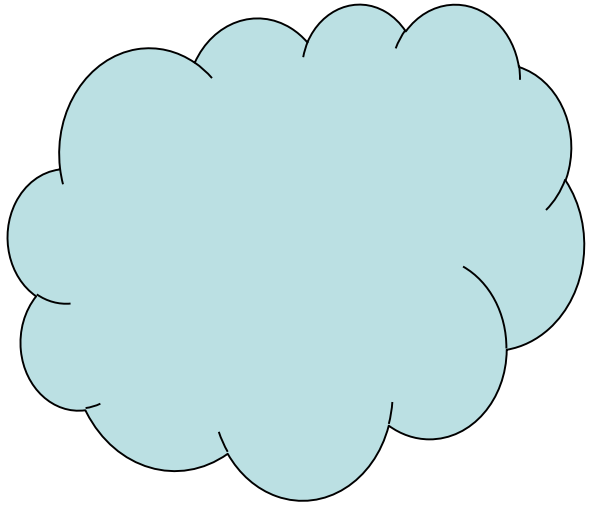
Introduction

How do we learn from Observations?



Given:

- Observations
- Experience
- Examples

Learn:

- Rules
- Cycles
- Repetitions
- Patterns

Learning from Data

Windsor, Ontario: Sunrise and set times							
			Length of day		Solar noon		
Date	Sunrise	Sunset	This day	Difference	Time	Altitude	Distance (10^6 km)
Sep 11, 2011	7:08 AM	7:49 PM	12h 41m 40s	− 2m 47s	1:29 PM	52.2°	150.592
Sep 12, 2011	7:09 AM	7:47 PM	12h 38m 51s	− 2m 48s	1:28 PM	51.8°	150.553
Sep 13, 2011	7:10 AM	7:46 PM	12h 36m 03s	− 2m 48s	1:28 PM	51.4°	150.514
Sep 14, 2011	7:11 AM	7:44 PM	12h 33m 14s	− 2m 48s	1:28 PM	51.1°	150.474
Sep 15, 2011	7:12 AM	7:42 PM	12h 30m 26s	− 2m 48s	1:27 PM	50.7°	150.434
Sep 16, 2011	7:13 AM	7:40 PM	12h 27m 36s	− 2m 49s	1:27 PM	50.3°	150.395
Sep 17, 2011	7:14 AM	7:39 PM	12h 24m 47s	− 2m 49s	1:27 PM	49.9°	150.355

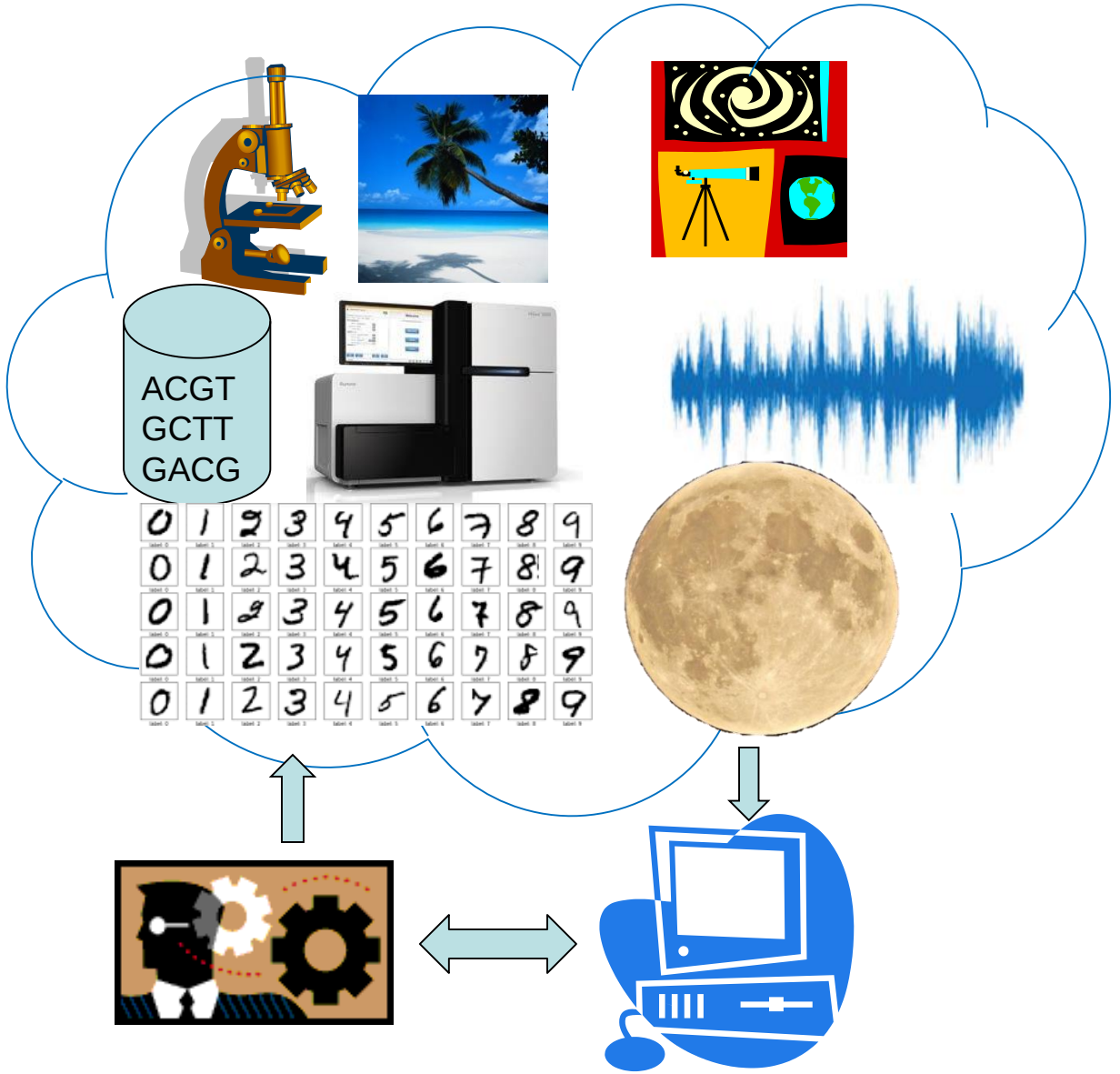
Can we predict what time the sun will rise on Sep 18?

How to name things

- How do we name things?
- One of the essential problems of humans
- It's the hardest problem in programming!
- My view:
  - group things (objects) that are similar
  - Assign a word to the group
  - Each object is an instance of the group
  - The essence of naming is that of finding a pattern
- Examples
  - apple, orange, adenocarcinoma, computer, drone, smartphone

Machine Learning

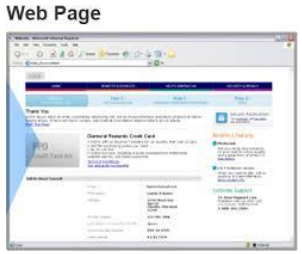
- We learn from observations, examples, images, signals, perception, data, experience, …
- Computers can learn from these as well
- Machine learning: A field of artificial intelligence that involves the design and implementation of algorithms for computers to evolve their behavior from observations, examples, images, sensors, data, experience, …



Representation Learning

Raw data:

labeled or unlabeled



… … … …

Machine learning model

- Classification: supervised
  - labeled samples
  - Predict class label
- Regression
  - Find model that best fits data
  - Predict future values (e.g., stocks)
- Clustering: unsupervised
  - Group objects on the basis of similarity
  - Gain knowledge from new groups
- Dimensionality reduction:
  - Find simpler representation of data
  - Preserve as much info as possible
- Embedding in graphs

Representation Learning – more formally

Input:

- Raw data – unstructured

Feature engineering

- Obtain structured data
- $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x} \in \mathbb{R}^d$
- Optional: class labels, $\{\omega_1, \omega_2, \ldots, \omega_c\}$
- $\mathbf{x}_i \in \omega_j$

Design a model: hypothesis

- $y = f(\mathbf{x})$
- y is a "representation" of $\mathbf{x}$

- Loss function:
  - $\ell(\mathbf{x}, y)$
  - Called "empirical loss"
- Aim: Design
  - Minimize empirical loss
- Aim: Prediction
  - Given a new unknown sample $\mathbf{x}$
  - Accurate prediction
  - Called "generalization power"

Machine Learning Lifecycle: Traditional vs Modern

Raw data

Structured data

Digit	x_1	x_2
0	2.1	.03
0	3.7	.07
0	4.2	.18
1	0.7	.15
1	0.9	.01
1	1.5	.08
…	…	…

Model

Output

0
1
2
3
…

Traditional: Feature Engineering

Modern: Automatically learn features

Learning algorithm

Prediction: $y = f(\mathbf{x})$

Classification

Given:

- $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x} \in \mathbb{R}^d$
- Class labels: $\boldsymbol{\omega} = \{\omega_1, \omega_2, \ldots, \omega_c\}$
- $\mathbf{x}_i \in \omega_j$

Problem: hypothesis

- Given new sample $\mathbf{x}$
- Find $\hat{y} = f(\mathbf{x})$ that assigns $\mathbf{x}$ to ω_j
- $f: \mathbb{R}^d \rightarrow \boldsymbol{\omega}$

Loss function:

- Simplest form:
  - $\ell(\mathbf{x}, \hat{y}) = \sum_{\mathbf{x}} \lambda(\mathbf{x}, \hat{y}) P(\mathbf{x})$
  - where ω_j is $\mathbf{x}$'s class, and
  - $\lambda(\mathbf{x}, \hat{y}) = \begin{cases} 0 \text{ if } \hat{y} = \omega_j \\ 1 \text{ if } \hat{y} \neq \omega_j \end{cases}$
  - Called 0-1 symmetrical loss
- Risk can be incorporated:
  - What if the loss of assigning $\mathbf{x}$ to a particular class ω_j is higher?
  - Example: classifying an important email as spam and deleting it

Regression

Given:

- $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x} \in \mathbb{R}^d$

Problem:

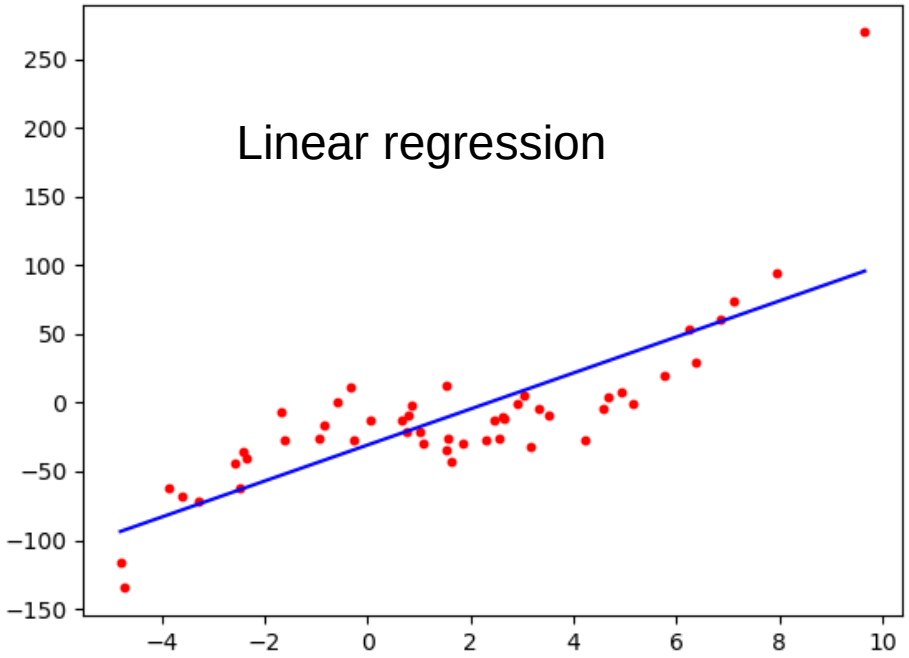
- Find $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$
- such that, given new sample $\mathbf{x}$
- $\mathbf{y} = f(\mathbf{x})$ gives a new value to $\mathbf{x}$

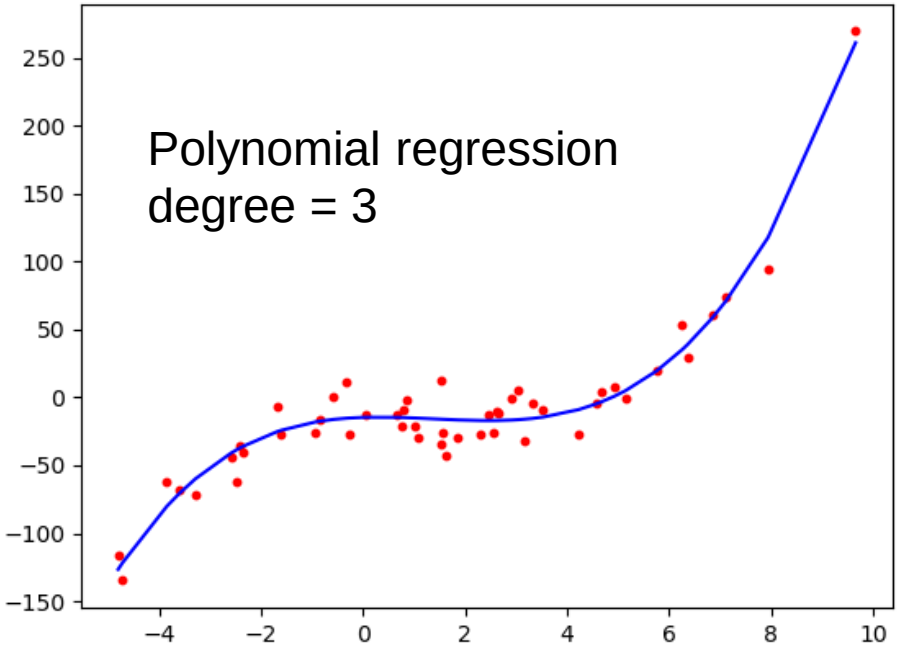
Loss function:

- Simplest form: MSE
  - $\ell(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{x} \in D} \|f(\mathbf{x}) - \mathbf{x}\|^2$

Prediction:

- Find $\mathbf{y} = f(\mathbf{x})$, where $\mathbf{x} \notin D$
- Ex: Future stock value





Clustering

Given:

- $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x} \in \mathbb{R}^d$

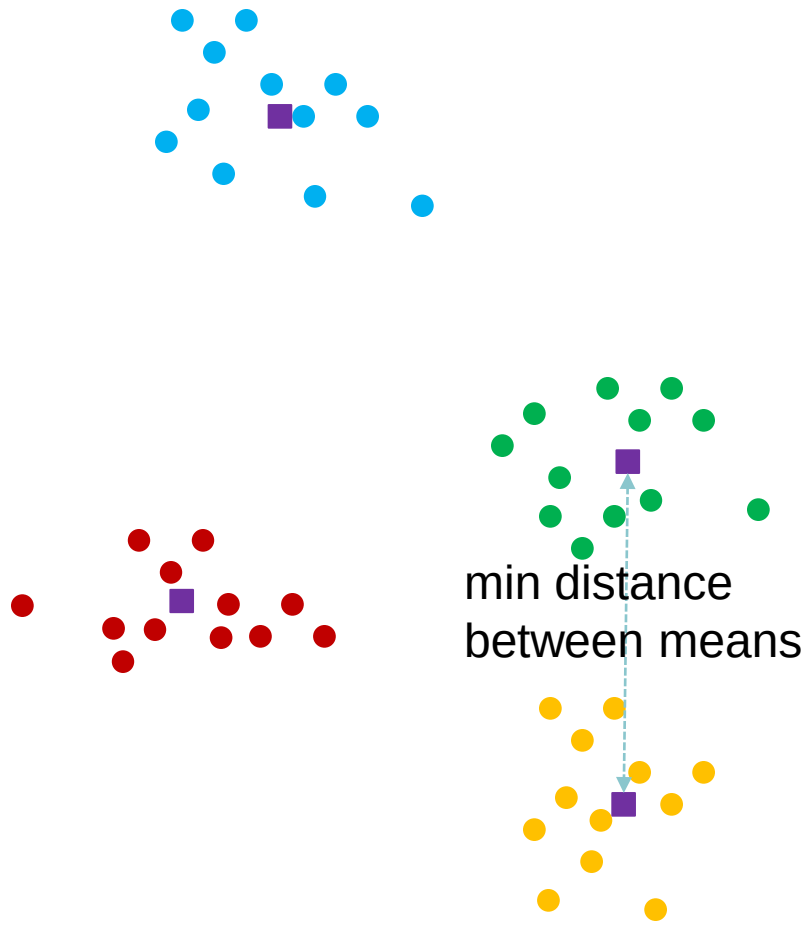
Problem:

- Find k subsets of D: $\{D_1, D_2, \ldots, D_k\}$
- D_i and D_j may (not) be disjoint

Loss function:

- Smaller (compact) clusters
- D_i and D_j as dissimilar as possible
- Example: Xie-Beni Index

$$XB(k) = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n} u_{ij}^2 \|x_j - \mu_i\|^2}{n \min_{i,j} \{\|\mu_i - \mu_j\|^2\}}$$



Dimensionality Reduction

Given:

- $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where $\mathbf{x} \in \mathbb{R}^d$

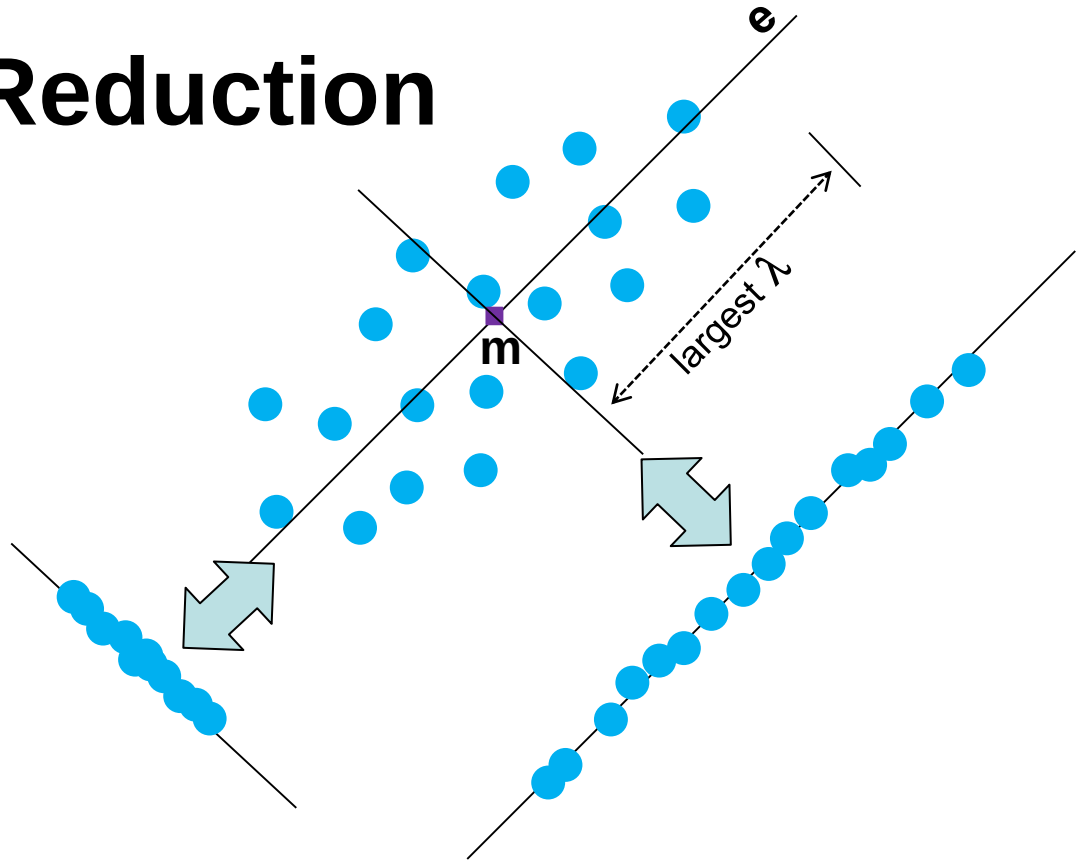
Problem:

- Find $f: \mathbb{R}^d \rightarrow \mathbb{R}^m$
- $\mathbf{y} = f(\mathbf{x})$, where $\boldsymbol{y} \in \mathbb{R}^m$
- Such that $m < d$ (or $m \lll d$)
- Decode $\hat{\mathbf{x}} = g(\mathbf{y})$
- Retain as much info as possible

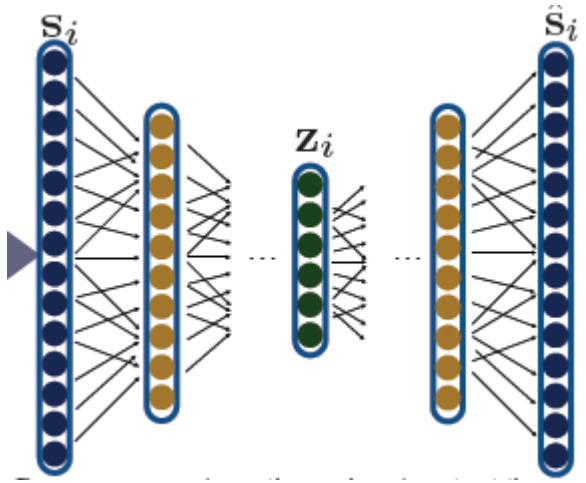
Loss function:

Examples:

- $\ell(\mathbf{x}, \hat{\mathbf{x}}) = \min \|\mathbf{x} - \hat{\mathbf{x}}\|^2$
- $\mathcal{L} = \sum_{v_i \in V} \|DEC(z_i) - s_i\|_2^2$

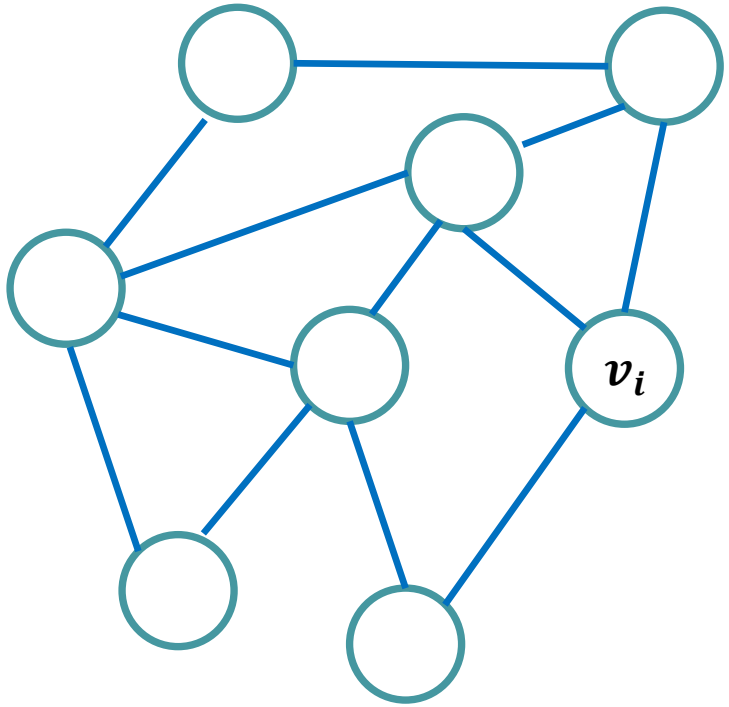


Autoencoder:
ref [5]



Representation Learning in Graphs - Embedding

- Automatically extract features from graphs
- Efficient for machine learning in networks

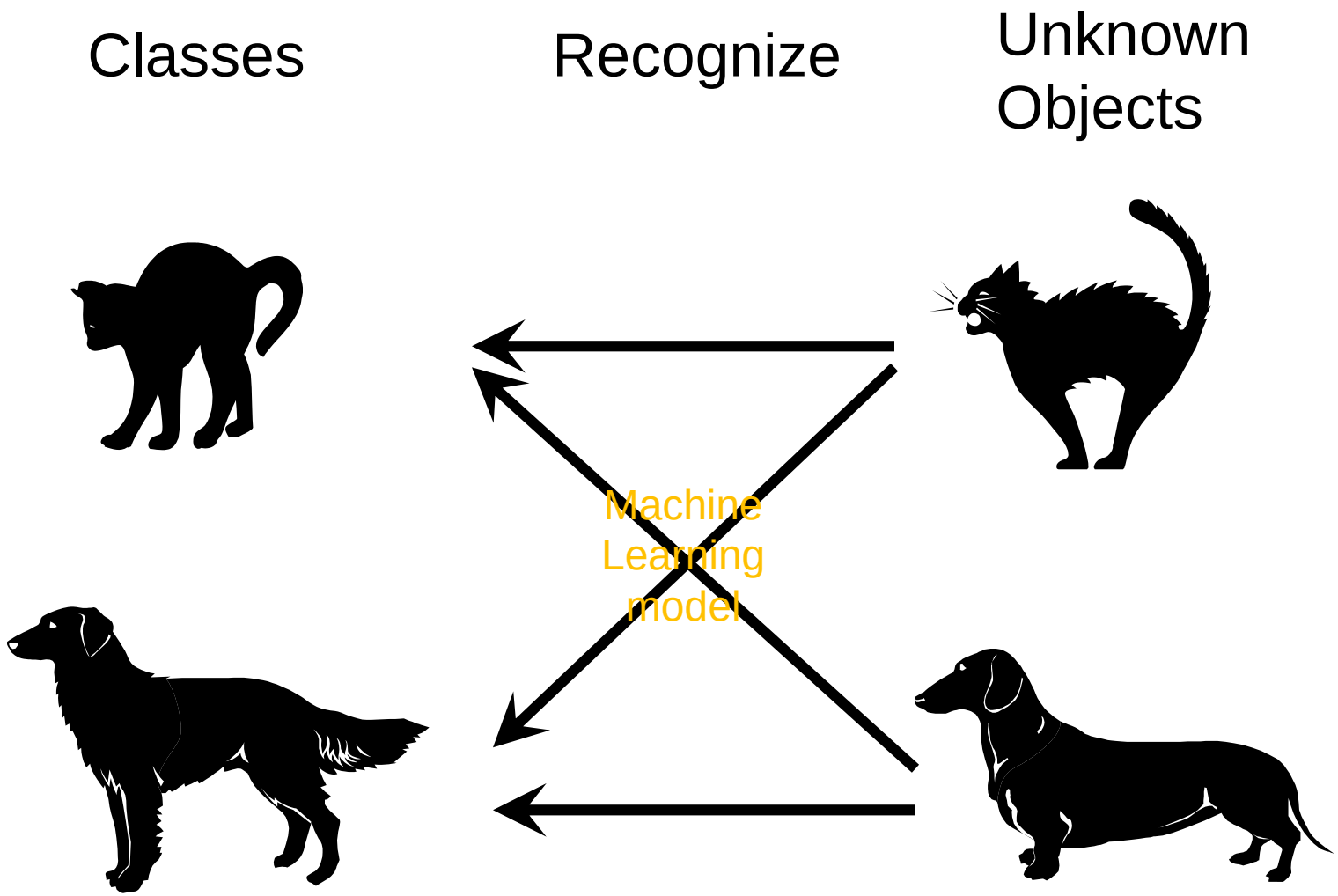


$f: v_i \rightarrow \mathbb{R}^d$

Feature representation (embedding)

$\mathbf{x}_i \in \mathbb{R}^d$

Traditional Pattern Recognition - Classification



Example: Two classes

Sea bass:

Salmon:

Generic System: Main steps

- Set up a camera, and take sample images
  - e.g., 100 samples of each class
- Note some physical differences between the two types of fish:
  - Length, width, lightness, number and shape of fins, position of the mouth, etc.
- Use these features or attributes to “explore” how to design our classifier.
- Problems:
  - Variations of images:
    - Lighting,
    - Position of the fish (rotation)
    - Background, etc.
- Questions:
  - How many features?
  - Which features?

Training dataset (labeled)

Samples or examples:

Class	length	lightness	width	…
salmon	5.4	2.3	16.2	…
salmon	8.2	4.8	18.3	…
salmon	6.0	5.3	19.0	…
...	…	…	…	…
…	…	…	…	…
…	…	…	…	…
sea bass	21.3	8.4	17.3	…
sea bass	24.9	5.0	21.5	…
sea bass	19.1	9.2	18.9	…
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

- Suppose we take *one* feature: the **length** of the fish
- Draw a histogram and design our classifier…

Class	length
salmon	5.4
salmon	8.2
salmon	6.0
...	…
…	…
…	…
sea bass	21.3
sea bass	24.9
sea bass	19.1
…	…
…	…
…	…

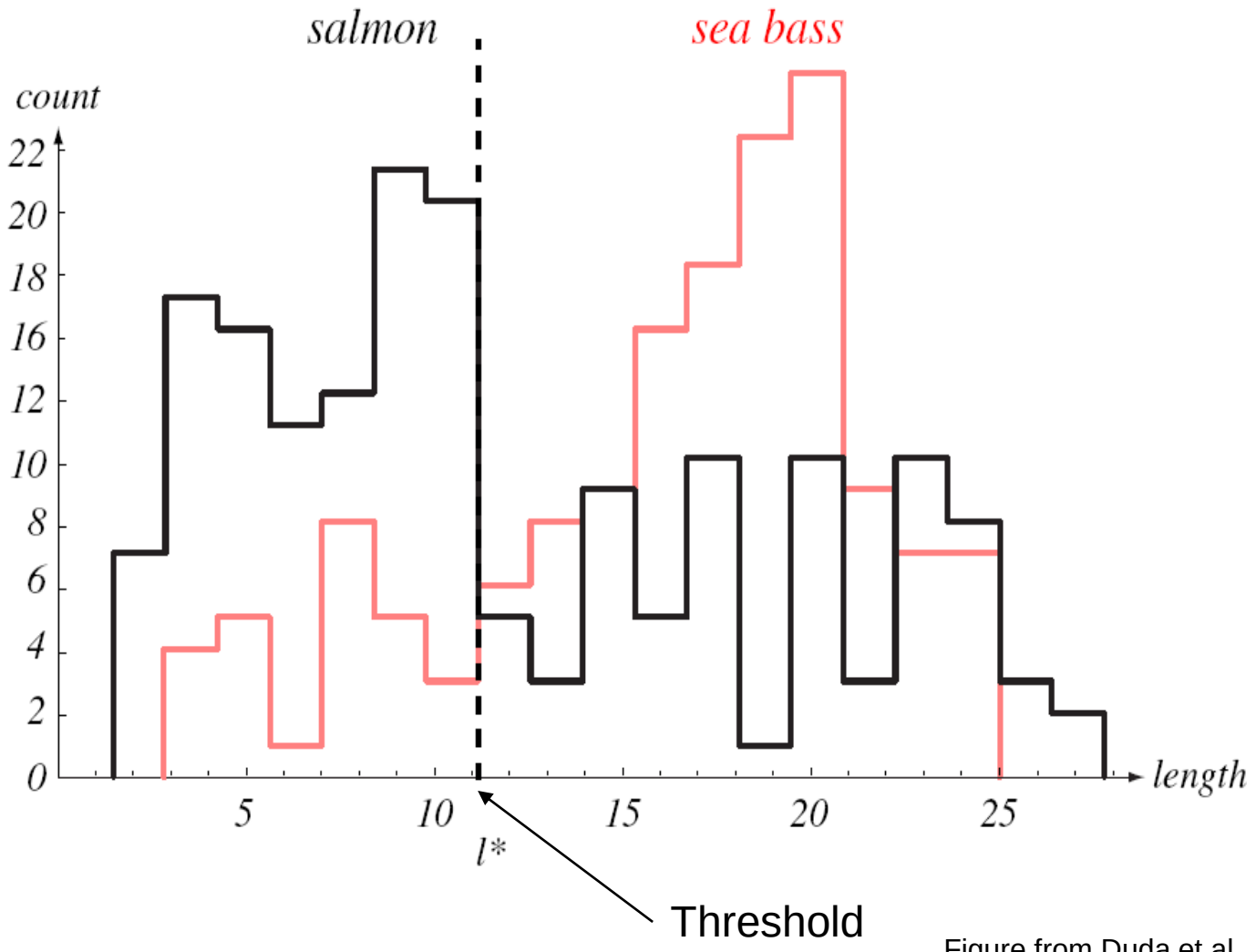


Figure from Duda et al.

- Lets take another feature: the **lightness** of the fish

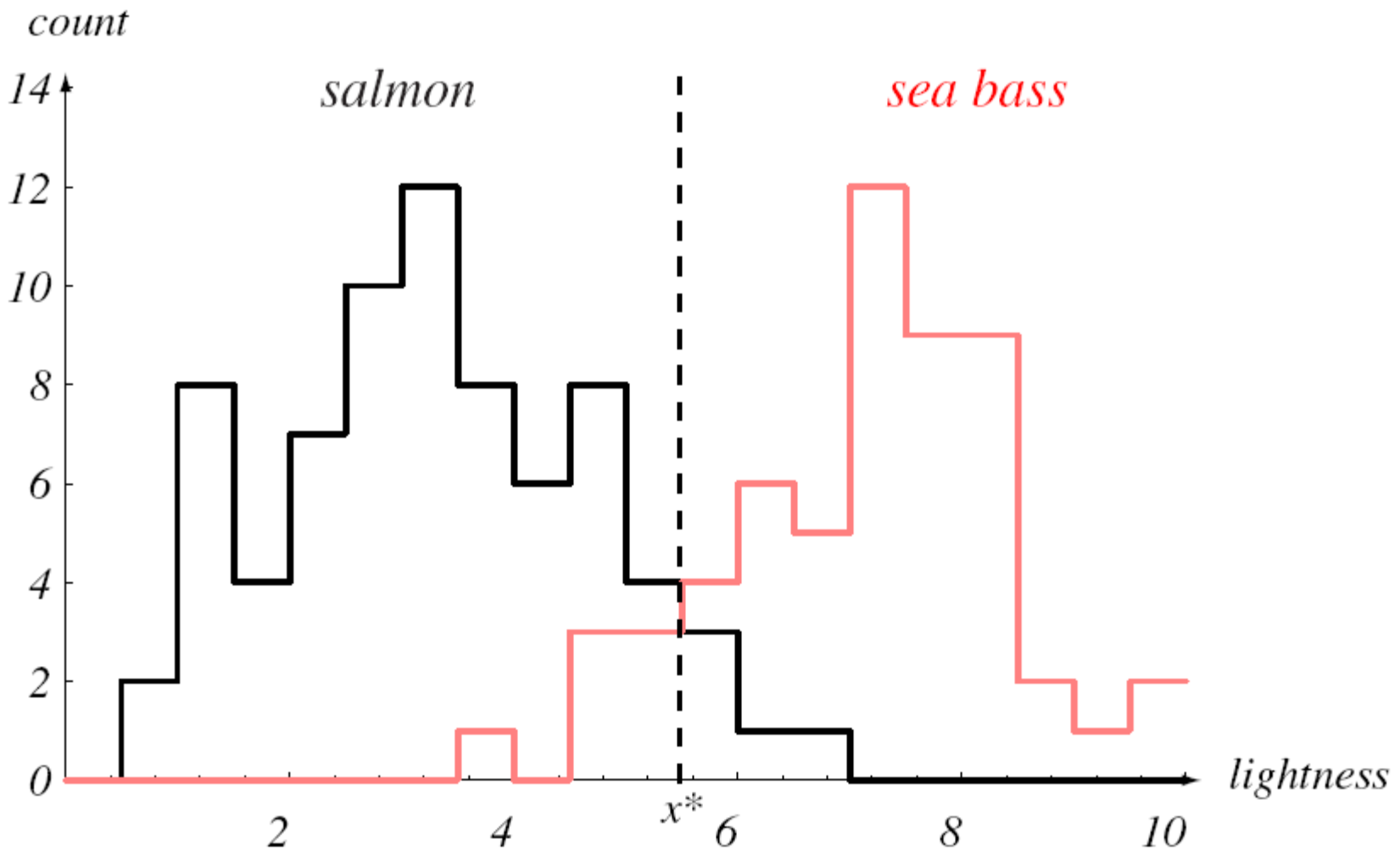


Figure from Duda et al.

Pattern Recognition: Decision Theory Approach

- Lets start with 2 classes
- Our task is to design a decision rule based on a
  decision boundary
- Decision rule:
  - Let *x* be an unknown object (represented by one feature)
    if $x < x^*$
    x is a **salmon**
    else
    x is a **sea bass**
- Ties are resolved arbitrarily
- There may be a cost associated with the decision
  - Decision theory = probability theory + utility theory
  - Assume no costs for now

- One feature may not be good enough… and,
- We may want to take advantage of having other features available…
- Lets take **width**, x_1, and **lightness**, x_2.
- So, each object, **x**, is represented by a vector:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Our feature space now is two-dimensional, i.e. the *plane*.
- Thus, a *threshold* is not good enough, and
- we want to take advantage of the "correlation" of the features. How?

- Our decision boundary could be a “linear combination” of the features:

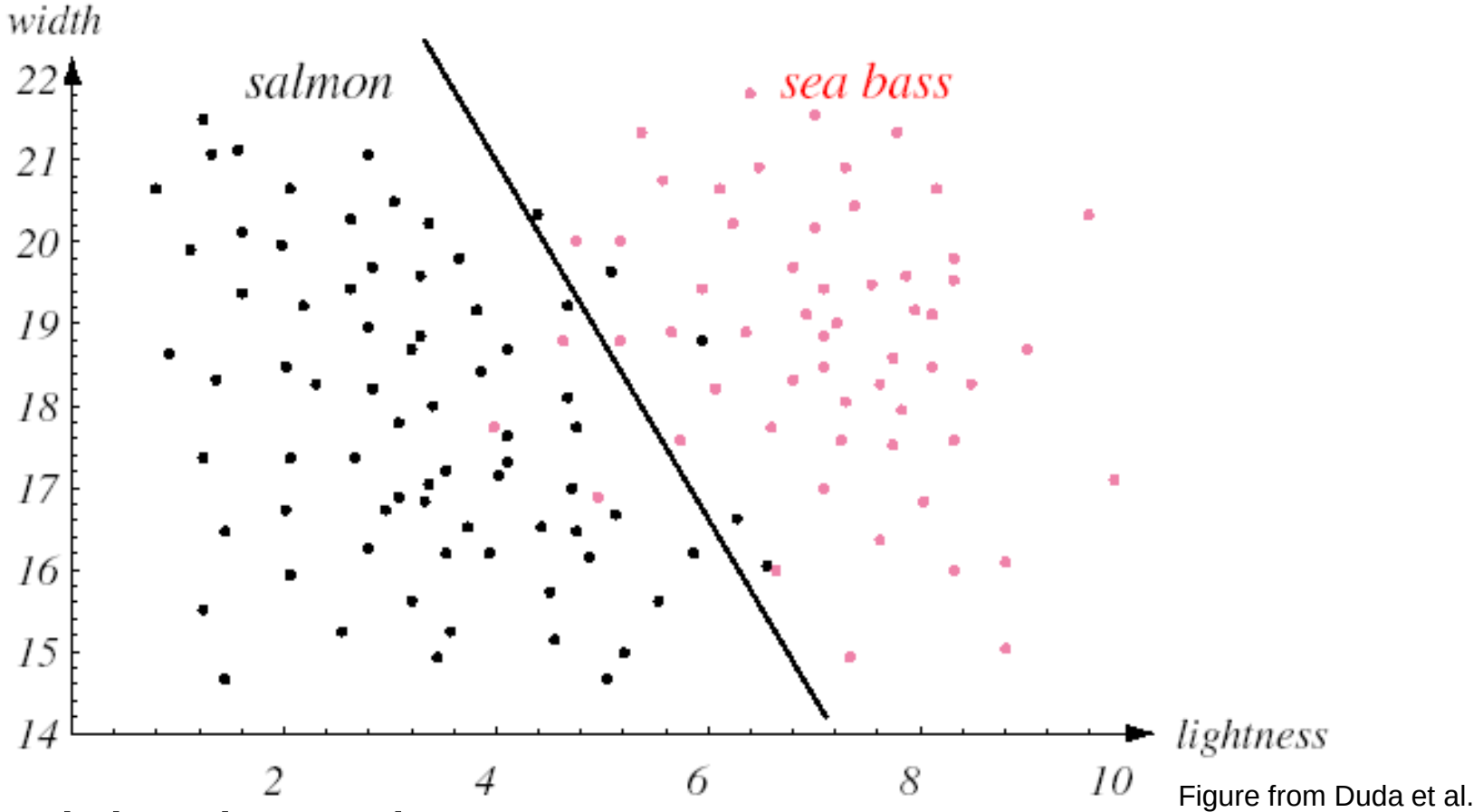


Figure from Duda et al.

- Decision boundary:

  a straight line: $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$

Decision Rule

- Let **x** be a *new* object (represented by *two* features)

  if $\mathbf{w}^t \mathbf{x} + w_0 < 0$

  x is a **salmon**

  else

  x is a **sea bass**
- Ties are resolved arbitrarily
- **w** is a 2D vector, and w_0 is a threshold
- Questions:
  - Is the new *linear* classifier better than the *threshold*?
  - Can we do better with *more than 2* features?
  - Can we do better with *these 2* features?
  - How do we deal with more than 2 classes?
  - What about 1 class?

- Lets design a “more complex” (polynomial?) classifier:

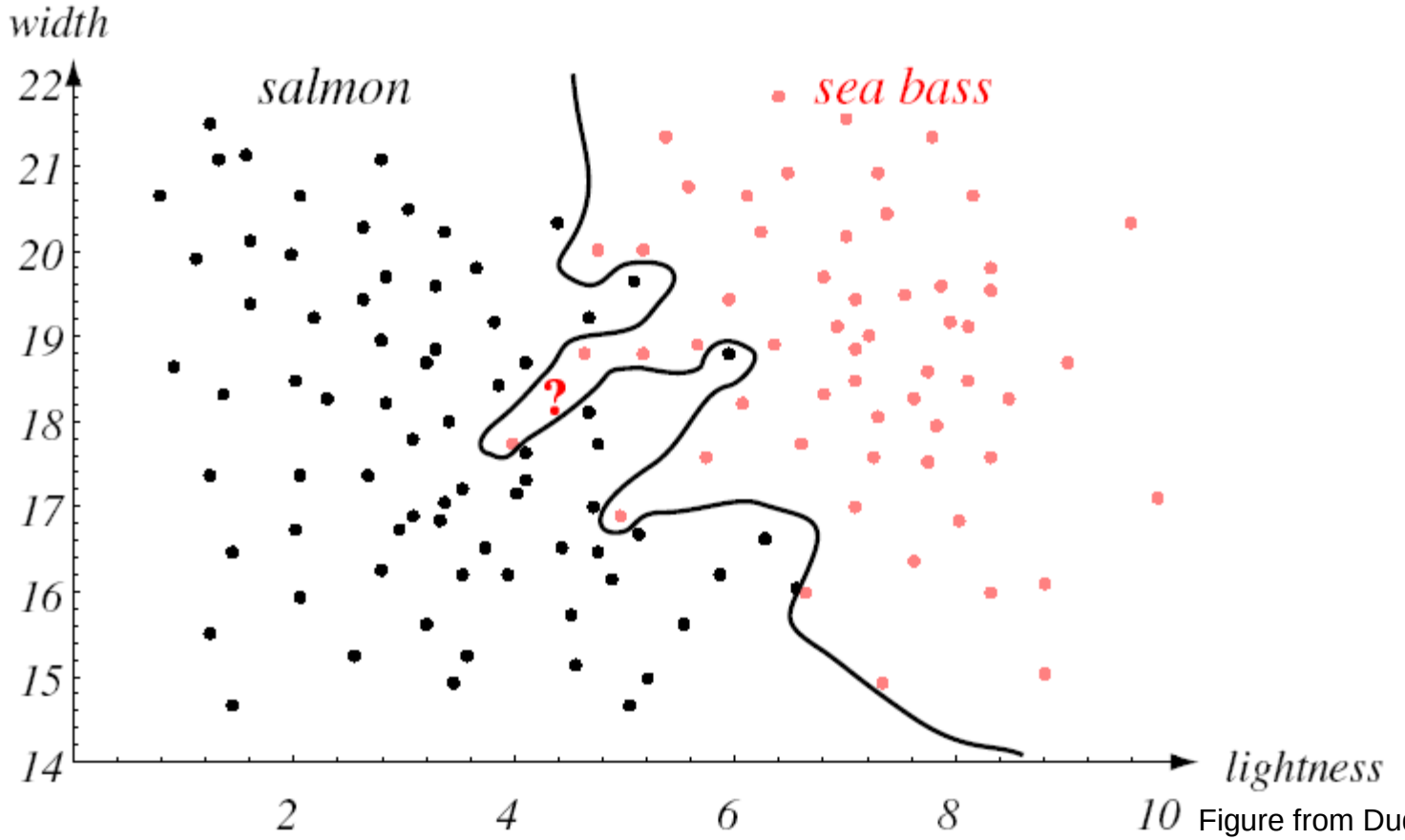


Figure from Duda et al.

- Is the new classifier better than the linear one?
- Is there a (even) better classifier?

- What if we design a **quadratic** classifier?

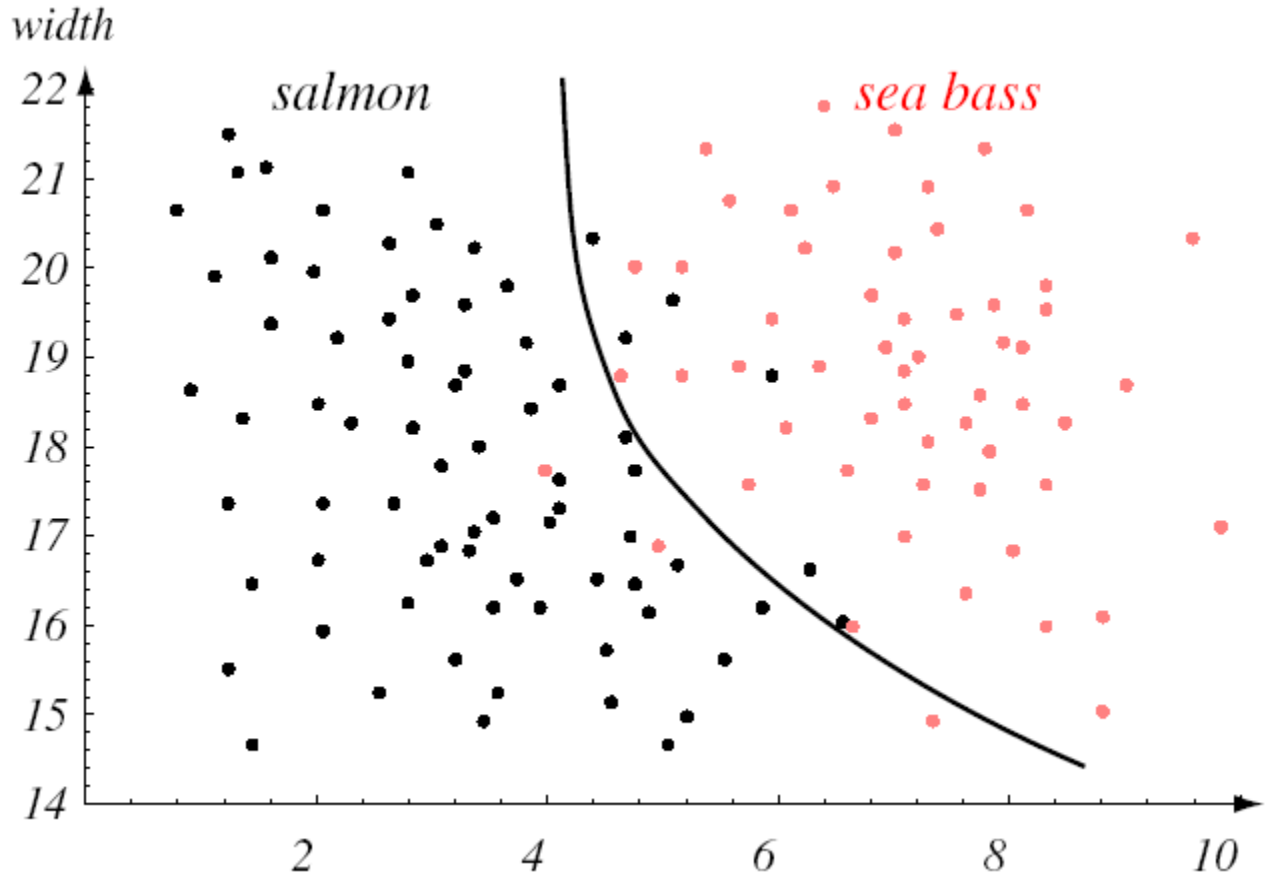


Figure from Duda et al.

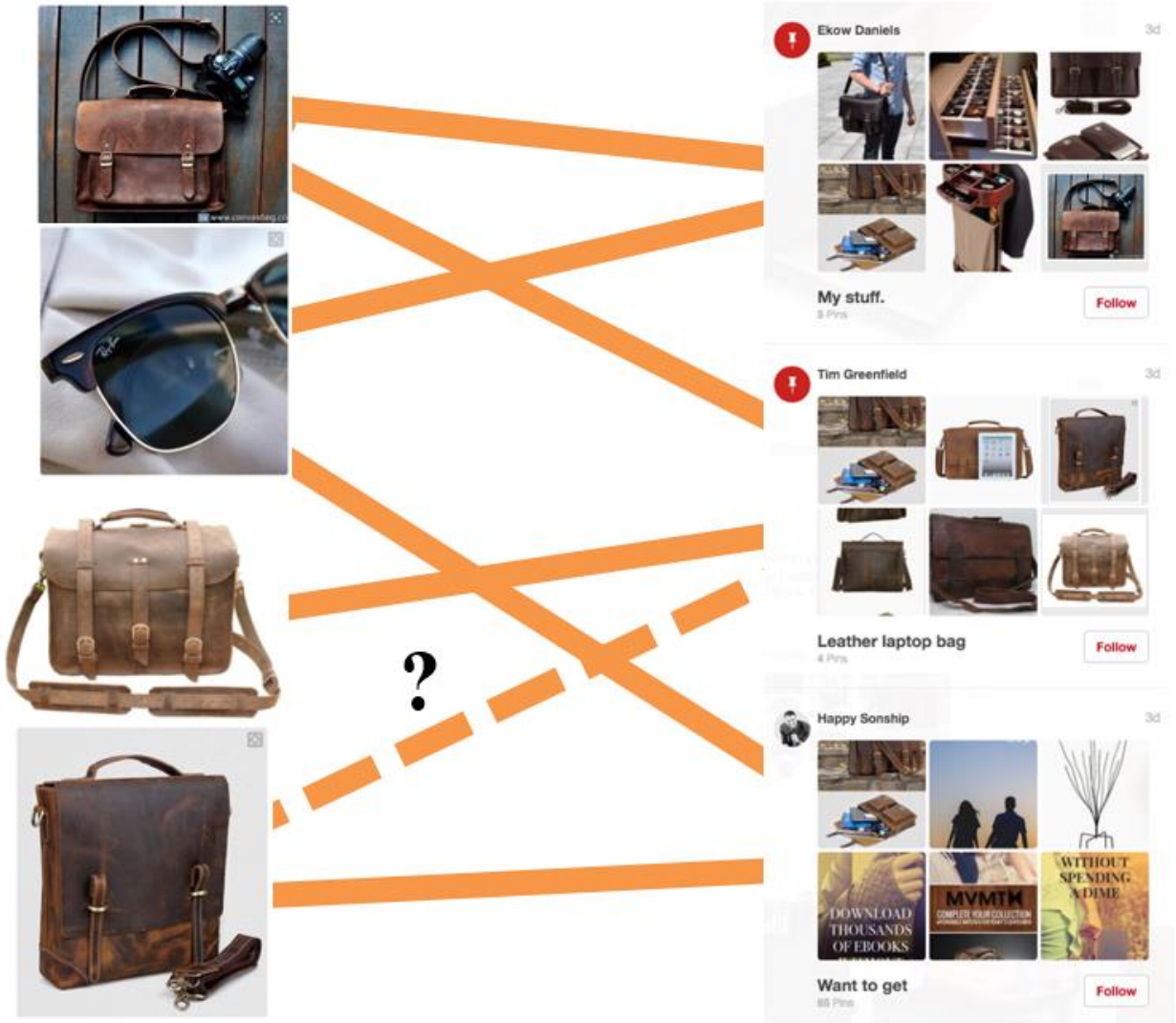
- Is this classifier better than the polynomial one?
- This one may “generalize” better for new samples

Applications of Machine Learning

- Document classification
- Prediction of proteins by function
- Prediction of PPI
- Tumour classification: Gleason score
- Email classification: spam filter
- Speech recognition
- Identification of cancer biomarkers
- User authentication via behavioural features
- Fake review detection
- Hand written digit recognition
- Face recognition
- Speech recognition
- Stock prediction
- Weather prediction
- Community detection in Networks
- Content recommendation …
- etc, etc, etc

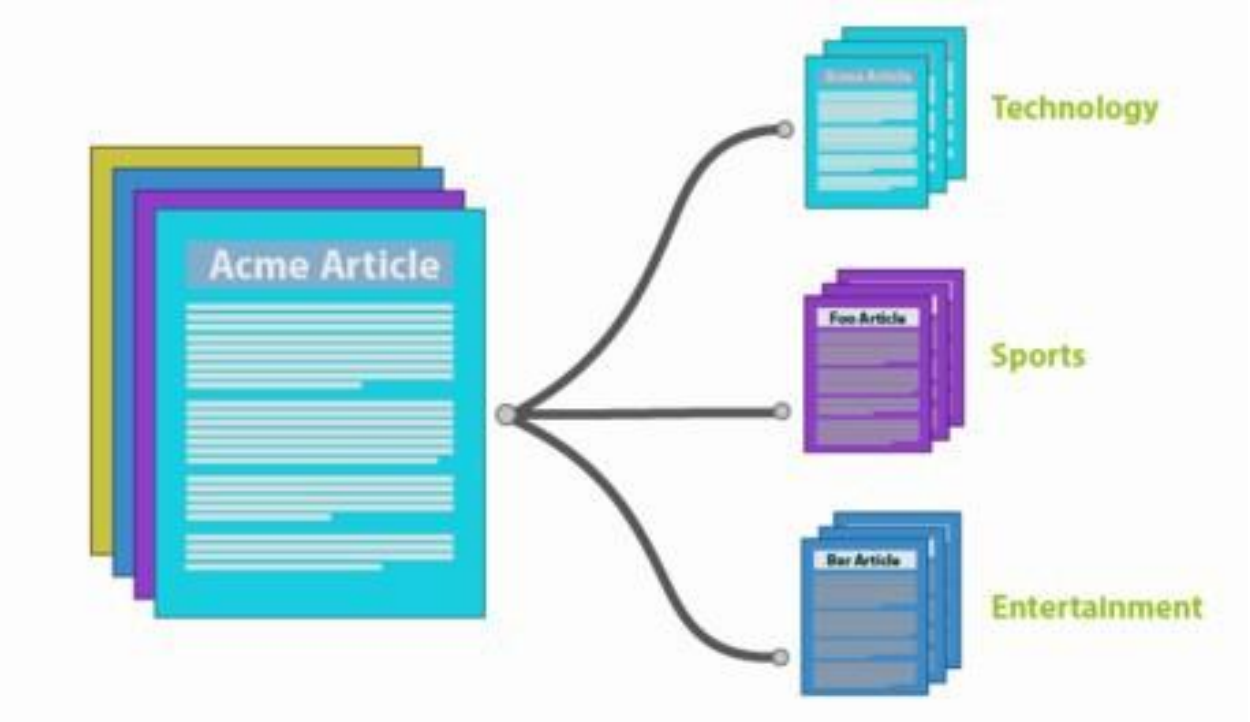
Link Prediction

- Content recommendation
- Done via link prediction in a network



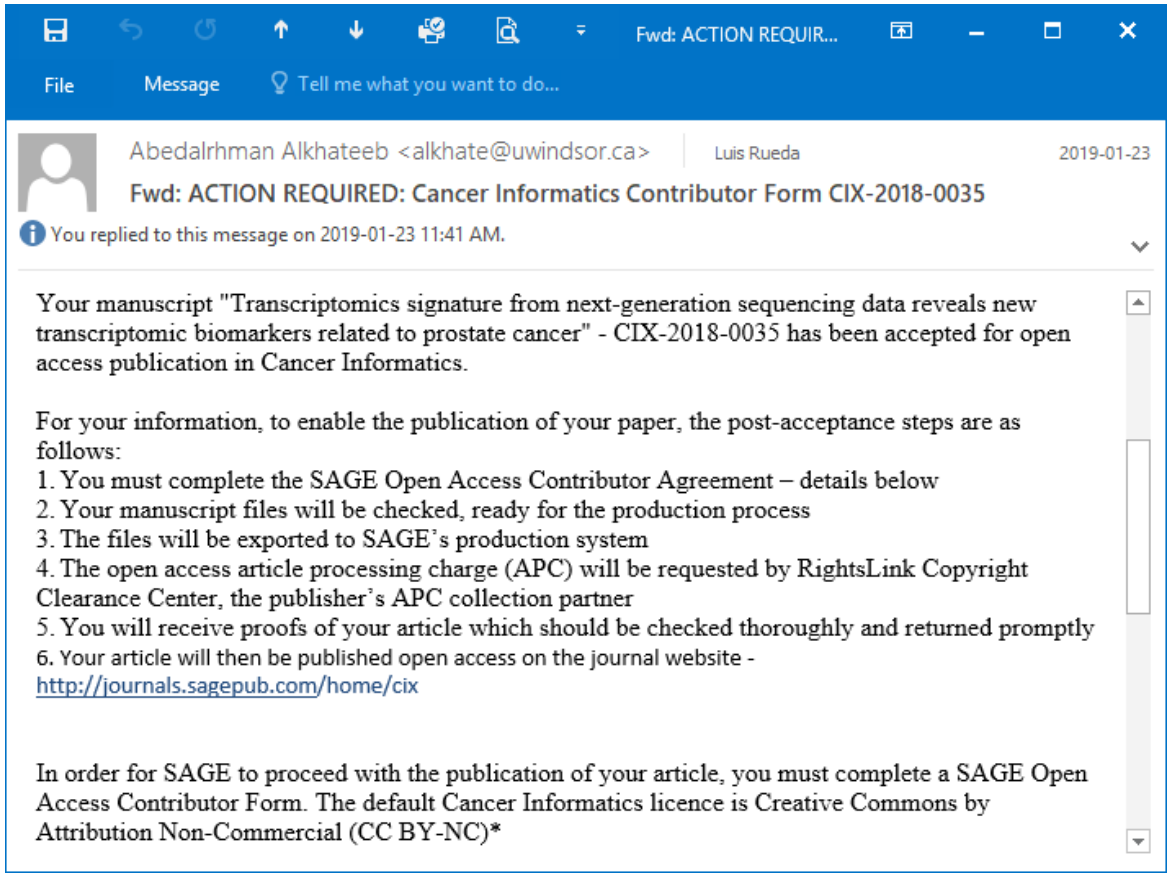
Document Classification

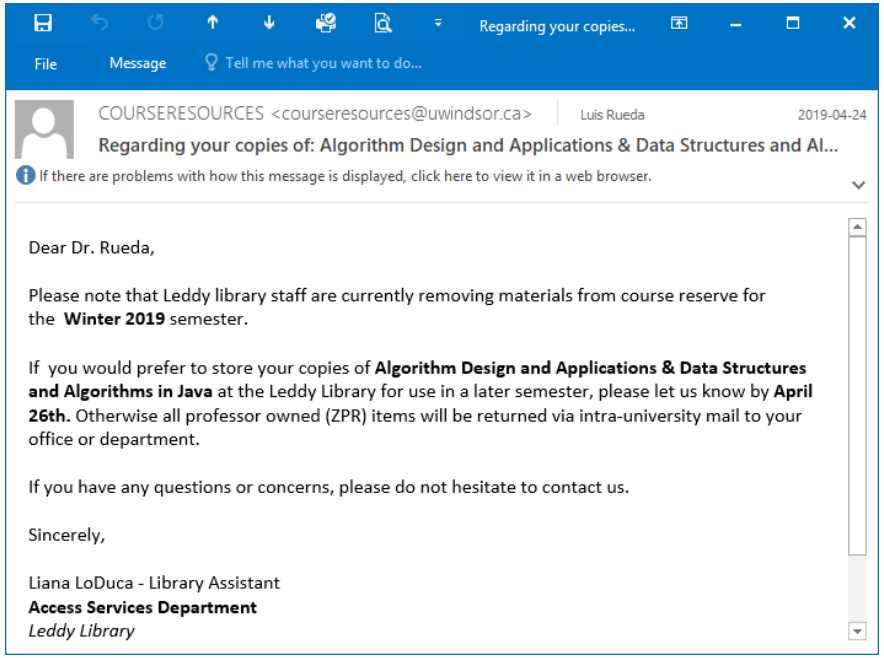
- Based on type of document
- Use of text, graphics
- Paragraph/sentence level
- Semantics
- NLP

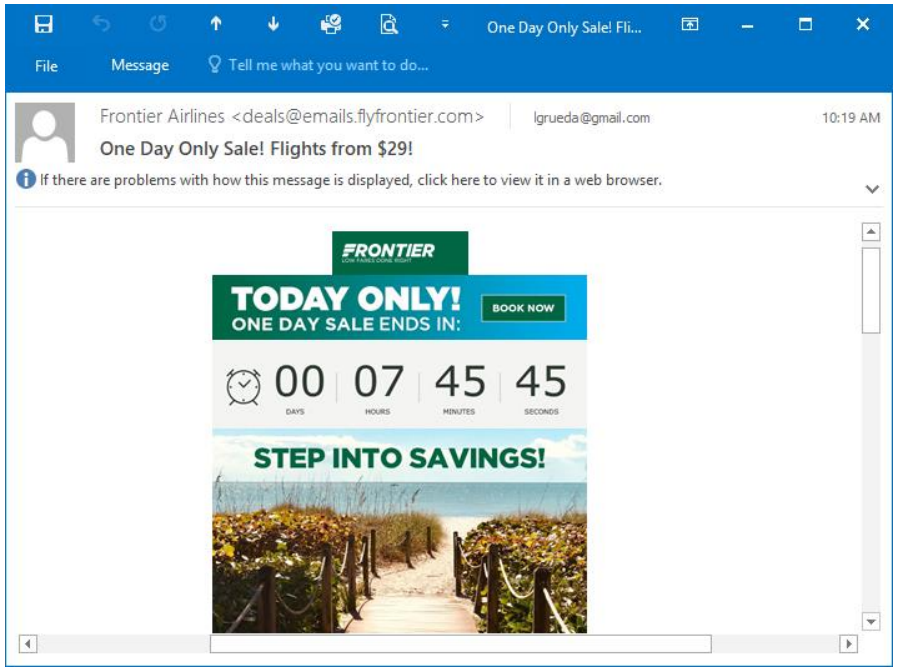


Spam Filter

- E-mail classification
- 2 classes: spam/ham
- Other classes: security threat, highly important

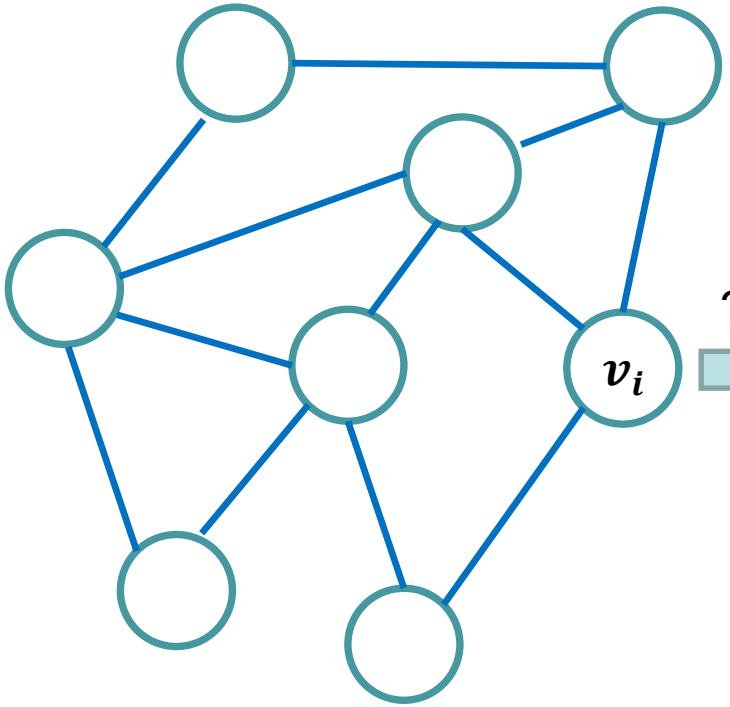




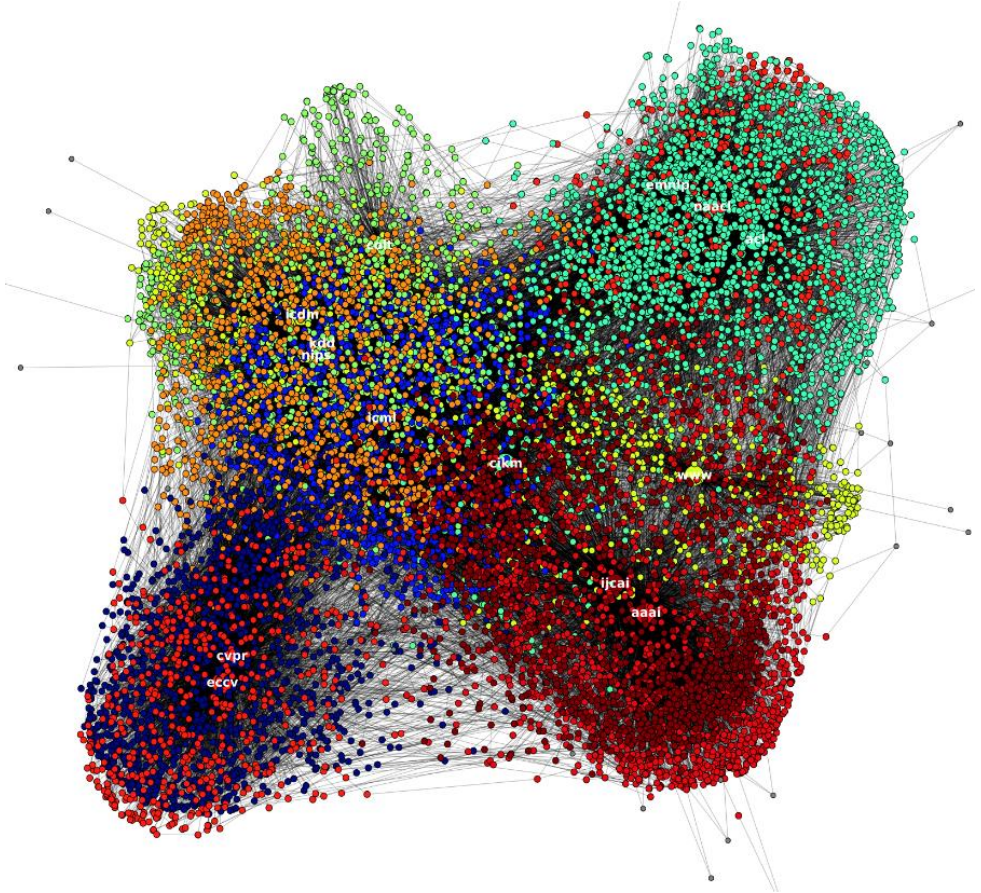


Graphs – Node Classification

- Classify nodes on basis of features



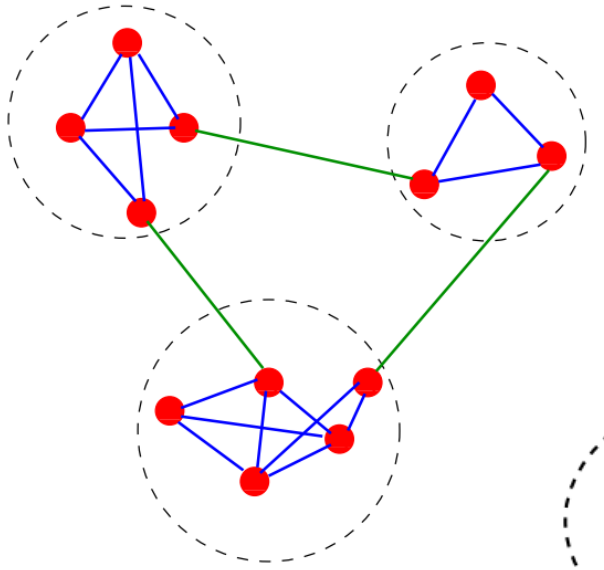
- Example: Citeseer; find topic of paper (node)



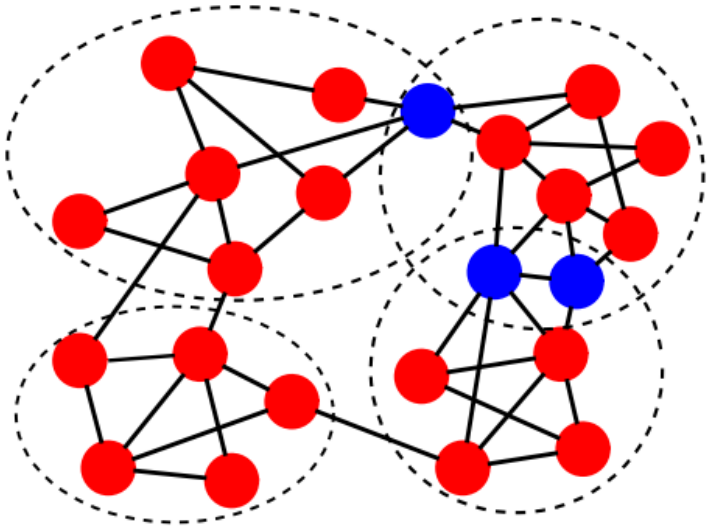
Community Detection in Graphs

- Given a graph
- Identify modules and their boundaries
- Applications
  - Social networks
  - Biological networks
    - E.g, proteins that are in the same pathway
  - Web mining
    - Customers with similar profile
    - Recommendation systems

Graph with 3 communities [3]:



Overlapping communities [3]

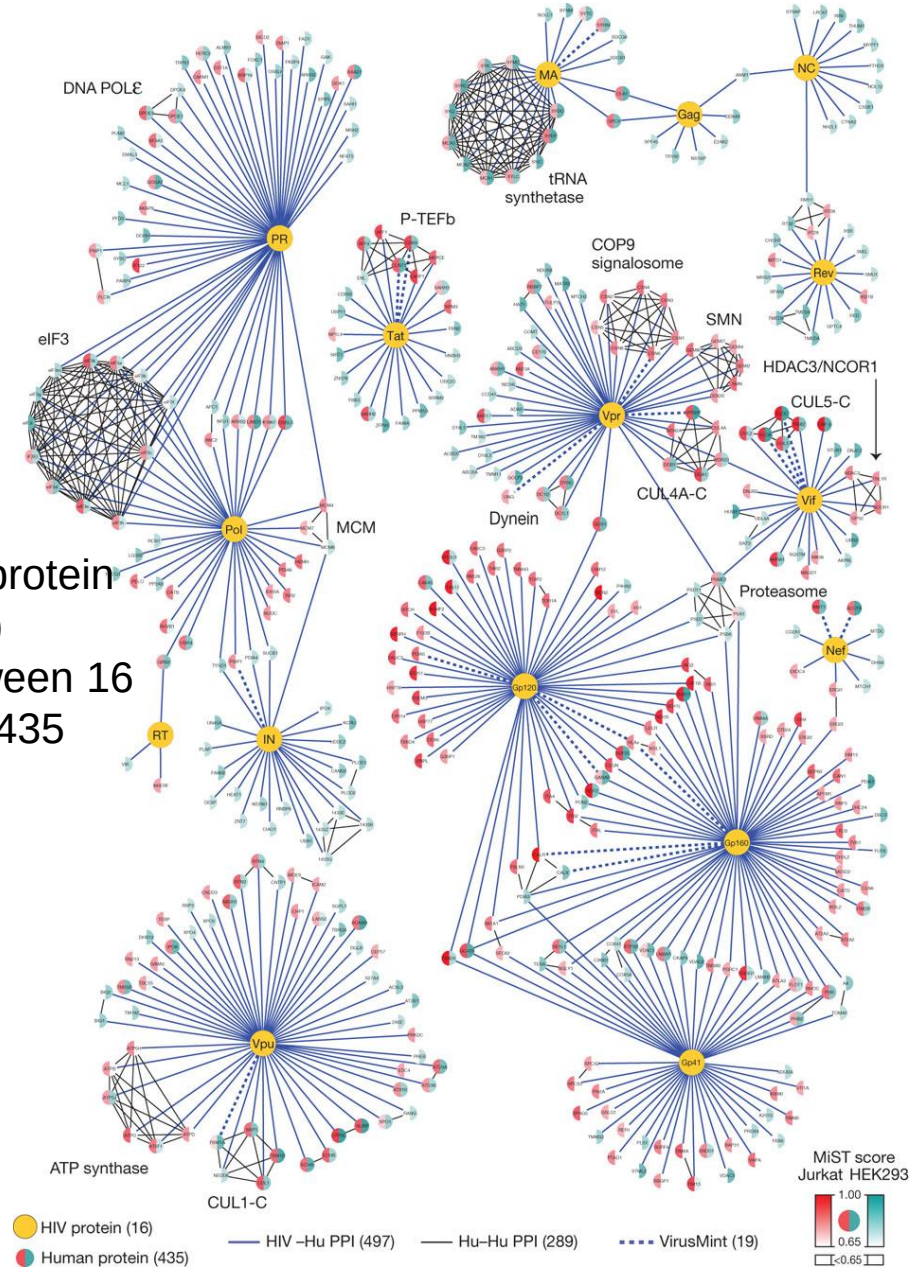


Protein Interaction Networks

Important problems in PPI networks

- Connected components:
  - Allows to find sub-networks of proteins with related functional activity
- Hubs:
  - Vertices that are connected to a large number of other vertices
- Clusters
  - Find groups of proteins interacting with each other
  - Proteins in a group may have related functional activity
- Visualization
  - Visualizing PPI networks in a way to better understand biological processes
- Alignment of graphs
  - Allows to compare two or more different networks
- Many of these problems discussed in bioinformatics/data mining courses

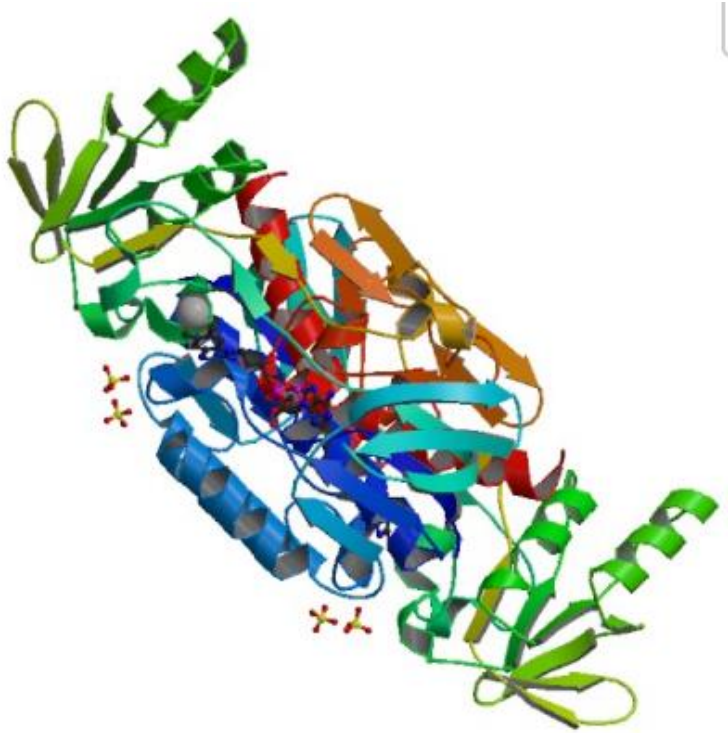
Example:
497 HIV–human protein interactions (blue) representing between 16 HIV proteins and 435 human factors [4]



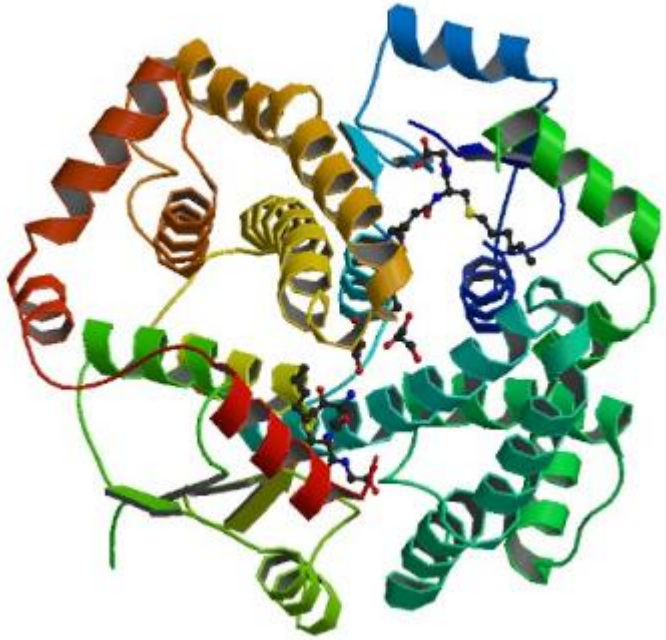
Enzyme Classification - PDB

Isomerase: 8TIM

Oxidoreductase: 2FBS

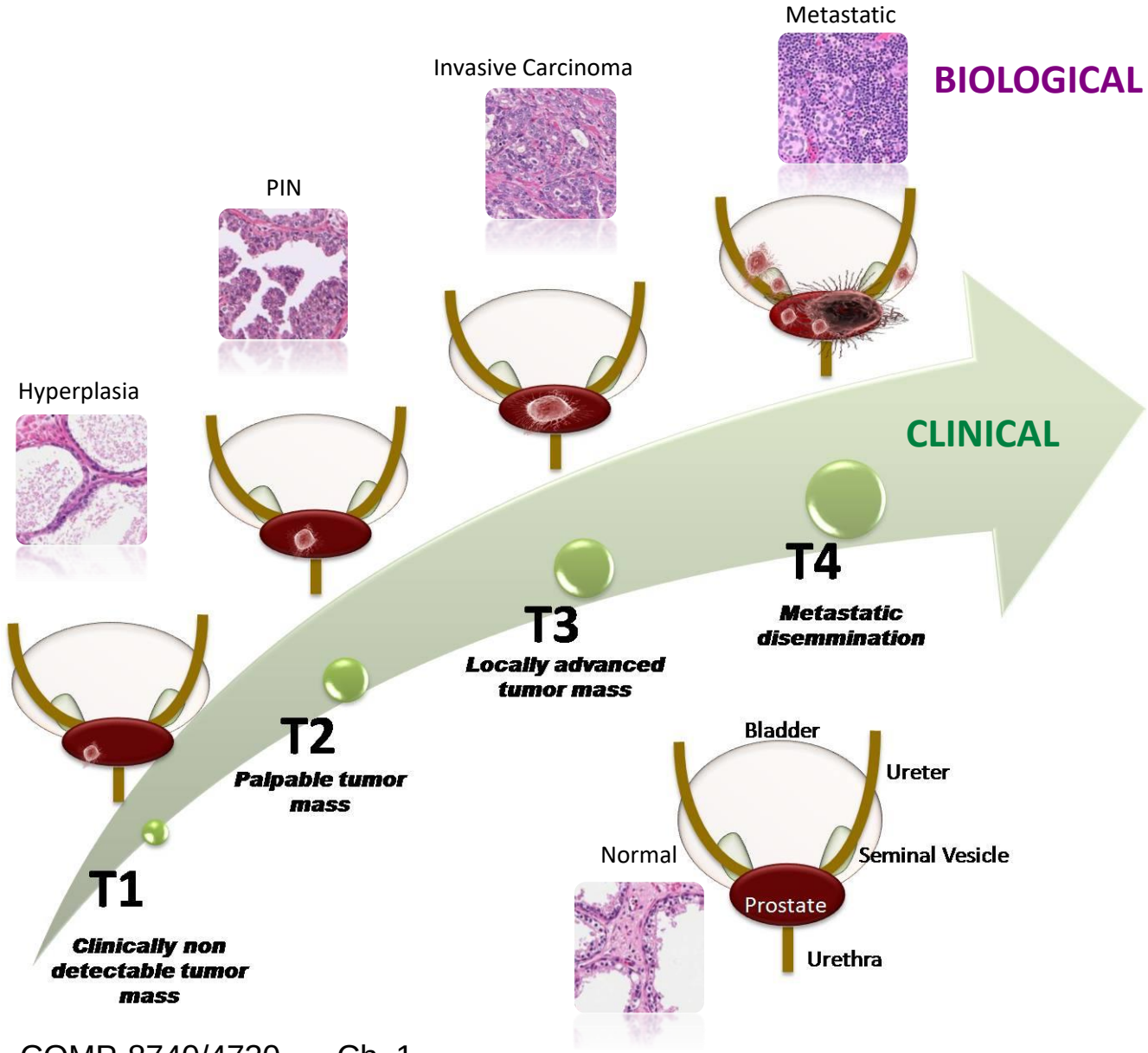


Transferase: 1K3Y

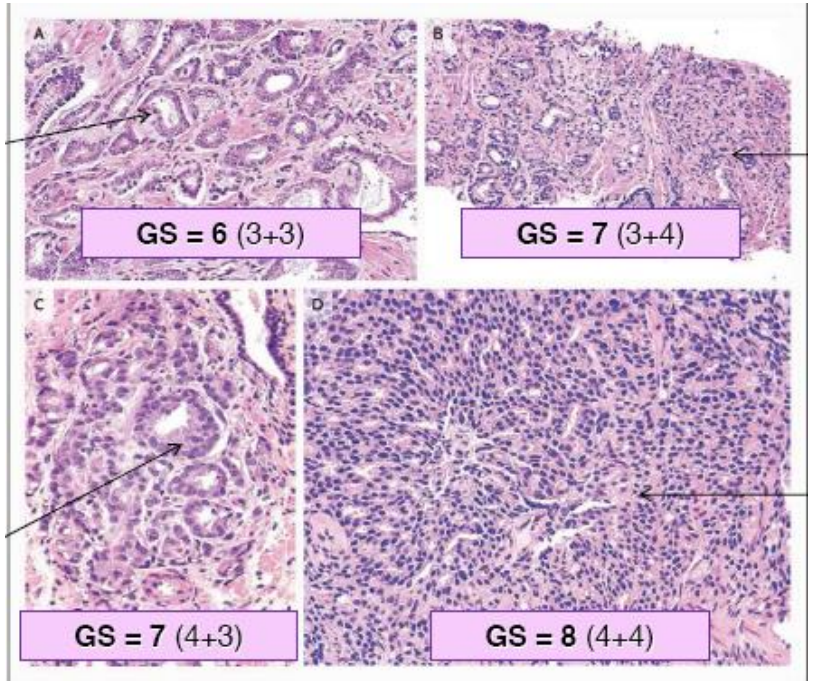


Progression of Prostate Cancer

Prostate Stage	Description
T1c	Cancer is detected using a needle biopsy. Not by imaging.
T2a	<= half of one of the prostate glands of two lobes
T2b	>= half of than half of one lobe, but not both.
T2c	Tumor in both lobes
T3b	The Tumor has invaded one or both of the seminal vesicles.
T4	Tumor rises to another organs.



Grading/Location of Prostate Tumours



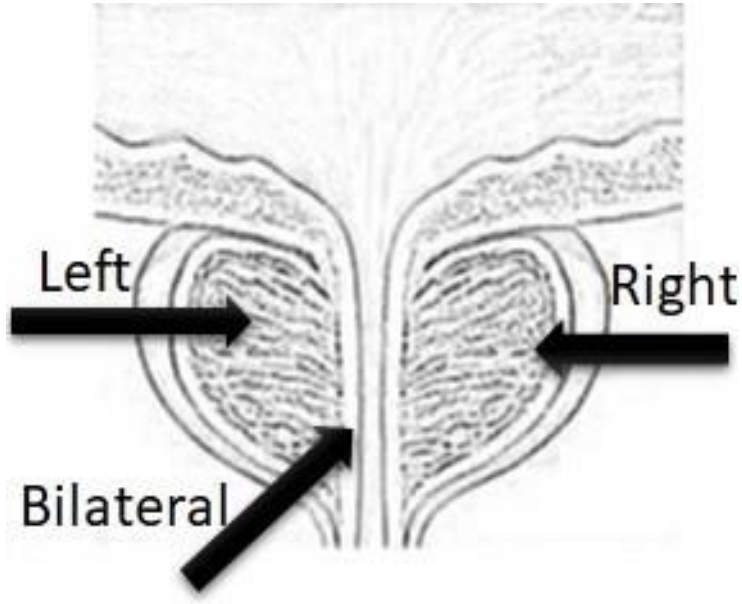
Grouping				
336	347	437	448-538	459-549

Grade of 1 is assigned to normal prostate tissue, while a very abnormal prostate tissue is graded as 5.

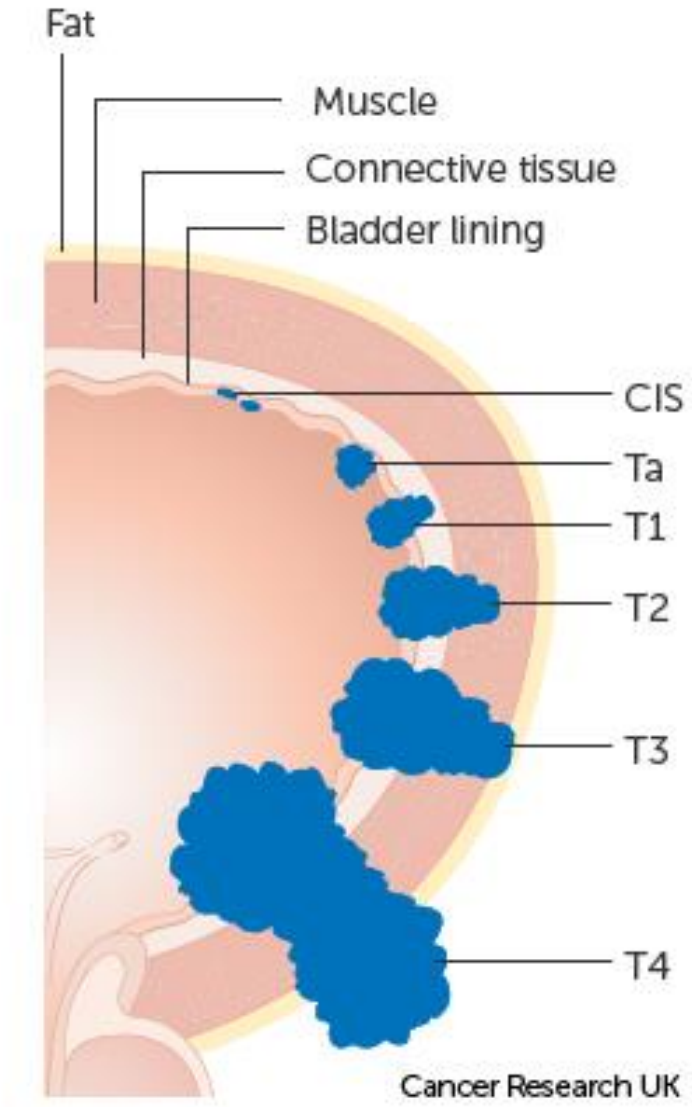
Since prostate cancer often have areas with different grades, a grade is assigned to the 2 areas that make up most of the cancer.

The higher the Gleason score, the more likely it is that the cancer will grow and spread quickly.

Location of the tumour:



Muscle Invasive Bladder Cancer (MIBC)



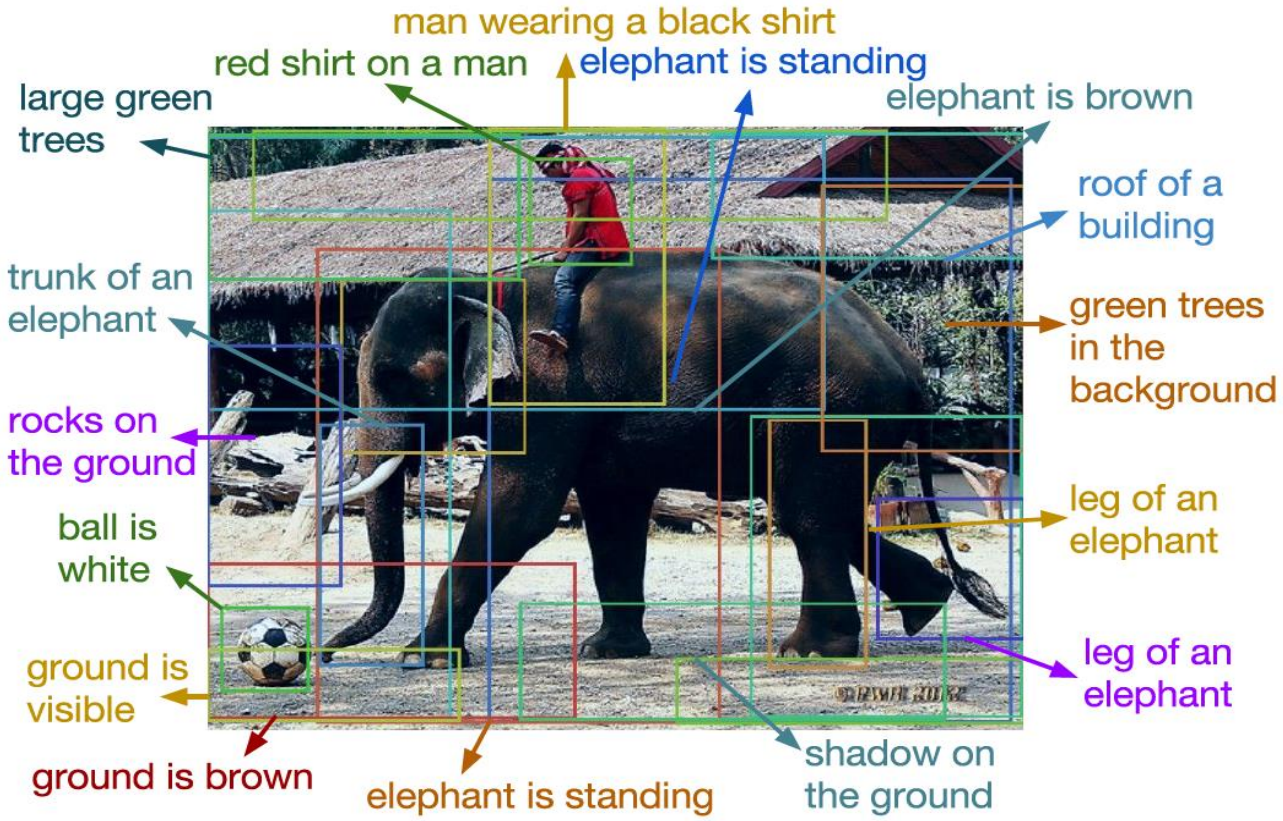
- Patients are initially diagnosed with non-muscle invasive bladder cancer (NMIBC), but 10-30% progress to muscle-invasive bladder cancer (MIBC) even with treatment.
- Copy number alteration(CNA) is a type of structural variation in the genome.
- Machine learning techniques are applied to identify predict MIBC
  - Can also identify genes with CNAs (biomarkers) that can be used to predict patients with MIBC against NMIBC.

Deep Learning

- Neural networks
  - Not a new theory
  - Date back from 1940s
  - McCulloch/Rosenblatt
- Emerged in the 1980s:
  - multi-layer perceptron
  - Convolutional NN
- Declined due to emergence of SVM
- Recently emerged due to:
  - Big data
  - Efficient hardware (GPUs, multi-processing, multicore)
  - Applications:
    - ➢ many

Example:

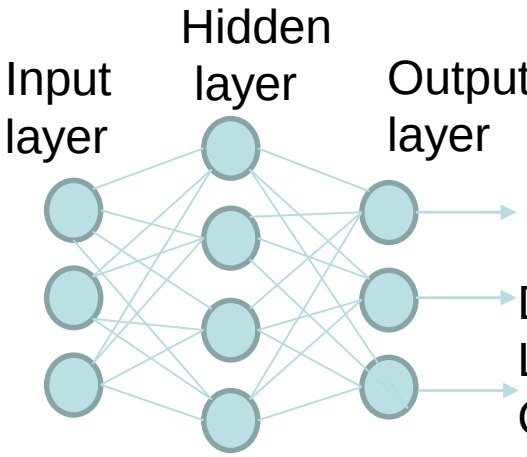
- DenseCap: From images to natural language
- Classification, object detection, to full sentences in natural language
- Uses convolutional and recurrent NN



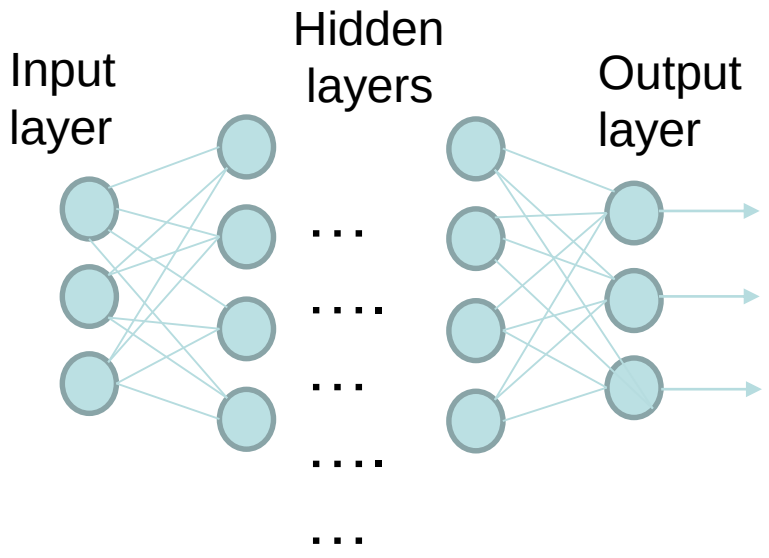
J. Johnson et al., DenseCap, IEEE CVPR 2016

Deep Learning - Artificial Neural Networks

- Deep Learning is based on deeper artificial neural networks



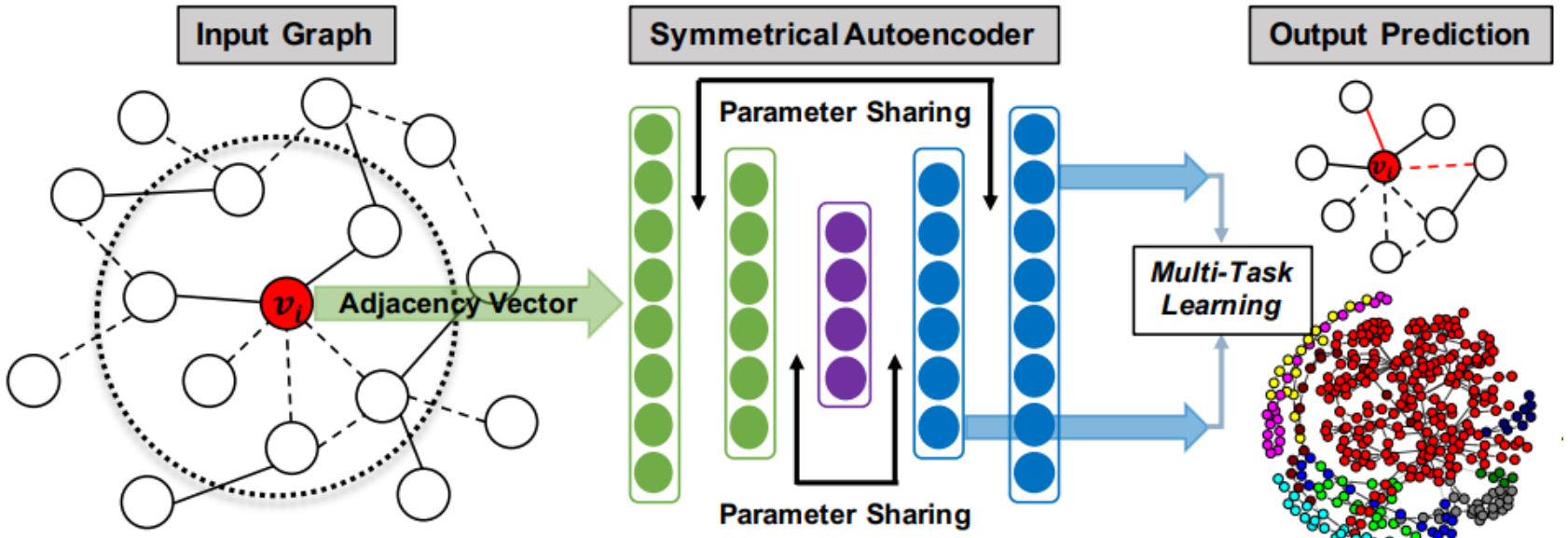
DenseCap: Fully Convolutional Localization Networks for Dense Captioning



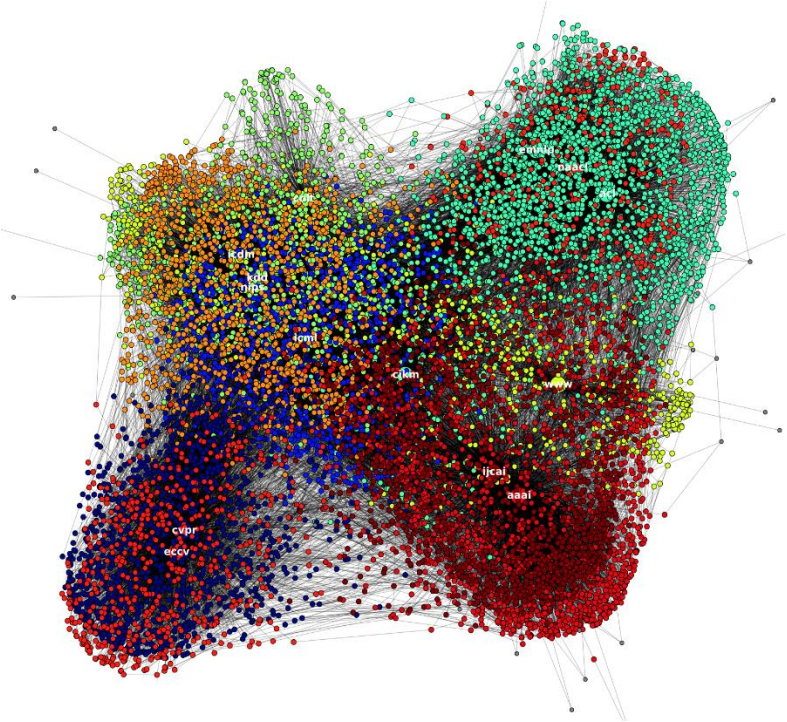
- Due to the advancement in computing resources (CPUs, GPUs, memory, etc..) ➔ Deep learning becomes feasible.
- More hidden layers ➔ More optimization & learning ➔ revealing more of the intrinsic features.

Dimensionality Reduction - Autoencoder

- Link prediction and node classification via deep Autoencoder
- Application: node/link classification/clustering in Citeseer



Citeseer:



- See ref [2].

References

1. R. Duda et al, Pattern Classification, 2nd Edition, Wiley, 2000.
2. P. Tran, “Multi-Task Graph Autoencoders.” Workshop on Relational Representation Learning, NIPS 2018, Montréal, Canada.
3. S. Fortunato et al. “Community detection in networks: A user guide.” Physics Reports, Elsevier, 2016, pages 1-44.
4. S. Jager et al. Global landscape of HIV–human protein complexes. Nature, 481:365–370, 2012.
5. W. Hamilton et al. Representation Learning on Graphs: Methods and Applications. IEEE Data Eng. Bull. 2017.