

CSE 152: Computer Vision

Hao Su

Lecture 5: Machine Learning Crash Course: Unsupervised Learning





Photo: CMU Machine Learning Department Protests G20

Our approach

- We will look at ML as a tool. We will not detail the underpinnings of each learning method.
- Please take a machine learning course if you want to know more!

Machine Learning

- Learn from and make predictions on data.
- Arguably the greatest export from computing to other scientific fields.
- Statisticians might disagree with CompScis on the true origins...

Agenda

- Principal Component Analysis (PCA)
- K-means

Agenda

- Principal Component Analysis (PCA)
- K-means

Suppose you have **10 million** face images,

Question:

1. How can you find the 5 faces closest to a query (maybe yours!) in just 0.1 sec?
2. How can you show all of them in a single picture?

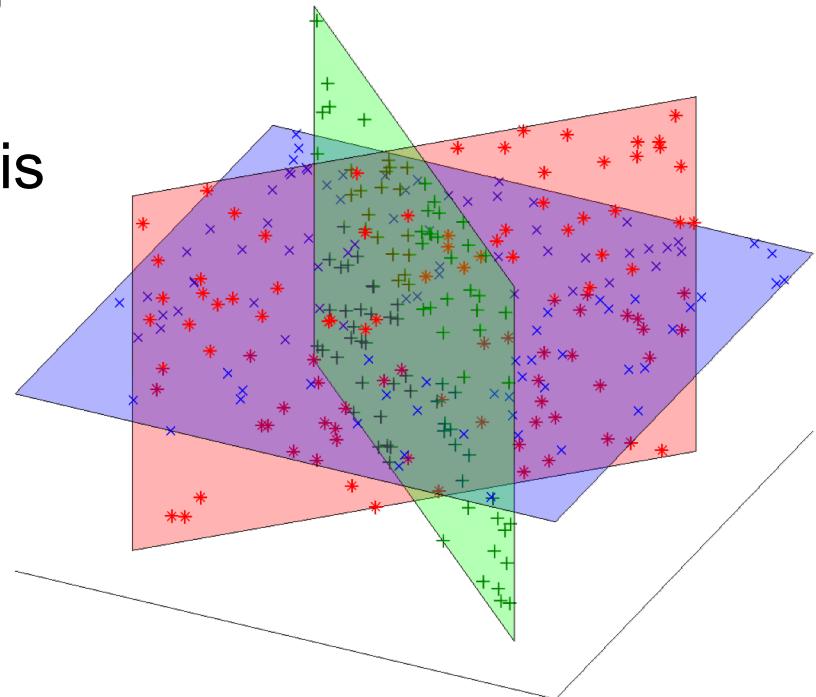
Suppose you have **10 million** face images,

Question:

1. How can you search for a specific person in the database
(maybe using a query image)?
 2. How can you show all of them in a single picture?
- SVD can help you do it!*

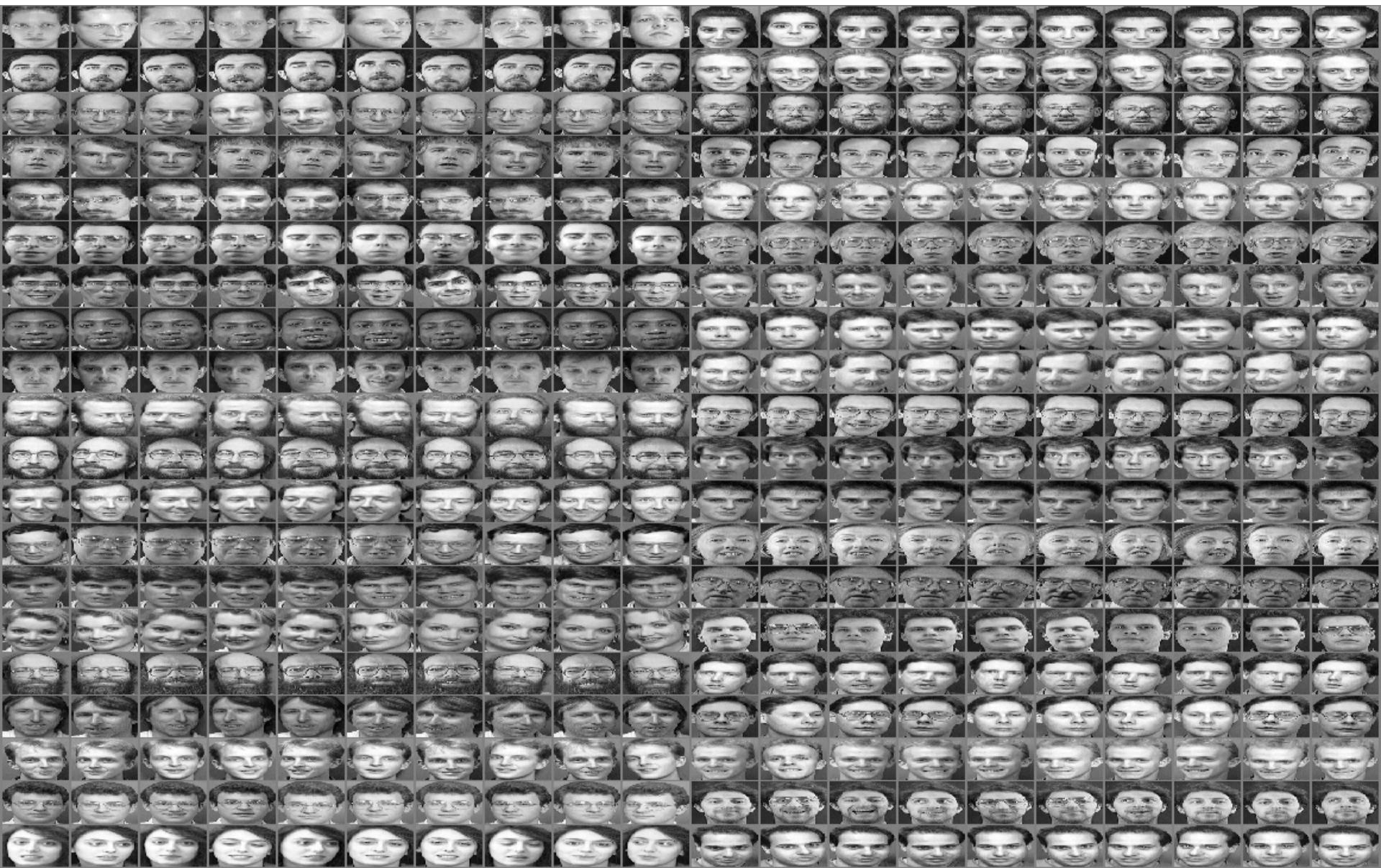
Data as Points in a Euclidean Space

- While data can be represented as a high-dimensional vector,
- Lower-dimensional structure is often present in the data



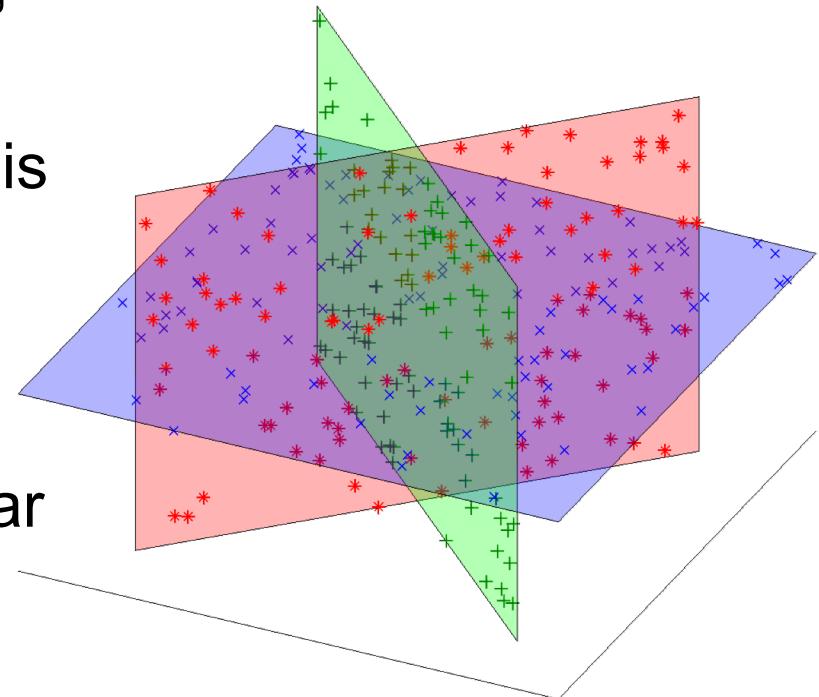
Eigenfaces

*The ATT face database (formerly the ORL database),
10 pictures of 40 subjects each*



Data as Points in a Euclidean Space

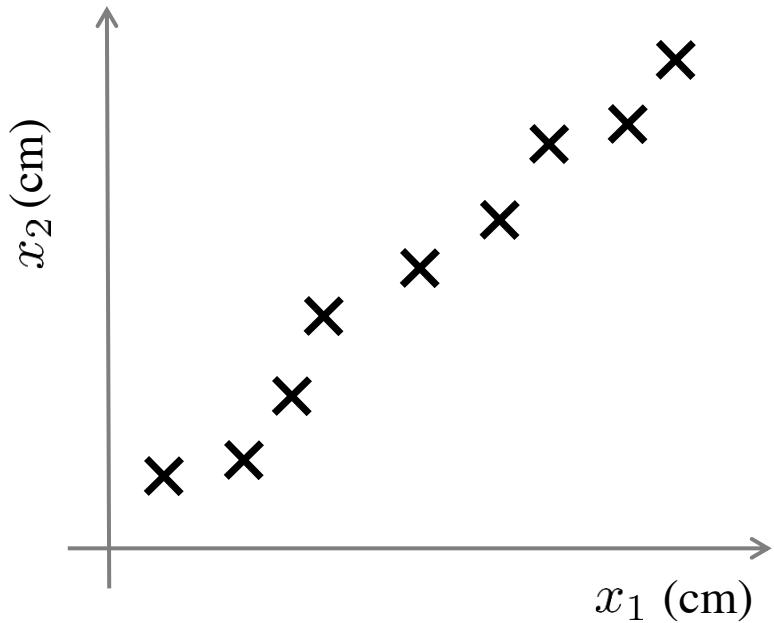
- While data can be represented as a high-dimensional vector,
- Lower-dimensional structure is often present in the data
- Sometimes this lower-d structure corresponds to linear subspaces within the original space.



Principal Component Analysis

- Given data points in d dimensions
- Convert them to data points in $r < d$ dimensions
- With minimal loss of information

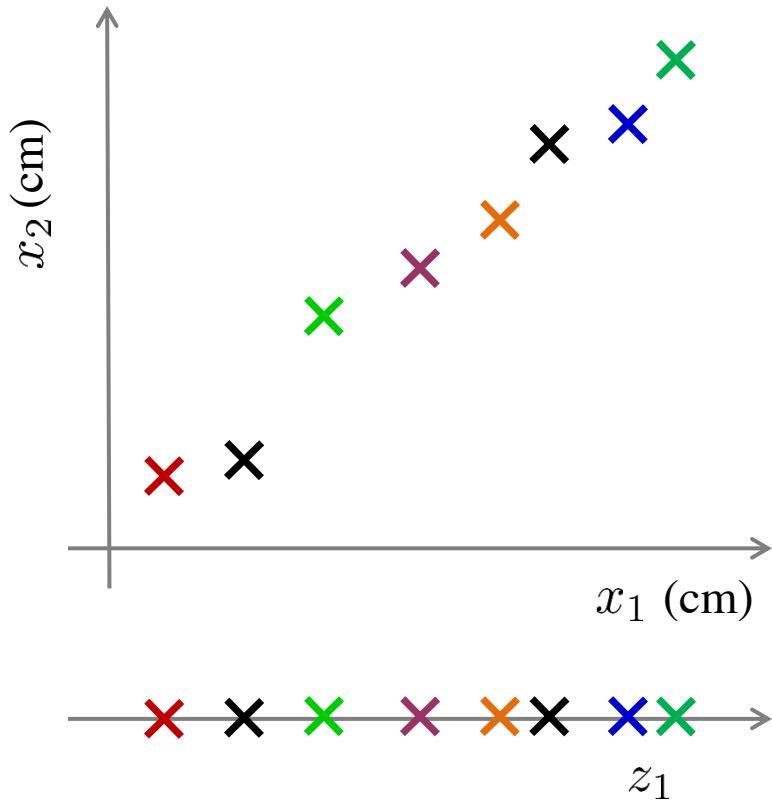
Data Compression



Reduce data from
2D to 1D

May be width and height of TVs

Data Compression



Reduce data from
2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$

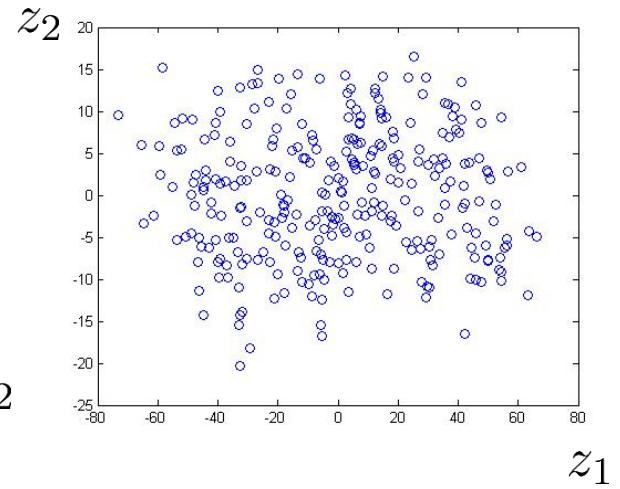
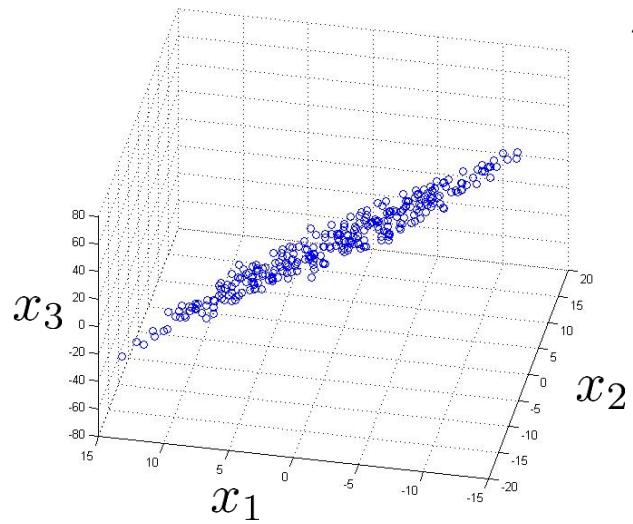
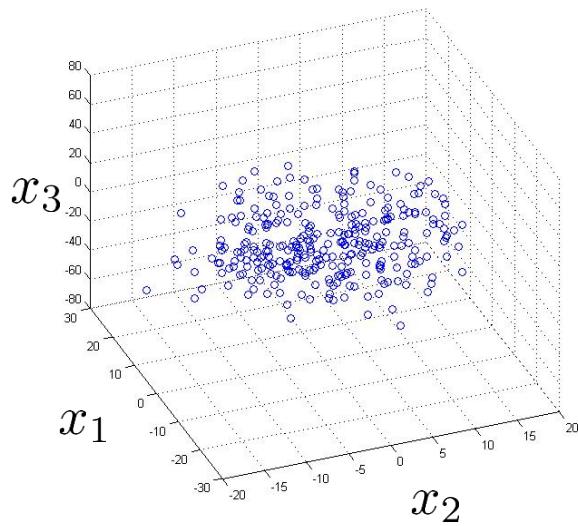
$$x^{(2)} \rightarrow z^{(2)}$$

⋮

$$x^{(m)} \rightarrow z^{(m)}$$

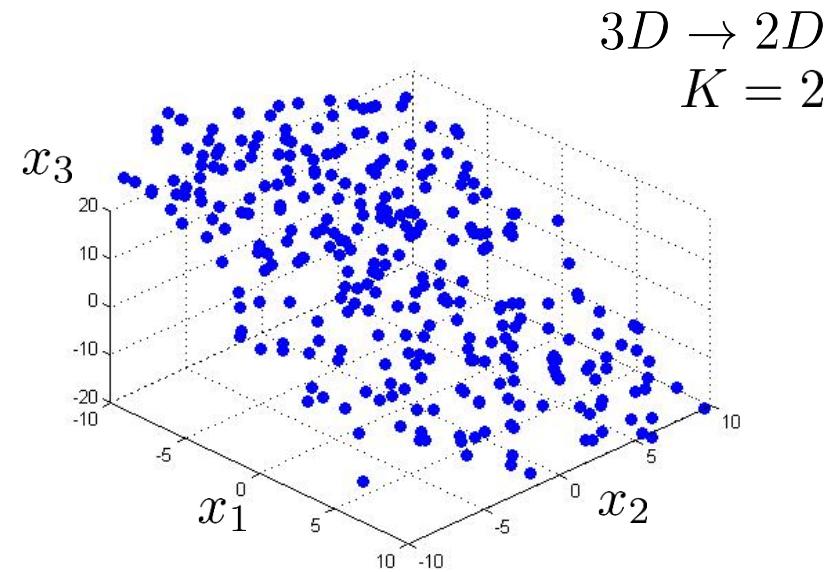
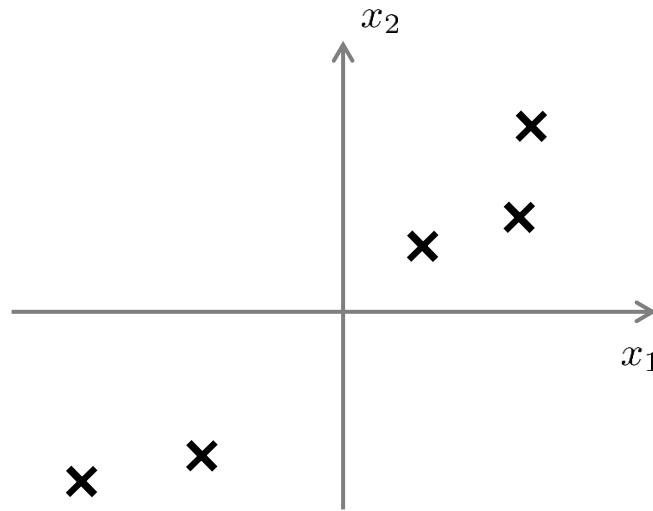
Data Compression

Reduce data from 3D to 2D



Andrew Ng

Principal Component Analysis (PCA) problem formulation



Reduce from 2-dimension to 1-dimension: Find a direction (a vector $u^{(1)} \in \mathbb{R}^n$) onto which to project the data so as to minimize the projection error.

Reduce from n-dimension to k-dimension: Find k vectors $u^{(1)}, u^{(2)}, \dots, u^{(k)}$ onto which to project the data, so as to minimize the projection error.

Principal Component Analysis

Goal: Find r -dim projection that best preserves variance

1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one,
and the rows of A are the eigenvectors

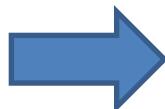
Variance

- Variance:
 - Measure of the deviation from the mean for points in one dimension



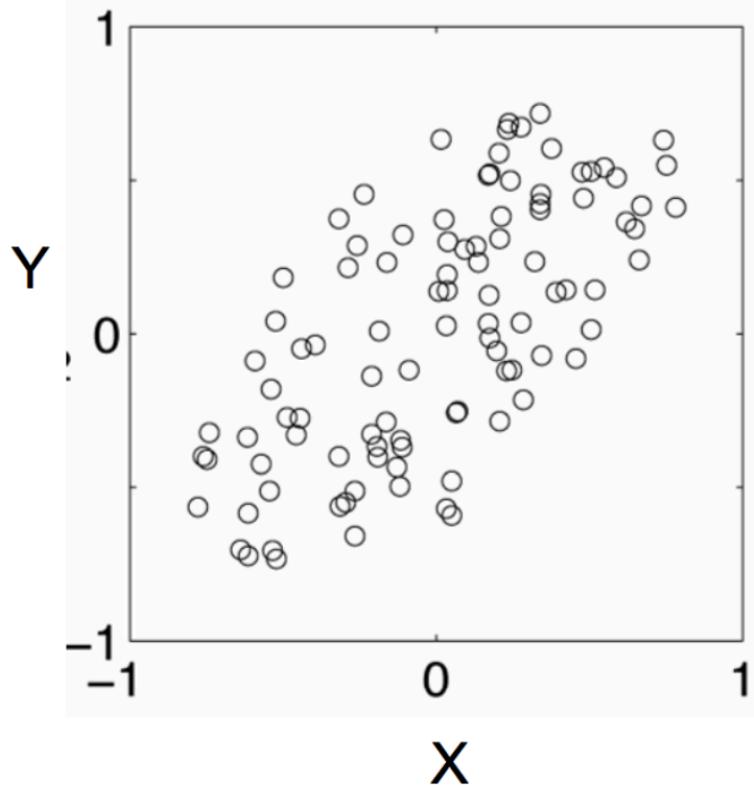
Covariance

- Variance:
 - Measure of the deviation from the mean for points in one dimension
- Covariance:
 - Measure of how much the variation of one dimension implies the variation of another

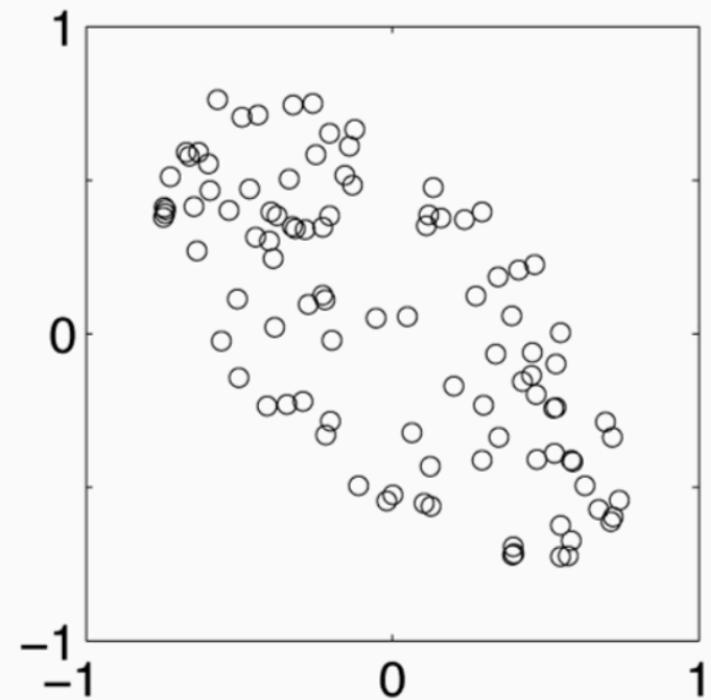


- Covariance is measured between two dimensions
 - Covariance sees if there is a relation between two dimensions
 - Covariance between one dimension is the variance

positive covariance



negative covariance



Positive: Both dimensions increase or decrease together

Negative: While one increase the other decrease

Covariance

- Used to find relationships between dimensions in high dimensional data sets

$$q_{jk} = \frac{1}{N} \sum_{i=1}^N (X_{ij} - E(X_j))(X_{ik} - E(X_k))$$



The Sample mean

Principal Component Analysis

Input:

$$\mathbf{x} \in \mathbb{R}^D: \mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$$

Set of basis vectors: $\mathbf{u}_1, \dots, \mathbf{u}_K$

Summarize a D dimensional vector X with K dimensional feature vector $h(x)$

$$h(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_1 \cdot \mathbf{x} \\ \mathbf{u}_2 \cdot \mathbf{x} \\ \vdots \\ \mathbf{u}_K \cdot \mathbf{x} \end{bmatrix}$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$$

Basis vectors are orthonormal

$$\mathbf{u}_i^T \mathbf{u}_j = 0$$

$$\|\mathbf{u}_j\| = 1$$

New data representation $h(\mathbf{x})$

$$z_j = \mathbf{u}_j \cdot \mathbf{x}$$

$$h(\mathbf{x}) = [z_1, \dots, z_K]^T$$

Principal Component Analysis

$$\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]$$

New data representation $h(\mathbf{x})$

$$h(\mathbf{x}) = \mathbf{U}^T \mathbf{x}$$

$$h(\mathbf{x}) = \mathbf{U}^T (\mathbf{x} - \mu_0)$$

Empirical mean of the data $\mu_0 = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$



Principal Component Analysis

Goal: Find r -dim projection that best preserves variance

1. Compute mean vector μ and covariance matrix Σ of original points
2. Compute eigenvectors and eigenvalues of Σ
3. Select top r eigenvectors
4. Project points onto subspace spanned by them:

$$y = A(x - \mu)$$

where y is the new point, x is the old one,
and the rows of A are the eigenvectors

PCA by SVD

Instead of computing the eigen-decomposition of the covariance $\mathbf{C} = \mathbf{X}^T \mathbf{X}$ (assume that the data matrix \mathbf{X} is centered), you can use SVD to solve PCA.

- Suppose that $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T$, note that $\mathbf{X}^T \mathbf{X} = \mathbf{V}\Sigma^2\mathbf{V}^T$.
- So eigenvectors are just the right singular vectors of \mathbf{X} !

Why the algorithm is correct?

Suppose that the data matrix is $X = \begin{bmatrix} x_1^T \\ \dots \\ x_n^T \end{bmatrix}$, i.e., each row is a data vector (already centered by mean).

Let $U = [u_1, \dots, u_k]$ be the projection matrix, whose columns are the basis of the low-dimensional space.

Then, the coordinate of data in the projected space is $Z = XU$, i.e., each row is the coordinate of a data point.

Using the coordinates and the basis, we can reconstruct the data in the original space as $X' = ZU^T = XUU^T$.

- So the sum of projection error is $\|X' - X\|_{fro}^2$, where the $\|\cdot\|_{fro}^2$ is the square of the Frobenius norm of a matrix, defined as the sum of squares of all elements in X .
- Our goal is to solve the optimization problem:

$$\begin{aligned} \text{minimize}_U \quad & \|XUU^T - X\|_{fro}^2 \\ \text{subject to} \quad & U^T U = I \end{aligned}$$

- The solution U is just the eigenvectors of $X^T X$ corresponding to the top k eigenvalues.
- Proof will be posted on the course website for those interested.

The space of all face images

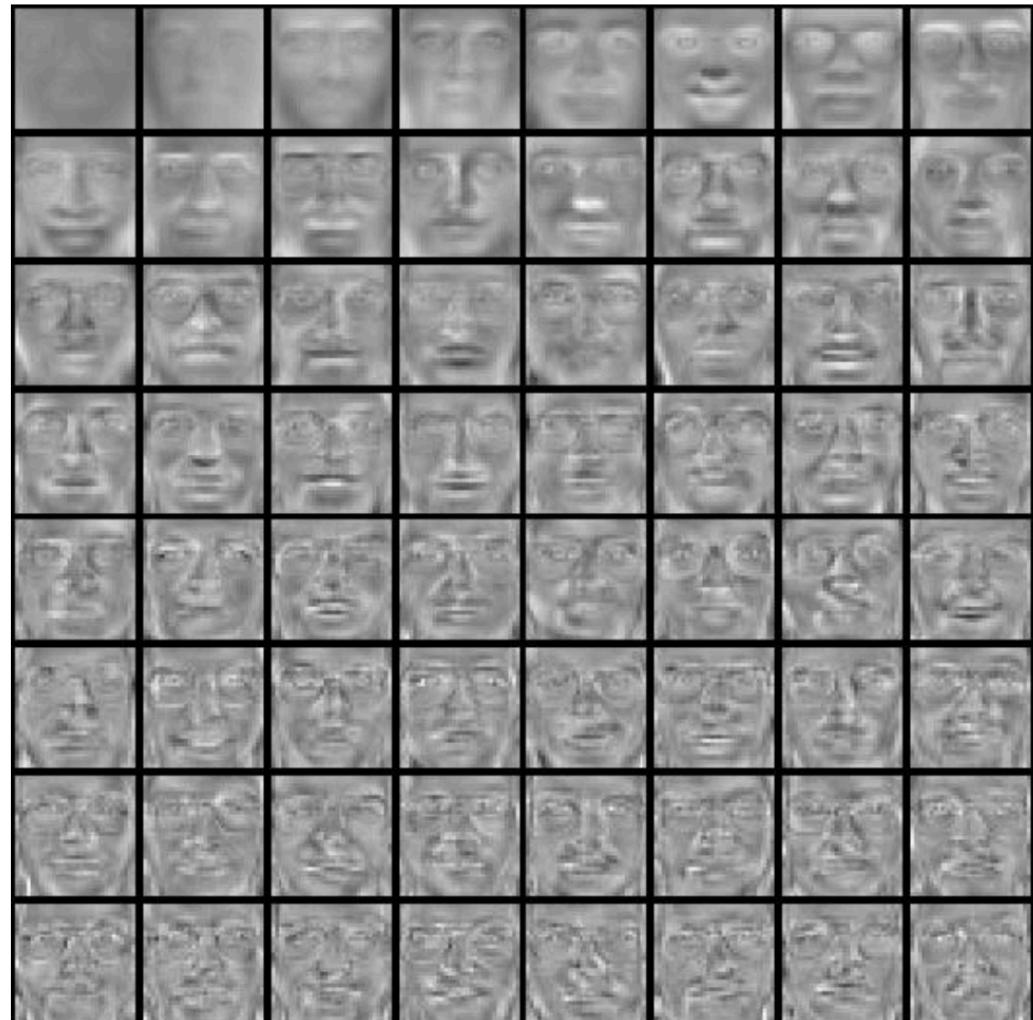
- When viewed as vectors of pixel values, face images are extremely high-dimensional
 - 100×100 image = 10,000 dimensions
 - Slow and lots of storage
- But very few 10,000-dimensional vectors are valid face images
- We want to effectively model the subspace of face images



Eigenfaces example

Top eigenvectors: u_1, \dots, u_k

Mean: μ



Representation and reconstruction

- Face \mathbf{x} in “face space” coordinates:



$$\begin{aligned}\mathbf{x} &\rightarrow [\mathbf{u}_1^T(\mathbf{x} - \mu), \dots, \mathbf{u}_k^T(\mathbf{x} - \mu)] \\ &= [z_1, \dots, z_K]\end{aligned}$$

- Reconstruction:

$$\begin{aligned}\hat{\mathbf{x}} &= \hat{\mu} + \hat{w}_1\mathbf{u}_1 + \hat{w}_2\mathbf{u}_2 + \hat{w}_3\mathbf{u}_3 + \hat{w}_4\mathbf{u}_4 + \dots \\ &= \hat{\mu} + \hat{w}_1\mathbf{u}_1 + \hat{w}_2\mathbf{u}_2 + \hat{w}_3\mathbf{u}_3 + \hat{w}_4\mathbf{u}_4 + \dots\end{aligned}$$

The equation shows the reconstruction of the face $\hat{\mathbf{x}}$ from its mean $\hat{\mu}$ and coefficients $\hat{w}_1, \hat{w}_2, \hat{w}_3, \hat{w}_4, \dots$ multiplied by the principal components $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \dots$. The components are shown as a horizontal stack of seven grayscale images.

Reconstruction

$P = 4$



$P = 200$



$P = 400$



After computing eigenfaces using 400 face images from ORL face database

Application: Image compression



Original Image

Divide the original 372x492 image into patches:

Each patch is an instance that contains 12x12 pixels on a grid

View each as a 144-D vector

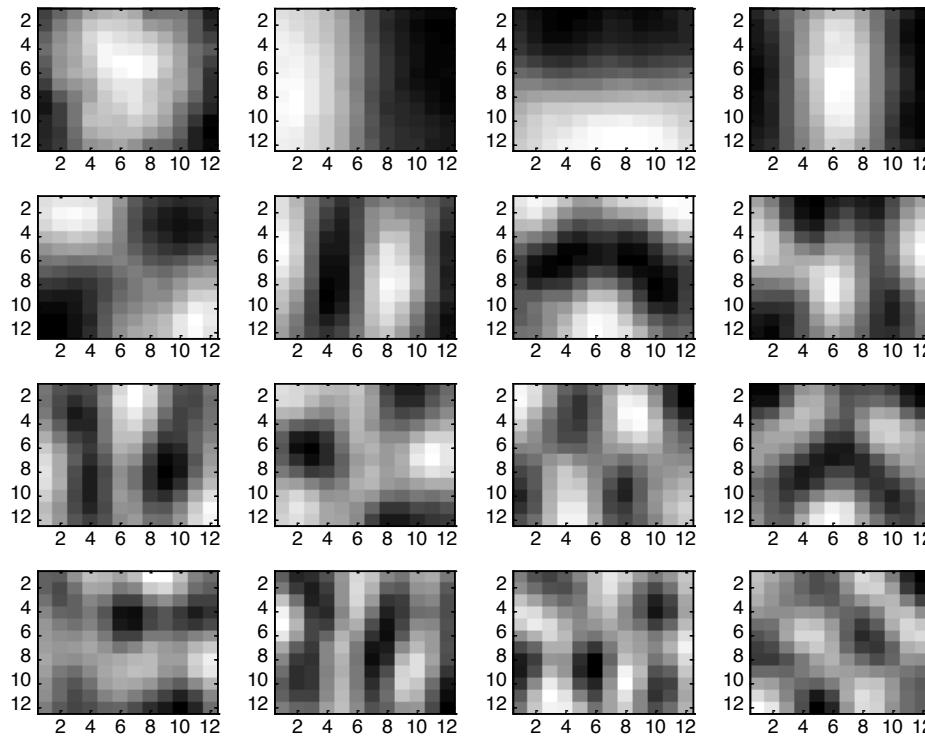
PCA compression: 144D → 60D



PCA compression: 144D → 60D



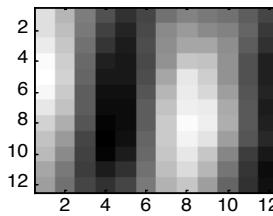
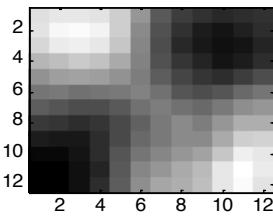
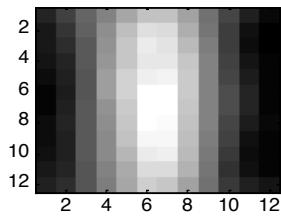
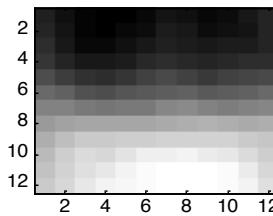
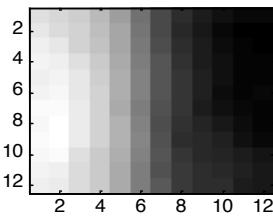
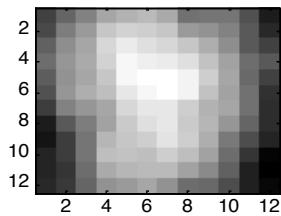
16 most important eigenvectors



PCA compression: 144D → 6D



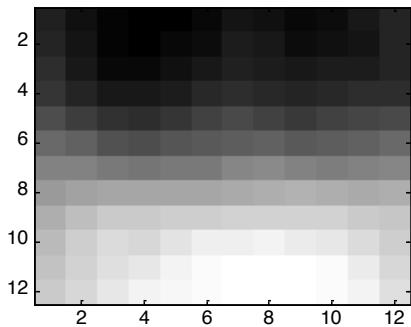
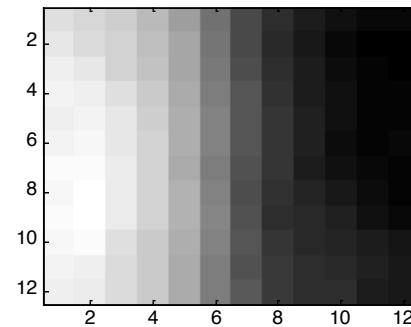
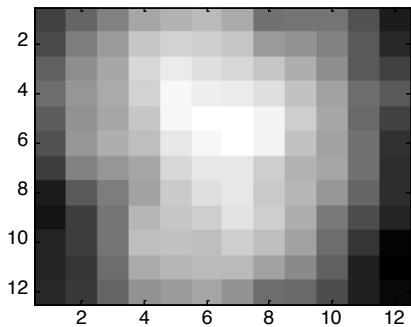
6 most important eigenvectors



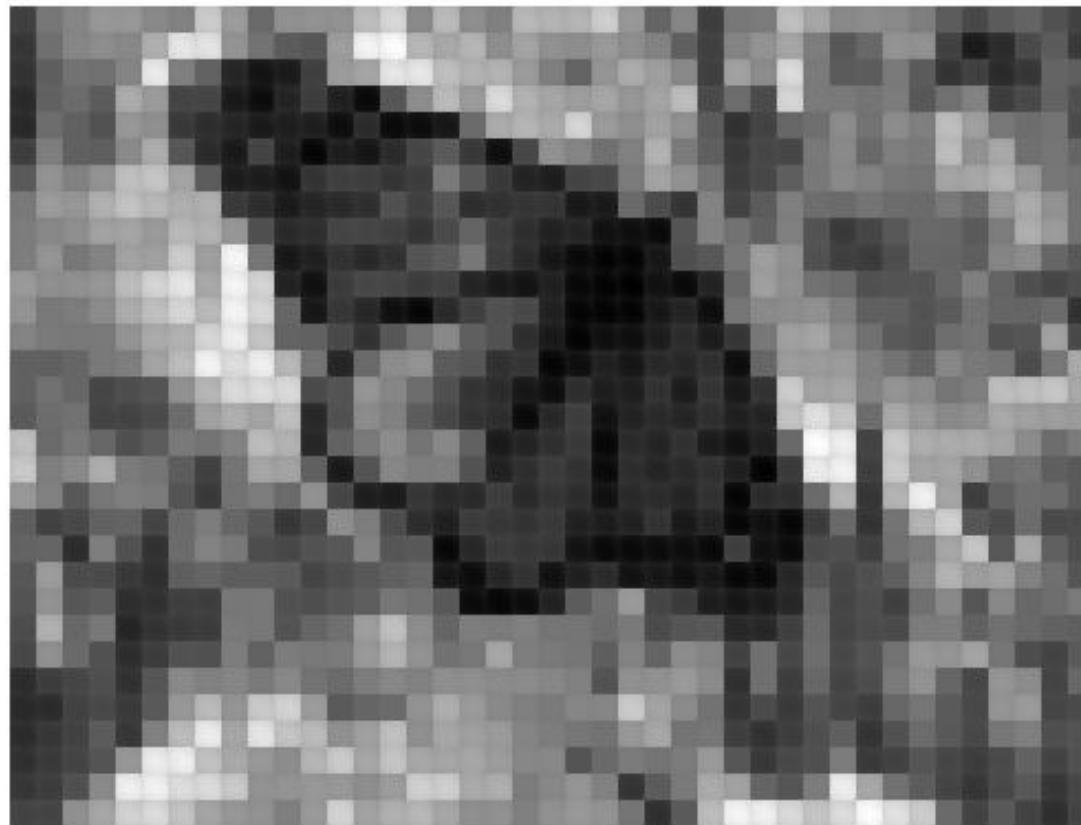
PCA compression: 144D \rightarrow 3D



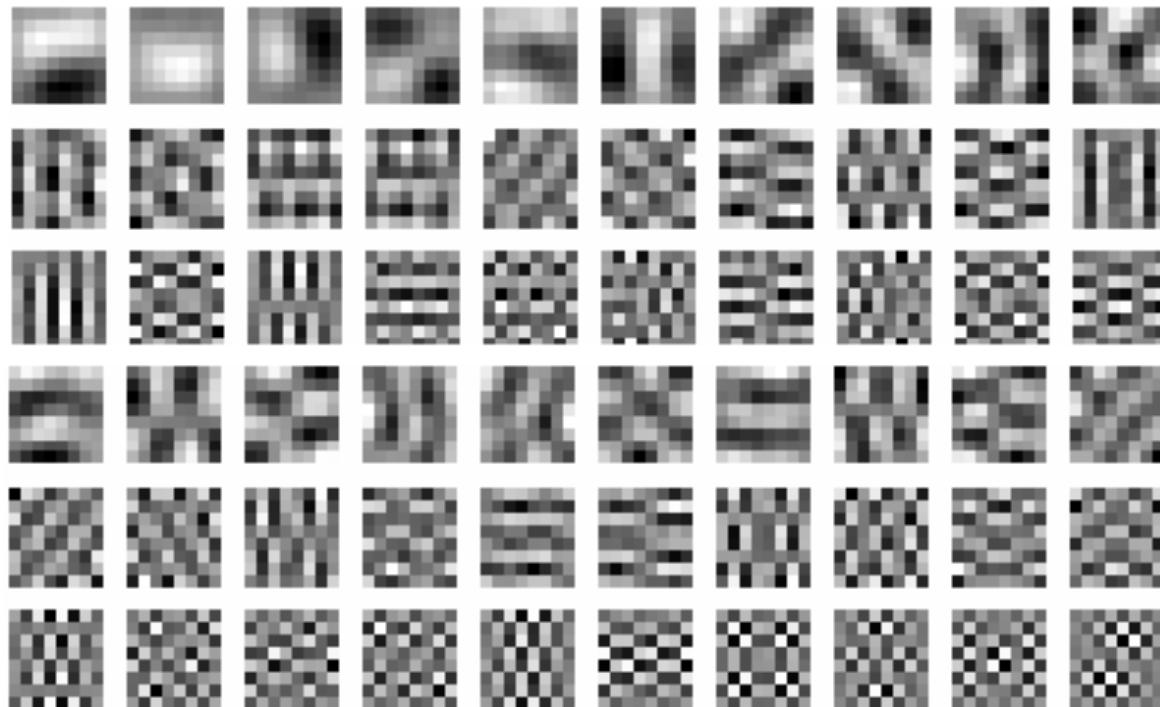
3 most important eigenvectors



PCA compression: 144D → 1D

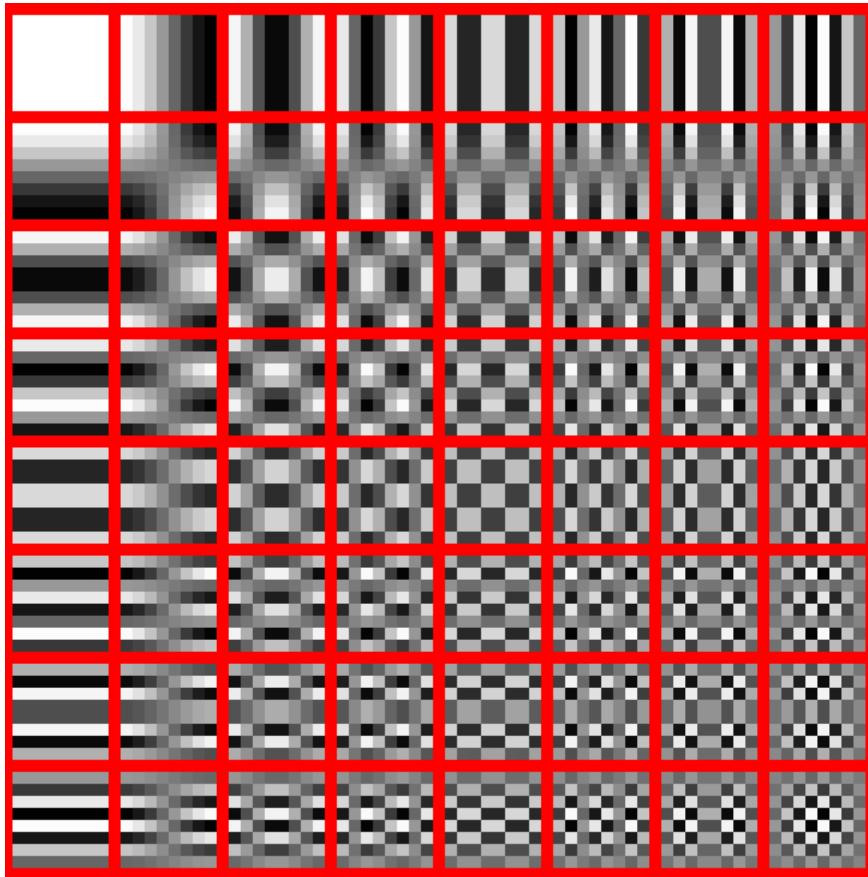


60 most important eigenvectors



Looks like the discrete cosine bases of JPG!...

2D Discrete Cosine Basis



http://en.wikipedia.org/wiki/Discrete_cosine_transform

Dimensionality reduction

- PCA (Principal Component Analysis):
 - Find projection that maximize the variance
- ICA (Independent Component Analysis):
 - Very similar to PCA except that it assumes non-Gaussian features
- Multidimensional Scaling:
 - Find projection that best preserves inter-point distances
- LDA(Linear Discriminant Analysis):
 - Maximizing the component axes for class-separation
- ...
- ...

Agenda

- Principal Component Analysis (PCA)
- K-means

You already know it!