

# Machine Learning

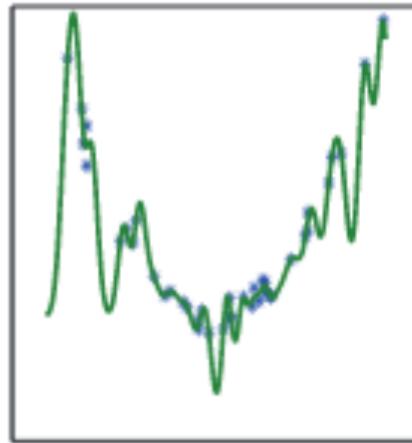
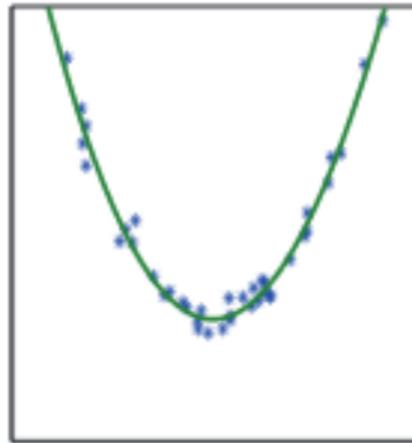
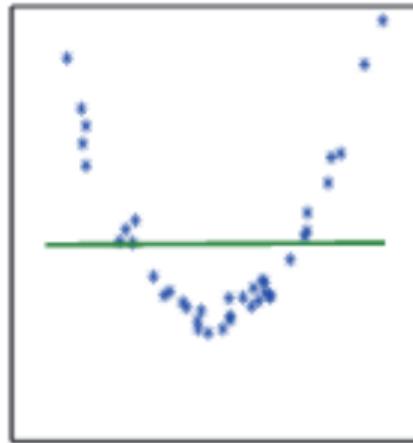
## Faces, Clustering, Anomalies

Jeff Abrahamson

octobre 2020

# Review

# Underfitting, overfitting



# Linear Regression

# Logistic Regression

# **SVM**

# CART

# **Random Forests**

# PCA

# PCA

# Motivation

?

# Motivation

**Remember the Curse of Dimensionality?**

# Principle

- Linear transformations have axes
- Find them (eigenvectors of the covariance matrix)
- Pick the biggest ones

# Principle

- Linear transformations have axes
- Find them (eigenvectors of the covariance matrix)
- Pick the biggest ones

Fitting an  $n$ -dimensional ellipsoid to the data

# Uses

- Exploratory data analysis
- Compression
- Visualisation

# Face Recognition

# Eigenfaces

- Sirovich and Kirby (1987)
- Turk and Pentland (1991)

*Turk, Matthew A and Pentland, Alex P. Face recognition using eigenfaces. Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on 1991.*

# Eigenfaces

Want: a low-dimensional representation of a face

Plan: cluster simplified faces

# Eigenfaces

Viewed as compression:

- Use PCA on face images to form a set of basis features
- Use eigenpictures to reconstruct original faces

# Eigenfaces



# Eigenfaces algorithm

Let  $X = \{x_1, x_2, \dots, x_n\}$  be a random vector with observations  $x_i \in \mathbb{R}^d$ .

Compute

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

*OpenCV*

# Eigenfaces algorithm

Compute the covariance matrix  $S$ :

$$\begin{aligned} S_{i,j} &= \mathbf{Cov}(x_i, x_j) \\ &= \mathbf{E}[(x_i - \mu_i)(x_j - \mu_j)^T] \end{aligned}$$

$$S = (S_{i,j})$$

# Eigenfaces algorithm

Compute the eigenvectors of  $S$ :

$$Sv_i = \lambda_i v_i \quad i = 1, 2, \dots, n$$

Sort the eigenvectors in decreasing order.

We want the  $k$  principal components, so take the first  $k$ .

# Eigenfaces algorithm

Compute the eigenvectors of  $S$ :

$$Sv_i = \lambda_i v_i \quad i = 1, 2, \dots, n$$

Sort the eigenvectors in decreasing order.

We want the  $k$  principal components, so take the first  $k$ .

This is PCA.

# Eigenfaces algorithm

The  $k$  principal components of the observed vector  $x$  are then given by

$$y = W^T(x - \mu)$$

where

$$W = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_k \\ | & | & & | \end{bmatrix}$$

# Eigenfaces algorithm

The reconstruction from the PCA basis is then

$$x = Wy + \mu$$

# Eigenfaces algorithm

So the plan is this:

- Project all training samples in the PCA subspace
- Project the query into the PCA subspace
- Find the nearest neighbour to the projected query image among the projected training images

# Eigenfaces algorithm



# Eigenfaces algorithm

Some advantages:

- Easy, relatively inexpensive
- Recognition cheaper than preprocessing
- Reasonably large database possible

# Eigenfaces algorithm

Some problems:

- Need controlled environment
- Needs straight-on view
- Sensitive to expression changes
- If lots of variance is external (e.g., lighting)...

# Handwriting Recognition

# Introduction to Handwriting Recognition

## Choices

- Online
- Offline

# Introduction to Handwriting Recognition

## Choices

- Get path information
- Get time data
- Get pressure information
- Only get image

# Introduction to Handwriting Recognition

## Major techniques

- Clustering (not great performance)
- SVM (until 2006 or so)
- Convolutional neural networks



questions?

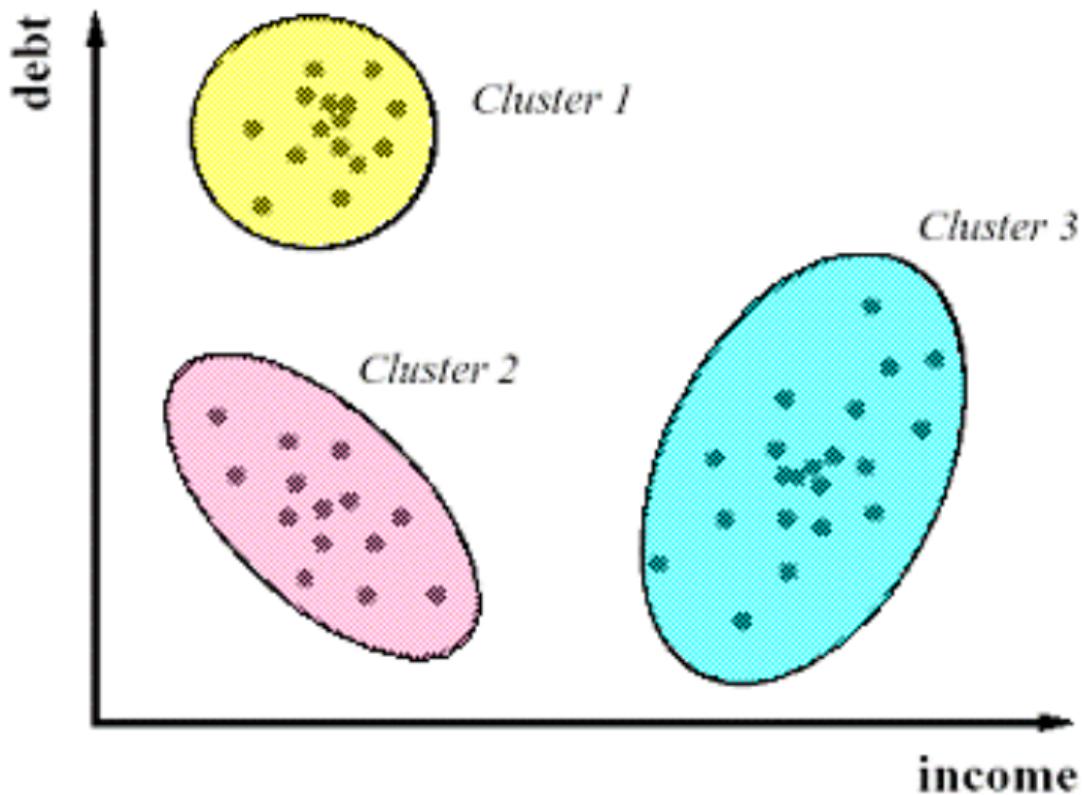
# Clustering

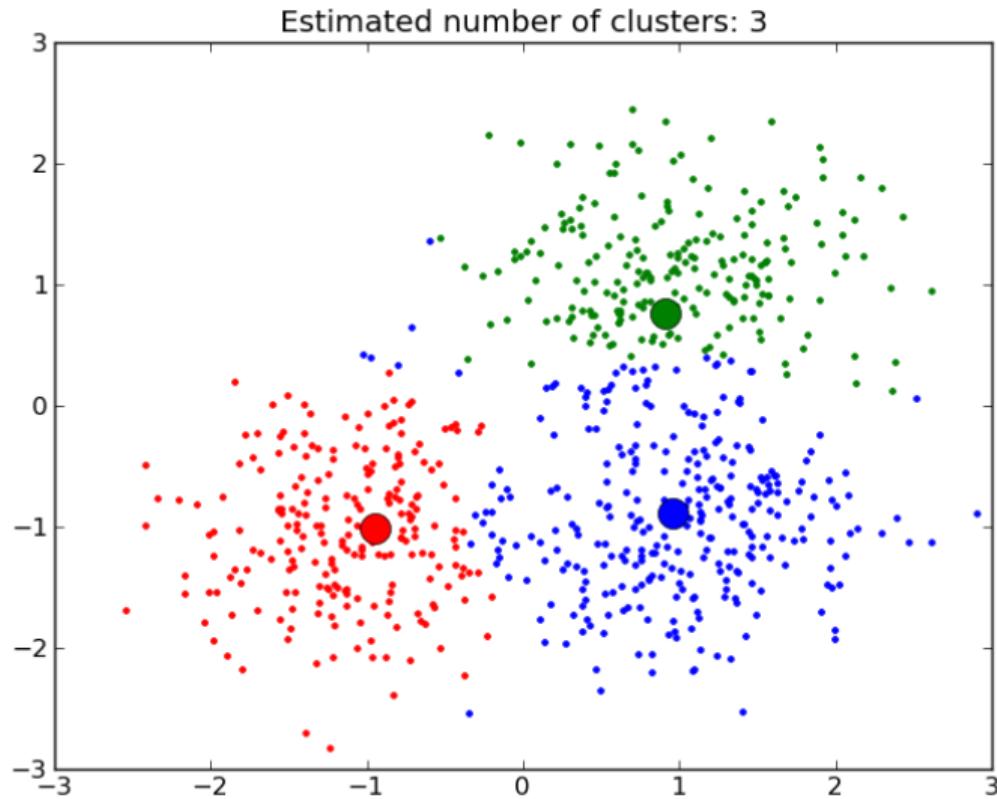
# The Problem

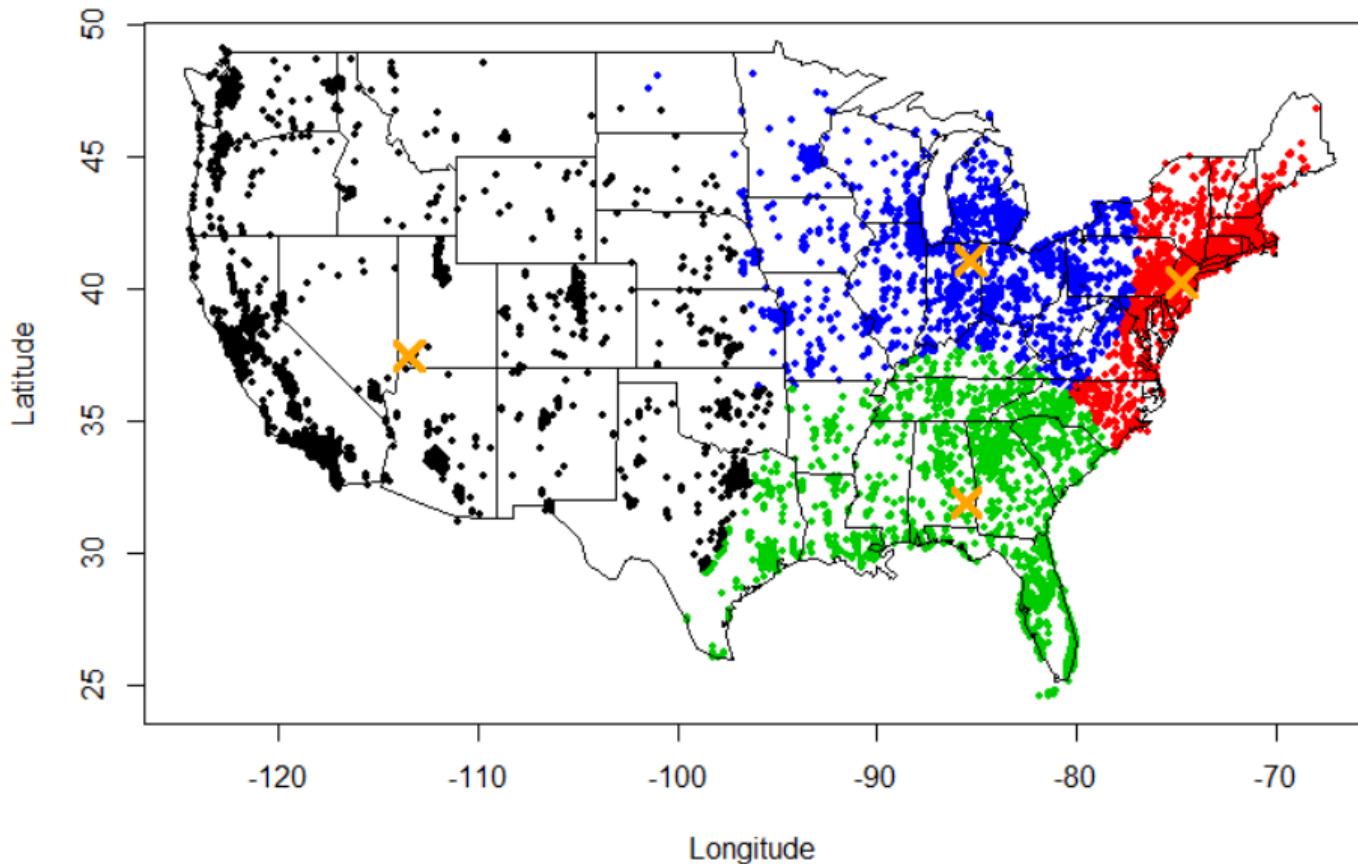
Have points  $d = \{d_1, \dots, d_n\}$ .

Have number of clusters  $k$ .

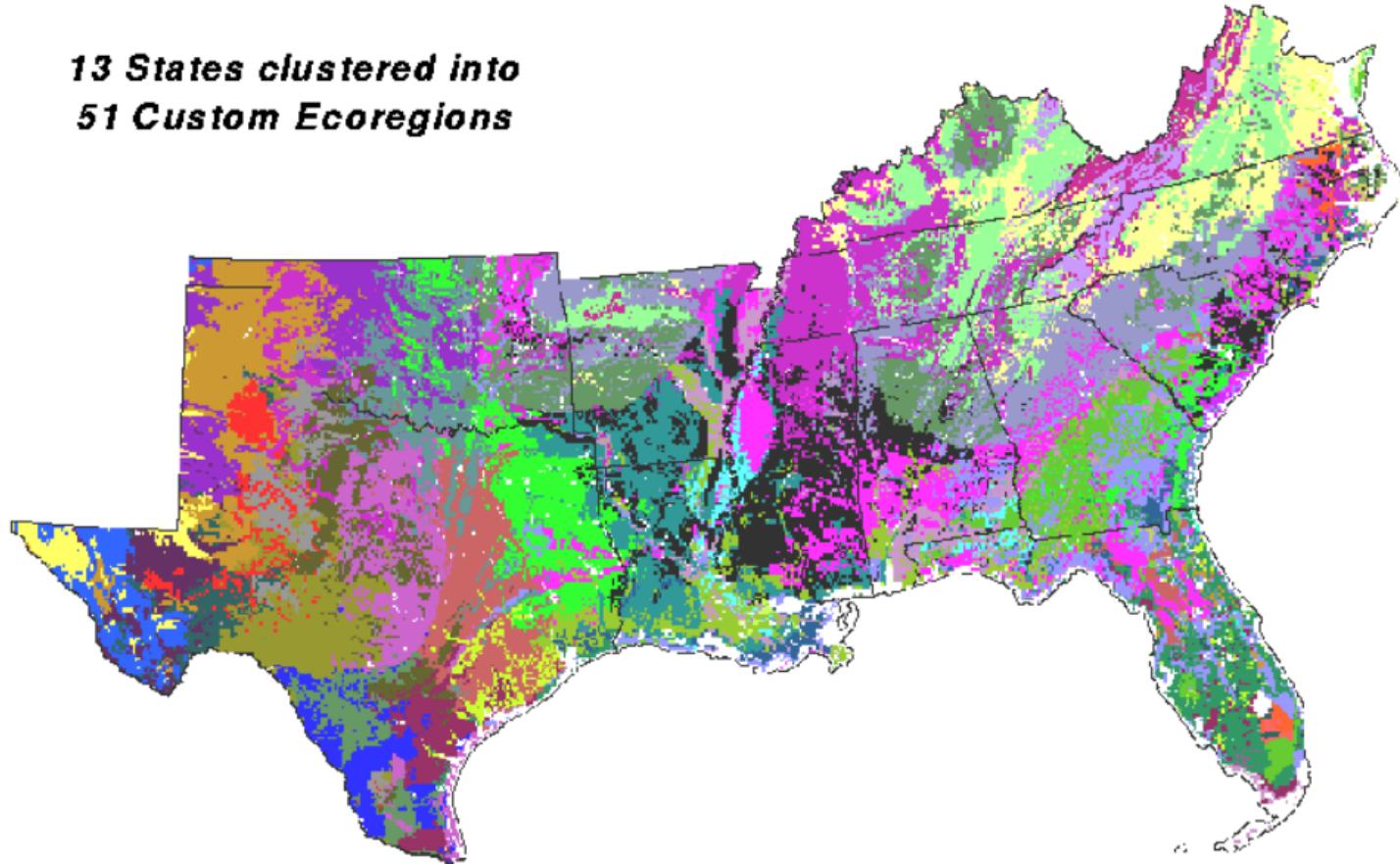
**Want:** an assignment of points to clusters







*13 States clustered into  
51 Custom Ecoregions*



# The Algorithm

- ① Assign points to clusters at random
- ② Repeat until stable:
  - ① Compute centroids of each cluster
  - ② Assign points to nearest centroid

# Cost function

$$\text{cost} = \sum_i \sum_j |x_j - \mu_i|$$

# Silhouette coefficient

Points  $d = \{d_1, \dots, d_n\}$

Clusters  $K = \{c_1, \dots, c_k\}$ .

Cluster  $c_{d_i}$  is the centroid of  $d_i$ .

# Silhouette coefficient

Points  $d = \{d_1, \dots, d_n\}$

Clusters  $K = \{c_1, \dots, c_k\}$ .

Cluster  $c_{d_i}$  is the centroid of  $d_i$ .

Let  $a_i$  be the average dissimilarity of  $d_i$  to all points in its cluster.

Let  $b_i$  be the least average dissimilarity of  $d_i$  to any cluster other than  $k_{d_i}$

# Silhouette coefficient

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

# Silhouette coefficient

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

$$s_i = \begin{cases} 1 - a_i/b_i & \text{if } a_i < b_i \\ 0 & \text{if } a_i = b_i \\ b_i/a_i - 1 & \text{if } a_i > b_i \end{cases}$$

# Silhouette coefficient

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

$$s_i = \begin{cases} 1 - a_i/b_i & \text{if } a_i < b_i \\ 0 & \text{if } a_i = b_i \\ b_i/a_i - 1 & \text{if } a_i > b_i \end{cases}$$

So  $s_i \in [-1, 1]$

# Silhouette coefficient

$s_i$  near 1  $\iff d_i$  well clustered

$s_i$  near 0  $\iff d_i$  on the border between two clusters

$s_i$  near -1  $\iff d_i$  poorly clustered

# Silhouette coefficient

Consider  $\bar{s}_i$  over  $i \in c_j$  for cluster  $c_j$

# Silhouette coefficient

Consider  $\bar{s}_i$

**video time**

# Anomaly Detection

# Introduction to Anomaly Detection

- Supervised
- Unsupervised

# Introduction to Anomaly Detection

Supervised anomaly detection:

- Training data: normal, abnormal
- Train a classifier

So reduced to existing problem of supervised classification.

# Introduction to Anomaly Detection

Unsupervised anomaly detection:

- Mostly, this is clustering
- Increasingly, this is neural networks in advanced applications

# Introduction to Anomaly Detection

Applications:

- Intrusion detection (physical or electronic)
- Fraud detection
- Health monitoring (people, animals, machines)

# Introduction to Anomaly Detection

Techniques:

- Density: kNN, local outlier factor
- SVM
- Clustering:  $k$ -Means

# Introduction to Anomaly Detection

## kNN techniques and variations

- Voronoi diagrams
- aNN

# Introduction to Anomaly Detection

## LOF

- Measure average density using kNN
- Points with low local density are suspect outliers
- There is no good thresholding technique

# Introduction to Anomaly Detection

## *k*-Means

# Examples

**ping times**

# Examples

**httpd response times**

# Examples

**single/multiple host access abuse (DOS/DDOS)**

# Examples

**bank card fraud**

# Examples

spam



questions?