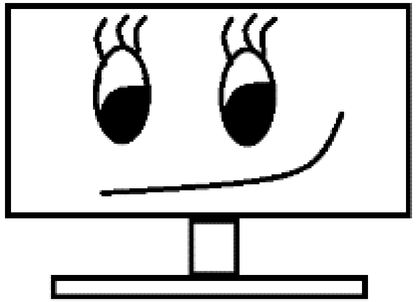


CVI620/ DPS920
Introduction to Computer Vision

Computer Vision
Projects & Evaluations

Seneca College

Vida Movahedi

Overview

- A simple computer vision problem
- Features
- Learning methods
  - Supervised, unsupervised, semi-supervised
  - Training and test sets
  - Ground Truth
  - Classification, Clustering, Regression

- Performance evaluation

Example: Assembly Line Inspection

- Design a vision system that can detect the objects on the assembly line

- How can you solve this problem?

https://binged.it/2TlZFLU

Solving a Computer Vision problem

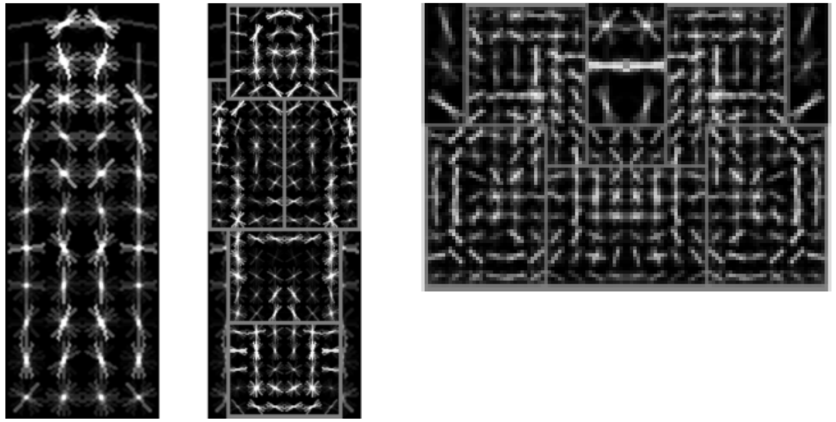
- Method 1: Start with intuition, implement it, use data to verify your intuition/ concept
  - Use color
  - Use size, shape, etc.

- Method 2: Start with data, automatically learn how to solve the problem (Machine Learning methods)
  - Give labelled training samples to a machine learning algorithm
  - The algorithm automatically builds a model for detection

- https://www.quora.com/What-is-the-alternative-to-machine-learning

Features

- In Machine learning, features are data measurements or information that can be used to predict a target value
- In computer vision, in addition to above, specific structures in images, e.g. edges, corners, etc. are also referred to as features.



http://cs.brown.edu/people/pfelzens/papers/lsvm-pami.pdf

- Examples of commonly used image features:
  - Color: e.g. color histogram (bin values)
  - HoG: Histogram of oriented gradients
  - SIFT: Scale-invariant feature transform
  - Shape features

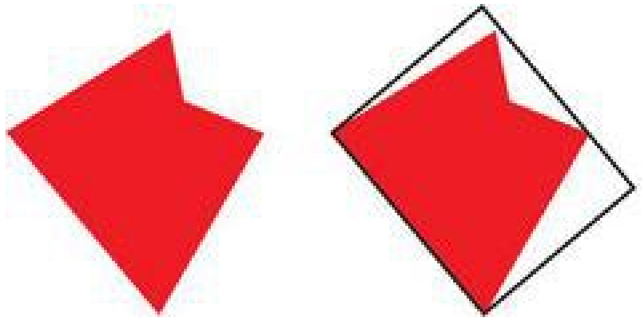
https://en.wikipedia.org/wiki/Scale-invariant_feature_transform

Shape features [3]

- Some features used for shape/ object detection
  - Area: number of pixels in the shape/ contour
    - OpenCV findContour and contourArea [3]
  - Elongatedness: how long the shape is

$$elongatedness = \frac{area}{(2d)^2}$$

    - d: number of iterations of erosion for the region to disappear
  - Length to width ratio of the *minimum bounding rectangle*



```
RotatedRect min_bounding_rectangle =
                    minAreaRect(contours[contour_number]);
```

Shape features (cont.)

- Area of convex hull

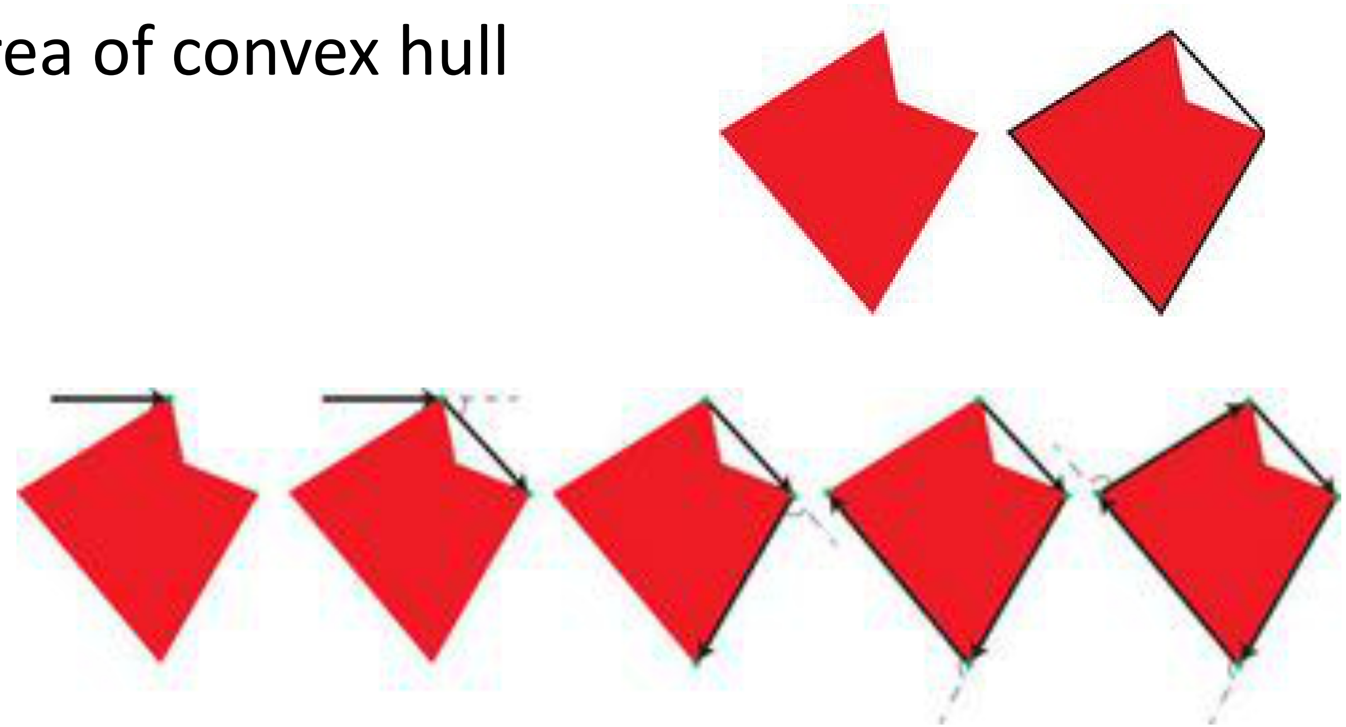


Figure 8.13: Stages in the creation of the convex hull. The START point and the initial 'previous vector' are shown(left). Then, in each of the other drawings, the previous vector from the previous step is extended and the next point on the convex hull is located as that with the minimum angle to the previous vector

More features

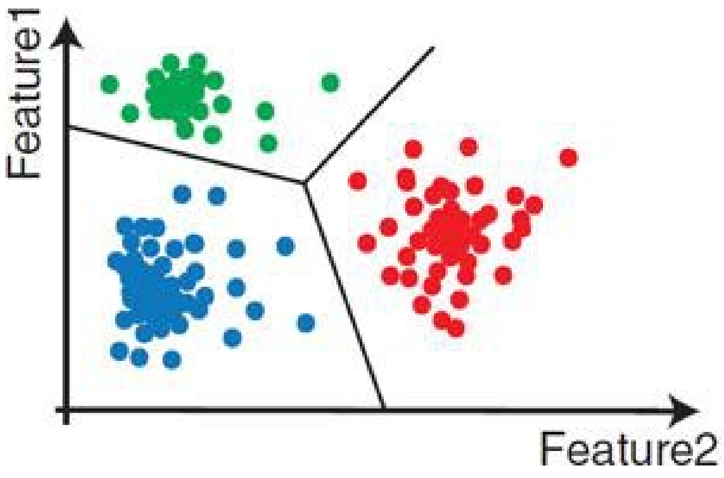
- Concavities and holes
- Rectangularity
  - ratio of region area to the area of the minimum bounding rectangle

- Circularity
  - Circularity = $\frac{4*\pi*\text{area}}{\text{perimeter}^2}$
  - A perfect circle has a circularity of 1 (maximum)

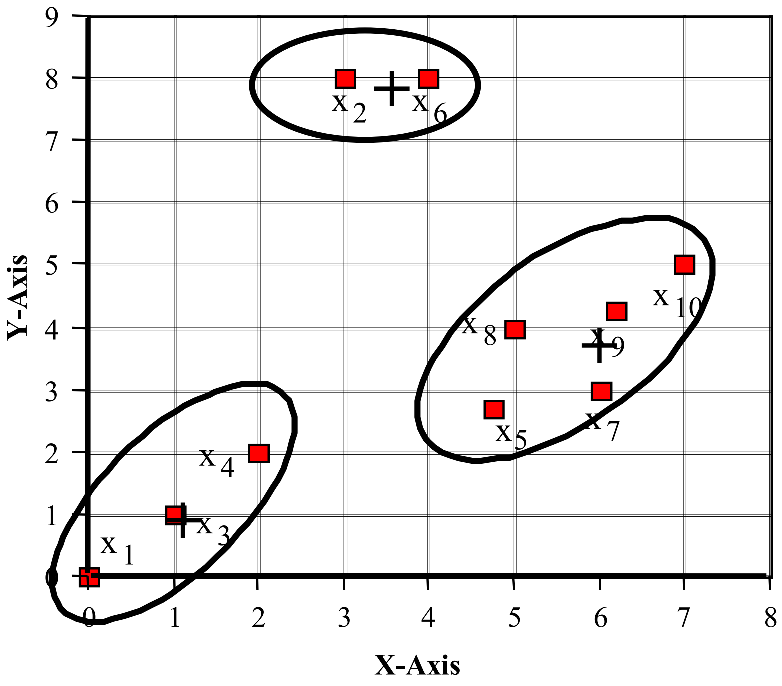
- Moments and moment invariants

Machine Learning Methods

- Classification [3]

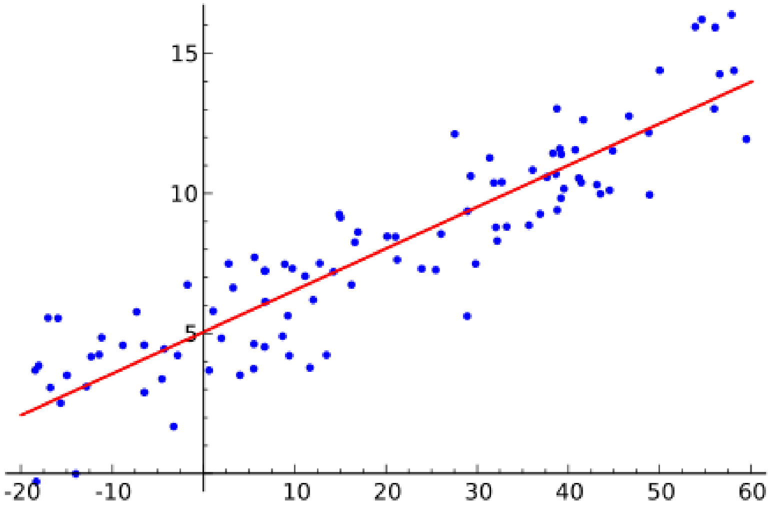


- Clustering



D. Abbott, Applied Predictive Analytics, Wiley, 2014

- Regression



https://en.wikipedia.org/wiki/Regression_analysis

Separable classes [3]

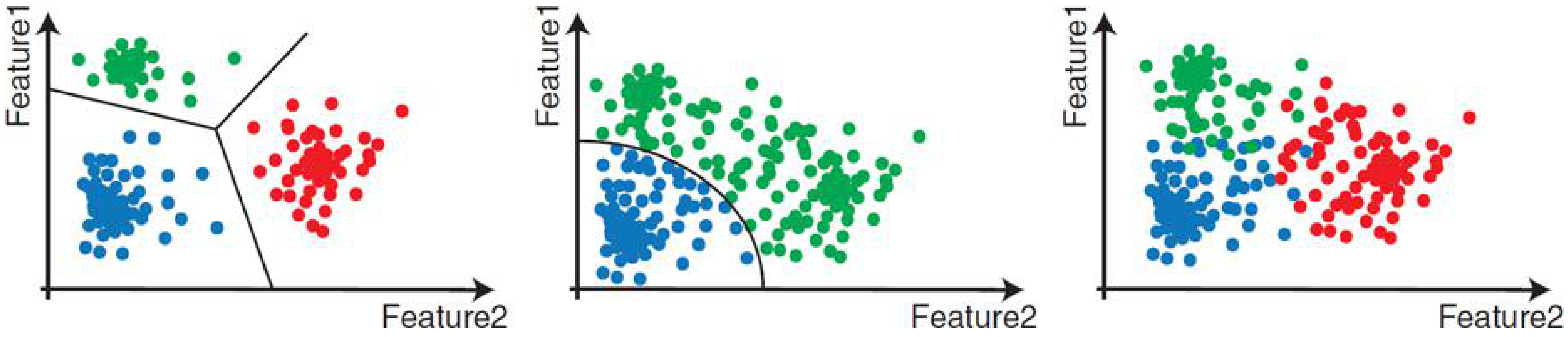


Figure 8.15: Objects of different classes (shown by dots of different colours) mapped into 2D feature space. On the left, the classes are linearly separable, in the centre, the classes are separable and a hyper-surface (a curve in this case) is shown between them, and on the right the classes are inseparable (using these features)

Supervised Learning (or training)

- In supervised learning, training data contains for each sample:
  - Feature vector of the sample
  - Given class or target value of the sample

- The classifier or predictor is trained based on the samples to learn the class or target value of each sample (training data)

- But also be able to generalize & predict the class (or target value) of samples not seen (test data)

Unsupervised & Semi-supervised Learning

- Unsupervised Training
  - Training data consists of only the feature vectors. The class, target, or cluster for each sample is unknown.
  - The classifier learns to classify (cluster) the data

- Semi-supervised training
  - Unsupervised, but given some feedback

Training vs. Test data

- A dataset of images, samples or examples
  - Should include sufficient variety
  - Noise, various background, lighting, occlusion, sizes, poses, etc.

- Often divided into two sets
  - **Training set**:

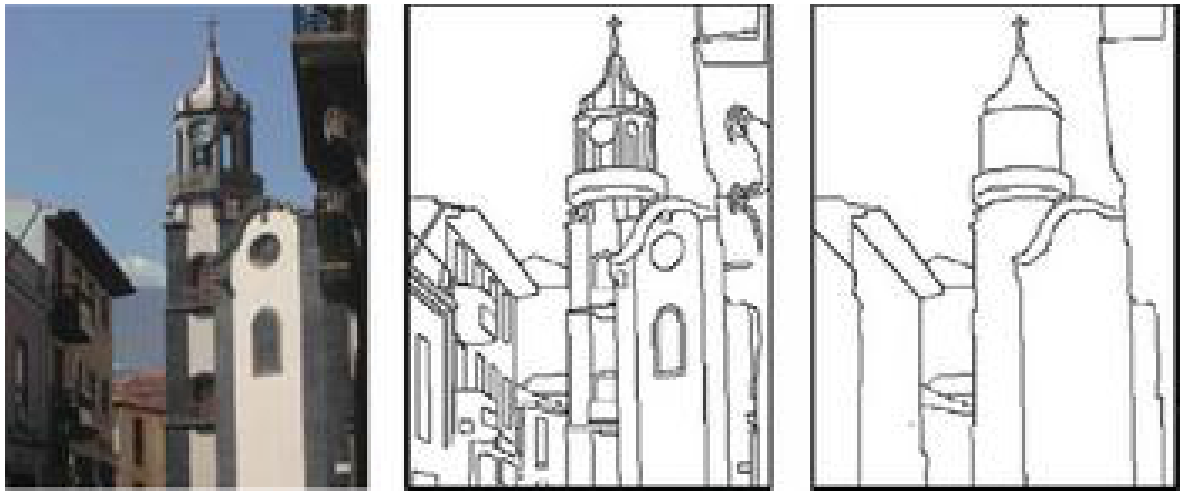
    Used for training the algorithm, setting the parameters, making design choices
  - **Test set**:

    Only used to compare performances between different methods, sometimes not even shared publicly

Ground Truth [3]

- The perfect answer
- Sometimes not possible to calculate/ measure
- Often annotated/ provided / labelled by multiple human subjects
- The more human subjects, the less it will be considered as 'subjective'

Example:
Berkeley segmentation dataset
30 human subjects
12 segmentations per image



Examples

1. Driver drowsiness
   - A set of images with drowsy and non-drowsy drivers
   - GT: a label of drowsy or non-drowsy (or 0/ 1)
2. Traffic sign detection
   - A set of images with traffic signs or no signs
   - GT: A label for each image indicating the sign in image (0: no sign, 1: stop, etc.)
3. Hockey player detection
   - A set of images of hockey rinks with players
   - GT: for each image, a list of player locations, i.e. (x, y) coordinate or bounding box features (x, y, w, h); or image masks with background pixels set to 0 and foreground pixels set to 1
4. Motion tracking
   - A set of videos with human motion or non-human or no motion
   - GT: a label of human motion or not (0/ 1)

Examples- cont.

5. Hand gesture passwords
   - A set of images with hand gestures
   - GT: a label of gesture (0 for none, 1 for …)
6. Driver's license detector
   - A set of images of driver's licenses at different angles and poses
   - GT: Location of card's bounding box (x, y, w, h) and info on cards
7. Cash counter
   - A set of images of coins and paper bills
   - GT: total amount of money in the image
8. Face or fake
   - A set of images of faces vs. face-like objects
   - GT: label of face or none-face (0 or 1)

Classifier

- Assign an input to one of possible classes (W_1, W_2, …, W_R)
- For example, object recognition (what's in the image?)
- Input: input pattern or feature vector (x_1, x_2, …., x_n)
- Output: class $r \in \{1, \dots, R\}$

Some well-known Methods

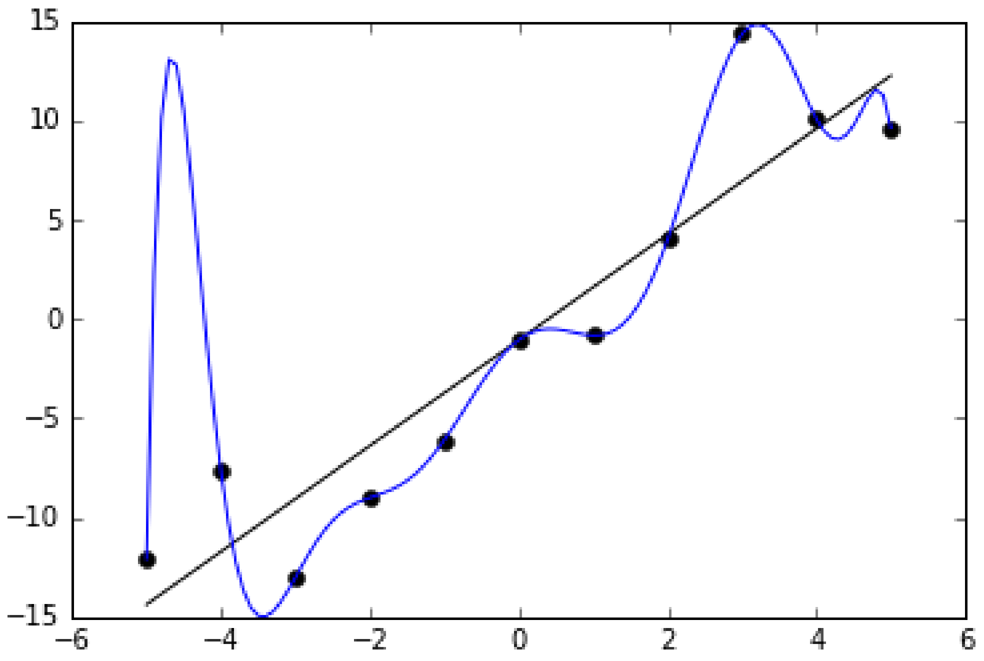
- Some classifiers
  - K-nearest neighbor
    - Get votes from the closest neighbors in the feature space among training data
  - Logistic regression
  - Support Vector Machines (SVM)
  - Naïve Bayes

- Clustering
  - K-means

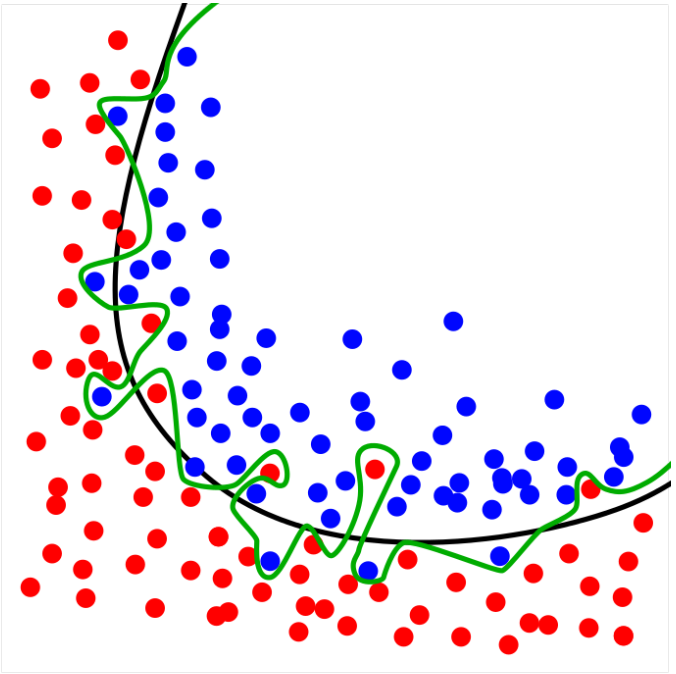
- Regression
  - Linear
  - Nonlinear

Overfitting

- Corresponding too closely or exactly to the training data
- Not generalizing → poor performance when seeing new data



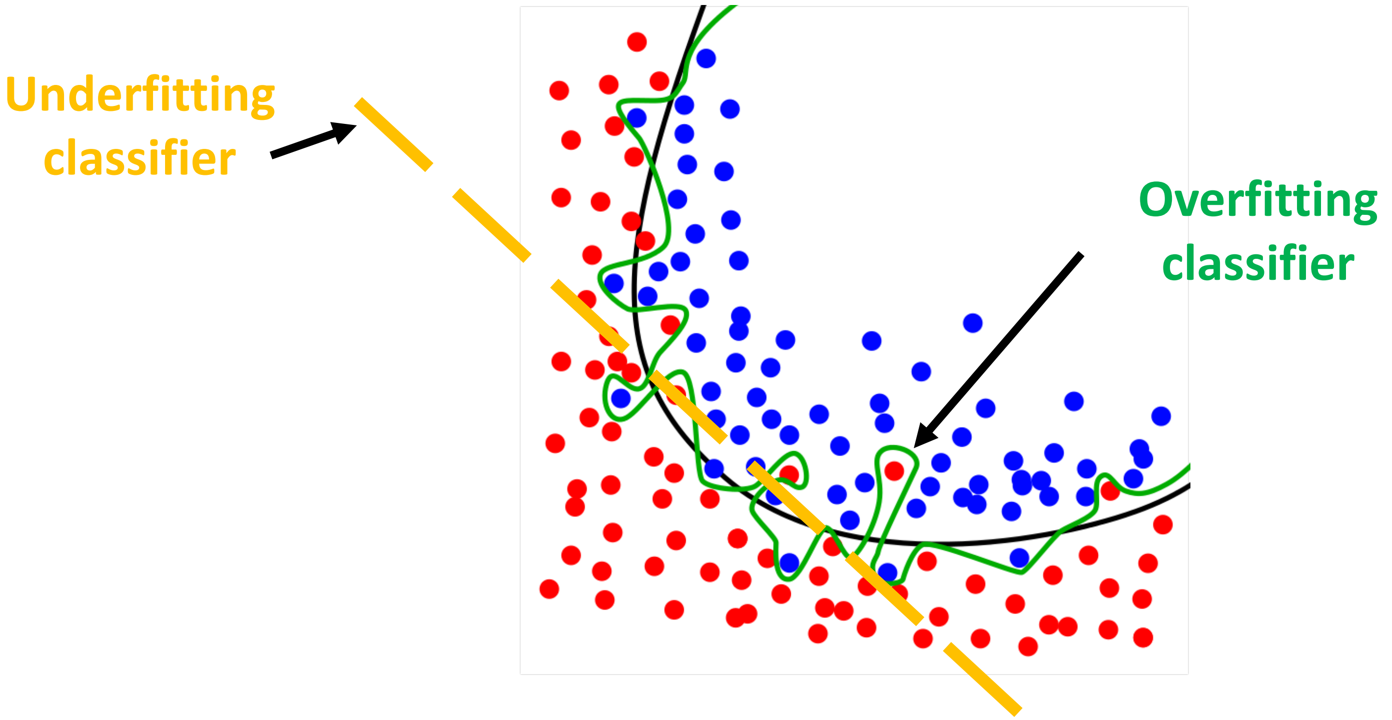
[Wikipedia] Overfitting in regression



[Wikipedia] Overfitting in classification

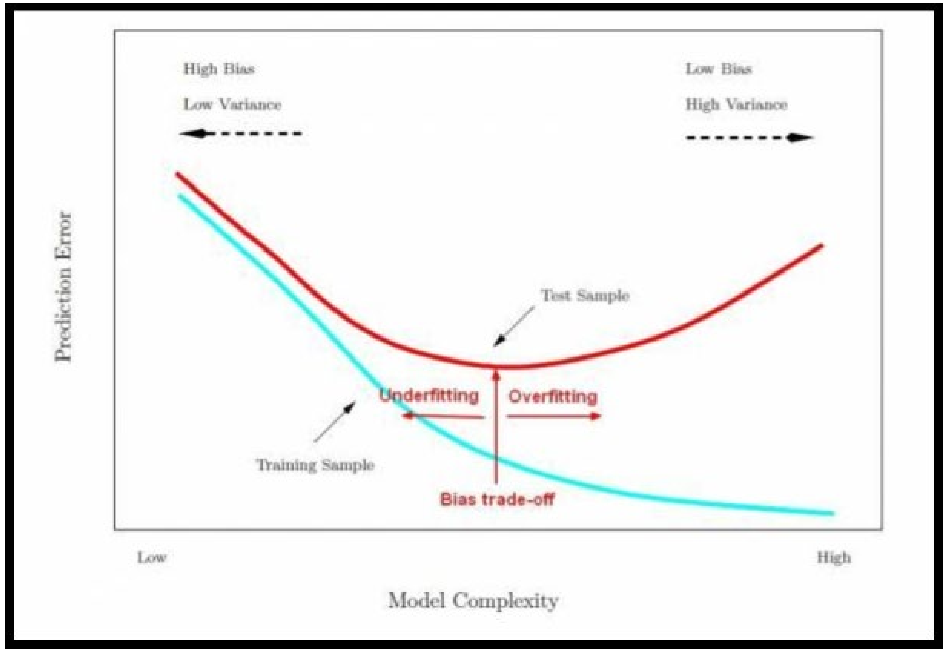
Underfitting

- The learned model, being too simple/ too general
- Not learning the details



Validation Set

- Optimize parameters based on data not used in training
- Validation set
  - A randomly selected subset of the training set
  - To avoid under or overfitting when optimizing classifiers or repressors



https://kevinbinz.com/tag/training-data/

- Cross validation
  - Leave-1-out
  - Leave-p-out
  - Repeat and take average

Performance Evaluation

How well is the algorithm working?

- After implementing an algorithm, we want to know
  - How well it is working?
  - How does it compare with other algorithms
- Qualitative performance evaluation
  - Show the performance of the algorithm (by examples)
  - Show cases where it works well (or better than competition)
  - Show cases where it fails
- Quantitative performance evaluation
  - Not subjective, based on a test set and an evaluation measure
  - Fair to all methods being compared

Example: (House) Cat-finder!

Is there a cat in the image?

Algorithm's output:

Yes	No	Yes	No
TP	FN	FP	TN

Metrics for recognition/ classification

- TP (true positives):
  - Number of samples identified correctly as belonging to a class/ category
- FP (false positive): (false alarms!)
  - Number of samples identified incorrectly as belonging to a class / category
- TN (true negative):
  - Number of samples identified correctly as NOT belonging to a class/ category
- FN (false negatives):
  - Number of samples identified incorrectly as NOT belonging to a class/ category
- Total number of samples = TP + FP + TN + FN

Precision / Recall

- Recall: Percentage of objects successfully identified

$$Recall = \frac{TP}{TP + FN}$$

- Precision: Percentage of identified objects which are actually correct (not false alarms)

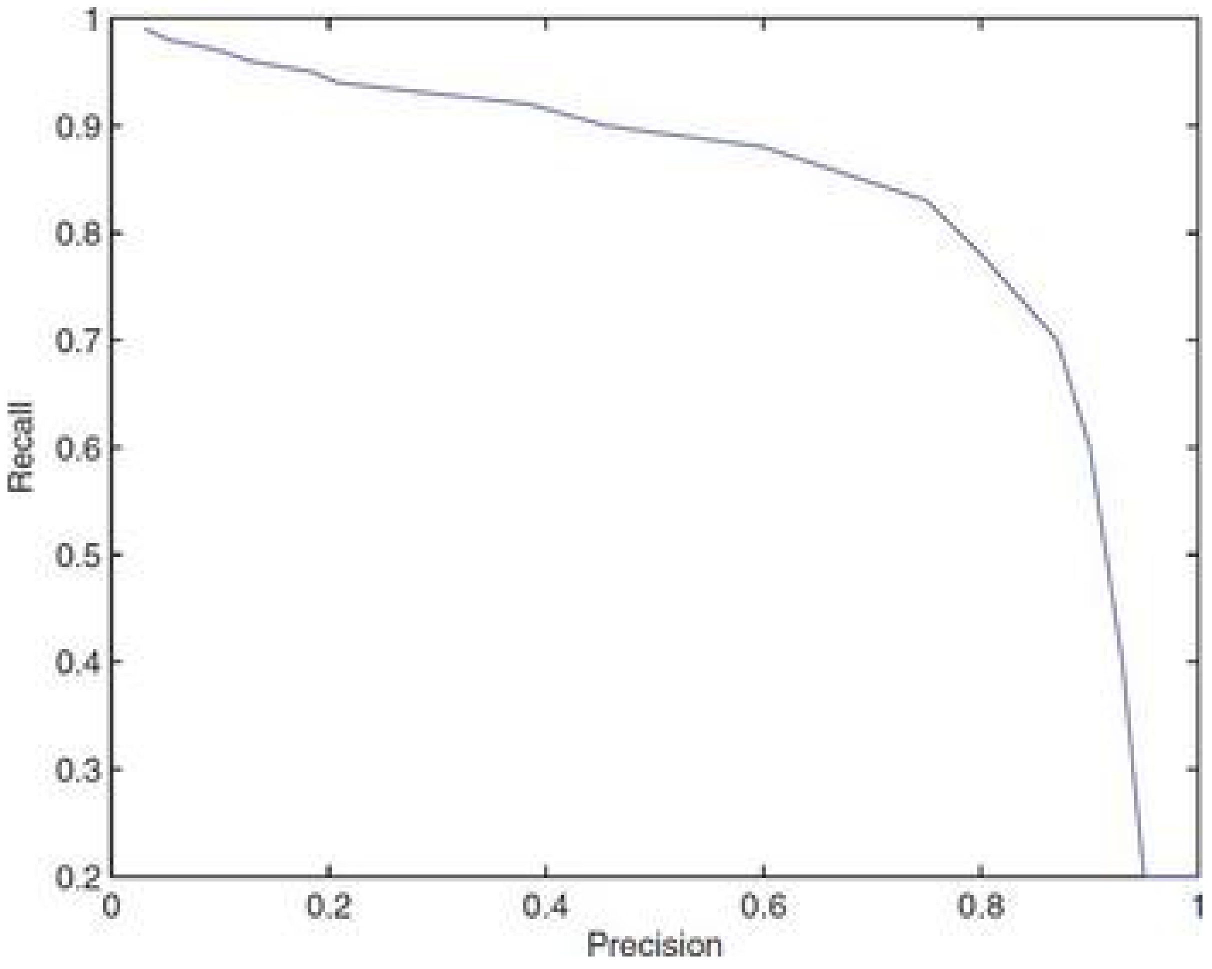
$$Precision = \frac{TP}{TP + FP}$$

- F- measure:

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}$$

F_1 (β=1) is the most commonly used

PR Curves



References

[1] Computer Vision: Algorithms and Applications, R. Szeliski (http://szeliski.org/Book)

[2] Learning OpenCV 3, A. Kaehler & G. Bradski

- Available online through Safari Books, Seneca libraries
- https://senecacollege-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01SENC_ALMA5153244920003226&context=L&vid=01SENC&search_scope=default_scope&tab=default_tab&lang=en_US

[3] Practical introduction to Computer Vision with OpenCV, Kenneth Dawson-Howe

- Available through Seneca libraries
- https://senecacollege-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=01SENC_ALMA5142810950003226&context=L&vid=01SENC&search_scope=default_scope&tab=default_tab&lang=en_US