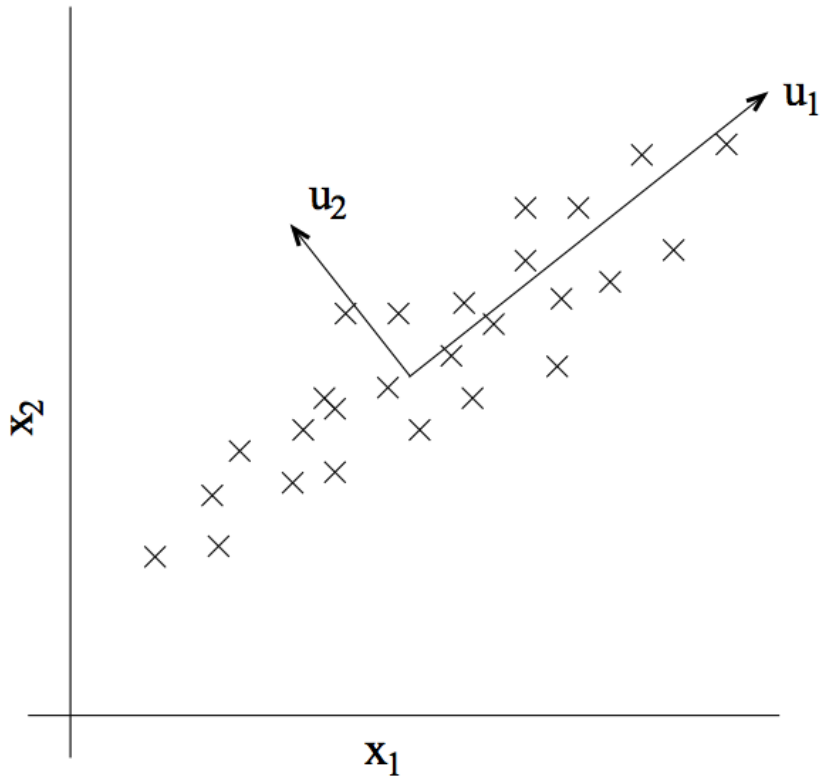
Image credit: Elements of Statistical Learning

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

Projection onto a subspace

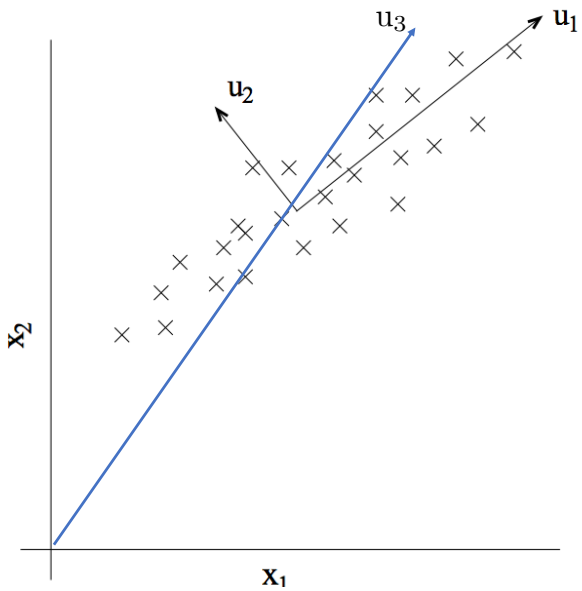
- Set-up: given a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\} \subset \mathbb{R}^D$
- Set $\hat{\boldsymbol{\mu}}$ to the sample mean of the data, $\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)}$
- Goal: find a K-dimensional subspace $\mathcal{S} \subset \mathbb{R}^D$ such that $\mathbf{x}^{(n)} - \hat{\boldsymbol{\mu}}$ is "well-represented" by its projection onto a K-dimensional $\mathcal{S}$
- Recall: The projection of a point $\mathbf{x}$ onto $\mathcal{S}$ is the point in $\mathcal{S}$ closest to $\mathbf{x}$. More on this coming soon.

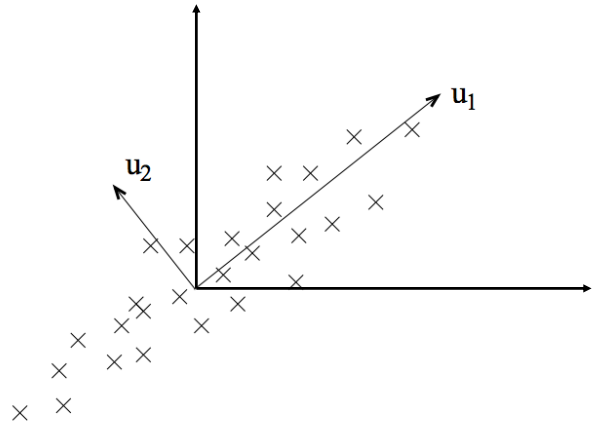
We are looking for directions



- For example, in a 2-dimensional problem, we are looking for the direction $\boldsymbol{u}_1$ along which the data is well represented: (?)
  - e.g. direction of higher variance
  - e.g. direction of minimum difference after projection
  - turns out they are the same!

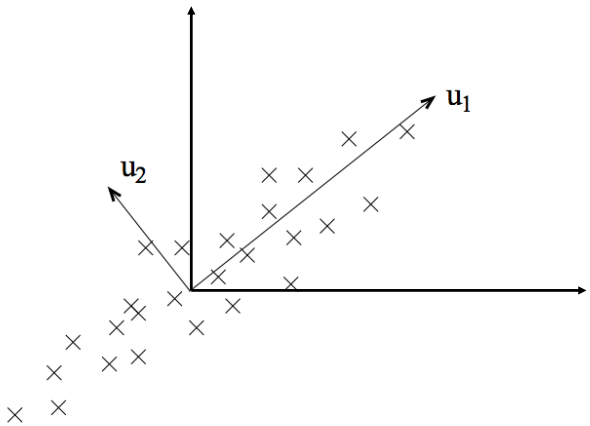
First step: Center data





- Directions we compute will pass through origin, and should represent the direction of highest variance.
- We need to center our data since we don't want location of data to influence our calculations. We are only interested in finding the direction of highest variance. This is independent from its mean.
- $\Longrightarrow$ We are **not** interested in u_3, we are interested in u_1.

Second step: Project onto lower dimensional space $\mathcal{S}$



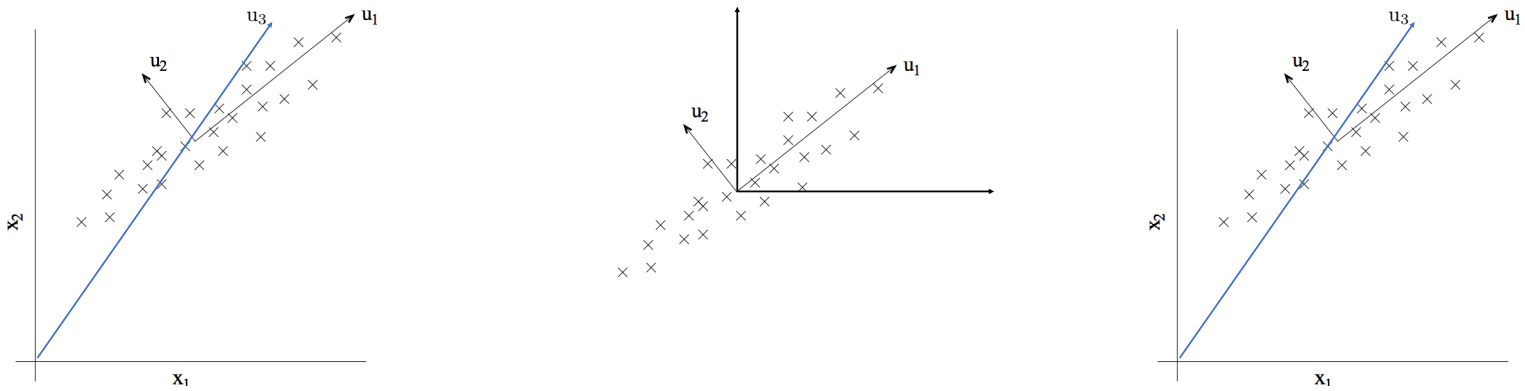
- A projection is just a multivariate "scale" by 0 in the pruned directions. You already know how to do this!
- Use positive semi-definite matrix:

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

- This is the same as:

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

Third step: Add back mean



Summary for a given point $\mathbf{x}$:

1. Subtract mean: $\mathbf{x} - \hat{\boldsymbol{\mu}}$
2. Project on $\mathcal{S}$: $\mathbf{U}\mathbf{U}^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$, where columns of $\mathbf{U}$ are unit eigenvectors for largest K eigenvalues of $\hat{\boldsymbol{\Sigma}}$ (K directions of highest variance)
3. Add back mean: $\tilde{\mathbf{x}} = \hat{\boldsymbol{\mu}} + \mathbf{U}\mathbf{U}^T(\mathbf{x} - \hat{\boldsymbol{\mu}})$

Here, $\mathbf{z} = \mathbf{U}^T(\mathbf{x} - \hat{\boldsymbol{\mu}})$ is a lower dimensional representation of $\mathbf{x}$.
And that's it! We've done Principal Components Analysis (PCA)!

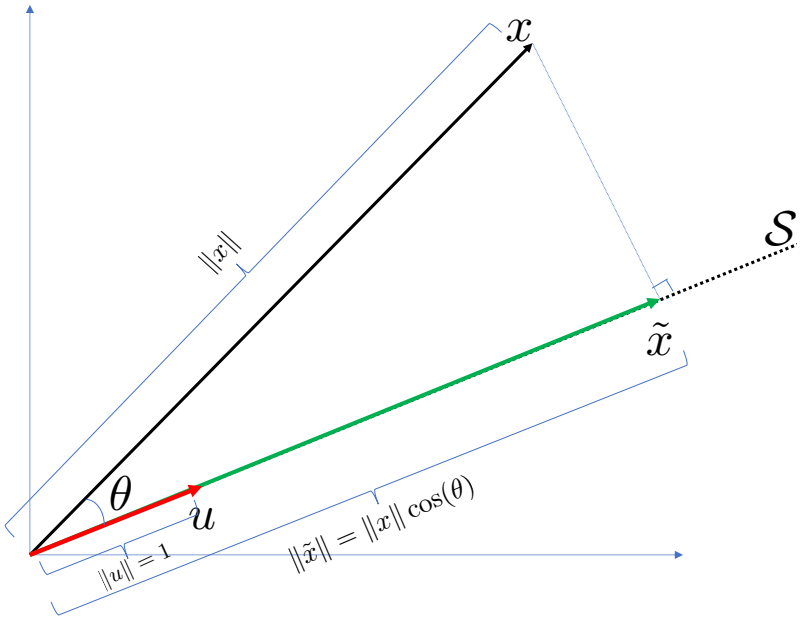
Goal: find a low dimensional represenation $\mathbf{z}$ of data $\mathbf{x}$.

Outline for PCA:

- Review projection onto a K dimensional subspace $\mathcal{S}$.
- Selecting the *best* affine space onto which to project.
- Project $\mathbf{x}$ onto the affine space to get our low dimensional representation $\mathbf{z}$.

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 $\mathcal{S}$ is denoted by $\text{Proj}_{\mathcal{S}}(\mathbf{x})$
- Recall: $\mathbf{x}^\top \mathbf{u} = \|\mathbf{x}\| \|\mathbf{u}\| \cos(\theta) = \|\mathbf{x}\| \cos(\theta)$
- $\text{Proj}_{\mathcal{S}}(\mathbf{x}) = \underbrace{\mathbf{x}^\top \mathbf{u}}_{\text{length of proj}} \cdot \underbrace{\mathbf{u}}_{\text{direction of proj}} = \|\tilde{\mathbf{x}}\| \mathbf{u}$

General 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:

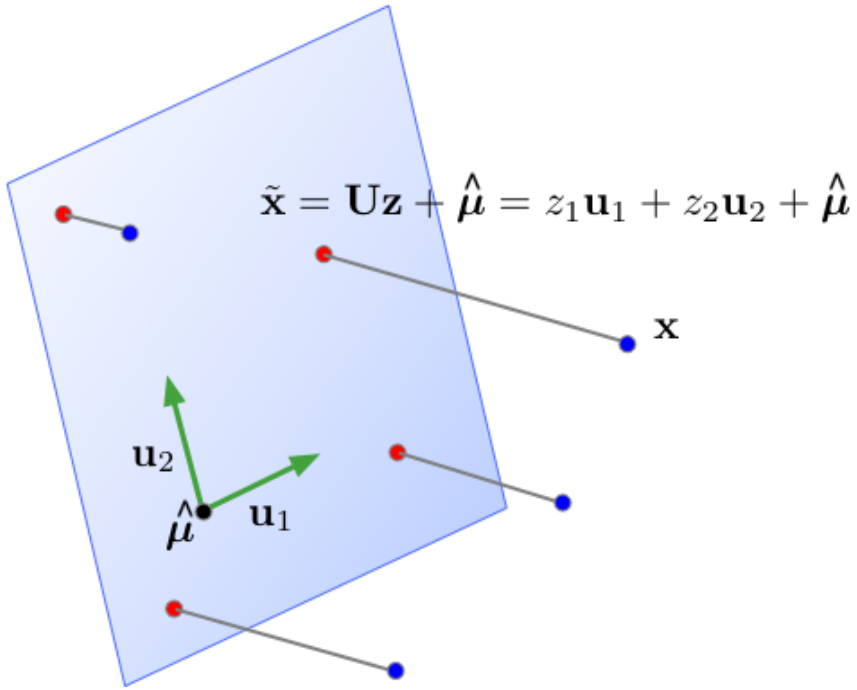
$$\text{Proj}_{\mathcal{S}}(\mathbf{x}) = \sum_{i=1}^{K} z_i \mathbf{u}_i \quad \text{where} \quad z_i = \mathbf{x}^\top \mathbf{u}_i.$$

- In vector form:

$$\text{Proj}_{\mathcal{S}}(\mathbf{x}) = \mathbf{U}\mathbf{z} \quad \text{where} \quad z_i = \mathbf{U}^\top \mathbf{x}$$

Projection onto a Subspace

- 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 $\hat{\boldsymbol{\mu}}$.



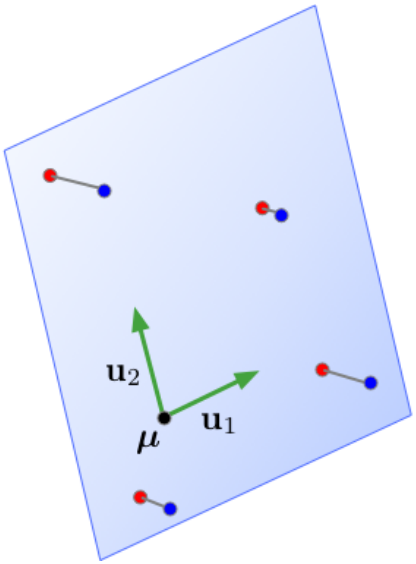
$$\mathbf{z} = \mathbf{U}^\top(\mathbf{x} - \hat{\boldsymbol{\mu}})$$

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

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

Projection onto a Subspace

- 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.



Learning a 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

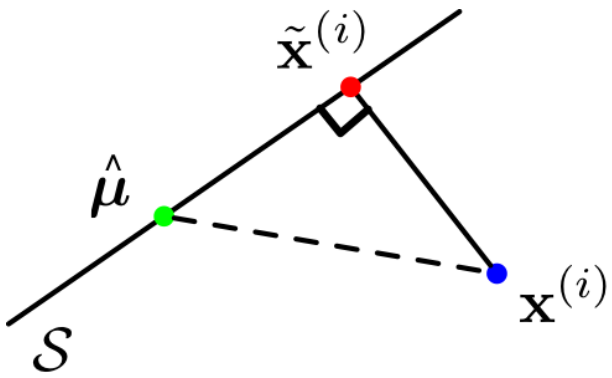
$$\|\tilde{\mathbf{x}} - \hat{\boldsymbol{\mu}}\|^2 = \|\mathbf{U}\mathbf{z}\|^2 = \mathbf{z}^\top \mathbf{U}^\top \mathbf{U}\mathbf{z} = \mathbf{z}^\top \mathbf{z} = \|\mathbf{z}\|^2.$$

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



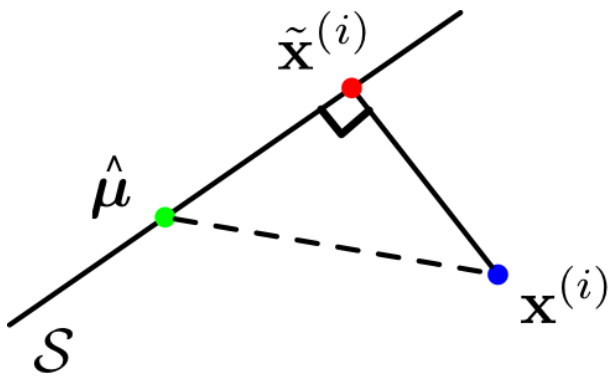
$$
\begin{aligned}
&(\tilde{\mathbf{x}}^{(i)} - \hat{\boldsymbol{\mu}})^\top (\tilde{\mathbf{x}}^{(i)} - \mathbf{x}^{(i)}) \\
&= (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}})^\top \mathbf{U}\mathbf{U}^\top (\hat{\boldsymbol{\mu}} - \mathbf{x}^{(i)} + \mathbf{U}\mathbf{U}^\top (\mathbf{x}^{(i)} - \hat{\boldsymbol{\mu}}))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

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,

$$\text{projected variance} = \text{constant} - \text{reconstruction error}$$

Maximizing the variance is equivalent to minimizing the reconstruction error!

Principal Component Analysis

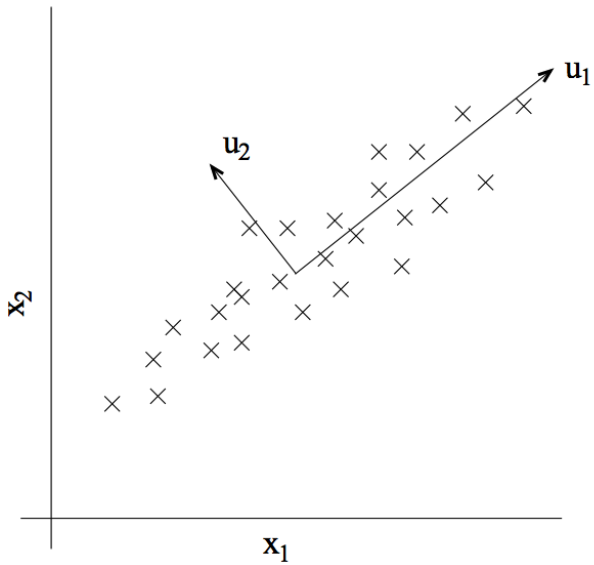
Choosing a subspace to maximize the projected variance, or minimize the reconstruction error, is called principal component analysis (PCA).

- Consider the empirical covariance matrix:

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

- 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

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

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

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

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

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

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

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

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 $\mathbf{\Sigma}$.
- 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 $\mathbf{\Sigma}$.

Decorrelation

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

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

- 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

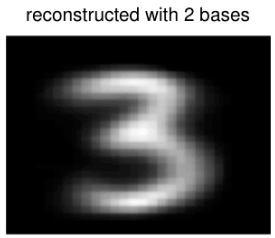
- Consider running PCA on 2429 19x19 grayscale images (CBCL data)
- Can get good reconstructions with only 3 components

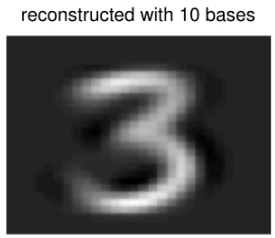
- PCA for pre-processing: can apply classifier to latent representation
  - Original data is 361 dimensional
  - For face recognition PCA with 3 components obtains 79% accuracy on face/non-face discrimination on test data vs. 76.8% for a Gaussian mixture model (GMM) with 84 states. (We'll cover GMMs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

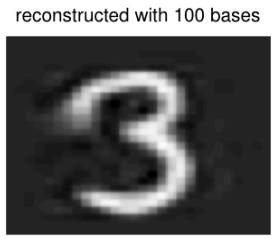
Applying PCA to faces: Learned basis

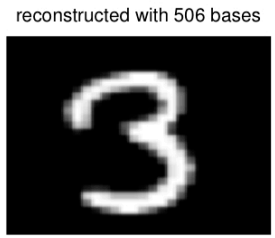
Principal components of face images ("eigenfaces")

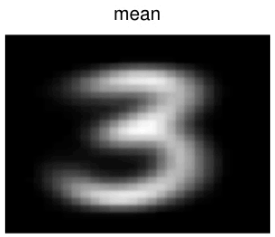
Applying PCA to digits

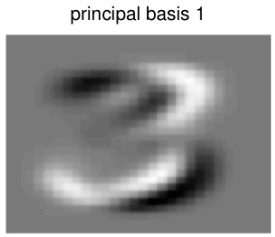


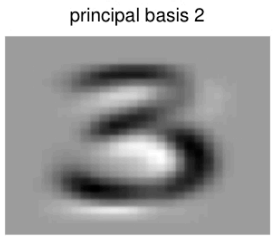


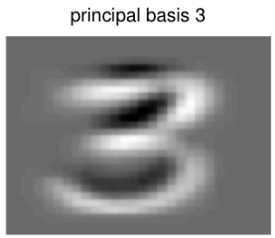












Next

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