

INF 554
Machine Learning and Deep Learning

Prof. M. Vazirgiannis

Labs/TDs: Prof. D. Buscaldi, Dr. J. Lutzer, S. Kosma

Data Science and Mining Team, LIX
<http://www.lix.polytechnique.fr/dascim/>

Sept 19, 2022

Course Syllabus

- **General Introduction to Machine Learning**

- Machine Learning paradigms
- The Machine Learning Pipeline

- **Supervised Learning**

- Generative and non generative methods
- Naive Bayes, KNN and regressions
- Tree based methods

- **Unsupervised Learning**

- Dimensionality reduction
- Clustering

- **Advanced Machine Learning Concepts**

- Regularization
- Model selection
- Feature selection
- Ensemble Methods

Course Syllabus

- **Kernels**
 - Introduction to kernels, Support Vector Machines
- **Neural Networks**
 - Introduction to Neural Networks
 - Perceptrons and back-propagation
- **Deep Learning I**
 - Convolutional Neural Networks
 - Recurrent Neural Networks
 - Applications
- **Deep Learning II**
 - Modern Natural Language Processing
 - Unsupervised Deep Learning
 - Embeddings, Auto-Encoders
- **Graph based ML**
 - Kernels, Node/Graph embeddings

Course Logistics

- Class: 14:00 – 16:00 – presentiel **PLEASE WEAR MASKS in CLASS**
- Labs: 16:15 – 18:30 - presentiel **PLEASE WEAR MASKS in CLASS**
- Interaction/Q&As
 - A **slack channel** was set up for individual questions to course/lab teachers:
<https://tinyurl.com/3mxbk6p4> - **please JOIN**
 - Different channels within will be set up for course/labs/other-admin issues
- **Install the software requested (last versions of Anaconda)!**
- Read carefully our announcements
- For **IPP master students & others** – please fill in the registration form:
<https://tinyurl.com/353ft47k>
 - Get access to the material and course communication
 - We are able to communicate your results to your home institution

Course Logistics

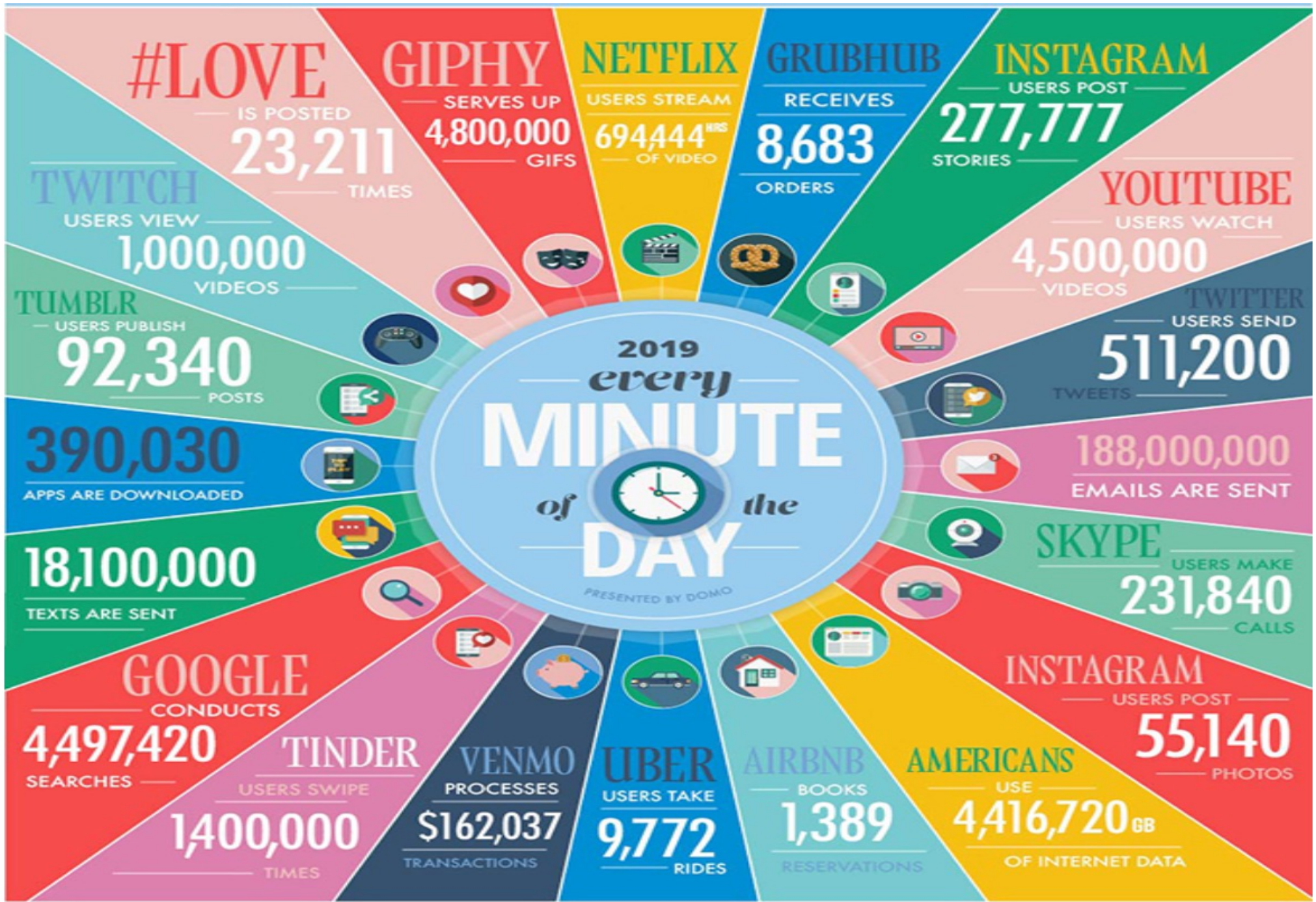
- **Magistral classes** 14:00-16:00 - Amphi Cauchy
- **Labs/TDs :**
- **Monday (16:15 - 18:15)**
 - Amphi Painlevé (Gr1), Amphi Poisson (Gr2), Amphi Sauvy (Gr3).
- **Wednesday (14:00 - 16:00)**
 - PC11 (Gr4), PC22 (Gr5), PC12 (Gr6). 1. Painlevé, 2. Poisson, 3. Sauvy.
 - **Distribution based on your surname initial letter:**
 - **1: A-G, 2: H -M, 3:N-Z**
 - If need (COVID) we will go online sometimes
- **Evaluation**
 - Assignment (A) - an individual take-home assessment handed out on Monday 3rd October with deadline on Monday 17th October
 - Course project (CP) – Kaggle data challenge – dates (tentative) handed out 7/11, submission deadline: 12/12. oral assessments in the last week of
 - Details on the slack channel.
 - Grading scheme
 - $\text{Final Grade} = A^* \sim 20\% + CP^* \sim 80\%$
- **Course/Lab Material:** @Moodle INF554 –2022

Suggested textbooks

Suggested textbooks

- **Foundations of Machine Learning**
Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar,
MIT Press, 2012.
- **Learning from Data** – Y. Abu-Mostafa, M. Magdon-Ismail,
Hsuan-Tien Lin
- **Pattern Recognition and Machine Learning** - Hardcover –
October 1, 2007, Christopher M. Bishop

+ course note/slides on advanced issues

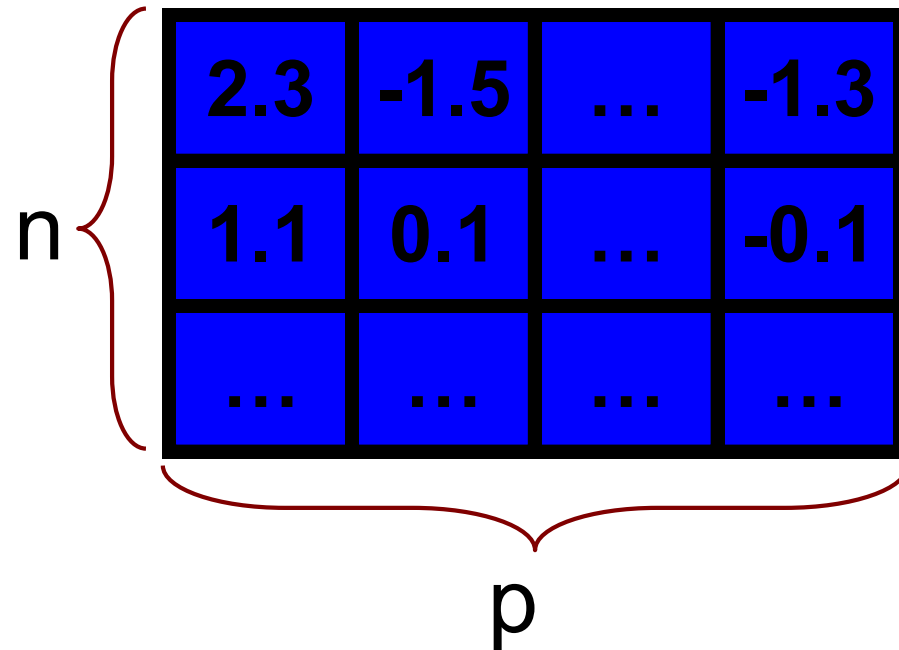


<https://www.domo.com/learn/data-never-sleeps-7>

Data are heterogeneous

- Data are considered as valuable resource
 - User behavior, Queries
 - User generated content in Social networks
 - Experimental/Scientific data – genome, ...
- Traditional: numerical, categorical, or binary
- Text: emails, tweets, *New York Times* articles
- Geo-based location data
- Network, Sensor data
- Images, Video

Flat File or Vector Data

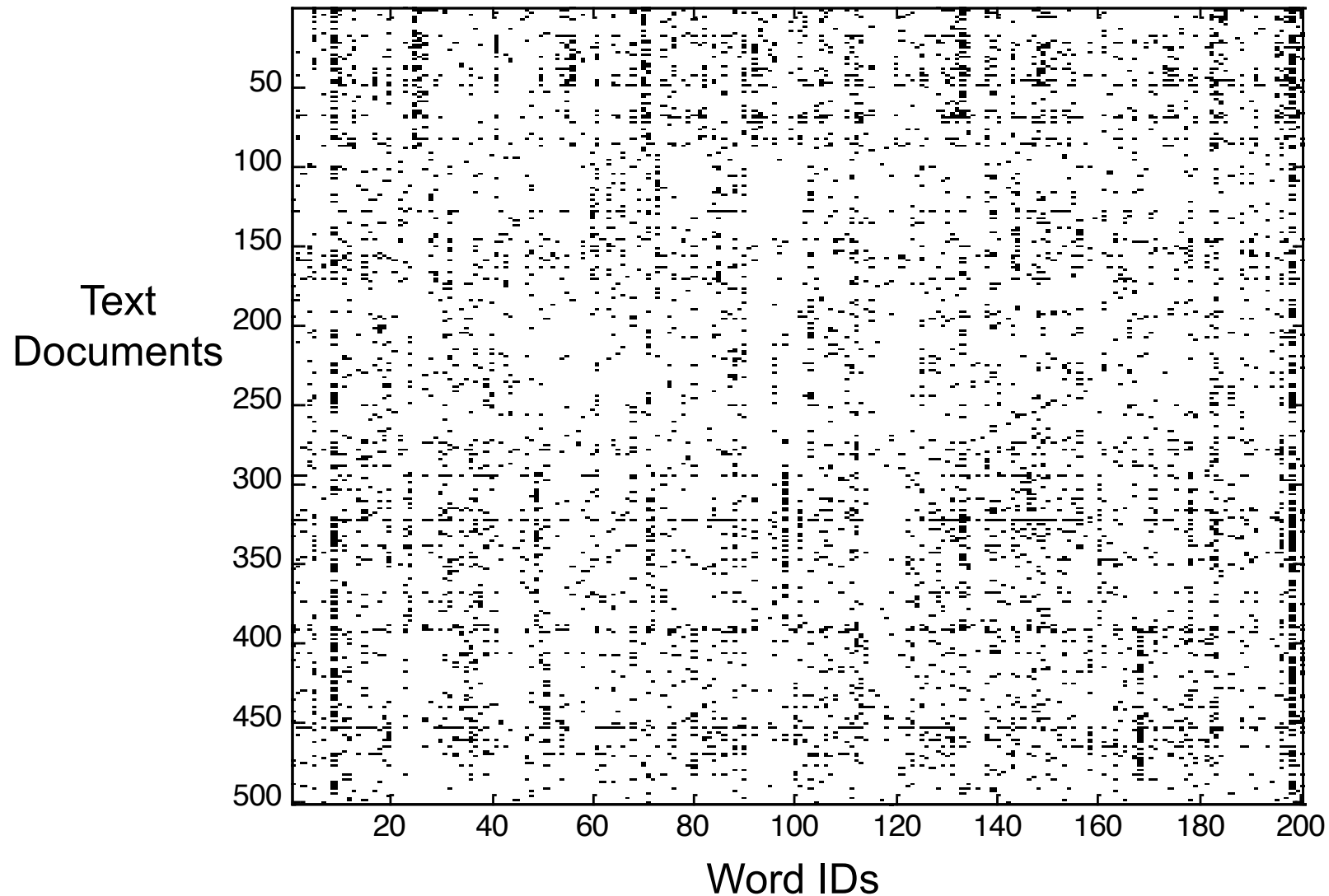


A diagram illustrating a data matrix. It consists of a 3x4 grid of blue squares with black borders. The first row contains the values 2.3, -1.5, ..., and -1.3. The second row contains 1.1, 0.1, ..., and -0.1. The third row contains four dots (...). To the left of the grid, a red curly brace spans the height of the three rows, with the letter 'n' next to it. Below the grid, a red curly brace spans the width of the four columns, with the letter 'p' below it.

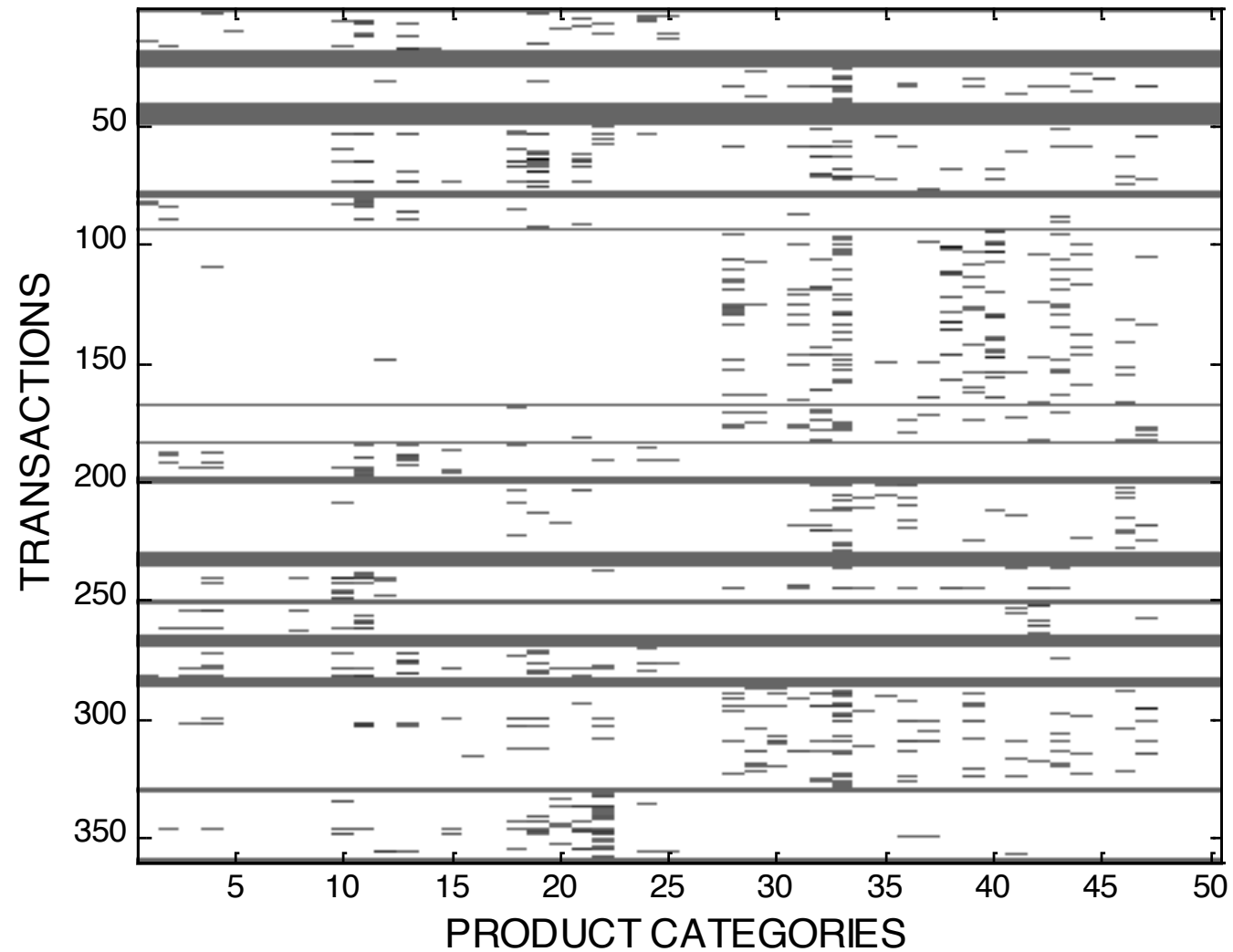
2.3	-1.5	...	-1.3
1.1	0.1	...	-0.1
...

- Rows = objects
- Columns = measurements on objects
 - Represent each row as a p -dimensional vector, where p is the dimensionality
 - In effect, embed our objects in a p -dimensional vector space
 - Often useful, but always appropriate
- Both n and p can be very large in certain data mining applications

Sparse Matrix (Text) Data



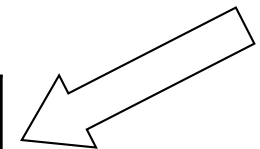
“Market Basket” Data



Sequence (Web) Data

128.195.36.195, -, 3/22/00, 10:35:11, W3SVC, SRVR1, 128.200.39.181, 781, 363, 875, 200, 0, GET, /top.html, -,
 128.195.36.195, -, 3/22/00, 10:35:16, W3SVC, SRVR1, 128.200.39.181, 5288, 524, 414, 200, 0, POST, /spt/main.html, -,
 128.195.36.195, -, 3/22/00, 10:35:17, W3SVC, SRVR1, 128.200.39.181, 30, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.195.36.101, -, 3/22/00, 16:18:50, W3SVC, SRVR1, 128.200.39.181, 60, 425, 72, 304, 0, GET, /top.html, -,
 128.195.36.101, -, 3/22/00, 16:18:58, W3SVC, SRVR1, 128.200.39.181, 8322, 527, 414, 200, 0, POST, /spt/main.html, -,
 128.195.36.101, -, 3/22/00, 16:18:59, W3SVC, SRVR1, 128.200.39.181, 0, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:54:37, W3SVC, SRVR1, 128.200.39.181, 140, 199, 875, 200, 0, GET, /top.html, -,
 128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 17766, 365, 414, 200, 0, POST, /spt/main.html, -,
 128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:55:07, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 1061, 382, 414, 200, 0, POST, /spt/main.html, -,
 128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:55:39, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:56:03, W3SVC, SRVR1, 128.200.39.181, 1081, 382, 414, 200, 0, POST, /spt/main.html, -,
 128.200.39.17, -, 3/22/00, 20:56:04, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
 128.200.39.17, -, 3/22/00, 20:56:33, W3SVC, SRVR1, 128.200.39.181, 0, 262, 72, 304, 0, GET, /top.html, -,
 128.200.39.17, -, 3/22/00, 20:56:52, W3SVC, SRVR1, 128.200.39.181, 19598, 382, 414, 200, 0, POST, /spt/main.html, -,

User 1	2	3	2	2	3	3	3	1	1	1	3	1	3	3	3	3
User 2	3	3	3	1	1	1										
User 3	7	7	7	7	7	7	7	7								
User 4	1	5	1	1	1	5	1	5	1	1	1	1	1	1	1	1
User 5	5	1	1	5												
...																



Time Series Data

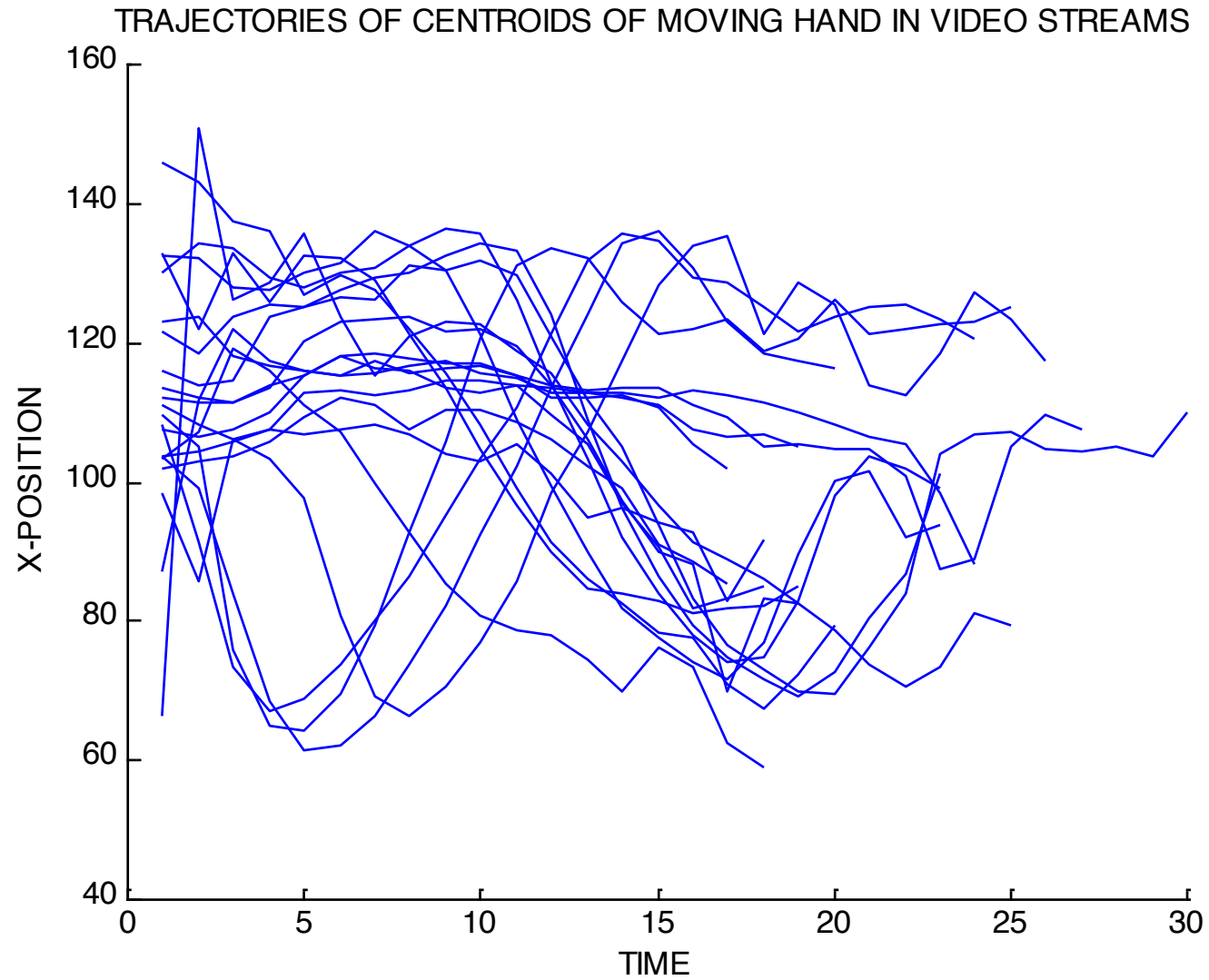
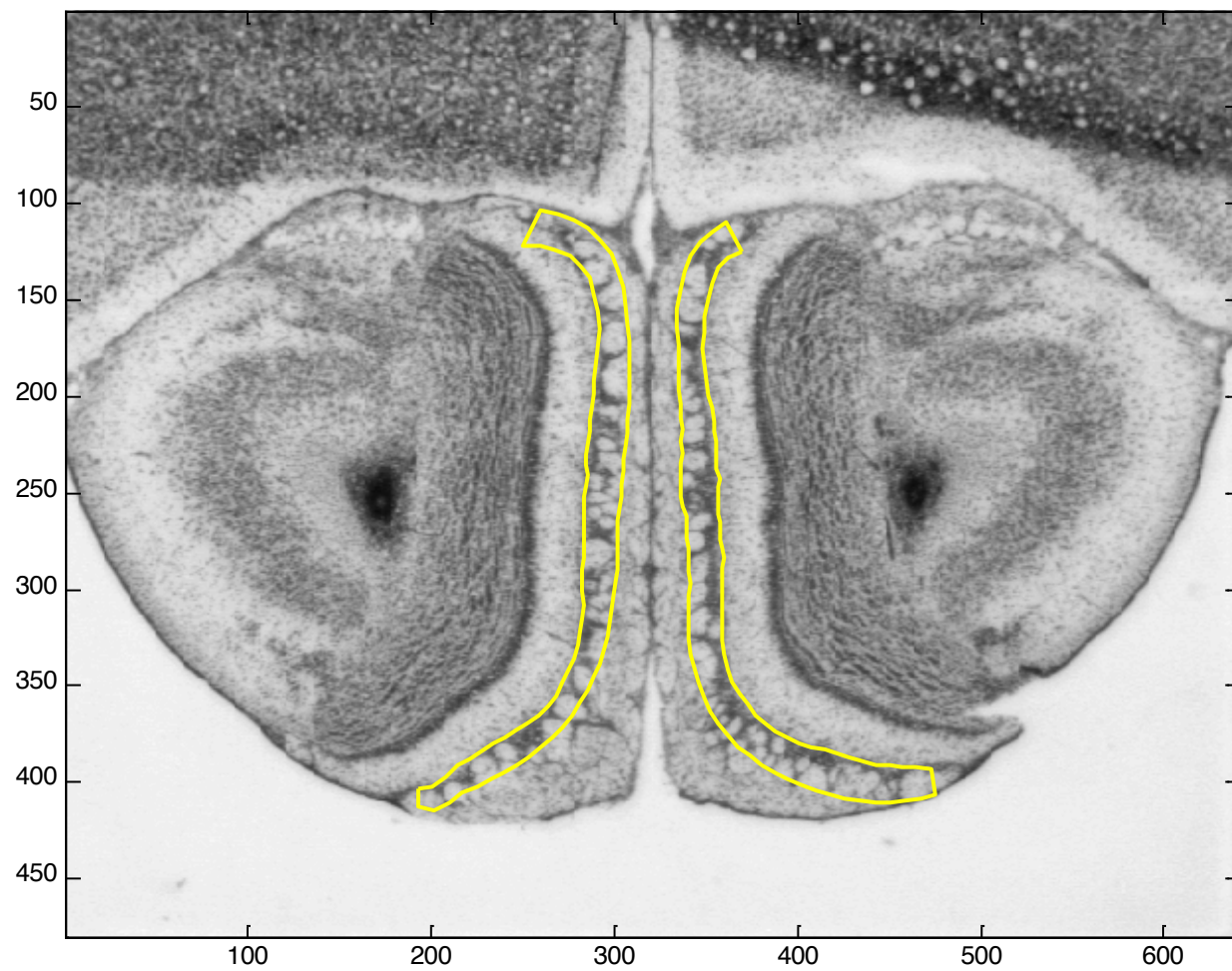
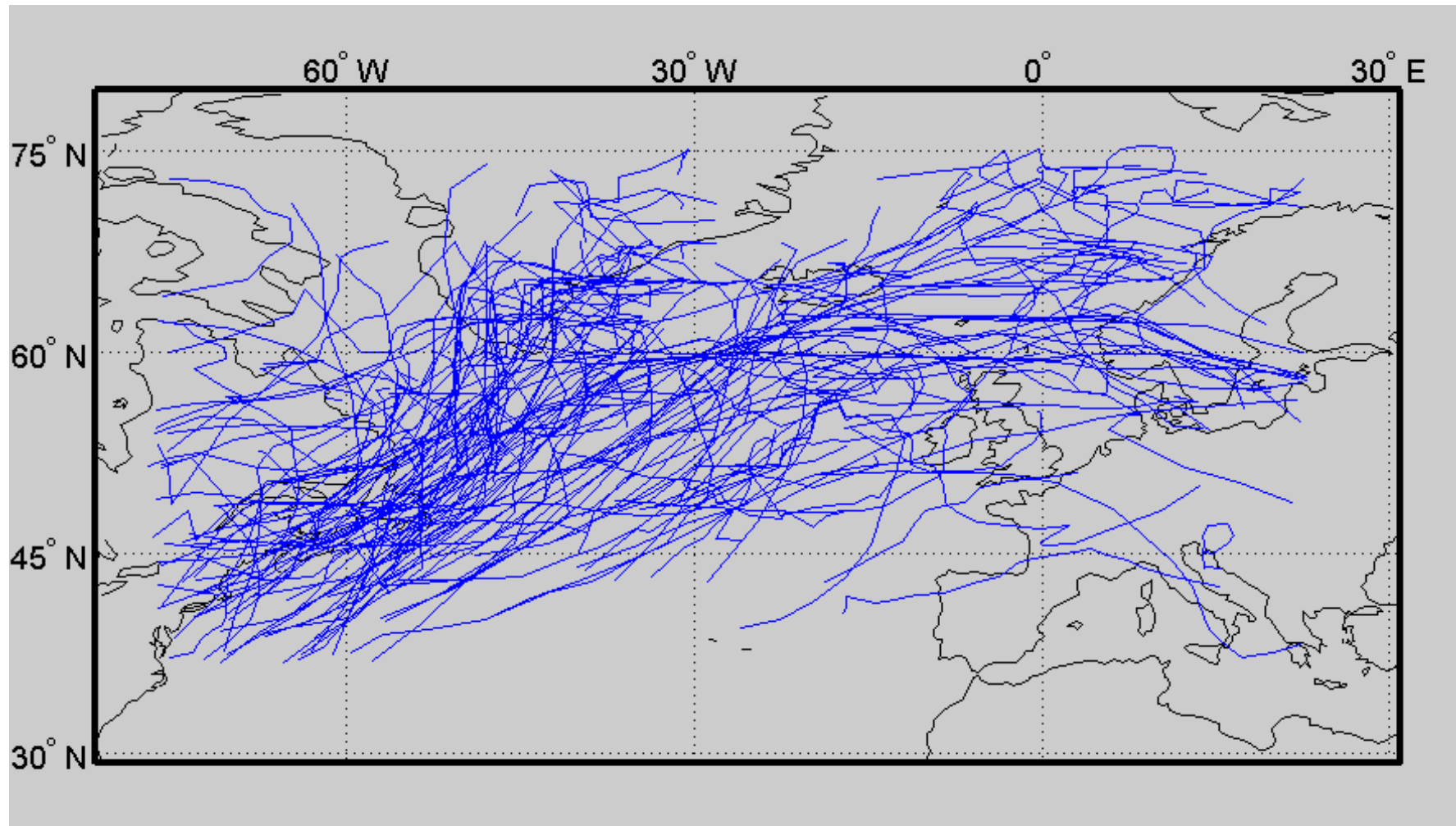


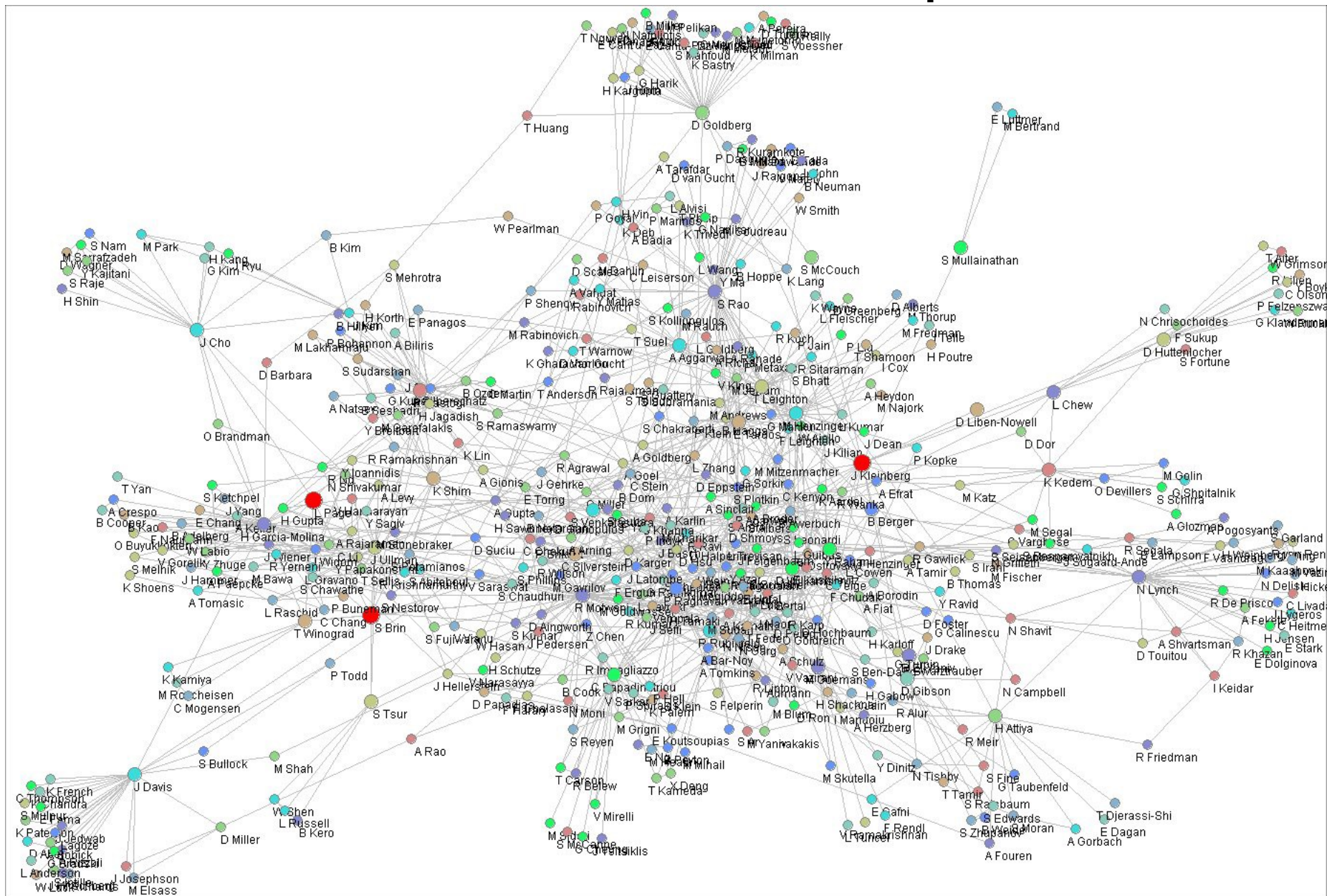
Image Data



Spatio-temporal data



Social Networks – Graphs

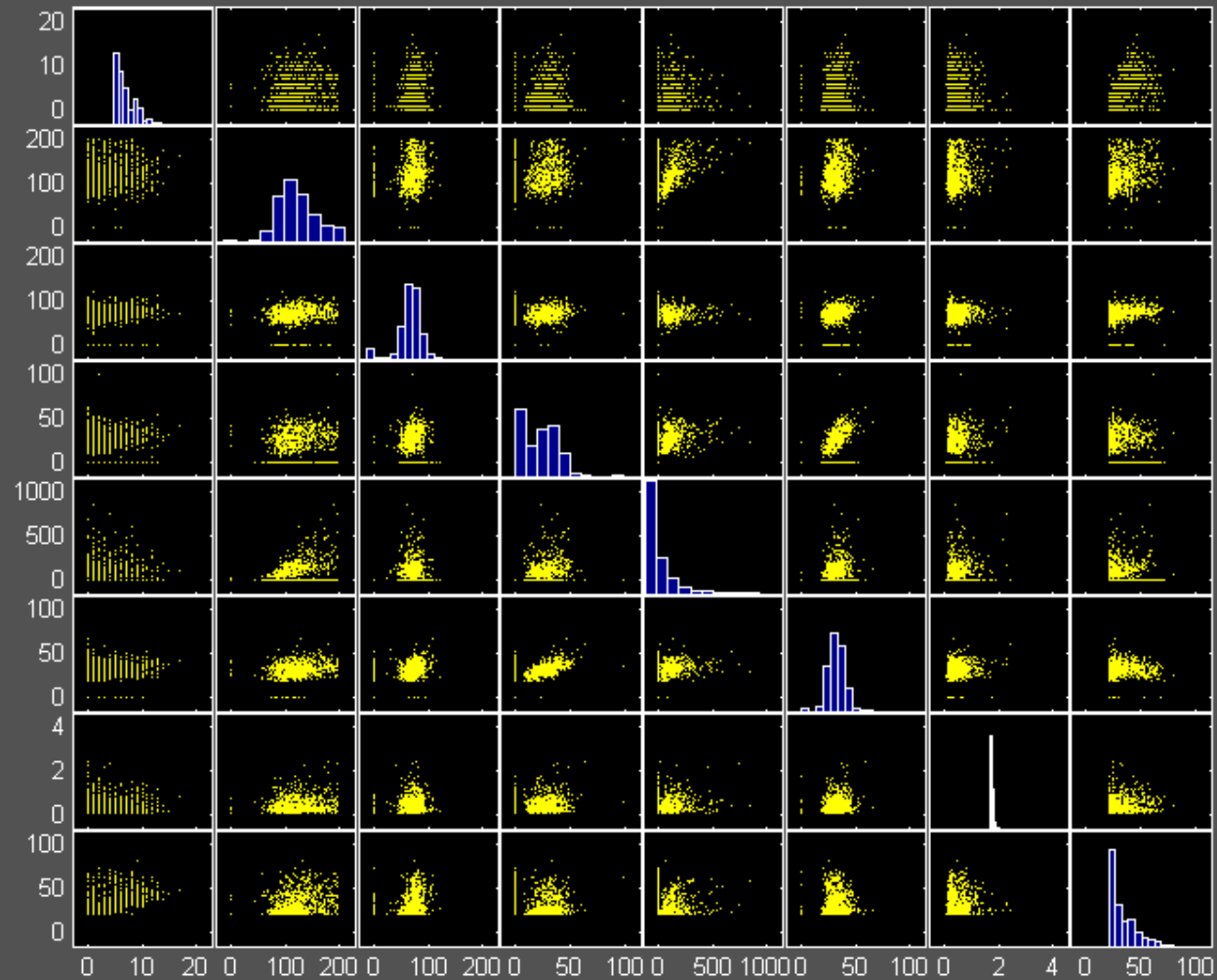


Exploratory Data Analysis


- Getting an overall sense of the data set
 - Computing summary statistics:
 - Number of distinct values, max, min, mean, median, variance, skewness,...
- Visualization is widely used
 - 1d histograms
 - 2d scatter plots
 - Higher-dimensional methods
- Useful for data checking
 - Finding the some variables are highly skewed
- Simple exploratory analysis is extremely valuable
 - You should always “look” at your data before applying any machine learning

Example of Exploratory Data Analysis

(Pima Indians data, scatter plot matrix)



Machine Learning for solving real problems



Competitions Datasets Notebooks Discussion Courses ...

Competitions

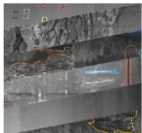







Documentation InClass

General InClass

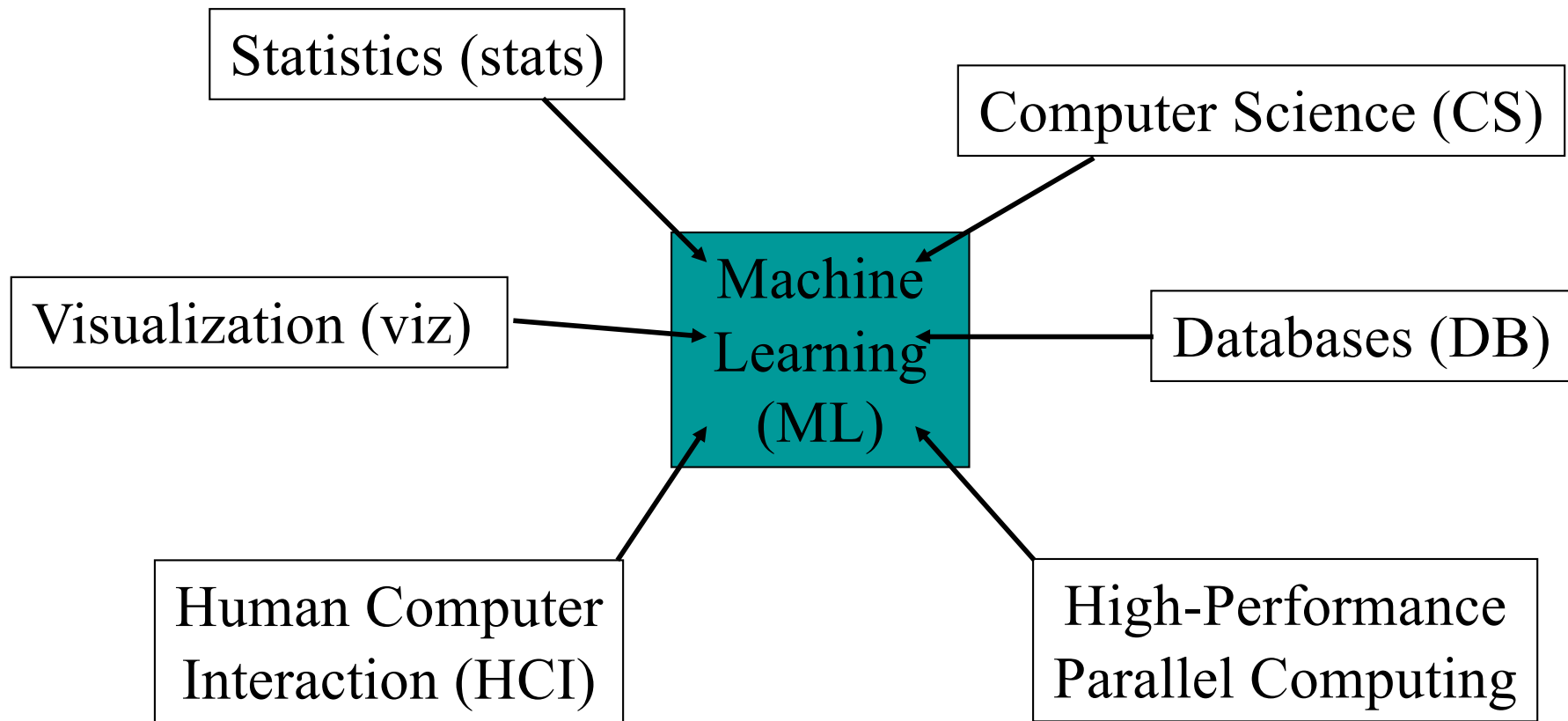
Sort by Grouped

All Categories Search competitions

15 Active Competitions

	Severstal: Steel Defect Detection Can you detect and classify defects in steel? Featured · Code Competition · 2 months to go ·  manufacturing, image data	\$120,000 1,191 teams
	The 3rd YouTube-8M Video Understanding Challenge Temporal localization of topics within video Research · a month to go ·  video data, object detection	\$25,000 247 teams
	Open Images 2019 - Object Detection Detect objects in varied and complex images Research · 22 days to go ·  image processing, image data	\$25,000 501 teams
	Open Images 2019 - Visual Relationship Detect pairs of objects in particular relationships Research · 22 days to go ·  image processing, image data	\$25,000 164 teams

ML: Intersection of Many Fields



Machine learning

- Tom Mitchell(1998): Well-posed Learning Problem: A computer program is said to learn from experience **E** with respect to some task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, *improves with experience* **E**.

email spam prediction

- **Task:** email classification to spam/no-spam
- **Experience:** the user's action to characterize emails
- **Performance:** # of emails characterized as spam correctly.

What is Machine Learning

- “...computational methods using experience to improve performance or to make accurate predictions” **Mohri et. al.** (2012)
- **experience**: past information available to the learner, in the form of electronic data collected and made available for analysis.
- **Data quality** and **size** are crucial to the success of the predictions made by the learner.
- Machine learning consists of
 - designing efficient & accurate prediction *algorithms* - time and space complexity.
 - Additionally - sample complexity to evaluate the sample size required for the algorithm to learn a family of concepts.

More generally, theoretical learning guarantees for an algorithm depend on the complexity of the concept classes and the size of the training sample.
- learning techniques are data-driven methods combining fundamental concepts in computer science with ideas from statistics, probability and optimization.

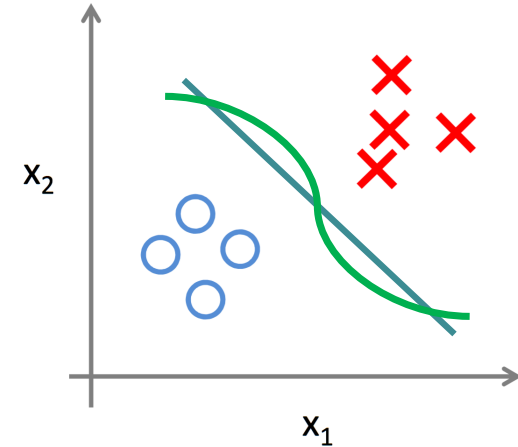
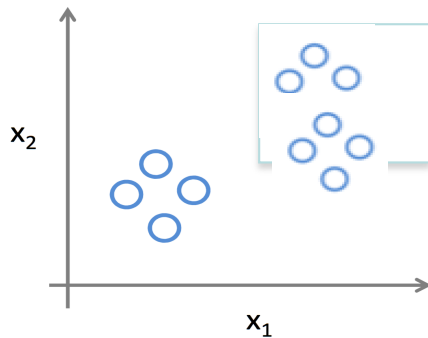
Applications of Machine/Deep Learning

- Text or document classification, e.g., spam detection;
- Natural language processing, e.g., morphological analysis, part-of-speech tagging, statistical parsing, named-entity recognition
- Recommendation systems, search engines, information extraction systems
- Fraud detection (credit card, telephone) and network intrusion
- Speech recognition, speech synthesis, speaker verification;
- Optical character recognition (OCR);
- Computational biology applications, e.g., protein function or structured prediction, Medical diagnosis;
- Computer vision tasks, e.g., image recognition, face detection;
- Games, e.g., chess, backgammon;
- Unassisted vehicle control (robots, navigation);
- ...

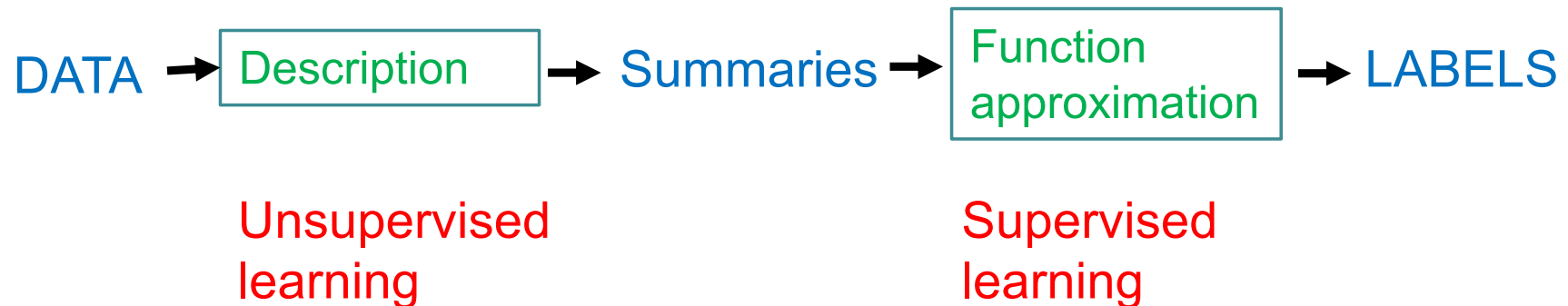
ML Tasks

Main Tasks

- Supervised Learning - Approximation
- Unsupervised Learning – Description



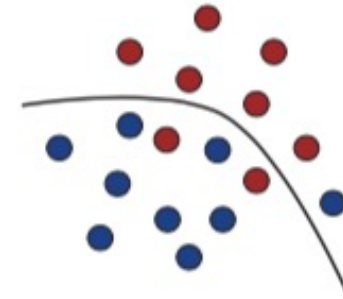
- Supervised & unsupervised learning synergy



More ML/DL Tasks

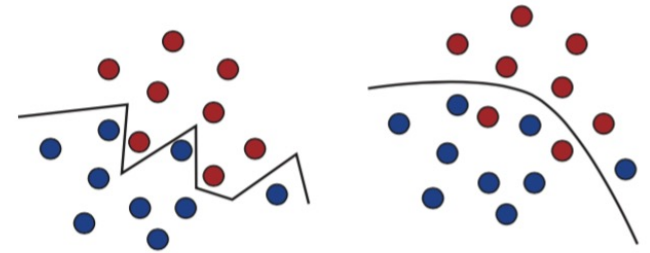
- Structured output
 - Transcription (optical character, speech recognition – out put is characters)
 - Machine translation
 - Summarization
- Anomaly detection
 - flags data as unusual/atypical – i.e. credit card fraud detection.
- Synthesis and sampling
 - generate new examples that are similar to those in the training data.
 - useful for media applications: video games automatically generate textures for landscapes
- Imputation of missing values
 - prediction of the values of the missing entries.
- De-noising

Machine Learning example



- Red and blue dots - **training set**
- Red/Blue - **labels/classes**
- **Features:** the space in which the training set is embedded (i.e. the (x,y) coordinates for this example)
- Objective: Learn a **model (a function)** f that based on the position of a sample decides the class of the point.
- Test sample: Examples to evaluate the performance of a learning algorithm - separate from the training and not made available in the learning stage

Machine Learning example



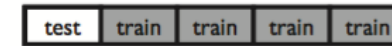
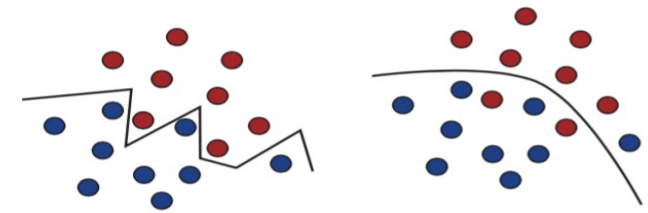
- **Loss function**: function L measures error, or loss, between a predicted and a true label. Let Y/Y' true/predicted labels:

$$L = Y x Y' \rightarrow R_+$$

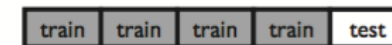
- Square loss: $E = \sum_{i=1}^k (y(i) - y'(i))^2$
- Other loss functions: Hinge, Logistic, Cross entropy...
- **Hypothesis set**: set of functions mapping features to labels (i.e. points to blue/red)
- **Over fitting vs generalization**: a function may be consistent (i.e. zero training error) but not generalize well.

Machine Learning example

- **Cross-validation**: in many cases not enough training data.
 - Split the m data into n subsets(folds) and let θ the model parameters
 - Train the algorithm for $n-1$ folds and test on the n -th
 - Compute the cross validation error
 - Choose parameters θ that minimize the cv. error



⋮



$$\hat{R}_{CV}(\theta) = \frac{1}{n} \sum_{i=1}^n \underbrace{\frac{1}{m_i} \sum_{j=1}^{m_i} L(h_i(x_{ij}), y_{ij})}_{\text{error of } h_i \text{ on the } i\text{th fold}} .$$

Error Optimization – gradient descent

- Learning & Optimization: Assume $J(\theta)$ the objective error function, θ hypothesis parameters.
- Objective: find θ that minimizes $J(\theta)$:
 - update the parameters in the opposite direction of the gradient of the objective function: $\nabla_{\theta} J(\theta)$ w.r.t. to the parameters
 - Batch gradient descent

$$\theta = \theta - \eta \nabla_{\theta} J(\theta)$$

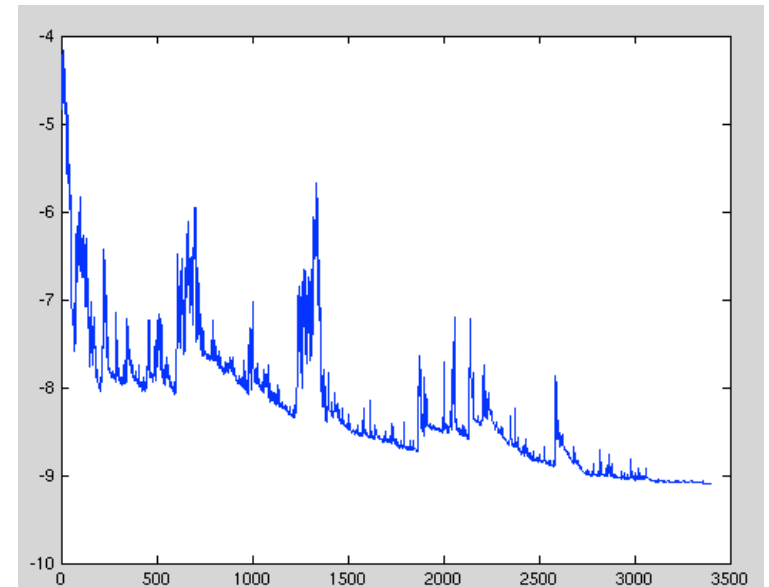
- η the learning rate
- *Redundant computations*: re-computes gradients for similar examples before each parameter update.

Error Optimization – gradient descent

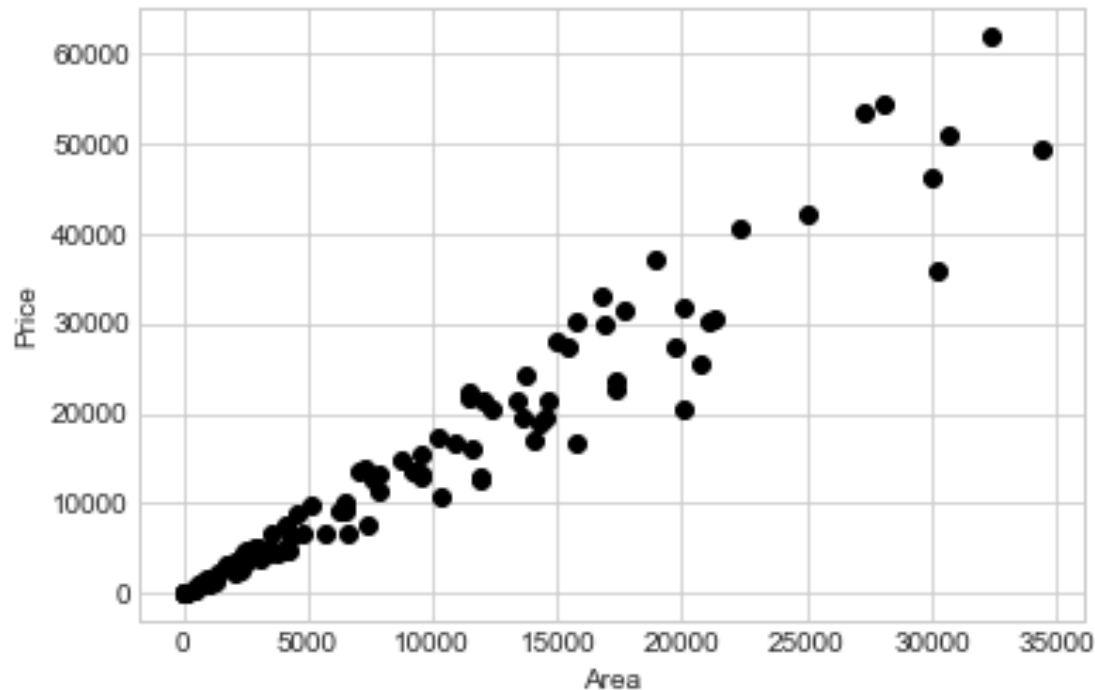
- **Stochastic gradient descent**: update the parameters for each training data point $(x(i), y(i))$

$$\theta = \theta - \eta \nabla_{\theta} J(\theta, x(i), y(i))$$

- One update at a time, faster
- High variance - fluctuation
- *Other optimization methods*
 - *Expectation Maximization*



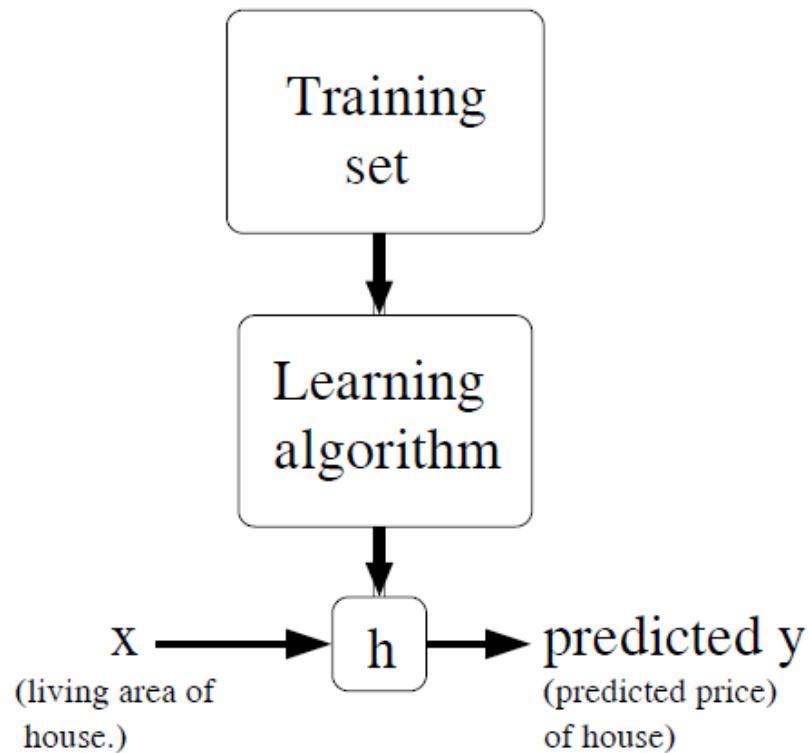
Supervised learning - prediction



- Can we predict the price of a house based on its size (surface in m^2) ?

Supervised learning - prediction

- y continuous value:
prediction
- y discrete value:
classification

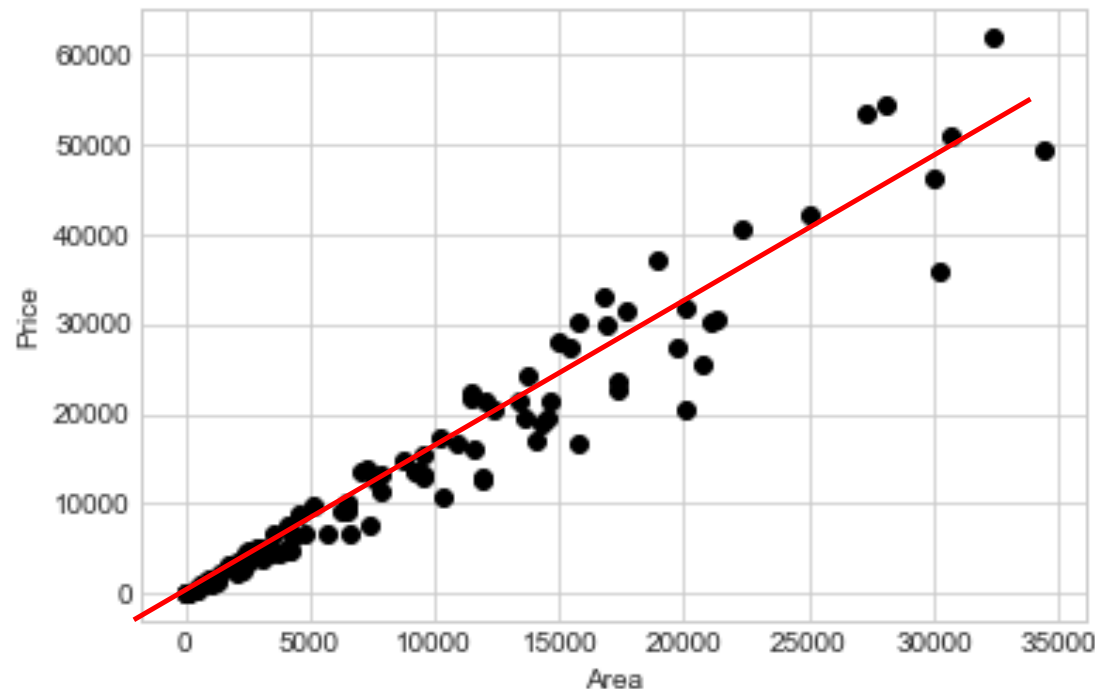


Prediction

- $x(i)$: “input” variables (input features)
- $y(i)$: “output” or target variable - trying to predict
- pair $(x(i), y(i))$: training example,
- *training set* - a list of m training examples $\{(x(i), y(i)); i = 1, \dots, m\}$.
- X : space of input values, Y : output values.
- The supervised learning problem:
 - given a training set,
 - learn a function $h : X \rightarrow Y : h(x)$ “good” predictor for the corresponding value of y - h also called *hypothesis*.

Prediction with Regression

- Aims at fitting a line to a set of observations $\{(x_1, y_1), \dots, (x_N, y_N)\}$, there is a straight line $y = ax + b$.



Regression

- the individual point error is: $y - (ax + b)$
- thus the error set is: $\{y_1 - (ax_1 + b), \dots, y_n - (ax_n + b)\}$
- the total error is:
$$E(a, b) = \sum_{n=1}^N (y_n - (ax_n + b))^2$$

Least Squares method

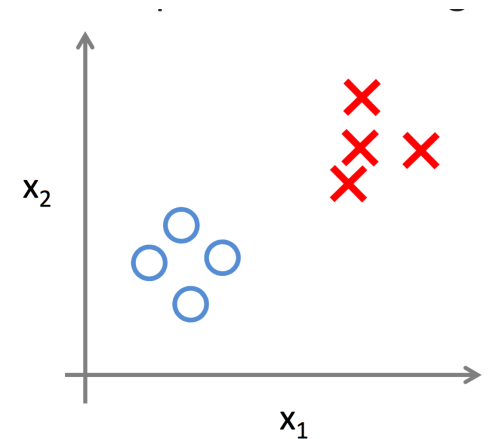
- The objective is to minimize $E(a, b) = \sum_{n=1}^N (y_n - (ax_n + b))^2$
- Thus to find values a, b such that: $\frac{\partial E}{\partial a} = 0, \quad \frac{\partial E}{\partial b} = 0.$
- Differentiation leads to:

$$\frac{\partial E}{\partial a} = \sum_{n=1}^N 2 (y_n - (ax_n + b)) \cdot (-x_n)$$

$$\frac{\partial E}{\partial b} = \sum_{n=1}^N 2 (y_n - (ax_n + b)) \cdot 1.$$

Supervised Learning - Classification

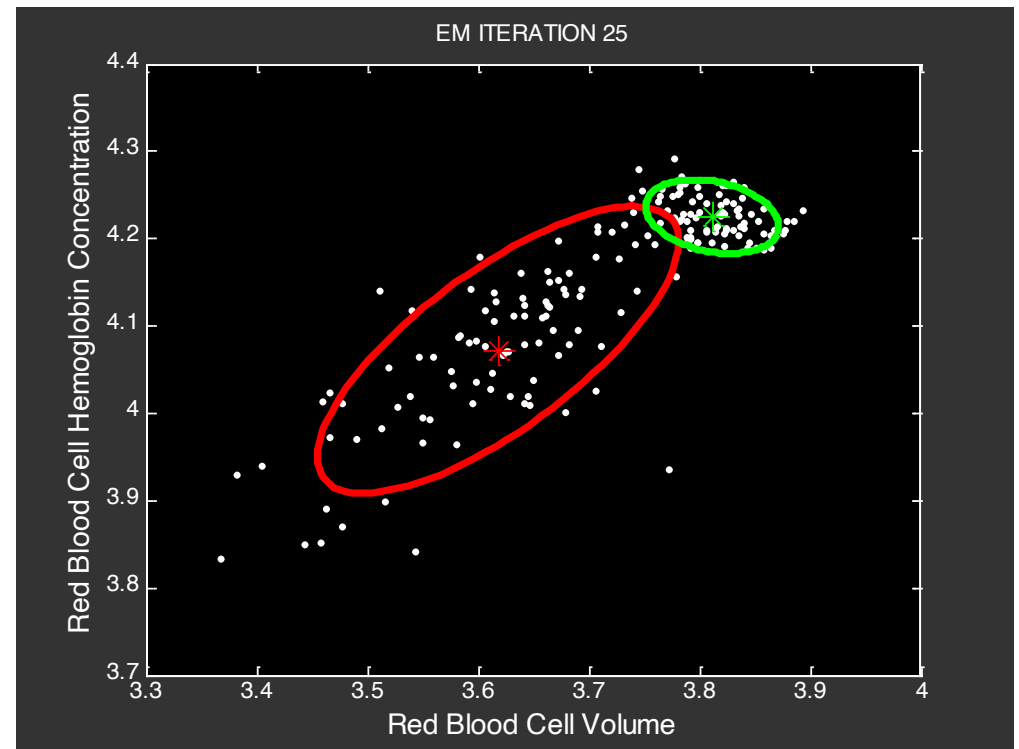
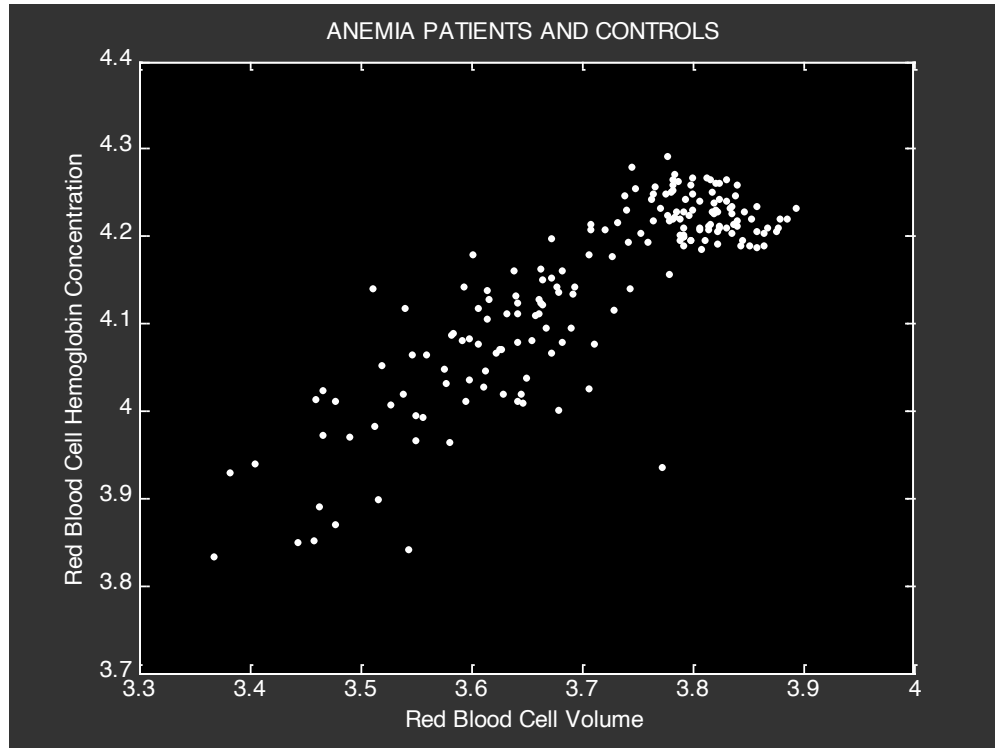
- Class-conditional/probabilistic, based on $p(\underline{x} | c_k)$,
 - **Naïve Bayes** (simple, but often effective in high dimensions)
 - **Parametric generative models**, e.g., Gaussian (can be effective in low-dimensional problems)
- Discriminative models, focus on locating optimal decision boundaries
 - **Linear discriminants**, perceptron: simple, sometimes effective
 - **Support vector machines**: generalization of linear discriminants, can be quite effective, computational complexity is an issue
 - **Nearest neighbor**: simple, can scale poorly in high dimensions
 - **Decision trees**: learning splitting of the data that maximize information learning, effective in high dimensions



Unsupervised learning

- Learn a “generative” or “descriptive” model,
 - E.g., a model to simulate/generate the data if needed
 - Model underlying processes
- Examples:
 - Density estimation:
 - estimate the joint data distribution $P(x_1, \dots, x_p)$
 - Cluster analysis:
 - Find natural groups in the data
 - Dimensionality reduction
 - Learn latent spaces (SVD, PCA,)
 - Word/document embeddings, auto-encoders....

Unsupervised learning - clustering



Unsupervised learning - Pattern Discovery

ADACABDABAABBDDBCADDDDBCDDBCCBBCCDADADAADABDBBDABABBCDD
DCDDABDCBBDBDBCBBABBBBCBBABCBBACBBDBAACCADDADBDBBCBBCCBB
BDCABDDBBADDBBBBCCACDABBABDDCDDBBABDBDDDBDDBCACDBBCCBBAC
DCADCBACCADCCCACCDDADCBCADADBAACCDDDCBDBDCCCCACACACCDAB
DDBCADADBCBDDADABCCABDAACABCABACBDDDCBADCBADDDDCDDCADC
CBBADABBAADAAABCCBCABDBAADCBCDACBCABABCCBACBDABDDDADAA
BADCDCCDBBCDBDADDCCBBCDBAADADBCAAAADBDCADBDBBBBCDCCBCCCD
CCADAADACABDABAABBDDBCADDDDBCDDBCCBBCCDADADACCCDABAABBC
BDBDBADB BBBCDADABABBDACDCDDDBBCDBBCBBCCDABCADDADBACBBBC
CDBAAADDDDBDDCABACBCADCDCBAAADCADDADAABBACCBB

Unsupervised learning - Pattern Discovery

ADACABDABAABBDDBCADDDDBCDDBC**CBBC**CDADADAADABDBBDABABBCDD
DCDDABDCBBDBDBCBBABBBCBBABCBBACBBDBAACCADDADBDBB**CBBC**BB
BDCABDDBBADDBBBBCCACDABBABDDCDDBBABDBDDDBDDBCACDBBCCBBAC
DCADCBACCADCCCACCDDADCBCADADBAACCDDDCBDBDCCCCACACACCDAB
DDBCADADBCBDDADABCCABDAACABCABACBDDDCBADCBADDDDCDDCADC
CBBADABBAADAAABCCBCABDBAADCBCDACBCABABCCBACBDABDDDADAA
BADCDCCDBBCDBDADDC**CBBCD**BAADADBCAAAADBDCADBDBBBBCD**CBBC**CD
CCADAADACABDABAABBDDBCADDDDBCDDBC**CBBC**CDADADACCCDABAABBC
BDBDBADB BBBCDADABABBDACDCDDDBBCDBBCBBCCDABCADDADBA**CBBC**
CDBAAADDDDBDDCABACBCADCDCBAAADCADDADAABBACCBB

Machine Learning – quiz..

Of the following examples, which would you address with *unsupervised/supervised* learning?

1. Given email labeled as spam/not spam, learn a spam filter.
2. Given data of patients diagnosed with cancer, learn to classify new patients for this disease.
3. Given a set of news articles found on the web, group them into set of articles about the same story.
4. Given a database of customer data, discover market segments and group customers into different market segments.

Machine Learning – quiz..

- Classification or regression?
 - Credit history -> offer a loan?
 - Human face picture -> {kid, adolescent, adult}
 - Human face picture -> age

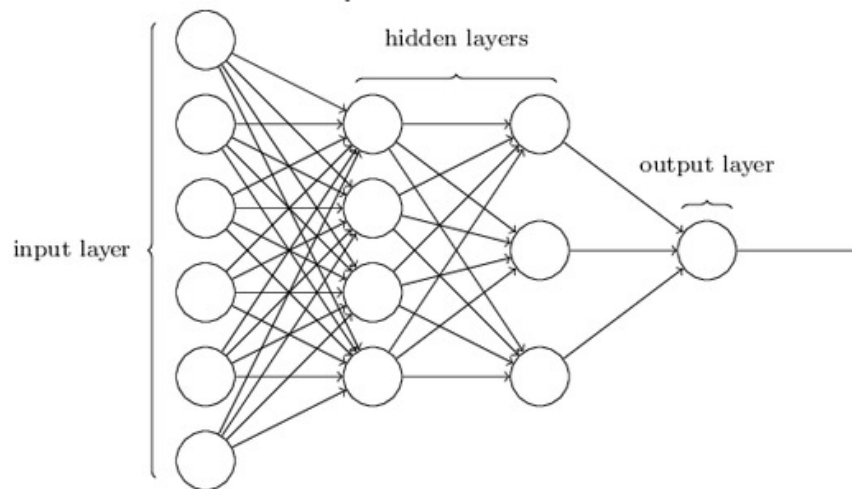
Discrete vs continuous output

More recent ML types of algorithms

- **Deep Learning**
- Reinforcement learning
 - Target, states, actions rewards, policy
 - Search for optimal policy..
- Adversarial learning
- ...

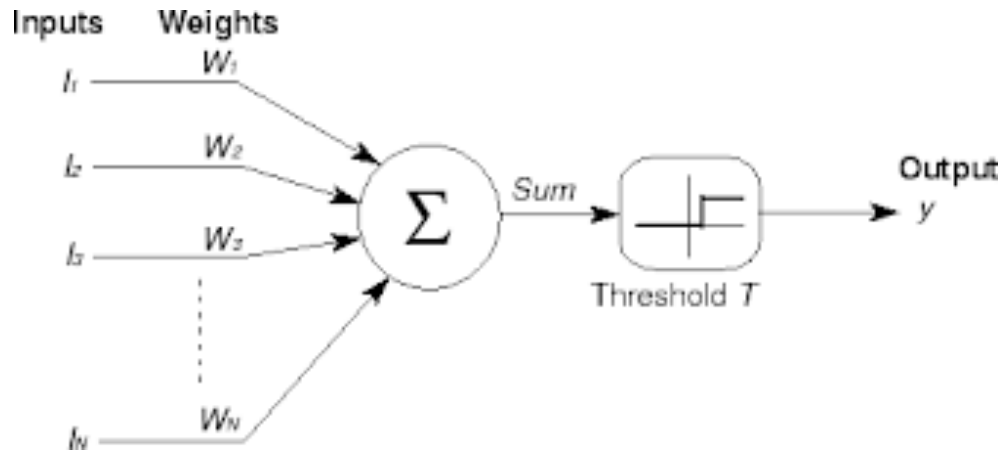
Deep Learning

- Based on perceptron – basic learning unit
- Layers of perceptrons learning a complex function $y=f(X)$



- *Learning features/embeddings*
- *Used for predictions*

Neural networks - perceptron



McCulloch-Pitts Neuron Model

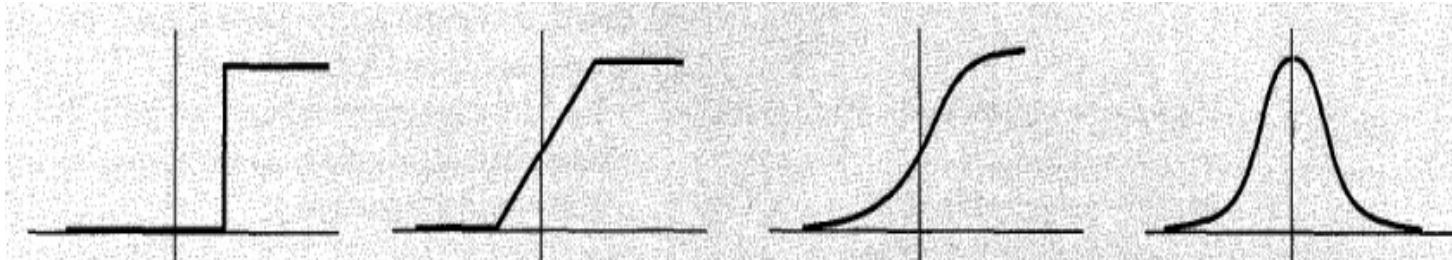
- f : activation function

- w_j : weight of the j -th input X_j

- b : bias

- activation functions: piecewise linear, sigmoid, or Gaussian

$$y = f\left(\sum_{j=1}^n w_j X_j + b\right)$$



Deep Learning

Dominant in recent years

Different architectures – non exhaustive

- Multilayer perceptrons (MLPs)
- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs) – for sequential data
 - GRUs
 - LSTMs
- Attention based Architectures
 - Self Attention, Transformer (Bert), ...
- Autoencoders
- **Graph Neural Networks**