CS 886
Applied Machine Learning
Audio Features, Dimensionality Reduction

Dan Lizotte

University of Waterloo

9 Oct 2012

Recall: SIFTs for image classification

- SIFTs code takes an image, produces a collection (set, bag, ...) of "features." SIFT descriptors stay largely the same even if you rotate or scale the image.

Recall: SIFTs for image classification

- SIFTs code takes an image, produces a collection (set, bag, ...) of "features." SIFT descriptors stay largely the same even if you rotate or scale the image.
- Two problems for classification:
  1. Space of possible SIFTs is $2^{8 \cdot 128} = 2^{1024}$
  2. Different images have different numbers of SIFTs

Recall: SIFTs for image classification

- SIFTs code takes an image, produces a collection (set, bag, ...) of "features." SIFT descriptors stay largely the same even if you rotate or scale the image.
- Two problems for classification:
  1. Space of possible SIFTs is $2^{8 \cdot 128} = 2^{1024}$
  2. Different images have different numbers of SIFTs
- Possible solution to the above problem:
  1. Vector Quantization to reduce to a "vocabulary size" of 100 (say) through clustering or sparse coding
  2. Any SIFT gets mapped to the closest "word" in the "vocabulary," creating a histogram of the "words" in each image, i.e. count how many times each quantized SIFT appears. This is the new feature vector, has length 100 (say)

Features for Audio (and other) Signals

- Sounds are just one-dimensional images
- Can construct global and local features
- Many of the same problems associated with pixels-as-features show up with samples-as-features: Lack of invariance.

Phase and Frequency

- Phase information is present in audio data, can have huge impact on simple similarity (distance, dot-product) but is perceptually irrelevant. (Except maybe stereo.)
- Can we derive features that are invariant to phase?
- [Matlab demo 1]

Phase and Frequency

- Phase information is present in audio data, can have huge impact on simple similarity (distance, dot-product) but is perceptually irrelevant. (Except maybe stereo.)
- Can we derive features that are invariant to phase?
- [Matlab demo 1]
- Fourier analysis takes a signal and re-writes it as a weighted sum of shifted sines and cosines
  - Weights correspond to importance of different frequencies
  - Shifts (phases) can be easily "thrown away"
  - Main methods: "Fourier Transform" (or Fast Fourier Transform, FFT), "Discrete Cosine Transform"

Windowing

- Discrete Fourier transform works perfectly for functions that have an integer number of periods over the samples
- No chance of this happening in practice
- Compensate by “windowing.” Essential for any kind of frequency analysis
- [Matlab demo 2]

The Frequency Domain: Spectrogram

- Many overlapping windows, one after the other
- Plot: Time on x-axis, frequency on y-axis, color indicates power
- Each “column” could be used as a feature vector for the audio playing at that instant in time.
- [Matlab demo 3]

Problems with the Spectrogram

1. All the information is scrunched way down at the bottom in the low frequencies. Perceptually, the frequency ranges 100-200Hz and 10kHz-20kHz should be approximately equally important.
2. Power distribution of natural sounds is... periodic!! Features become redundant.

Problems with the Spectrogram

1. All the information is scrunched way down at the bottom in the low frequencies. Perceptually, the frequency ranges 100-200Hz and 10kHz-20kHz should be approximately equally important.
2. Power distribution of natural sounds is... periodic!! Features become redundant.

1. Warp the frequency axis to "enlarge" the region where the information is (i.e. low frequencies.) Converts from Hz to "mels"
2. Take the log of the result then take Fourier transform *AGAIN* to remove periodicity

- Produces the Mel-Frequency Cepstral Coefficients (MFCC), which are *the* feature vectors for natural sounds. Much like SIFTs in this respect.
- (But, see `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5618550`)
- [Matlab Demo 4]

The Cepstrum

- The spectrum of the spectrum.
- Mapping from Hz to “mels” first gives the Mel-frequency Cepstrum

The Cepstrum

- The spectrum of the spectrum.
- Mapping from Hz to “mels” first gives the Mel-frequency Cepstrum
- Warp the frequency axis to “enlarge” the region where the information is (i.e. low frequencies)
- Take the log of the result then take Fourier transform *AGAIN* to identify periodicity in the spectrogram

Audio signal classification

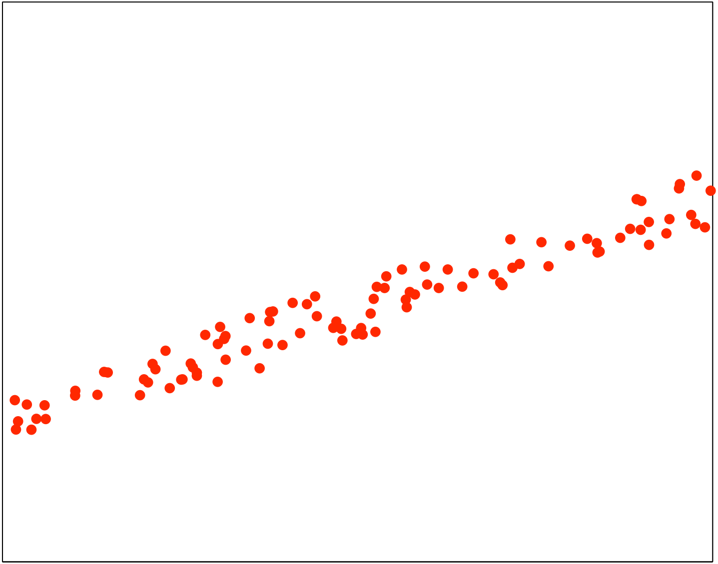
Talk by Anssi Klapuri at
`http://mtg.upf.edu/ismir2004/graduateschool/`

- Concrete problems
  - musical instrument classification
  - musical genre classification
  - percussive instrument transcription
  - music segmentation
  - speaker recognition, language recognition, sound effects retrieval, context awareness, video segmentation using audio,...
- Closely related to sound source recognition in humans
  - includes segmentation (perceptual sound separation) in polyphonic signals
- Many efficient methods have been developed in the speech / speaker recognition field

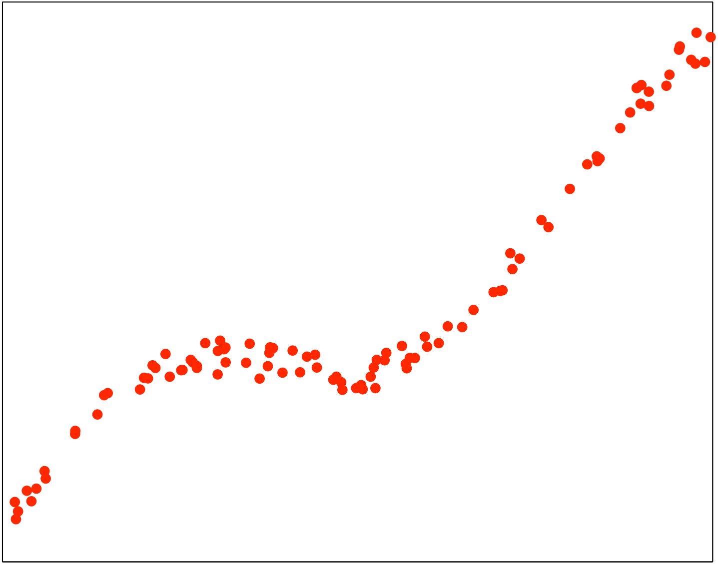
What is dimensionality reduction?

- Dimensionality reduction (or embedding) techniques:
    - Assign instances to real-valued vectors, in a space that is much smaller-dimensional (even 2D or 3D for visualization).
    - Approximately preserve similarity/distance relationships between instances
    - Sometimes, retain the ability to (approximately) reconstruct the original instances
- Some techniques:
    - Axis-aligned: Feature selection
    - Linear: Principal components analysis
    - Non-linear
        - Kernel PCA
        - Independent components analysis
        - Self-organizing maps
        - Multi-dimensional scaling

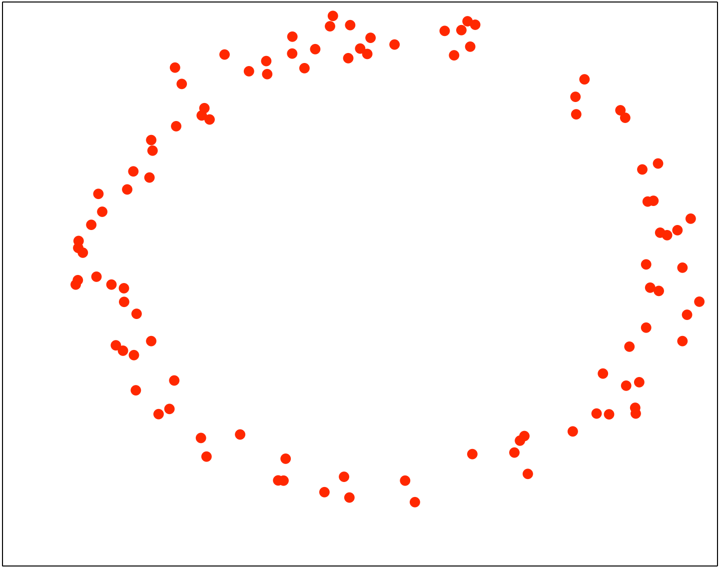
What is the true dimensionality of this data?



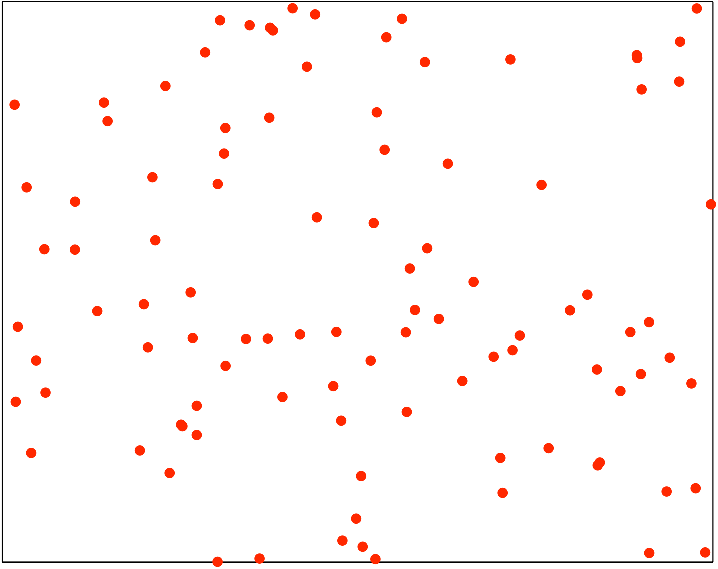
What is the true dimensionality of this data?



What is the true dimensionality of this data?



What is the true dimensionality of this data?



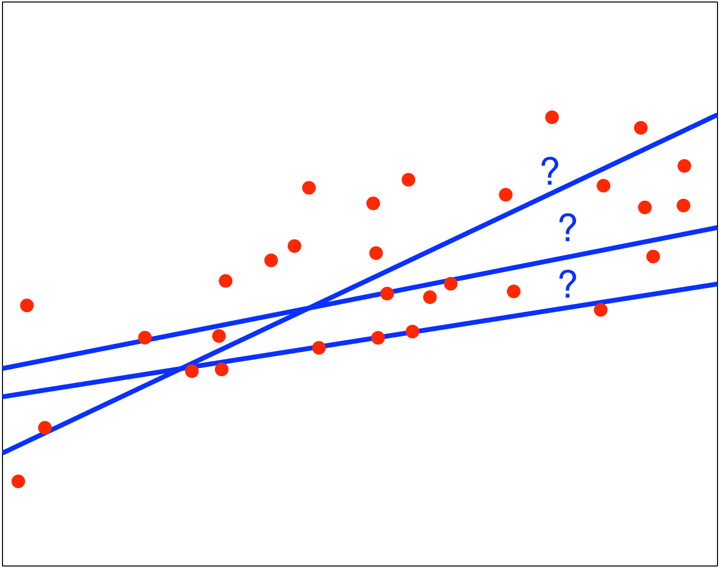
Remarks

- All dimensionality reduction techniques are based on an implicit assumption that the data lies along some *low-dimensional manifold*
- This is the case for the first three examples, which lie along a 1-dimensional manifold despite being plotted in 2D
- In the last example, the data has been generated randomly in 2D, so no dimensionality reduction is possible without losing information
- The first three cases are in increasing order of difficulty, from the point of view of existing techniques.

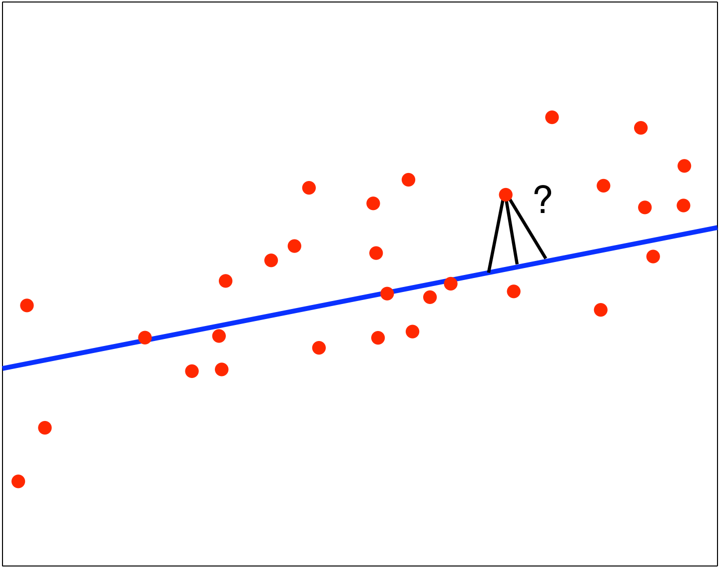
Simple Principal Component Analysis (PCA)

- Given: n instances, each being a length-p real vector.
- Suppose we want a 1-dimensional representation of that data, instead of p-dimensional.
- Specifically, we will:
  - Choose a line in $\mathbb{R}^p$ that "best represents" the data.
  - Assign each data object to a point along that line.
  - (Identifying a point on a line just requires a scalar: How far along the line is the point?)

Which line is best?



How do we assign points to lines?



Reconstruction error

- Let the line be represented as $\mathbf{b} + \alpha\mathbf{v}$ for $\mathbf{b}, \mathbf{v} \in \mathbb{R}^p$, $\alpha \in \mathbb{R}$. For convenience assume $\|\mathbf{v}\| = 1$.
- Each instance $\mathbf{x}_i$ is associated with a point on the line $\hat{\mathbf{x}}_i = \mathbf{b} + \alpha_i\mathbf{v}$.
  - Instance $\mathbf{x}_i$ is *encoded* as a scalar α_i
- We want to choose $\mathbf{b}$, $\mathbf{v}$, and the α_i to minimize the total reconstruction error over all data points, measured using Euclidean distance:

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

$$\begin{array}{rl} \text{min} & \sum_{i=1}^{n} \|\mathbf{x}_i - (\mathbf{b} + \alpha_i\mathbf{v})\|^2 \\ \text{w.r.t.} & \mathbf{b}, \mathbf{v}, \alpha_i, i = 1, \ldots n \\ \text{s.t.} & \|\mathbf{v}\|^2 = 1 \end{array}$$

Solving the optimization problem [HTF Ch. 14.5]

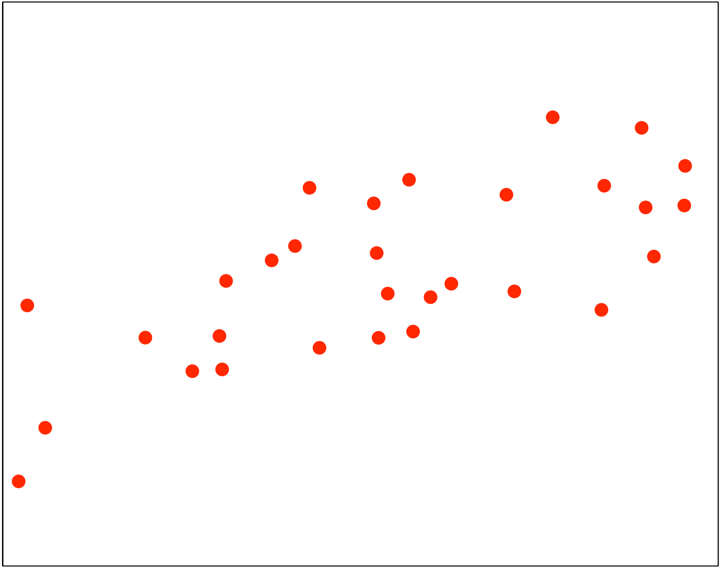
$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} \|\mathbf{x}_i - (\mathbf{b} + \alpha_i \mathbf{v})\|^2 \\ \text{w.r.t.} \quad & \mathbf{b}, \mathbf{v}, \alpha_i, i = 1, \ldots n \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Turns out the optimal $\mathbf{b}$ is just the sample mean of the data, $\mathbf{b} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i$
- This means that the best line goes through the mean of the data. Typically, we subtract the mean first. Assuming it's zero:

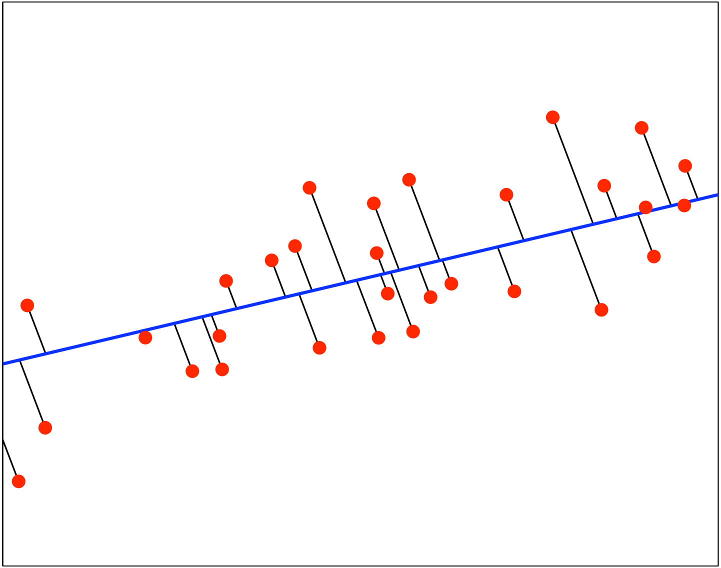
$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2 \\ \text{w.r.t.} \quad & \mathbf{v}, \alpha_i, i = 1, \ldots n \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Consider fixing $\mathbf{v}$. The optimal α_i is given by *projecting* $\mathbf{x}_i$ onto $\mathbf{v}$.

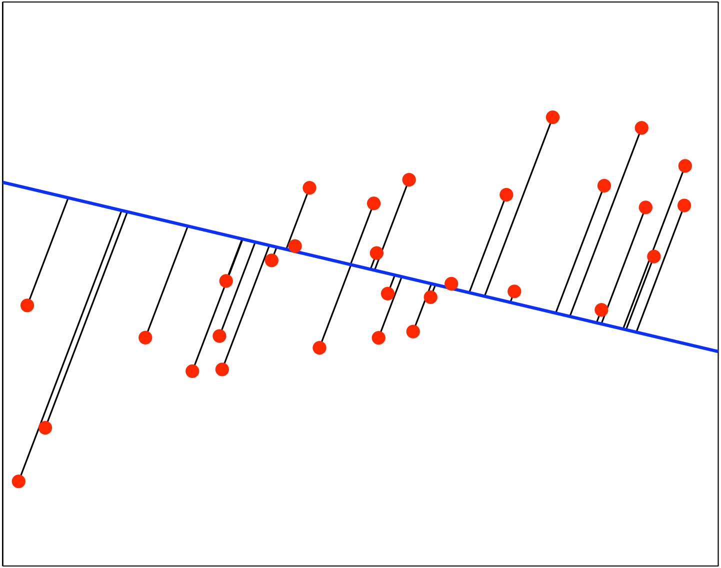
Example data



Example with $\mathbf{v} \propto (1, 0.3)$



Example with $\mathbf{v} \propto (1, -0.3)$



Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \text{max} \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{array}{rl} \max & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} & \mathbf{v} \\ \text{s.t.} & \|\mathbf{v}\|^2 = 1 \end{array}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \text{max} \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{array}{rl} \max & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} & \mathbf{v} \\ \text{s.t.} & \|\mathbf{v}\|^2 = 1 \end{array}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \text{max} \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \text{max} \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimizing...

Let's look at the objective we want to minimize:

- $\sum_{i=1}^{n} \|\mathbf{x}_i - \alpha_i \mathbf{v}\|^2$, min over $\mathbf{v}, \alpha_i$ s.t. $\|\mathbf{v}\| = 1$
- $\sum_{i=1}^{n} (\mathbf{x}_i - \alpha_i \mathbf{v})^\mathsf{T} (\mathbf{x}_i - \alpha_i \mathbf{v})$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2 \mathbf{v}^\mathsf{T} \mathbf{v}$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\alpha_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \alpha_i^2$ (Assumed $\mathbf{v}$ was a unit vector.)
- $\implies \alpha_i^* = \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - 2\mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i + \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\sum_{i=1}^{n} \mathbf{x}_i^\mathsf{T} \mathbf{x}_i - \sum_{i=1}^{n} \mathbf{v}^\mathsf{T} \mathbf{x}_i \mathbf{v}^\mathsf{T} \mathbf{x}_i$
- $\mathrm{tr}(X^\mathsf{T} X) - \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v}$

$$\begin{aligned} \text{max} \quad & \mathbf{v}^\mathsf{T} (X^\mathsf{T} X) \mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

Optimal choice of $\mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^{\mathsf{T}}(X^{\mathsf{T}}X)\mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Forming the Lagrangian of the above problem and setting derivative to zero gives $(X^{\mathsf{T}}X)\mathbf{v} = \lambda\mathbf{v}$ as feasible solutions. (Left to reader.)

Optimal choice of $\mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T}(X^\mathsf{T}X)\mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Forming the Lagrangian of the above problem and setting derivative to zero gives $(X^\mathsf{T}X)\mathbf{v} = \lambda\mathbf{v}$ as feasible solutions. (Left to reader.)
- Recall: an *eigenvector* $\mathbf{u}$ of a matrix A satisfies $A\mathbf{u} = \lambda\mathbf{u}$, where $\lambda \in \mathbb{R}$ is the *eigenvalue*.

Optimal choice of $\mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T}(X^\mathsf{T}X)\mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Forming the Lagrangian of the above problem and setting derivative to zero gives $(X^\mathsf{T}X)\mathbf{v} = \lambda\mathbf{v}$ as feasible solutions. (Left to reader.)
- Recall: an *eigenvector* $\mathbf{u}$ of a matrix A satisfies $A\mathbf{u} = \lambda\mathbf{u}$, where $\lambda \in \mathbb{R}$ is the *eigenvalue*.
- Fact: The matrix $X^\mathsf{T}X$ has p non-negative eigenvalues and p orthogonal eigenvectors.

Optimal choice of $\mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T}(X^\mathsf{T}X)\mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Forming the Lagrangian of the above problem and setting derivative to zero gives $(X^\mathsf{T}X)\mathbf{v} = \lambda\mathbf{v}$ as feasible solutions. (Left to reader.)
- Recall: an *eigenvector* $\mathbf{u}$ of a matrix A satisfies $A\mathbf{u} = \lambda\mathbf{u}$, where $\lambda \in \mathbb{R}$ is the *eigenvalue*.
- Fact: The matrix $X^\mathsf{T}X$ has p non-negative eigenvalues and p orthogonal eigenvectors.
- Thus, $\mathbf{v}$ must be an eigenvector of $(X^\mathsf{T}X)$.

Optimal choice of $\mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T}(X^\mathsf{T}X)\mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Forming the Lagrangian of the above problem and setting derivative to zero gives $(X^\mathsf{T}X)\mathbf{v} = \lambda\mathbf{v}$ as feasible solutions. (Left to reader.)
- Recall: an *eigenvector* $\mathbf{u}$ of a matrix A satisfies $A\mathbf{u} = \lambda\mathbf{u}$, where $\lambda \in \mathbb{R}$ is the *eigenvalue*.
- Fact: The matrix $X^\mathsf{T}X$ has p non-negative eigenvalues and p orthogonal eigenvectors.
- Thus, $\mathbf{v}$ must be an eigenvector of $(X^\mathsf{T}X)$.
- The $\mathbf{v}$ that maximizes $\mathbf{v}^T(X^\mathsf{T}X)\mathbf{v}$ is the eigenvector of $(X^\mathsf{T}X)$ with the largest eigenvalue

Another view of $\mathbf{v}$

$$\begin{aligned} \max \quad & \mathbf{v}^\mathsf{T}(X^\mathsf{T}X)\mathbf{v} \\ \text{w.r.t.} \quad & \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{v}\|^2 = 1 \end{aligned}$$

- Recall $\mathbf{x}_i^\mathsf{T}\mathbf{v}$ is our low-dimensional representation of $\mathbf{x}_i$
- $\mathbf{v}^\mathsf{T}(X^\mathsf{T}X)\mathbf{v} = \sum_i(\mathbf{x}_i^\mathsf{T}\mathbf{v})^2 = \mathrm{Var}(\mathbf{x}_i^\mathsf{T}\mathbf{v})$
- The optimal $\mathbf{v}$ produces an encoding that has as much variance as possible
- (Dan you should draw some pictures.)

Recall: Covariance

- $(X^{\mathsf{T}}X)$ is an $p \times p$ matrix whose i, j entry is proportional to the *estimated covariance* between the ith and jth feature
- Covariance quantifies a *linear relationship* (if any) between two random variables X and Y.

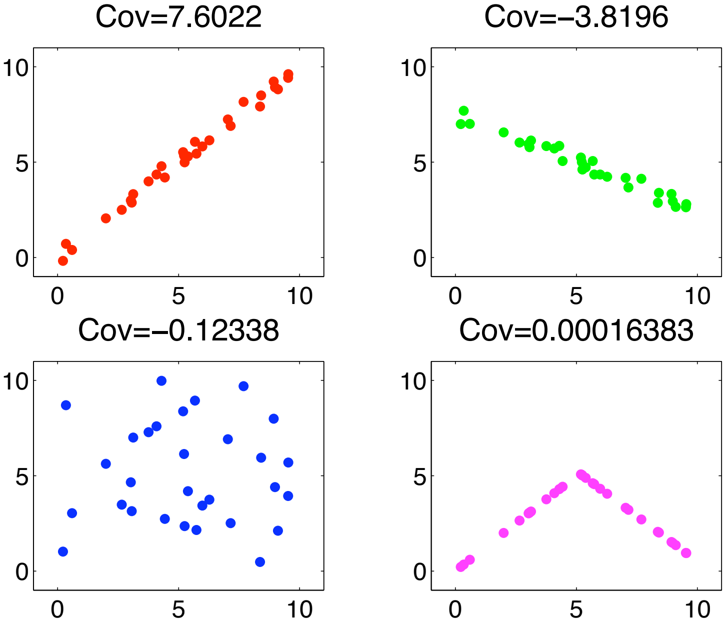
$$Cov(X, Y) = E\{(X - E(X))(Y - E(Y))\}$$

- Given n samples of X and Y, covariance can be estimated as

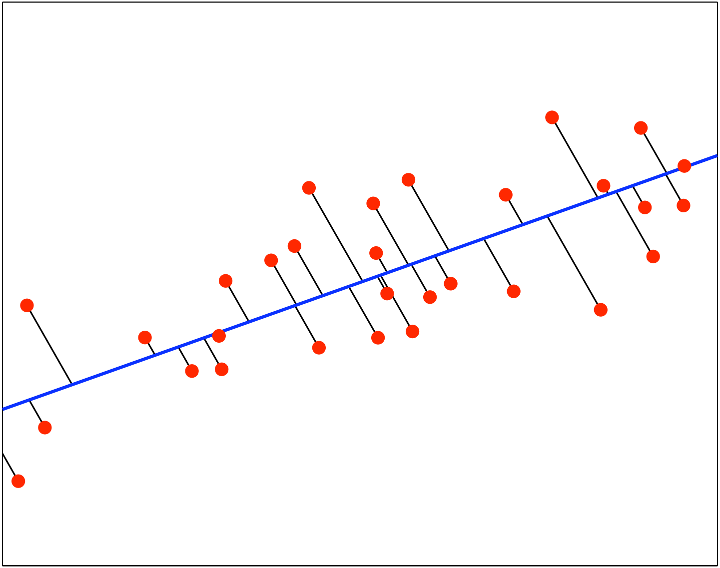
$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_X)(y_i - \mu_Y) \ ,$$

where $\mu_X = (1/n)\sum_{i=1}^{n} x_i$ and $\mu_Y = (1/n)\sum_{i=1}^{n} y_i$.
- Note: $Cov(X, X) = Var(X)$.

Covariance example



Example with optimal line: $\mathbf{b} = (0.54, 0.52)$, $\mathbf{v} \propto (1, 0.45)$



Remarks

- The line $\mathbf{b} + \alpha\mathbf{v}$ is the *first principal component*.
- The variance of the data along projected onto the line $\mathbf{b} + \alpha\mathbf{v}$ is as large as if they are projected onto any other line.
- $\mathbf{b}$, $\mathbf{v}$, and the α_i can be computed easily in polynomial time.

Reduction to d dimensions

- More generally, we can create a d-dimensional representation of our data by projecting the instances onto a hyperplane $\mathbf{b} + \alpha^1 \mathbf{v}_1 + \ldots + \alpha^d \mathbf{v}_d$.
- If we assume the $\mathbf{v}_j$ are of unit length and orthogonal, then the optimal choices are:
  - $\mathbf{b}$ is the mean of the data (as before)
  - The $\mathbf{v}_j$ are orthogonal eigenvectors of S corresponding to its d largest eigenvalues.
  - Each instance is projected orthogonally on the hyperplane.

Remarks

- **b**, the eigenvalues, the $\mathbf{v}_j$, and the projections of the instances can all be computing in polynomial time, e.g. using Singular Value Decomposition.

$$X_{n\times p} = U_{n\times n} D_{n\times p} V^{\mathsf{T}}_{p\times p}$$

- Columns of U are left-eigenvectors, diagonal of D are sqrts of eigenvalues ("singular values"), V are right-eigenvectors

Remarks

- **b**, the eigenvalues, the $\mathbf{v}_j$, and the projections of the instances can all be computing in polynomial time, e.g. using Singular Value Decomposition.

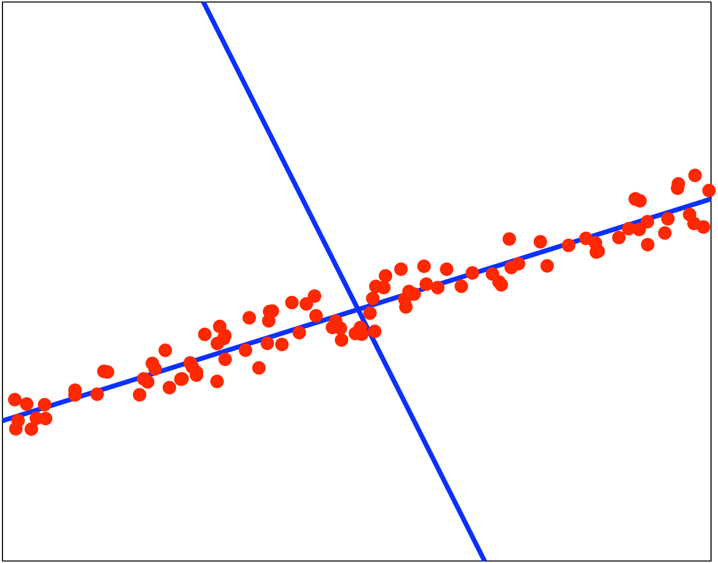
$$X_{n\times p} = U_{n\times n} D_{n\times p} V^{\mathsf{T}}_{p\times p}$$

- Columns of U are left-eigenvectors, diagonal of D are sqrts of eigenvalues ("singular values"), V are right-eigenvectors
- The magnitude of the j^{th}-largest eigenvalue, λ_j, tells you how much variability in the data is captured by the j^{th} principal component
- So you have feedback on how to choose d!
- When the eigenvalues are sorted in decreasing order, the proportion of the variance captured by the first d components is:

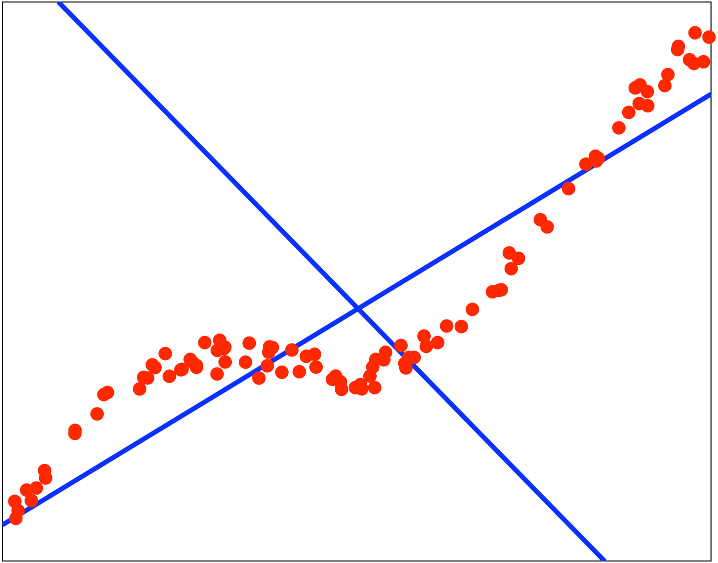
$$\frac{\lambda_1 + \cdots + \lambda_d}{\lambda_1 + \cdots + \lambda_d + \lambda_{d+1} + \cdots + \lambda_n}$$

- So if a "big" drop occurs in the eigenvalues at some point, that suggests a good dimension cutoff

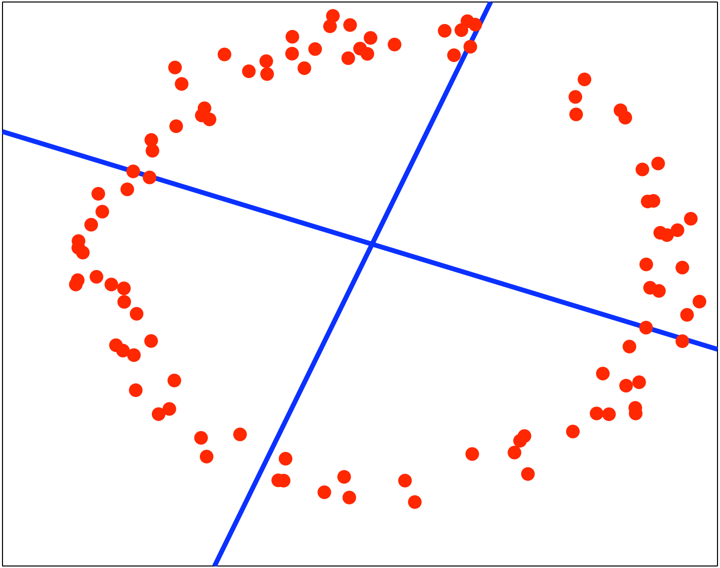
Example: $\lambda_1 = 0.0938, \lambda_2 = 0.0007$



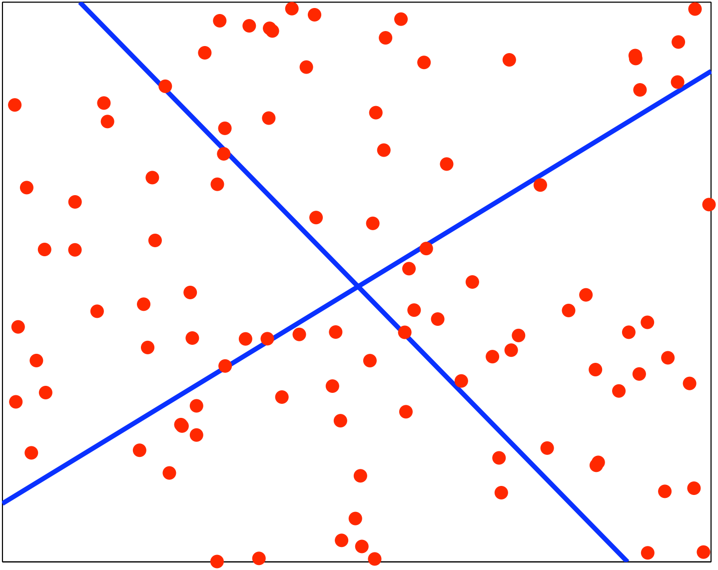
Example: $\lambda_1 = 0.1260, \lambda_2 = 0.0054$



Example: $\lambda_1 = 0.0884, \lambda_2 = 0.0725$



Example: $\lambda_1 = 0.0881, \lambda_2 = 0.0769$



More remarks

- Outliers have a big effect on the covariance matrix, so they can affect the eigenvectors quite a bit
- A simple examination of the pairwise distances between instances can help discard points that are very far away (for the purpose of PCA)
- If the variances in the original dimensions vary considerably, they can "muddle" the true correlations. There are two solutions:
  - work with the correlation of the original data, instead of covariance matrix
  - normalize the input dimensions individually before PCA
- In certain cases, the eigenvectors are meaningful; e.g. in vision, they can be displayed as images ("eigenfaces")

Uses of PCA

- Pre-processing for a supervised learning algorithm, e.g. for image data, robotic sensor data
- Used with great success in image and speech processing
- Visualization
- Exploratory data analysis
- Removing the linear component of a signal (before fancier non-linear models are applied)

Eigenfaces

- L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America A 4 (3): 519-524.
- Adapted from Wikipedia: `http://en.wikipedia.org/wiki/Eigenface`
  1. Prepare a training set of face images taken under the same lighting conditions, normalized to have the eyes and mouths aligned, resampled to a common pixel resolution. Each image is treated as one vector, by concatenating the rows of pixels

Eigenfaces

- L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America A 4 (3): 519-524.
- Adapted from Wikipedia: `http://en.wikipedia.org/wiki/Eigenface`
    1. Prepare a training set of face images taken under the same lighting conditions, normalized to have the eyes and mouths aligned, resampled to a common pixel resolution. Each image is treated as one vector, by concatenating the rows of pixels
    2. Subtract the mean vector.

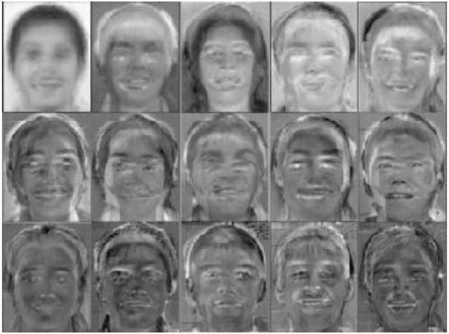
Eigenfaces

- L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America A 4 (3): 519-524.
- Adapted from Wikipedia: `http://en.wikipedia.org/wiki/Eigenface`
  1. Prepare a training set of face images taken under the same lighting conditions, normalized to have the eyes and mouths aligned, resampled to a common pixel resolution. Each image is treated as one vector, by concatenating the rows of pixels
  2. Subtract the mean vector.
  3. Calculate the eigenvectors and eigenvalues of $(X^T X)$. Each eigenvector has the same dimensionality (number of components) as the original images, and thus can itself be seen as an image. The eigenvectors are called eigenfaces. They are the directions in which the images differ from the mean image.

Eigenfaces

- L. Sirovich and M. Kirby (1987). "Low-dimensional procedure for the characterization of human faces". Journal of the Optical Society of America A 4 (3): 519-524.
- Adapted from Wikipedia: `http://en.wikipedia.org/wiki/Eigenface`
    1. Prepare a training set of face images taken under the same lighting conditions, normalized to have the eyes and mouths aligned, resampled to a common pixel resolution. Each image is treated as one vector, by concatenating the rows of pixels
    2. Subtract the mean vector.
    3. Calculate the eigenvectors and eigenvalues of $(X^T X)$. Each eigenvector has the same dimensionality (number of components) as the original images, and thus can itself be seen as an image. The eigenvectors are called eigenfaces. They are the directions in which the images differ from the mean image.
    4. Choose the principal components. The eigenvectors (eigenfaces) with largest associated eigenvalue are kept.

Eigenfaces



Beyond PCA: Nonlinear dimensionality reduction

- Kernel PCA (but you don't get the eigenvectors)
- Self-Organizing Maps
- Isomap
- Locally Linear Embedding
- http://en.wikipedia.org/wiki/Nonlinear_dimensionality_reduction

Application: Netflix Recommender

- Given: An enormous matrix $Y_{n \times p}$ containing the ratings by n users of p movies. Ratings are all $\in \{1, 2, 3, 4, 5\}$.

Application: Netflix Recommender

- Given: An enormous matrix $Y_{n \times p}$ containing the ratings by n users of p movies. Ratings are all $\in \{1, 2, 3, 4, 5\}$.
- Most of the ratings are missing. The point is to reconstruct Y. "As well as possible."

Application: Netflix Recommender

- Given: An enormous matrix $Y_{n \times p}$ containing the ratings by n users of p movies. Ratings are all $\in \{1, 2, 3, 4, 5\}$.
- Most of the ratings are missing. The point is to reconstruct Y. "As well as possible."
- Recall, SVD gives you:

$$Y_{n \times p} = U_{n \times n} D_{n \times p} V^{\mathsf{T}}_{p \times p}$$

but requires a complete Y, which we don't have.

Application: Netflix Recommender

- Given: An enormous matrix $Y_{n \times p}$ containing the ratings by n users of p movies. Ratings are all $\in \{1, 2, 3, 4, 5\}$.
- Most of the ratings are missing. The point is to reconstruct Y. "As well as possible."
- Recall, SVD gives you:

$$Y_{n \times p} = U_{n \times n} D_{n \times p} V^{\mathsf{T}}_{p \times p}$$

but requires a complete Y, which we don't have.

- First, let's rearrange the decomposition (assume $n > p$):

$$Y_{n \times p} = U_{n \times p} V^{\mathsf{T}}_{p \times p}$$

Application: Netflix Recommender

- Given: An enormous matrix $Y_{n \times p}$ containing the ratings by n users of p movies. Ratings are all $\in \{1, 2, 3, 4, 5\}$.
- Most of the ratings are missing. The point is to reconstruct Y. "As well as possible."
- Recall, SVD gives you:

$$Y_{n \times p} = U_{n \times n} D_{n \times p} V^{\mathsf{T}}_{p \times p}$$

  but requires a complete Y, which we don't have.
- First, let's rearrange the decomposition (assume $n > p$):

$$Y_{n \times p} = U_{n \times p} V^{\mathsf{T}}_{p \times p}$$

- Solving this is way too easy: $U = Y$, $V = I$.

SVD with Missing Data

$$Y_{n \times p} = U_{n \times p} V_{p \times p}^{\mathsf{T}}$$

- Solving this is way too easy: $U = Y$, $V = I$. We have $n \times p$ plus $p \times p$ parameters to fit $n \times p$ targets (elements of Y). Massive overfitting.

SVD with Missing Data

$$Y_{n\times p} = U_{n\times p} V^{\mathsf{T}}_{p\times p}$$

- Solving this is way too easy: $U = Y$, $V = I$. We have $n \times p$ plus $p \times p$ parameters to fit $n \times p$ targets (elements of Y). Massive overfitting.
- "Force" generalization by choosing a $c \ll p$ and asserting

$$Y_{n\times p} \approx U_{n\times c} V^{\mathsf{T}}_{p\times c}$$

SVD with Missing Data

$$Y_{n\times p} = U_{n\times p} V^{\mathsf{T}}_{p\times p}$$

- Solving this is way too easy: $U = Y$, $V = I$. We have $n \times p$ plus $p \times p$ parameters to fit $n \times p$ targets (elements of Y). Massive overfitting.
- "Force" generalization by choosing a $c \ll p$ and asserting

$$Y_{n\times p} \approx U_{n\times c} V^{\mathsf{T}}_{p\times c}$$

- What do we mean by $\approx$? Minimize squared error over the observed data:

$$\min_{U,V} \sum_{i=1}^{n} \sum_{j=1}^{p} \mathbf{1}_{ij} (\mathbf{u}_i \mathbf{v}_j^T - Y_{ij})^2$$

Final Touch: Regularization

- What do we mean by ≈? Minimize squared error over the observed data, don't let the matrices entries grow too large:

$$\min_{U,V} \sum_{i=1}^{n} \sum_{j=1}^{p} \mathbf{1}_{ij}(\mathbf{u}_i \mathbf{v}_j^T - Y_{ij})^2 + \lambda \sum_{ij} \mathbf{1}_{ij} \left(\|\mathbf{u}_i\|^2 + \|\mathbf{v}_j\|^2 \right)$$

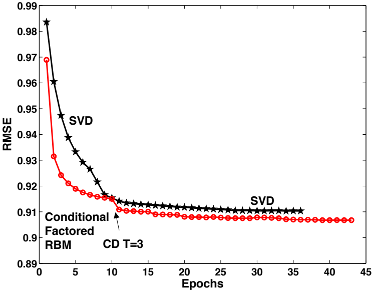


Figure 4. Performance of the conditional factored RBM vs. SVD with $C = 40$ factors. The y-axis displays RMSE (root mean squared error), and the x-axis shows the number of epochs, or passes through the entire training dataset.

- Salakhutdinov, Mnih, Hinton, "Restricted Boltzmann Machines for Collaborative Filtering" `http://www.machinelearning.org/proceedings/icml2007/papers/407.pdf` presents an alternative model also