GENERAL ASSEMBLY

INTRO TO DATA SCIENCE
SESSION 16: DIMENSIONALITY REDUCTION

Rob Hall
DAT13 SF // April 29, 2015

AGENDA

I. OVERVIEW OF DIMENSIONALITY REDUCTION
II. PRINCIPAL COMPONENTS ANALYSIS
III. SINGULAR VALUE DECOMPOSITION

EXERCISE:
IV. PCA

I. DIMENSIONALITY REDUCTION

	continuous	categorical
supervised	???	???
unsupervised	???	???

	continuous	categorical
supervised	regression	classification
unsupervised	dimension reduction	clustering

Q: What is dimensionality reduction?

Q: What is dimensionality reduction?
A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

Q: What is dimensionality reduction?
A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.

Q: What is dimensionality reduction?
A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.

Dimensionality reduction is frequently performed as a pre-processing step before another learning algorithm is applied.

Q: What is the goal of dimensionality reduction?

- reduce computational expense
- reduce susceptibility to overfitting
- reduce noise in the dataset
- enhance our intuition

Q: What are the motivations for dimensionality reduction?

Q: What are the motivations for dimensionality reduction?

The number of features in our dataset can be difficult to manage, or even misleading (e.g., if the relationships are actually simpler than they appear).

For example, suppose we have a dataset with some features that are related to each other.

For example, suppose we have a dataset with some features that are related to each other.

Ideally, we would like to eliminate this redundancy and consolidate the number of variables we’re looking at.

For example, suppose we have a dataset with some features that are related to each other.

Ideally, we would like to eliminate this redundancy and consolidate the number of variables we’re looking at.
If these relationships are *linear*, then we can use well-established techniques like PCA/SVD.

To say this more intuitively, we want to go from a more complex representation of our data to a less complex one (while retaining as much of the signal in our data as possible).

We can do this by looking at our data “from another angle”.

In doing this, we tease out the “principal components” of our data.

Q: What is the goal of dimensionality reduction?

Q: What is the goal of dimensionality reduction?

We'd like to analyze the data using the most meaningful basis (or coordinates) possible.

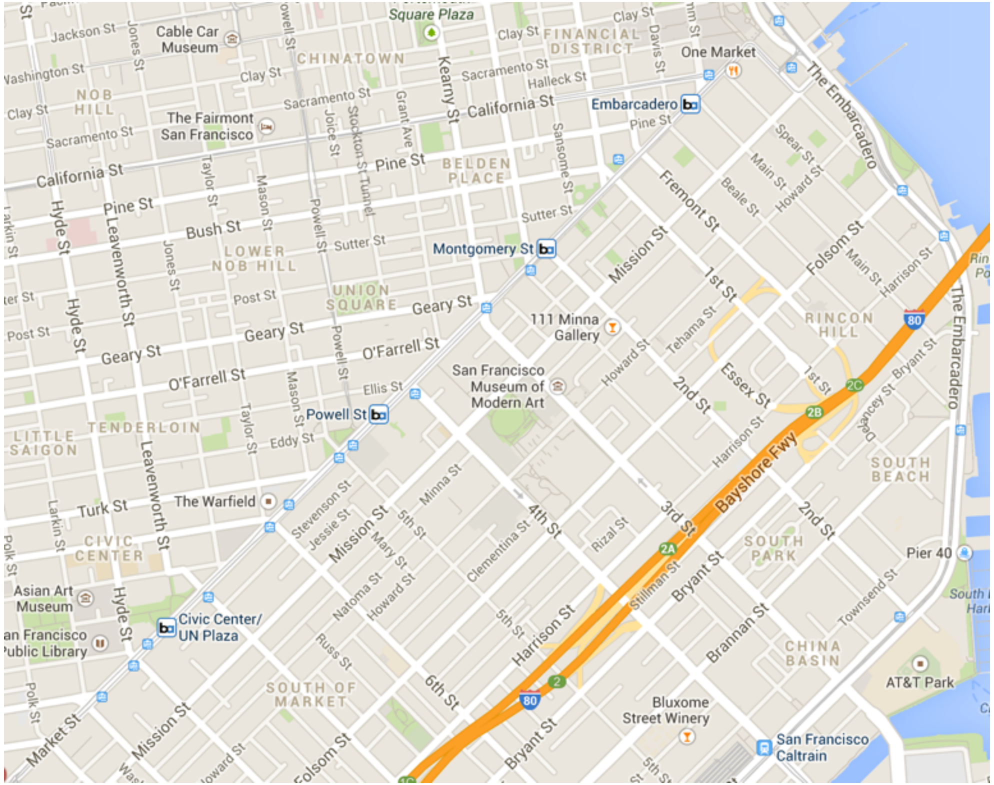
Q: What is the goal of dimensionality reduction?

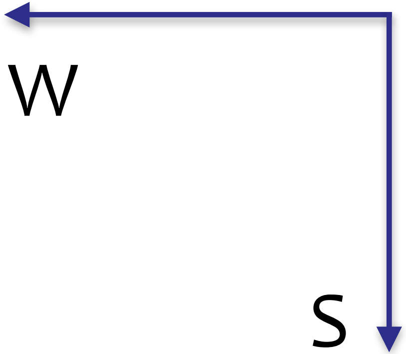
We'd like to analyze the data using the most meaningful basis (or coordinates) possible.

More precisely: given an n x d matrix A (encoding n observations of a d-dimensional random variable), we want to find a k-dimensional representation of A ($k < d$) that captures the information in the original data, according to some criterion.

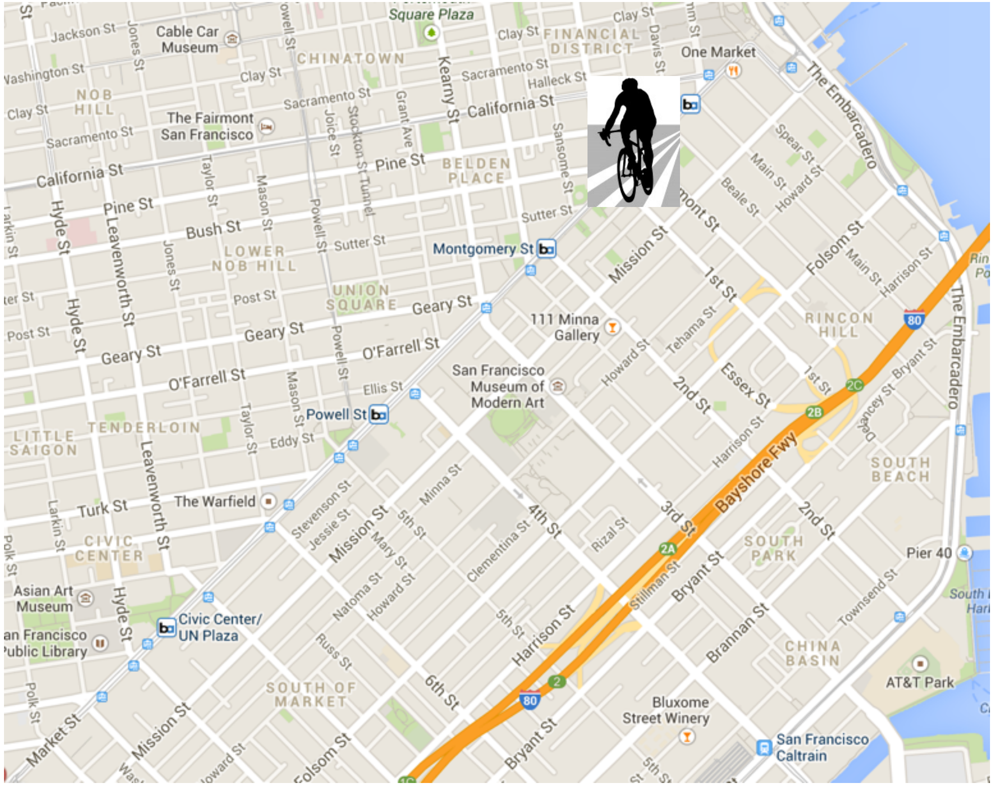
The goal of dimensionality reduction is to create a new set of coordinates that *simplify the representation* of the data.

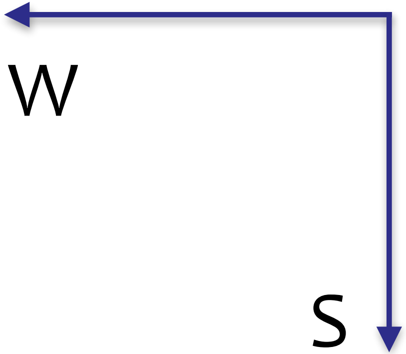
INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET



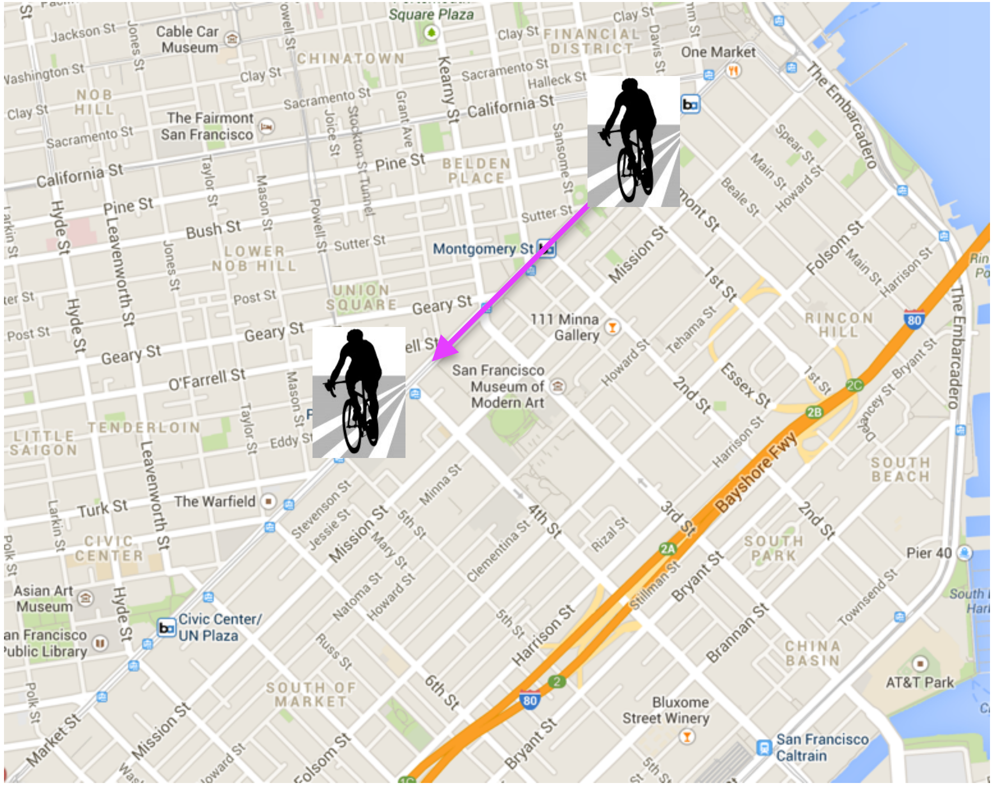


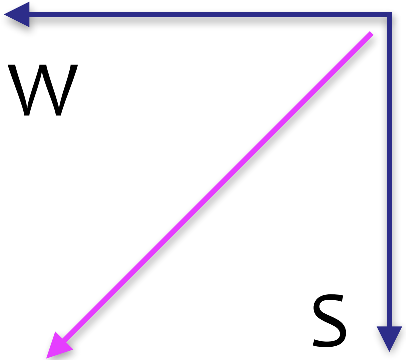
INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET



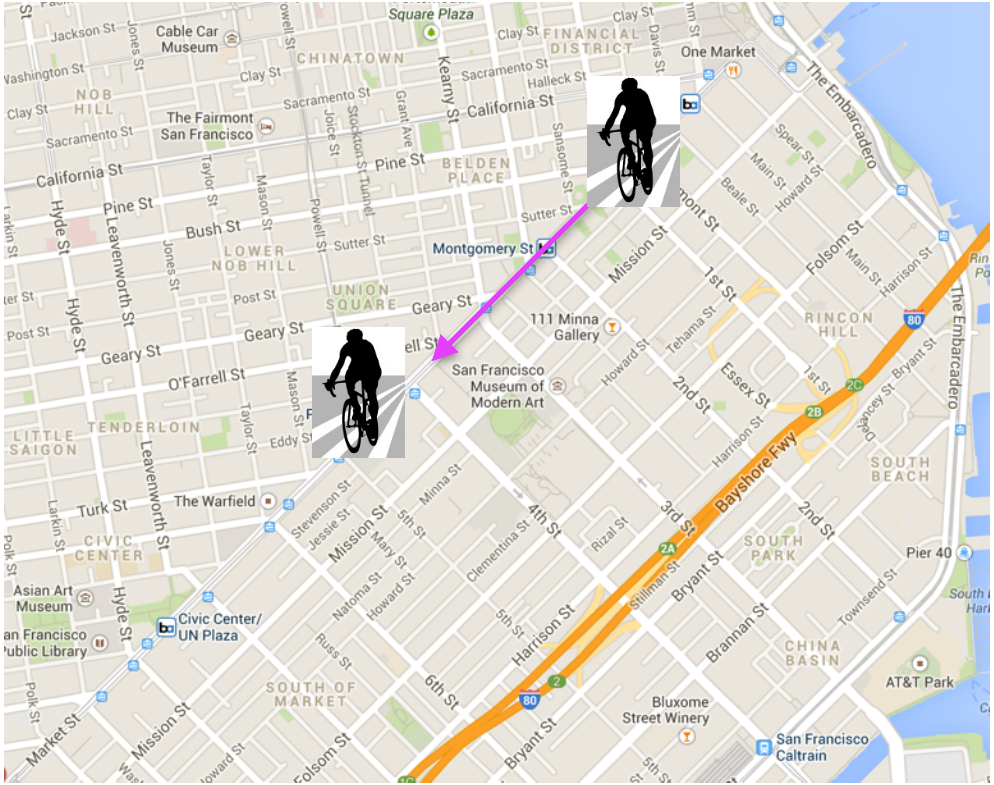


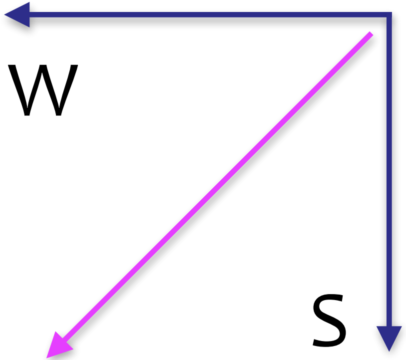
INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET





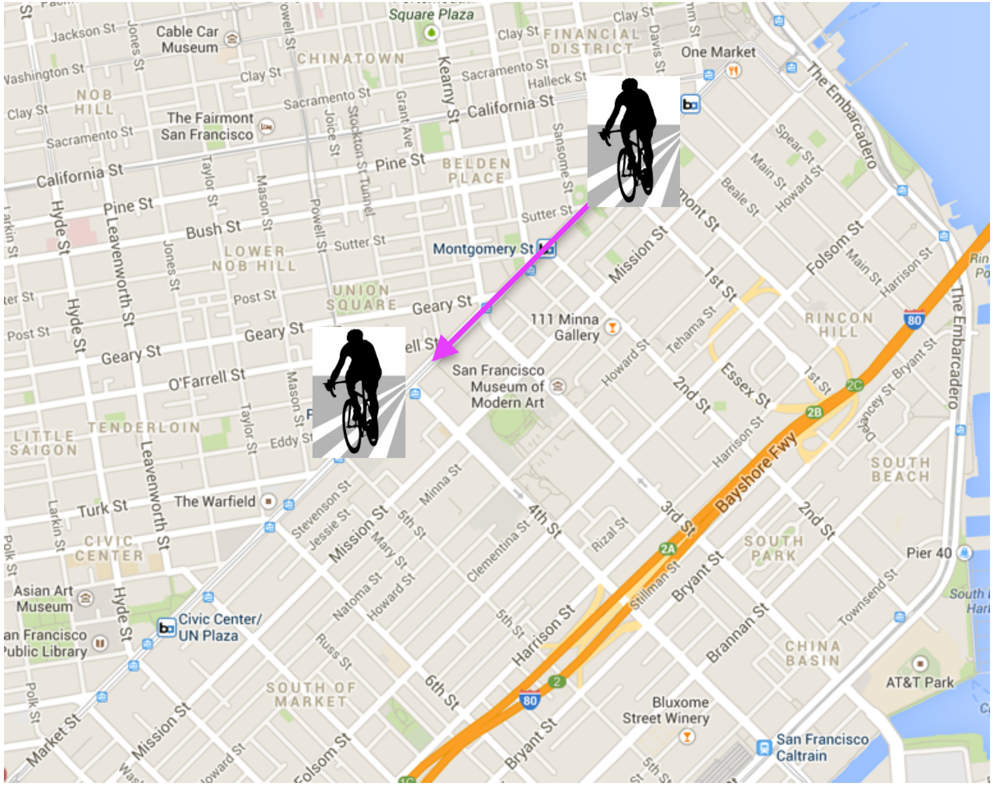
INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET

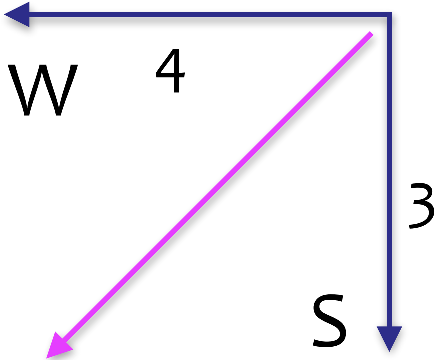




How many dimensions do we need to specify the position of this bike?

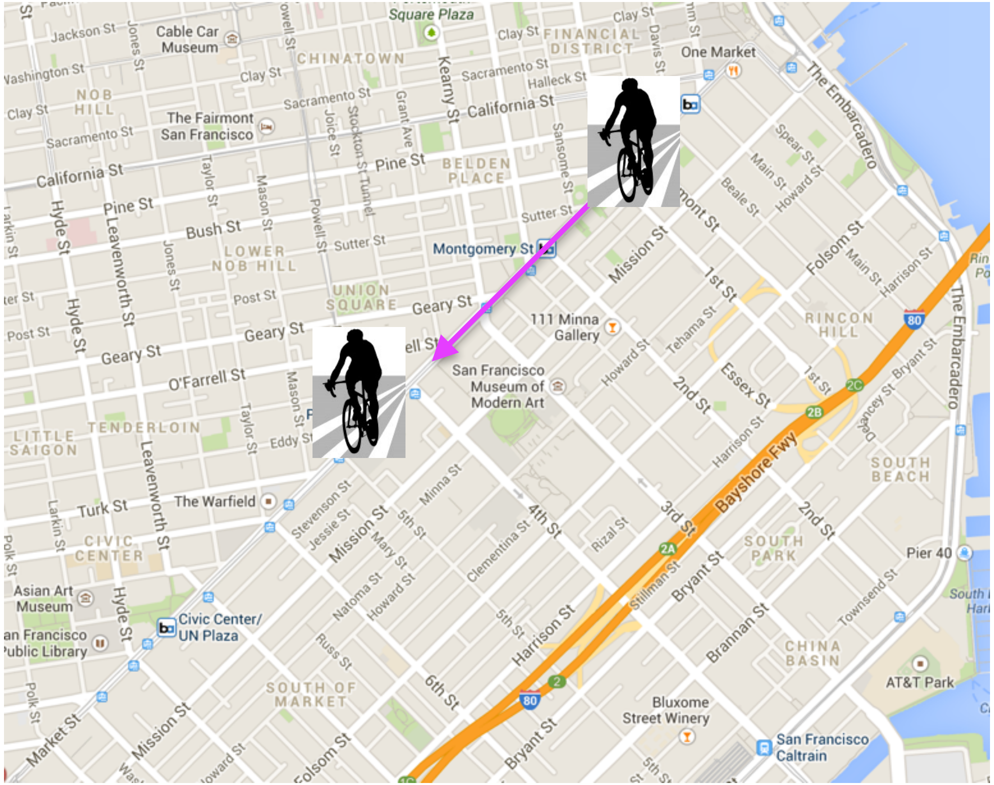
INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET

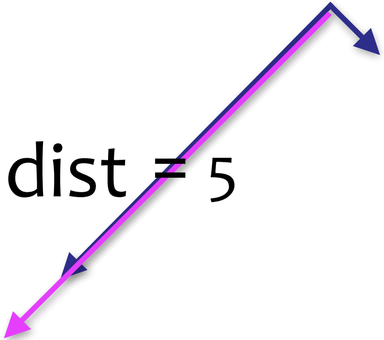




Yep, two. But could we represent the biker's position with fewer dimensions? How?

INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET

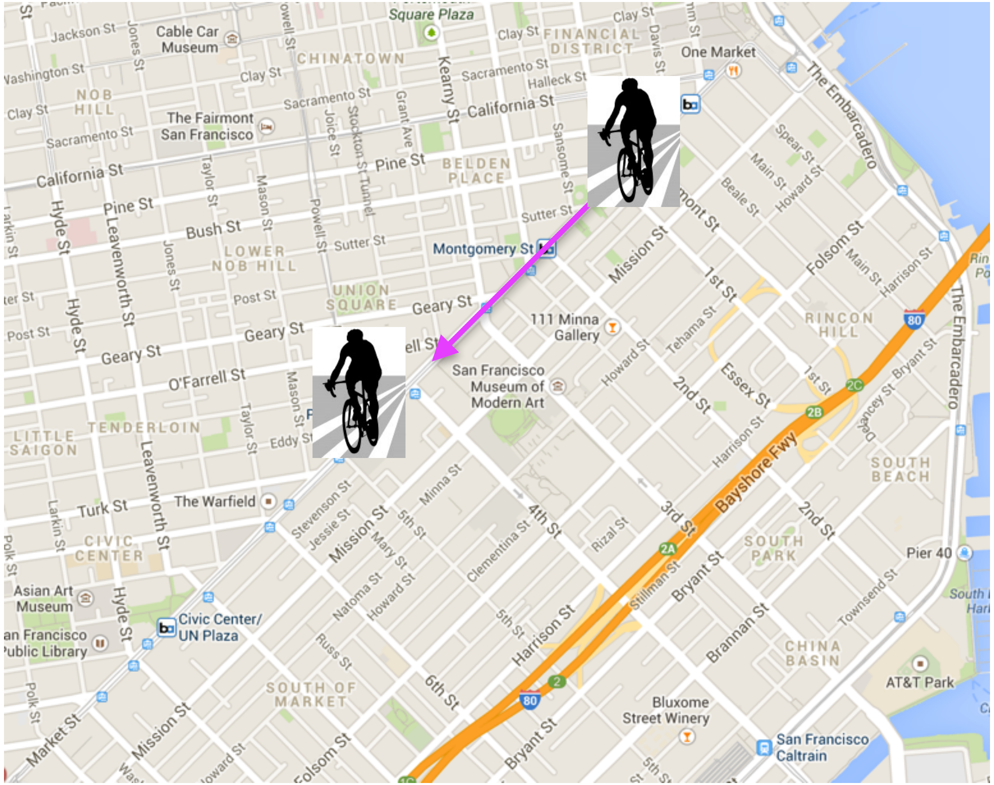


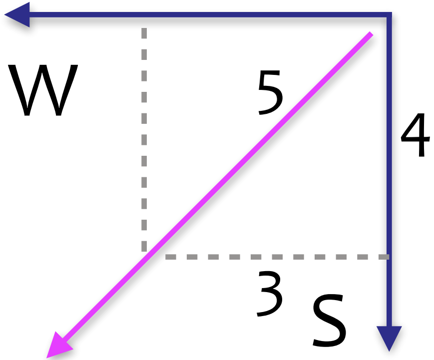


dist = 5

What if we just used distance down Market St.?

INTUITIVE EXAMPLE - BIKING DOWN MARKET STREET





Of course, we can always map back to the original coordinate system!

EXAMPLE: 1D HARMONIC OSCILLATOR

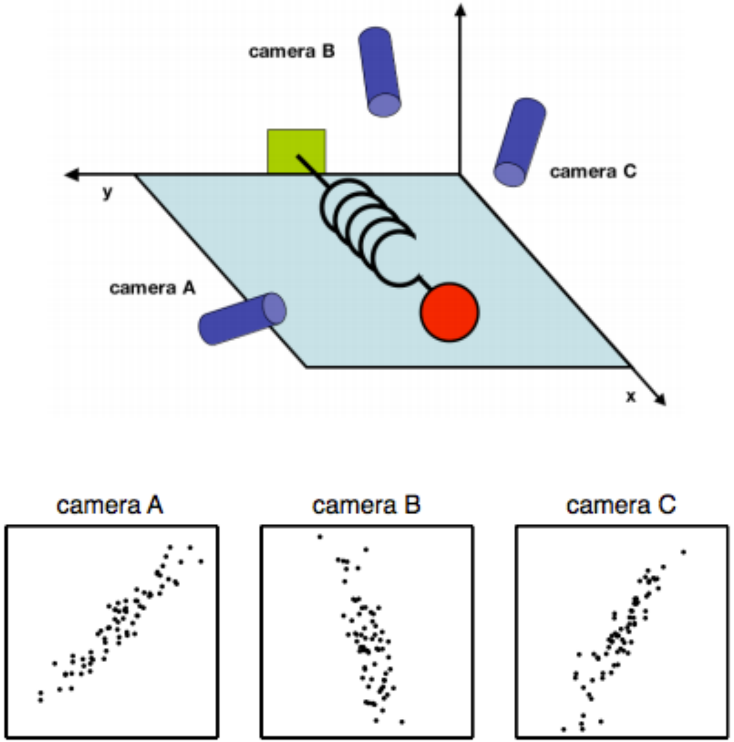
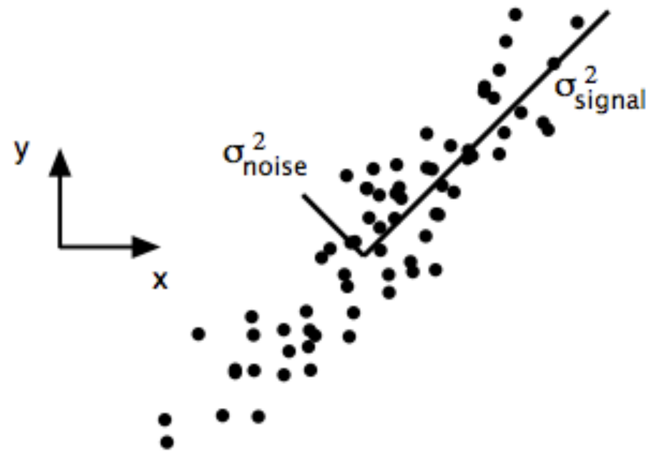


FIG. 1 A toy example. The position of a ball attached to an oscillating spring is recorded using three cameras A, B and C. The position of the ball tracked by each camera is depicted in each panel below.

source: http://www.snl.salk.edu/~shlens/pca.pdf



$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}.$$

FIG. 2 Simulated data of (x, y) for camera A. The signal and noise variances σ^2_{signal} and σ^2_{noise} are graphically represented by the two lines subtending the cloud of data. Note that the largest direction of variance does not lie along the basis of the recording (x_A, y_A) but rather along the best-fit line.

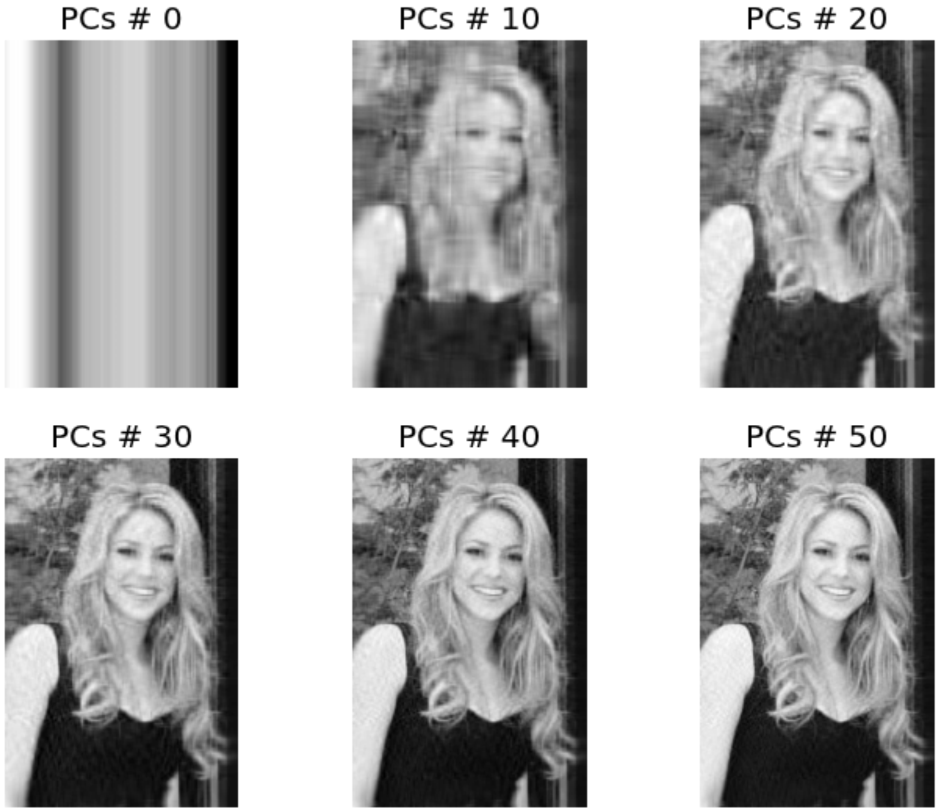
source: http://www.snl.salk.edu/~shlens/pca.pdf

Q: What are some applications of dimensionality reduction?

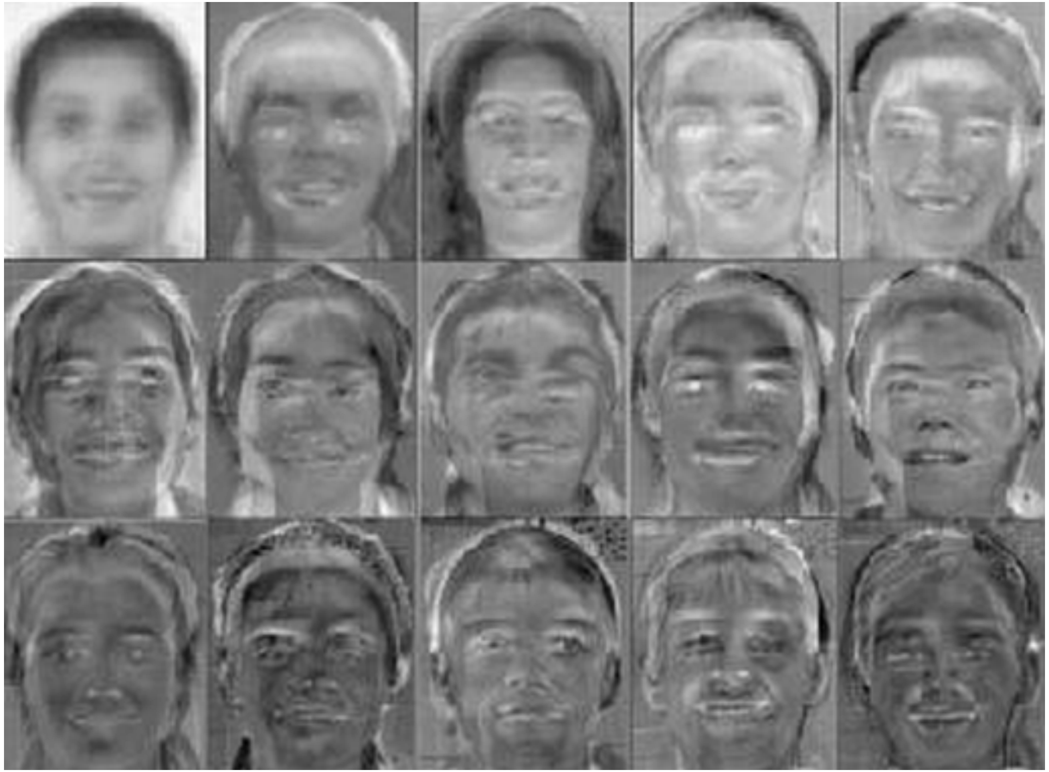
Q: What are some applications of dimensionality reduction?

- topic models (document clustering)
- image recognition/computer vision
- bioinformatics (microarray analysis)
- speech recognition
- astronomy (spectral data analysis)
- recommender systems

DIMENSIONALITY REDUCTION



source: http://glowingpython.blogspot.it/2011/07/pca-and-image-compression-with-numpy.html



II. PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

This procedure produces a new basis (a new coordinate system), each of whose components retain as much variance from the original data as possible.

Principal component analysis is a dimension reduction technique that can be used on a matrix of any dimensions.

This procedure produces a new basis, each of whose components retain as much variance from the original data as possible.

The PCA of a matrix A boils down to the eigenvalue decomposition of the covariance matrix of A.

The covariance matrix *C* of a matrix *A* is always square:

$$\mathrm{C} = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

off-diagonal elements C_{ij} give the *covariance* between X_i, X_j $(i \neq j)$

diagonal elements C_{ii} give the *variance* of X_i

Wait a minute, what's a covariance matrix?

$$\mathrm{C} = \begin{bmatrix} \mathrm{E}[(X_1-\mu_1)(X_1-\mu_1)] & \mathrm{E}[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_1-\mu_1)(X_n-\mu_n)] \\ \mathrm{E}[(X_2-\mu_2)(X_1-\mu_1)] & \mathrm{E}[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}[(X_n-\mu_n)(X_1-\mu_1)] & \mathrm{E}[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & \mathrm{E}[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

For that matter, what is covariance?

Remember variance?

Remember variance?

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}$$

Variance is the average distance from the mean of a data set to a point in that data set.

In other words, it is a measure of the *spread* of the data.

Recall that standard deviation is the square root of variance.

Standard deviation and variance only operate on 1 dimension, so that you could only calculate the standard deviation for each dimension of the data set *independently* of the other dimensions. However, it is useful to have a similar measure to find out how much the dimensions vary from the mean *with respect to each other*.

This is called covariance.

Variance:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} \qquad var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

Covariance:

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)}$$

Covariance is always measured between two dimensions. If you calculate the covariance between a dimension and itself, you get the variance.

The covariance matrix *C* of a matrix *A* is always square:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

off-diagonal elements C_{ij} give the *covariance* between X_i, X_j $(i \neq j)$

diagonal elements C_{ii} give the *variance* of X_i

The covariance matrix C of a matrix A is always square:

$$C = \begin{bmatrix} E[(X_1-\mu_1)(X_1-\mu_1)] & E[(X_1-\mu_1)(X_2-\mu_2)] & \cdots & E[(X_1-\mu_1)(X_n-\mu_n)] \\ E[(X_2-\mu_2)(X_1-\mu_1)] & E[(X_2-\mu_2)(X_2-\mu_2)] & \cdots & E[(X_2-\mu_2)(X_n-\mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ E[(X_n-\mu_n)(X_1-\mu_1)] & E[(X_n-\mu_n)(X_2-\mu_2)] & \cdots & E[(X_n-\mu_n)(X_n-\mu_n)] \end{bmatrix}.$$

off-diagonal elements C_{ij} give the *covariance* between X_i, X_j $(i \neq j)$

diagonal elements C_{ii} give the *variance* of X_i

The *eigenvalue decomposition* of a square matrix A is given by:

$$A = Q\Lambda Q^{-1}$$

The *eigenvalue decomposition* of a square matrix A is given by:

$$A = Q\Lambda Q^{-1}$$

The columns of Q are the eigenvectors of A, and the values in Λ are the associated eigenvalues of A.

The *eigenvalue decomposition* of a square matrix A is given by:

$$A = Q\Lambda Q^{-1}$$

The columns of Q are the eigenvectors of A, and the values in Λ are the associated eigenvalues of A.

For an eigenvector v of A and its eigenvalue λ, we have the important relation:

$$Av = \lambda v$$

The *eigenvalue decomposition* of a square matrix A is given by:

$$A = Q\Lambda Q^{-1}$$

The columns of Q are the eigenvectors of A, and the
are the associated eigenvalues of A.

NOTE
This relationship *defines* what it means to be an eigenvector of A.

For an eigenvector v of A and its eigenvalue λ, we have the important relation:

$$Av = \lambda v$$

The eigenvectors form a basis of the vector space on which A acts (e.g., they are orthogonal).

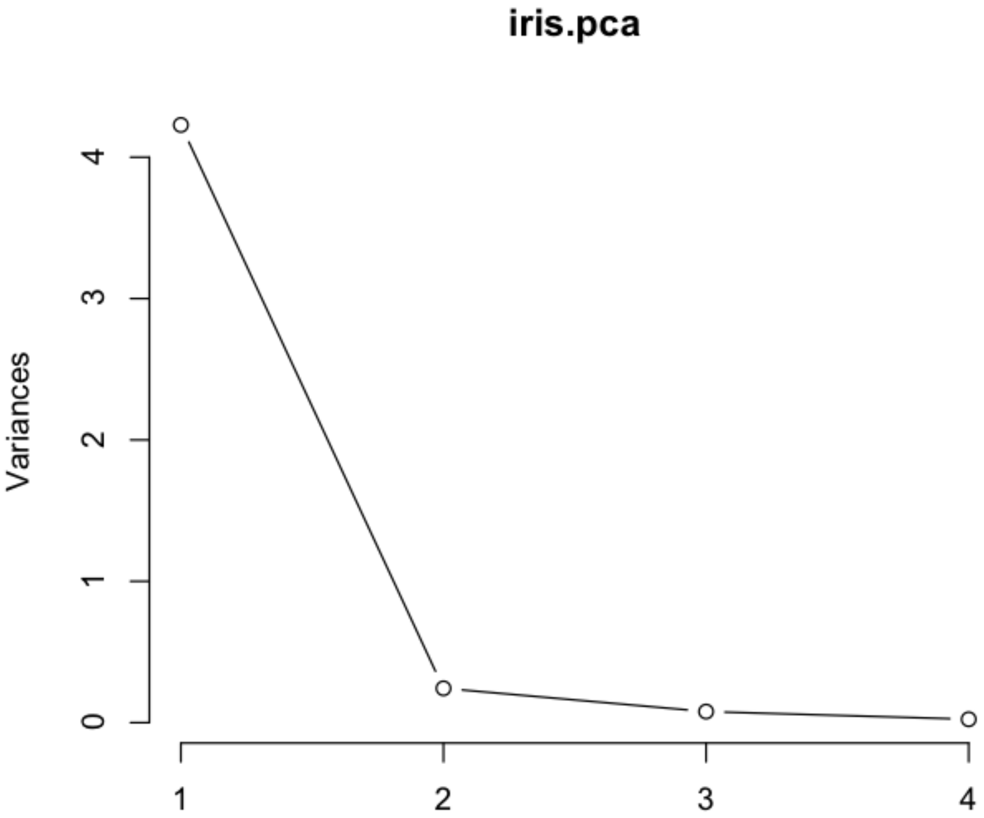
The eigenvectors form a basis of the vector space on which A acts (e.g., they are orthogonal).

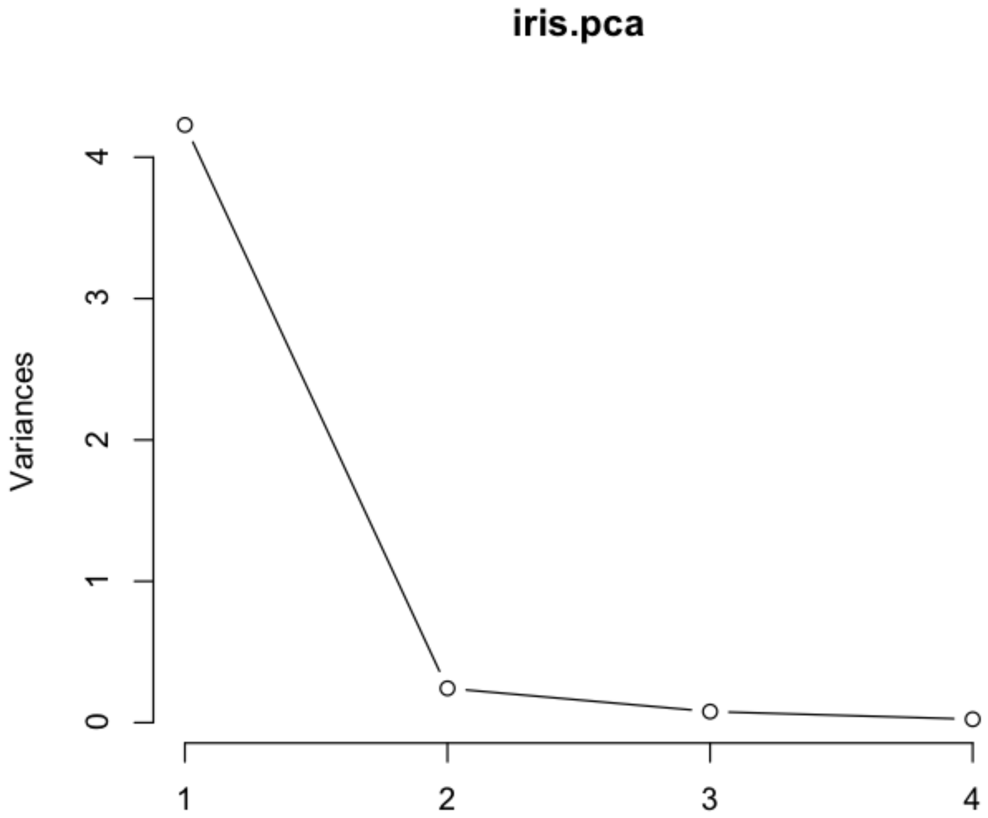
Furthermore the basis elements are ordered by their eigenvalues (from largest to smallest), and these eigenvalues represent the amount of variance explained by each basis element.

The eigenvectors form a basis of the vector space on which A acts (eg, they are orthogonal).

Furthermore the basis elements are ordered by their eigenvalues (from largest to smallest), and these eigenvalues represent the amount of variance explained by each basis element.

This can be visualized in a scree plot, which shows the amount of variance explained by each basis vector.


iris.pca
Variances
0
1
2
3
4
1
2
3
4



NOTE

Looking at this plot also gives you an idea of how many principal components to keep.

Apply the *elbow test*: keep only those pc's that appear to the left of the elbow in the graph.

1. **Linearity** – The change in basis is a linear projection

2. **Large variances have important structure** – e.g. large signal-to-noise ratio. In other words, we assume that principal components with larger associated variances are signal, while those with lower variances represent noise. NOTE: this is a strong (and not always correct) assumption!

3. **The principal components are orthogonal** – A simplification that makes PCA soluble with linear algebra matrix decomposition techniques

III. SINGULAR VALUE DECOMPOSITION

Notice: Lots of math / linear algebra notation ahead!

It's okay if it does not all immediately make sense.

Take a deep breath…

Notice: Lots of math / linear algebra notation ahead!

It's okay if it does not all immediately make sense.

Take a deep breath…

That's better! Okay, then…

Consider a matrix A with n rows and d features.

Consider a matrix A with n rows and d features.

The singular value decomposition of A is given by:

$$A = U \Sigma V^T$$

Consider a matrix *A* with *n* rows and *d* features.

The singular value decomposition of *A* is given by:

$$A = U \Sigma V^T$$

(n x d) (n x n) (n x d) (d x d)

Consider a matrix A with n rows and d features.

The singular value decomposition of A is given by:

$$\underset{(n \times d)}{A} = \underset{(n \times n)}{U} \; \underset{(n \times d)}{\Sigma} \; \underset{(d \times d)}{V^T}$$

st. U, V are orthogonal matrices and Σ is a diagonal matrix.

Consider a matrix A with n rows and d features.

The singular value decomposition of A is given by:

$$\underset{(n \times d)}{A} = \underset{(n \times n)}{U} \; \underset{(n \times d)}{\Sigma} \; \underset{(d \times d)}{V^T}$$

st. U, V are orthogonal matrices and Σ is a diagonal matrix.

→ $UU^T = I_n$, $VV^T = I_d$ → $\Sigma_{ij} = 0 \;\; (i \neq j)$

The singular value decomposition of *A* is given by:

$$\underset{(n \times d)}{A} = \underset{(n \times n)}{U} \; \underset{(n \times d)}{\Sigma} \; \underset{(d \times d)}{V^T}$$

The columns of *U* & *V* are the (left- and right-) singular vectors of *A*.

The singular value decomposition of *A* is given by:

$$\underset{(n \times d)}{A} = \underset{(n \times n)}{U} \; \underset{(n \times d)}{\Sigma} \; \underset{(d \times d)}{V^T}$$

The columns of *U* & *V* are the (left- and right-) singular vectors of *A*.

These singular vectors provide orthonormal bases for the spaces K_n & K_d (columns of *U* & *V*, respectively).

The singular value decomposition of *A* is given by:

$$A = U \Sigma V^T$$

$$\underset{(n \times d)}{A} \qquad \underset{(n \times n)}{U} \quad \underset{(n \times d)}{\Sigma} \quad \underset{(d \times d)}{V^T}$$

The nonzero entries of Σ are the singular values of *A*. These are real, nonnegative, and *rank-ordered* (decreasing from left to right).

Q: How do you interpret the SVD?

Q: How do you interpret the SVD?

A: Recall that given a set of n points in d-dimensional space (e.g., a matrix A), we want to find the best $k < d$ dimensional subspace to represent the data.

Q: How do you interpret the SVD?

A: Recall that given a set of n points in d-dimensional space (eg, a matrix A), we want to find the best $k < d$ dimensional subspace to represent the data.

For $k = 1$, this subspace is a line passing through the origin.

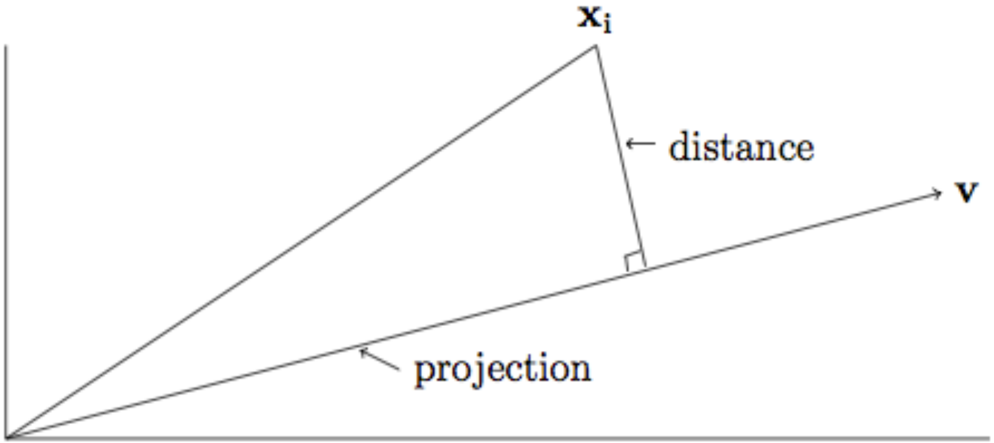


Figure 4.1: The projection of the point $\mathbf{x_i}$ onto the line through the origin in the direction of $\mathbf{v}$

source: http://www.cs.princeton.edu/courses/archive/spring12/cos598C/svdchapter.pdf

IV. EXERCISE: DIMENSIONALITY REDUCTION IN SKLEARN