Machine Learning

Images, Faces, Clustering, Anomalies

Jeff Abrahamson

Cours sur l'année, 2017–2018

Review

What is Machine Learning?

Learning is what we do when we can't explain how.

?

What is Machine Learning?

Learning is what we do when we can't explain how.

- Supervised
- Unsupervised
- Reinforcement

Lots of maths

We'll try to ignore it, but it's there...

- Vector spaces and linear algebra
- Probability
- Statistics
- Optimisation theory
- Differential calculus

The curse of dimensionality.

Data Science

?

Data Science

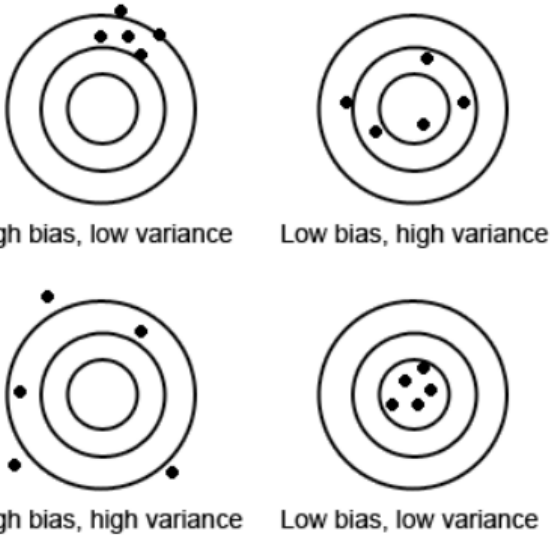
1. Define the question of interest
2. Get the data
3. Clean the data
4. Explore the data
5. Fit statistical models
6. Communicate the results
7. Make your analysis reproducible

Observational vs experimental

Data

Anecdote: it doesn't accumulate to be data.

Data



Features

Feature Engineering

Data

One of K = one-hot encoding

Outliers: don't ignore them!

Feature Engineering

1. Brainstorm
2. Pick some
3. Make them
4. Evaluate
5. Repeat

Easy Features

Text

bag of words

Easy Features

Images

corners, edges, point matching

Linear Regression

?

Linear Regression

Problem: $\{(x_i, y_i)\}$.

Given x, predict $\hat{y}$.

Here y is continuous.

Linear Regression

x: **explanatory** or **predictor** variable.

y: **response** variable.

For some reason, we believe a linear model is a good idea.

Residuals

What's left over.

?

Residuals

What's left over.

$$\text{data} = \text{fit} + \text{residual}$$

Residuals

What's left over.

$$y_i = \hat{y}_i + e_i$$

Residuals

What's left over. Goal: small residuals.

$$\sum e_i^2$$

Logistic regression

?

Logistic regression

- Binary output
- Classification

Logistic regression

- Have: continuous and discrete inputs
- Want: class (0 or 1)

Logistic regression

Logistic (sigmoid, logit) function

$$g(z) = \frac{1}{1 + e^{-z}}$$

One vs Rest, One vs One

One vs Rest, One vs One

- OvR (OvA): compute k classifiers
- OvO: compute $k(k-1)/2$ classifiers

The classifiers give scores, not just in/out answers.

One vs Rest, One vs One

One vs Rest:

Accept the judgement of the classifier with the highest score.

One vs Rest, One vs One

One vs One:

Classifiers vote. Accept the class that gets the most votes.

One vs Rest, One vs One

Advantage: Reduces multi-class classification to single-class classification.

Disadvantage: Classifier scores aren't necessarily comparable. For example, classes may have very different numbers of members.

Hyperparameters

?

Hyperparameters

- The word hyperparameter is not well-defined.
- In most contexts, it is the parameters of the underlying distribution
- In training, we learn the parameters of the model
- We choose the hyperparameters to govern the training
- So we may want to experiment to learn the distribution parameters that best optimise our learned model's performance

Testing

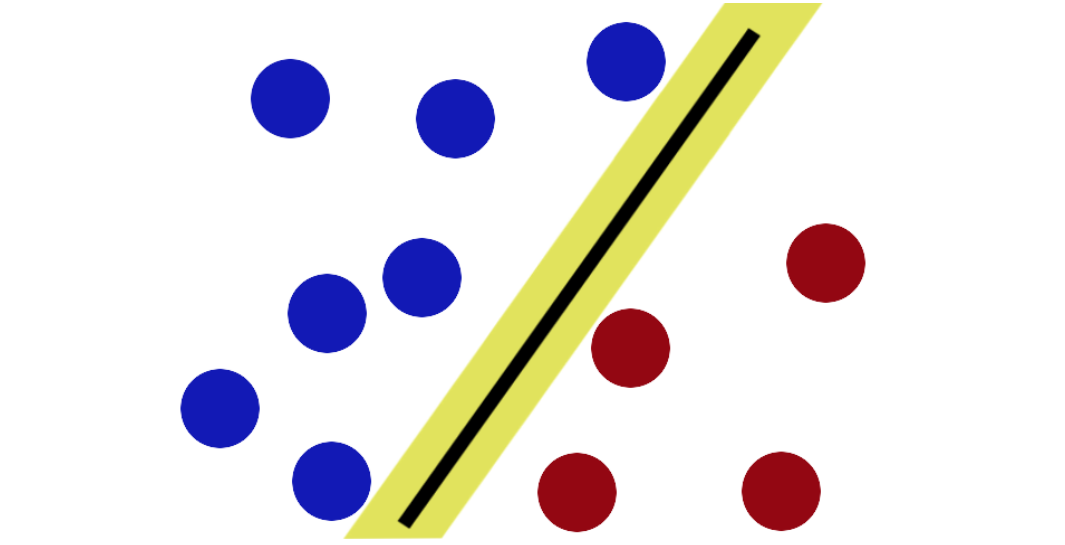
- Set aside (partition) data for testing (e.g., 70% / 30%)
- Learn on training set, test on testing set
- When searching hyperparameters, set aside again (e.g., 60% / 20% / 20%)

SVM

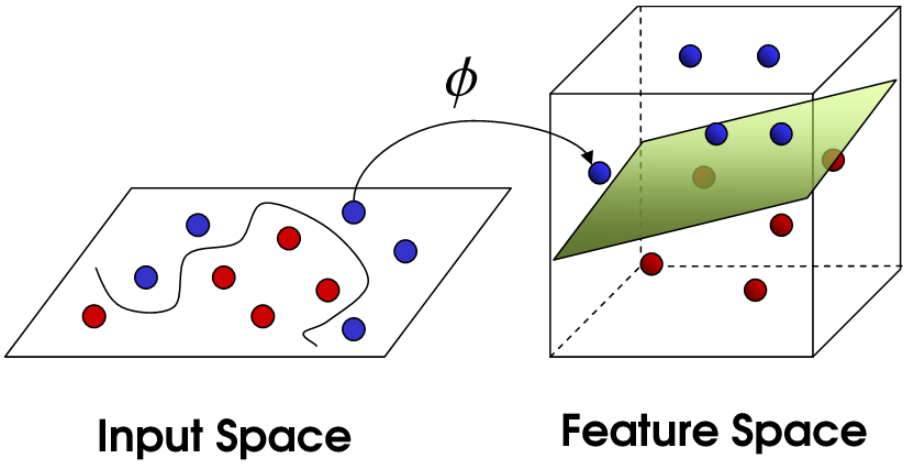
?

copperking@reddit

SVM



SVM



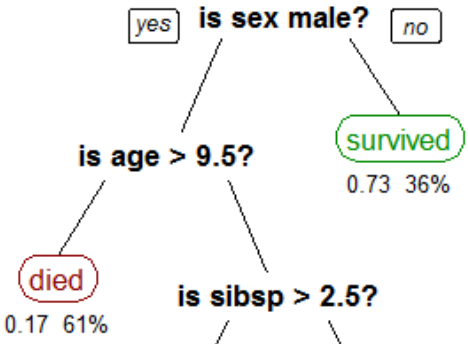
CART

?

Random Forest

?

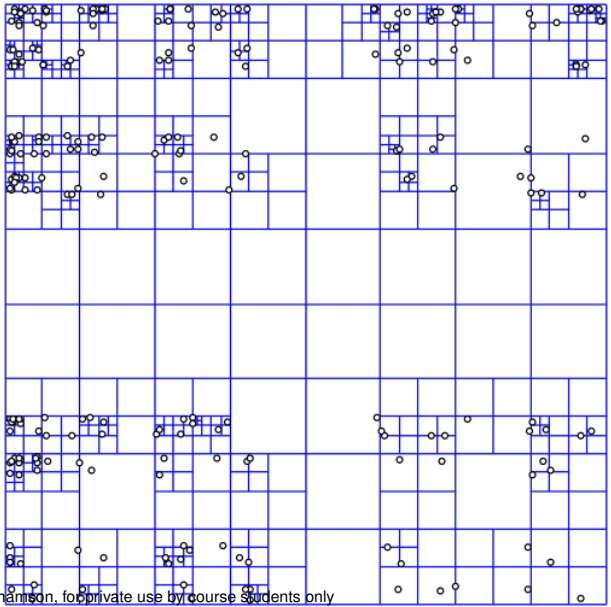
Decision Trees



E.g., passengers died with probability .17 which is 61% of observations

Stephen Milborrow

Quadtree



Decision Trees

Variations

- Classification tree
- Regression tree

CART = classification and regression trees

Decision Trees

Ensemble methods

- Bagging
- Random forest
- Boosted trees *(gradient boosted trees)*
- Rotation forest

Boostrap aggregating = bagging

Bagging

- Increase stability
- Increase accuracy
- Reduce variance
- Avoid overfitting

A type of model averaging (ensemble method).

Boostrap aggregating = bagging

- Training set D of size n
- Sample D *with replacement* to create $D_1, \ldots, D_k$ of size n'
- If $n = n'$, expect $1 - 1/e \approx 63.2\%$ repeats

- Train k models
- Average (regression) or vote (classification)

Random subspace method

attribute bagging = feature bagging

Random subspace method

Bagging (bootstrap aggregation) = resampling to create more data sets, train models on different samples

Attribute bagging = project to create more data sets, train models on different samples

Random forests

Combine bagging with random subspace method

Images

Signal processing

in 2 or 3 dimensions

Images

Details that can matter:

- Illumination
- White balance
- Resolution
- Camera settings (e.g., depth of field)
- Sensor noise
- Compression technology

Images

Challenges:

- Segmentation
- Area of interest detection
- Perspective shifting

Images

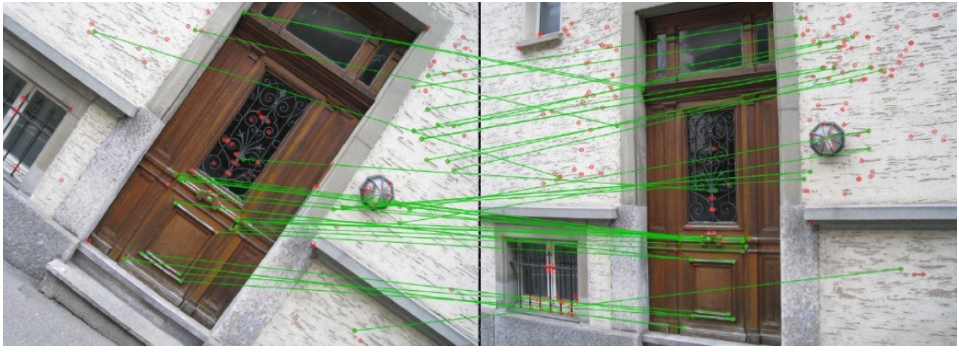
Applications:

- Agriculture: fruit ripening, automated harvesting
- Security: detecting specific people
- Security: detecting accidents (e.g., falls)
- Art: counterfeit detection
- Medicine: assisted surgery
- Image search

Images

Image search (at first):

- Texture
- Colour
- Shape, simple objects



Eddie Bell @ Lyst

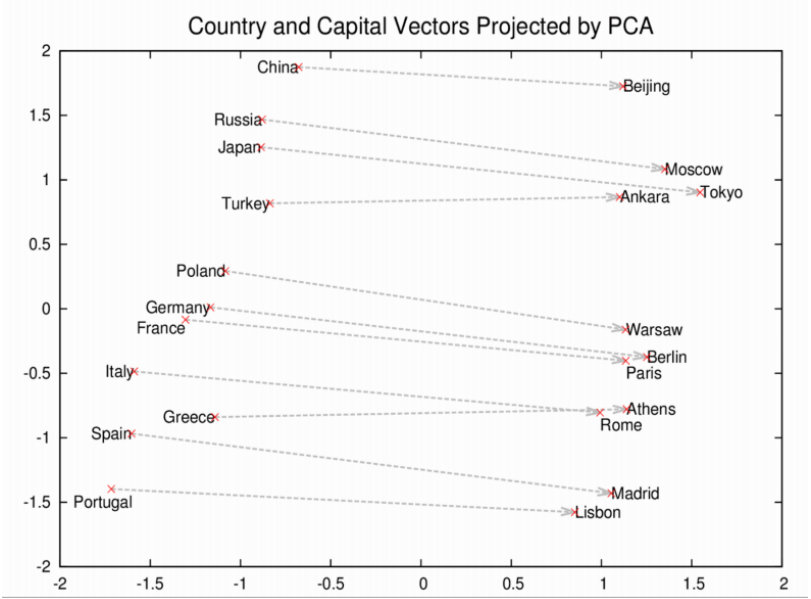
Eddie Bell @ Lyst

Eddie Bell @ Lyst

Eddie Bell @ Lyst

-1	0	+1
-2	0	+2
-1	0	+1

-1	-2	-1
0	0	0
+1	+2	+1



	Term	Similarity
	"shift"	0.933104
	"gown"	0.887743
	"skirt"	0.881672
	"bandage"	0.880162
	"midi"	0.869786

Similar to ‘dress’

Eddie Bell @ Lyst

a group of young girls standing next to each other on the beach

A clock tower with a clock on top of it

Google?

A bunch of bananas hanging from a tree

Google?

PCA

Principle component analysis

Analyse en composantes principales

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

Also known as

- Discrete Kosambi-Karhunen–Loève transform (KLT) (signal processing)
- Hotelling transform (multivariate quality control)
- Proper orthogonal decomposition (POD) (ME)
- Singular value decomposition (SVD), Eigenvalue decomposition (EVD) (linear algebra)
- Etc.

History

- Invented by Karl Pearson in 1901
- Invented (again) and named by Harold Hotelling in 1930's
- Also known as...

Also known as

- It's a long list, every field uses a different name...

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 $X = \{x_1, x_2, \ldots, x_n\}$ 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:

$$S = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T$$

Eigenfaces algorithm

Compute the eigenvectors of S:

$$Sv_i = \lambda_i v_i \qquad i = 1, 2, \ldots, n$$

Sort the eigenvectors in decreasing order.

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

Eigenfaces algorithm

Compute the eigenvectors of S:

$$Sv_i = \lambda_i v_i \qquad i = 1, 2, \ldots, n$$

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
- Offlne

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

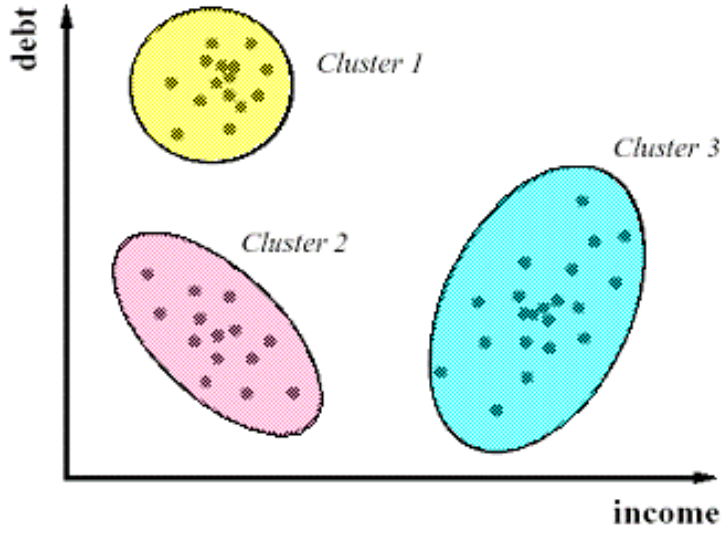
Clustering

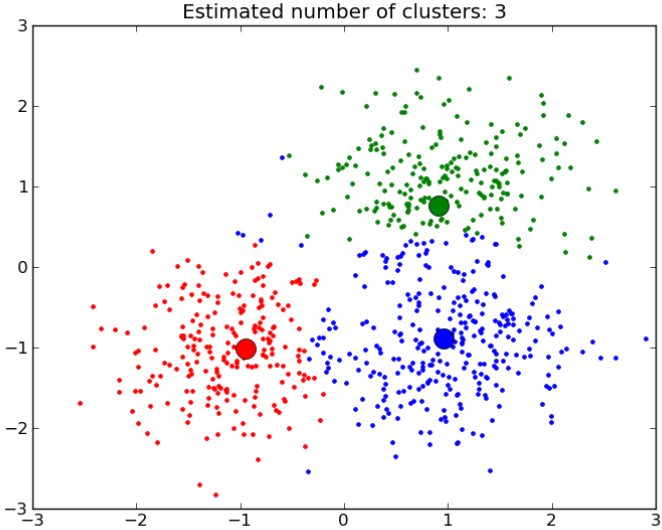
The Problem

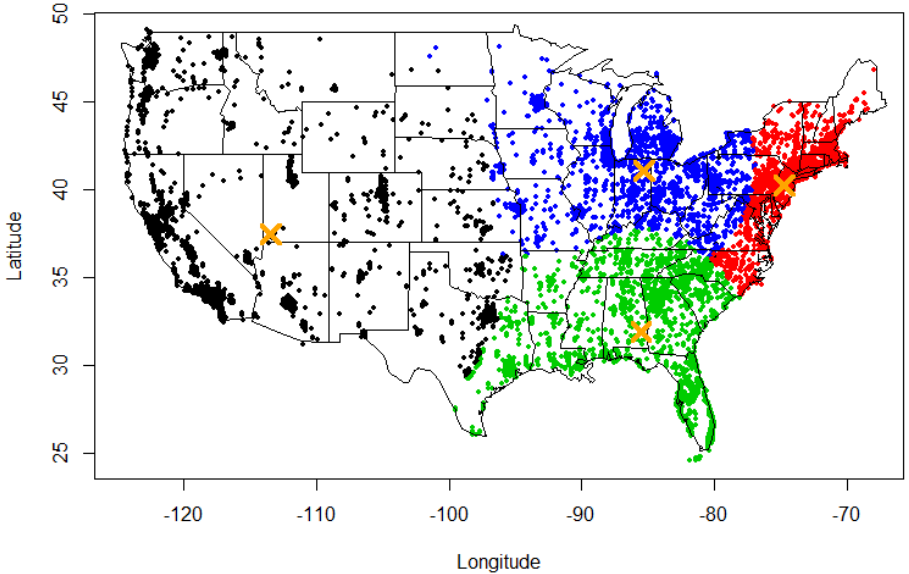
Have points $d = \{d_1, \ldots, d_n\}$.

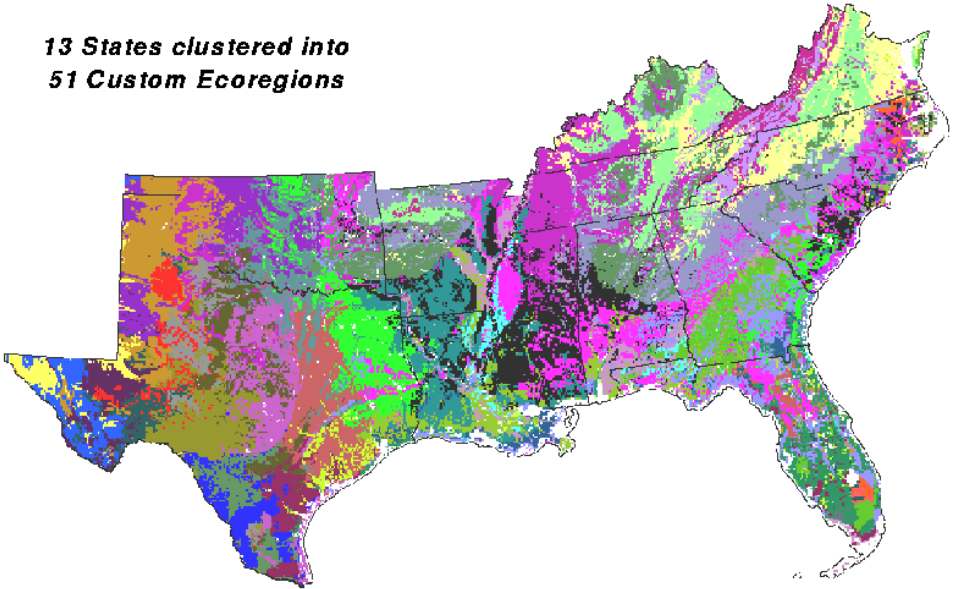
Have number of clusters k.

Want: an assignment of points to clusters









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 $d = \{d_1, \ldots, d_n\}$

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

Cluster c_{d_i} is the centroid of d_i.

Silhouette coefficient

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

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

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 $\overline{s_i}$ over $i \in c_j$ for cluster c_j

Silhouette coefficient

Consider $\overline{s_i}$

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

LOF

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

k-Means

Examples

ping times

Examples

httpd response times

Examples

single/multiple host access abuse (DOS/DDOS)

Examples

bank card fraud

Examples

spam