Data mining & Machine Learning

CS 373
Purdue University

Dan Goldwasser
dgoldwas@purdue.edu

Today's Lecture

It's Complicated!

- *The data we get is often too complicated simply "eyeball"*
- It's too complex to visualize in a meaningful way
- *We don't understand it enough to start annotating*

How can make sense of complex data?

But first, some thoughts on distance/similarity

Distance

- If data objects have the same fixed set of numeric attributes, then the **data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute**
- Many data mining techniques then **use similarity/dissimilarity measures to characterize relationships between the instances**, e.g.,
- We talked about distances for continuous values (e.g., L2 distance) and for binary values (e.g., Hamming distance).

Distance

- We talked about distances for **continuous values** (e.g., *L2 distance*) and for **binary values** (e.g., *Hamming distance*).

L1 and L2 Distance:

$$d_M(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

$$d_E(x,y) = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

Hamming Distance:

$$\mathbf{n_{10}+n_{01}}$$

Recall:

$$n_{11} = \sum_{i}^{p} \mathbb{I}(x_i + y_i = 2)$$

How different are these two really?

Distance

1. Netflix popular shows and movies are currently available for streaming world wide.

2. Netflix provides a video streaming service, with popular shows such as OitB

3. Manchester united is a football team popular world wide

Which pair of sentences are the most similar?

Which distance measures would you use?

Does it identify the correct pair?

Distance

1. Netflix's popular shows and movies are currently available for streaming world wide.

2. Netflix video content, now available everywhere, is well-liked by many.

3. Manchester united is a football team popular world wide

How about now?

What went wrong?

Distance

How about now?

How would you fix this problem?

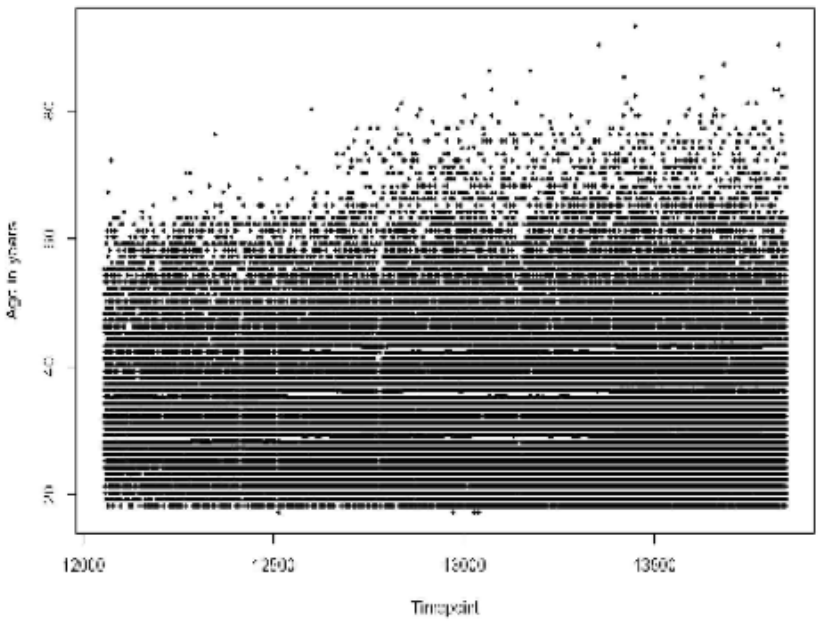
Today's Lecture

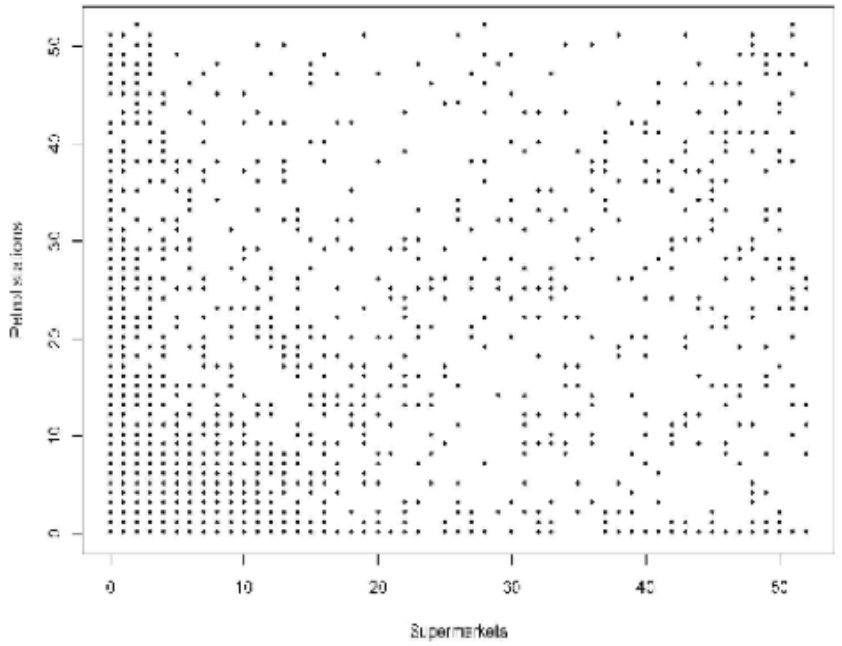
It's Complicated!

- *The data we get is often too complicated simply "eyeball"*
- It's too complex to visualize in a meaningful way
- *We don't understand it enough to start annotating*

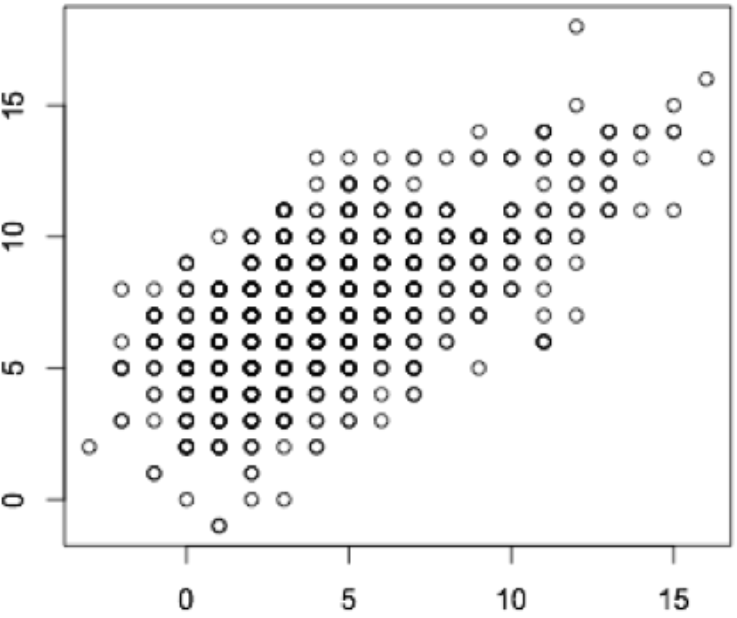
How can make sense of complex data?

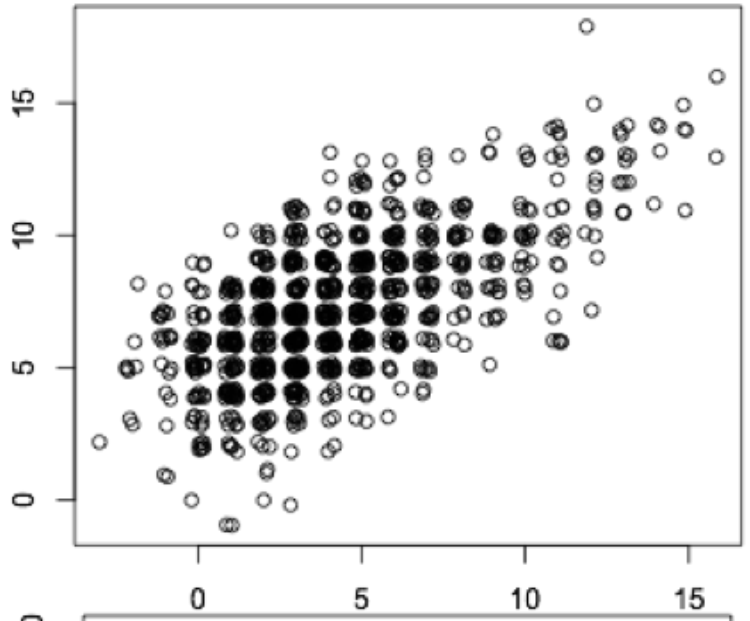
It’s complicated!

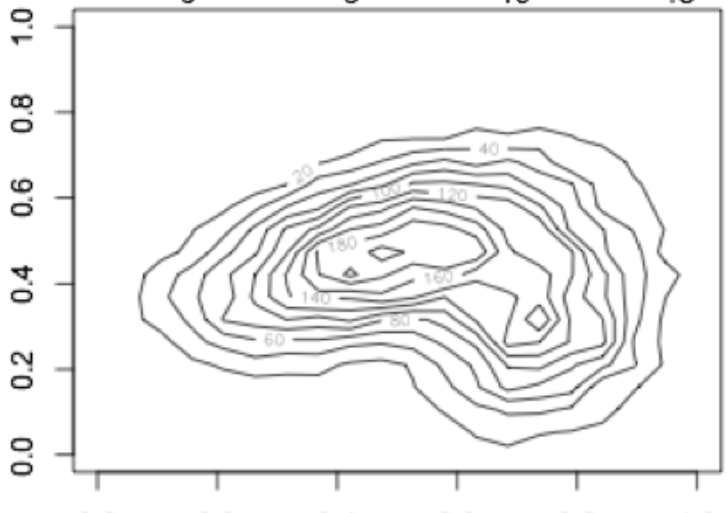




In some case we have easy solutions..







Unsupervised methods

- **In other cases we would like to use algorithms to make sense of the data.**
  - Can we simplify the data, such that preserve as much of the its properties?
  - Can we identify patterns in the data? Can we simplify the data based on these patterns?
    - Automatically identify that there are 3 groups in the data, and analyze them separately.
  - Can we learn meaningful representations, such that these patterns can be identified?
  - Can we identify outliers? Detect anomalous behavior?
- **In all these cases, we want to exploit the properties of the data, rather than relying on an external judgment (annotation)**

Dimensionality Reduction using PCA

Dimensionality reduction

- Identify and describe the “dimensions” that underlie the data
  - May be more fundamental than those directly measured but hidden to the user
- Reduce dimensionality of modeling problem
  - Benefit is simplification, it reduces the number of variables you have to deal with in modeling
- Can identify set of variables with similar behavior

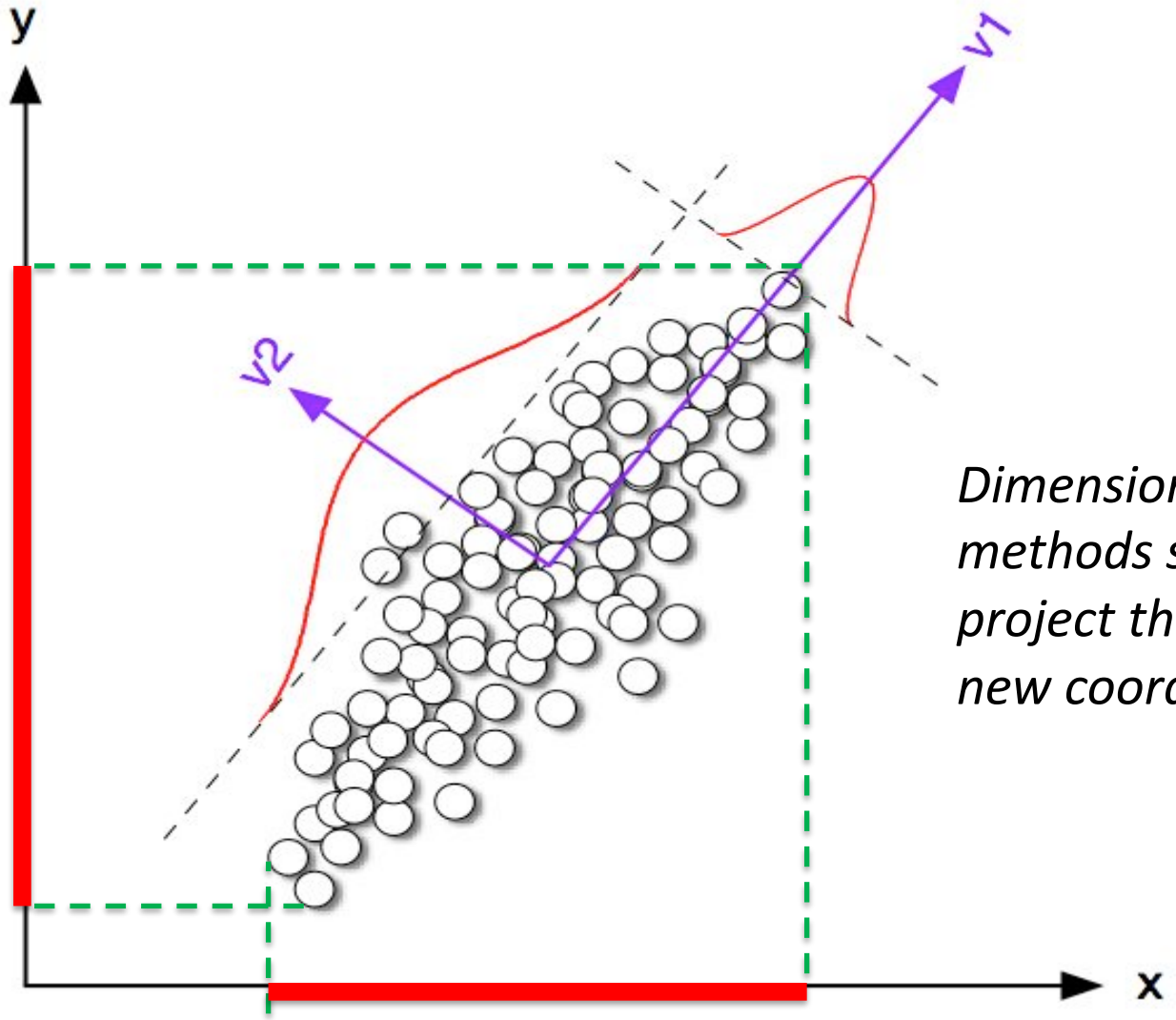
Dimensionality reduction methods

- ***Feature Selection***
  - Select a subset of the features (manual process!)

VS.

- **Principal component analysis (PCA)**
  - Linear transformation, minimize unexplained variance

Dimensionality Reduction Intuition

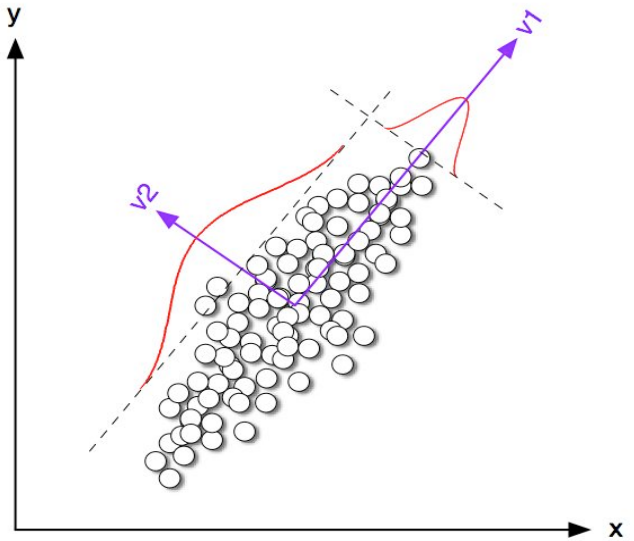


Dimensionality Reduction methods such as PCA project the data into a new coordinate system.

*Projecting the data points to the x or y axis is **feature selection***

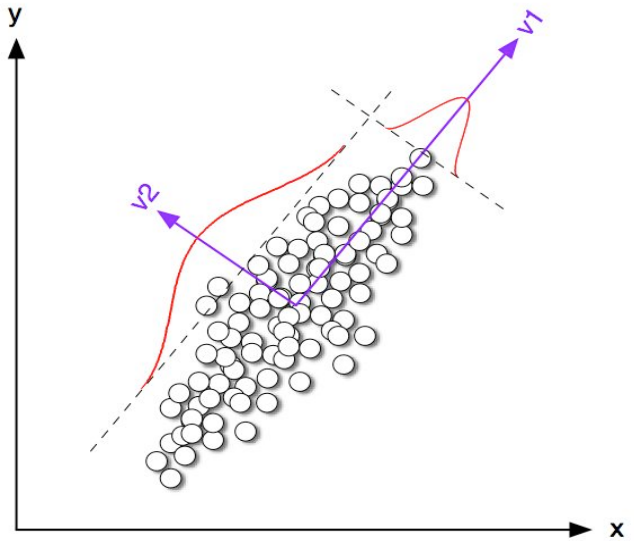
PCA Intuition

- High dimensional data can record many observations that are **highly correlated**
  - *E.g., Income and education, temperature and precipitation*
  - These variables capture the "same information"
- **PCA**: *a transformation to new variables, that are linearly uncorrelated*
- Each variable:
  - Captures as much of the variability in the data
  - Orthogonal to the other variables



PCA Intuition

- If you maintain all the principle components, this is a linear transformation of the data, so you can reconstruct the original dataset.
- Choosing a subset of principle components is similar to lossy compression, reducing the dimensionality of the data while maintain as much of its variance.



Principal component analysis (PCA)

- Given data matrix **D** with **p** dimensions:
  - Preprocess **D** so that the mean of each attribute is 0, call this matrix **X**
  - Compute pxp covariance matrix: $\Sigma = X^T X$
  - Compute eigenvectors/eigenvalues of covariance matrix:

$$\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$$

$$(\Sigma - \lambda\mathbf{I})\mathbf{a} = 0$$

$\mathbf{A}$: matrix of eigenvectors

Λ : diagonal matrix of eigenvalues

$\mathbf{a}$: 1st principal component, eigenvector assoc. with largest eigenvalue (λ)

  - Eigenvectors $\mathbf{A}$ are the **principal component** vectors, where each $\mathbf{a}$ is a px1 column vector of projection weights

Framework for learning methods

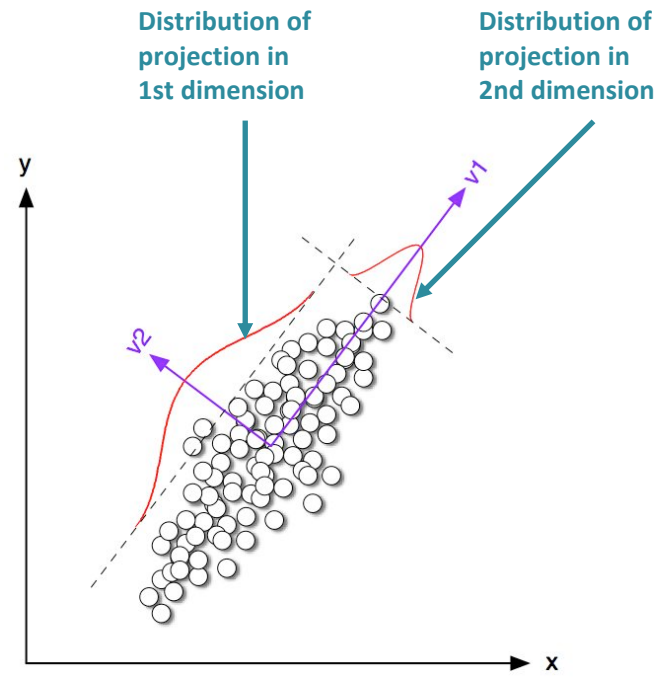
- **Model space**
  - Choice of knowledge representation defines a set of possible models or patterns
- **Scoring function**
  - Associates a numerical value (score) with each member of the set of models/patterns
- **Search technique**
  - Defines a method for generating members of the set of models/patterns and determining their score

What is the *model space* for PCA?

$$\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$$

$\mathbf{A}$: matrix of eigenvectors
Λ : diagonal matrix of eigenvalues

- **A** is a *p x p* matrix of principal components (if data is p-dimensional) each column is a basis vector, each cell is a projection weight
- **Model space**: is defined by **A**
  - Method needs to choose the p^2 weights that populate **A**, i.e., set of p basis vectors
  - **Constraints**: Basis vectors must be **orthonormal**, i.e., each has a norm of 1 and any pair of basis vectors have dot-product of 0
  - E.g., any orthogonal set of v_1 and v_2



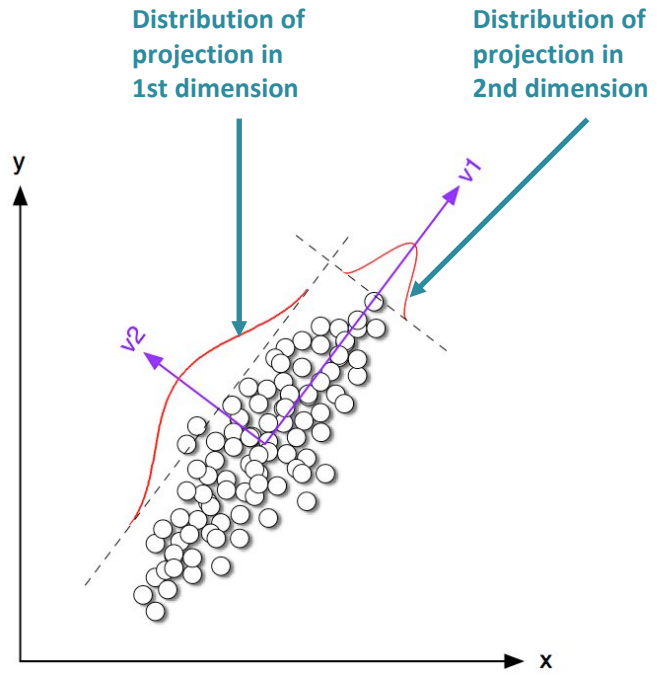
What is the score function for PCA?

$$\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$$

$\mathbf{A}$: matrix of eigenvectors
Λ : diagonal matrix of eigenvalues

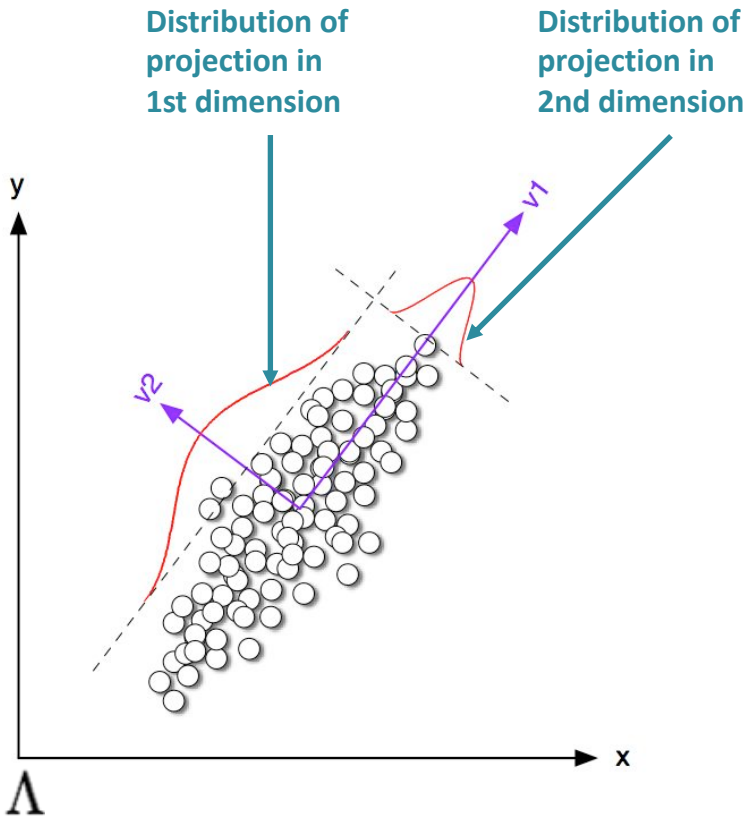
- **Λ** is diagonal matrix of *p* eigenvalues
  - Each eigenvalue λ_i corresponds to the variance of dimension **i**
- **Score function**: sum of eigenvalues in **Λ** $\sum_{j=1}^{p} \lambda_j$
- Sum of eigenvalues is equal to the sum of the variances of the original attributes:

$$\sum_{j=1}^{p} \sigma_j^2 = \sum_{j=1}^{p} \lambda_j$$



What is the search method for PCA?

- **Goal:** find basis vectors that maximize variance
  - 1st basis (eg. v_1) ***maximizes*** variance of projected data
  - 2nd basis (eg. v_2) again ***maximizes*** variance of projected data, but has to be orthogonal to previous bases, …
  - New dimensions are orthogonal, thus transformed features have 0 covariance
- **Search**: Solving eigensystem corresponds to finding the $\mathbf{A}\Sigma\mathbf{A}^{-1} = \Lambda$ orthonormal basis that **maximize** variance of projected data



Applying PCA

- Choose number of target dimensions (i.e., select m < p)
  - Transform data vectors by projecting them onto the first *m* principal components, which correspond to top m eigenvectors)

$$\mathbf{x} = [x_1, x_2, \ldots, x_p] \text{ (original instance)}$$
$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p] \text{ (principal components)}$$

$$x'_1 = \mathbf{a}_1\mathbf{x} = \sum_{j=1}^{p} a_{1j}x_j$$

Linear transformation from the original coordinates to the first PC

$$\ldots$$

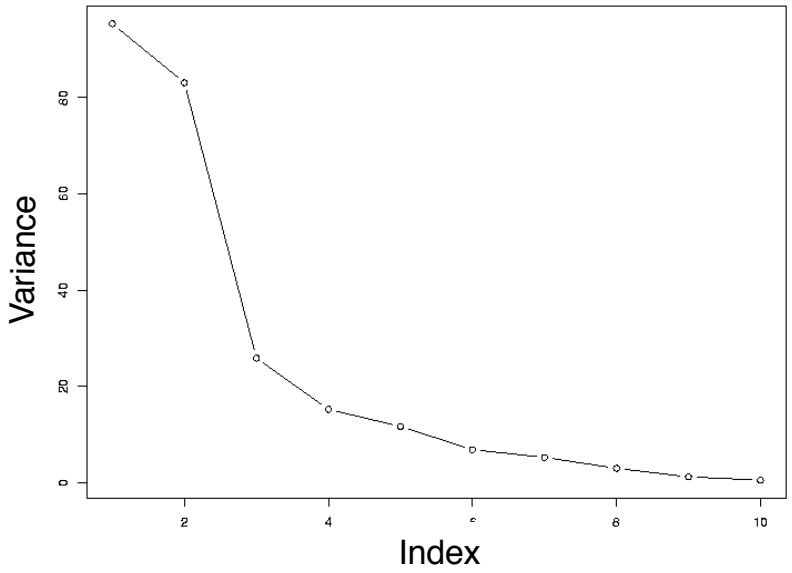
$$x'_m = \mathbf{a}_m\mathbf{x} = \sum_{j=1}^{p} a_{mj}x_j \quad for\ m < p$$

If ***m=p*** then data is transformed
If ***m<p*** then transformation is lossy and dimensionality is reduced

$$\mathbf{x}' = [x'_1, x'_2, \ldots, x'_m] \text{ (transformed instance)}$$

Applying PCA (cont')

- **Goal**: *Find a new (smaller) set of dimensions that captures most of the variability of the data*
- Can use **scree plot** to choose number of dimensions
  - Choose $m < p$ so projected data captures much of the variance of original data

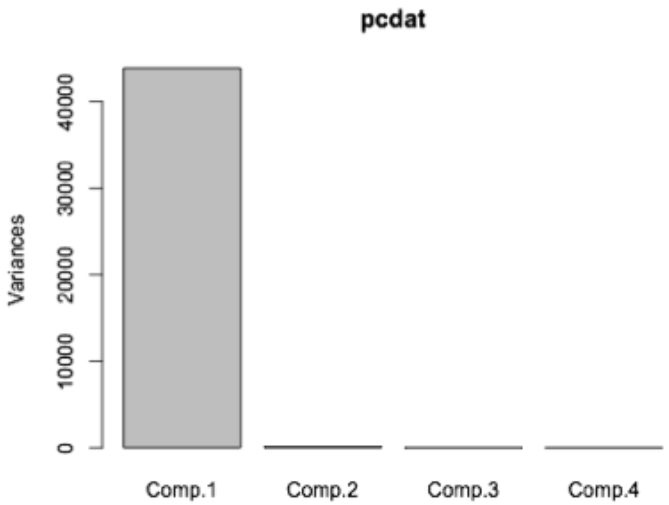


PCA example on Iris data

Choose m=1

```
> x <- scale(as.matrix(d[,1:4]),scale=FALSE)
> sigma <- t(x)%*% x
> sigma
           V1       V2       V3        V4
V1 102.16833  -5.8510 189.7787  77.01867
V2  -5.85100  28.0126 -47.9352 -17.57920
V3 189.77867 -47.9352 463.8637 193.16173
V4  77.01867 -17.5792 193.1617  86.77973

> pcdat <- princomp(d[,1:4])
> summary(pcdat)
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0485788 0.49053911 0.27928554 0.153379074
Proportion of Variance 0.9246162 0.05301557 0.01718514 0.005183085
Cumulative Proportion  0.9246162 0.97763178 0.99481691 1.000000000
> plot(pcdat)
> loadings(pcdat)

Loadings:
   Comp.1 Comp.2 Comp.3 Comp.4
V1  0.362 -0.657 -0.581  0.317
V2        -0.730  0.596 -0.324
V3  0.857  0.176        -0.480
V4  0.359         0.549  0.751

               Comp.1 Comp.2 Comp.3 Comp.4
SS loadings      1.00   1.00   1.00   1.00
Proportion Var   0.25   0.25   0.25   0.25
Cumulative Var   0.25   0.50   0.75   1.00
```

First component explains 92% of data variance



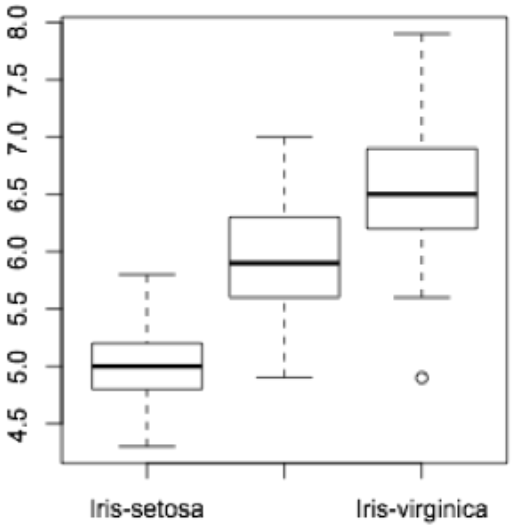
PCA example on Iris data

m=1, transform data to one dimension

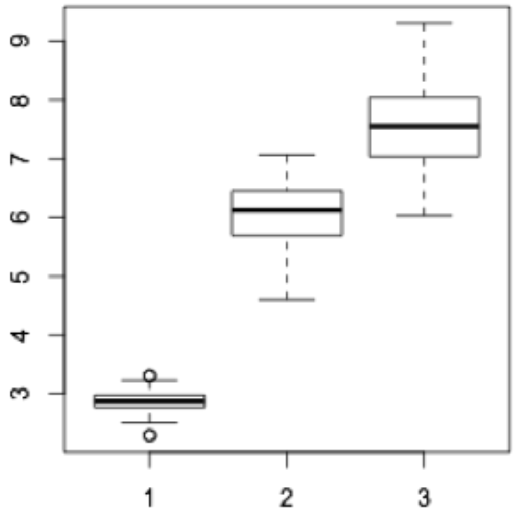
$$\mathbf{x} = [x_1, x_2, \ldots, x_p] \quad \text{(original instance)}$$
$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p] \quad \text{(principal components)}$$
$$x'_1 = \mathbf{a}_1\mathbf{x} = \sum_{j=1}^{p} a_{1j}x_j$$

```
### Single example
> pcdat$loadings[,1]
         V1          V2          V3          V4
 0.36158968 -0.08226889  0.85657211  0.35884393

> d[1,1:4]
   V1  V2  V3  V4
1 5.1 3.5 1.4 0.2

> as.matrix(d[1,1:4]) %*% pcdat$loadings[,1]
      [,1]
1 2.827136

### All data
> as.matrix(d[,1:4]) %*% pcdat$loadings[,1]
```

Original data (1st variable)



Transformed data



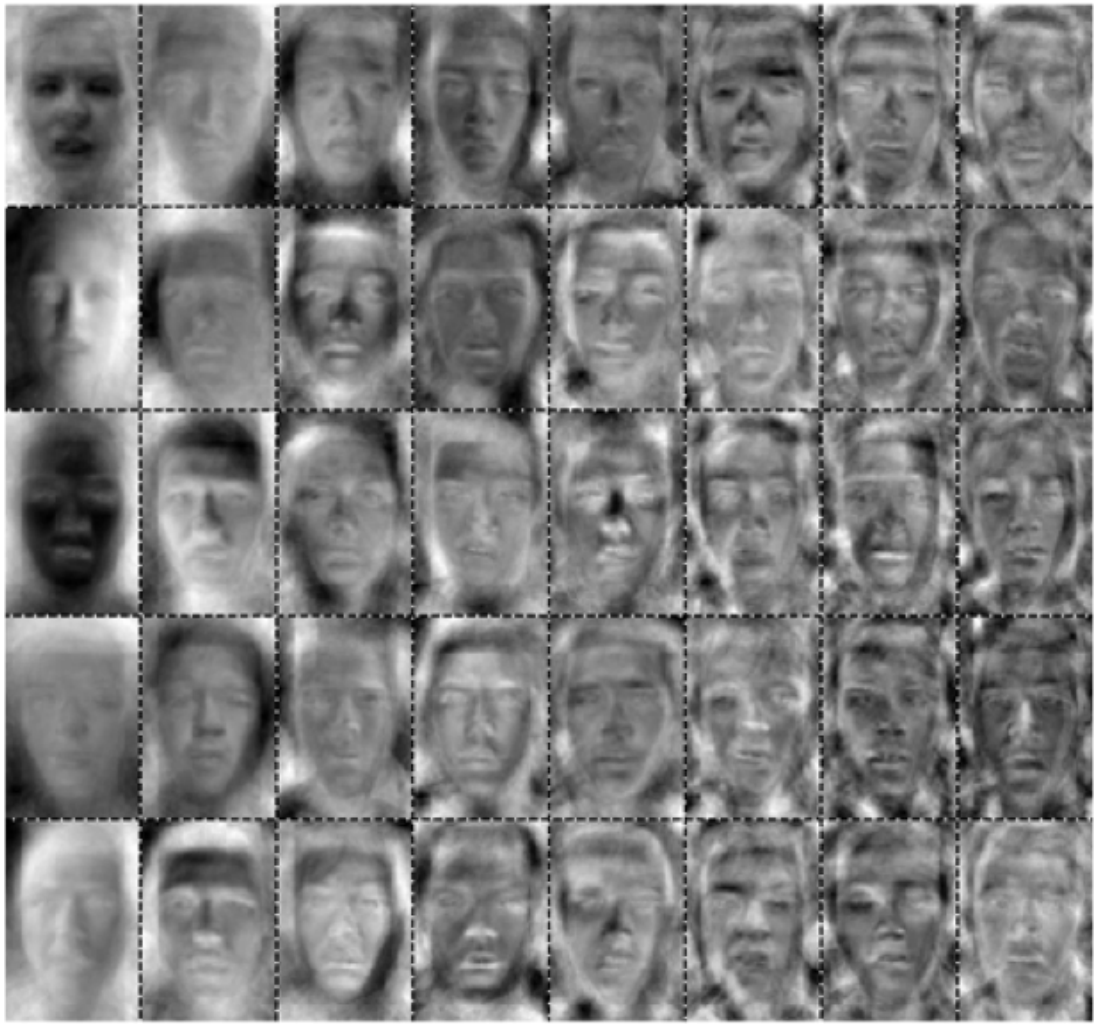
Example: Eigenfaces

PCA applied to images of human faces.

Reduce dimensionality to set of basis images.

All other images are linear combo of these “eigenpictures”.

Used for facial recognition.

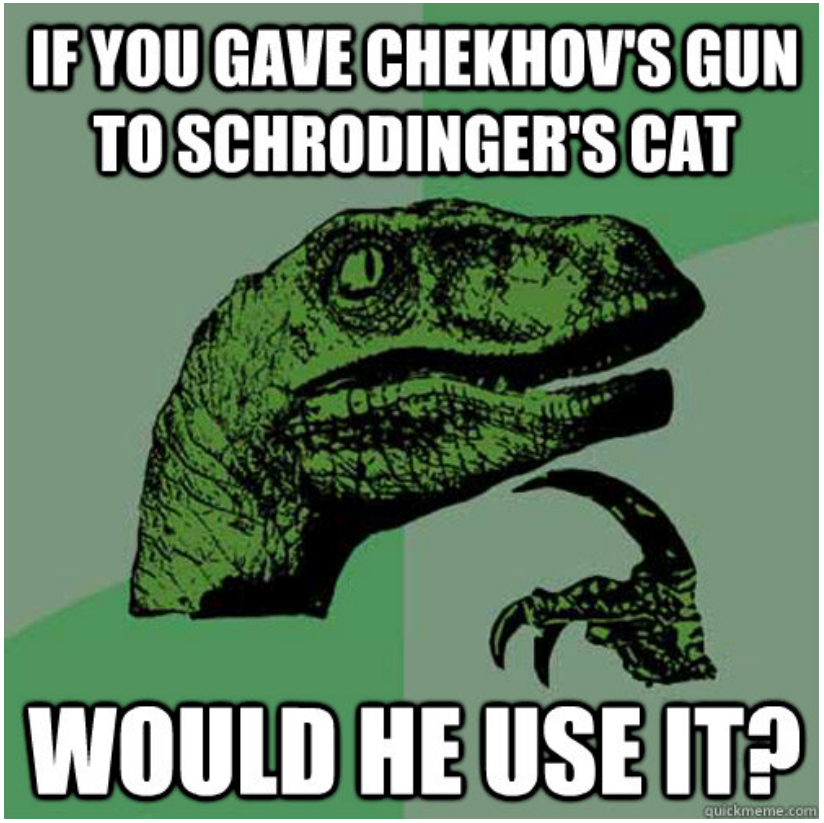


First 40 PCA dimensions

PCA Summary

- Automatic method for dimensionality reduction
  - Linear transformation from the original coordinates, such that the variance of the data is maximized
  - The Principal components are decorrelated and used as the new feature representation
- Can be viewed as an unsupervised learning technique.
  - Define a model space, scoring function, search procedure that maximizes the scoring function.
- One of the oldest and most popular methods
  - Key limitation – **Linear transformation**
  - We will revisit these methods when discussing DL

Back to Distances..



Back to Distances..

- See quiz on Piazza

1. Netflix's popular shows and movies are currently available for streaming world wide.

2. Netflix video content, now available everywhere, is well-liked by many.

3. Manchester united is a football team popular world wide