PA164 Natural Language Learning

Lecture 02: Quick and Dirty Intro to ML

Vít Nováček

Faculty of Informatics, Masaryk University

Autumn, 2023

MUNI

Outline

Outline

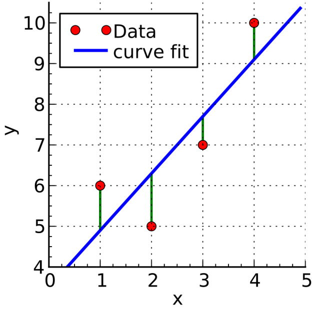
Supervised Learning – the Gist

- Learning to predict output class labels (classification) or numerical values (regression) that are associated with input objects (typically represented by so called feature vectors of predictor variables)
- Typically trained and tested on two independent sets, with the correct output values hidden in the test set
- Some popular methods:
  - Linear regression, logistic regression
  - Support vector machines
  - Decision/regression trees
  - Neural networks

A Sample Method – Linear Regression

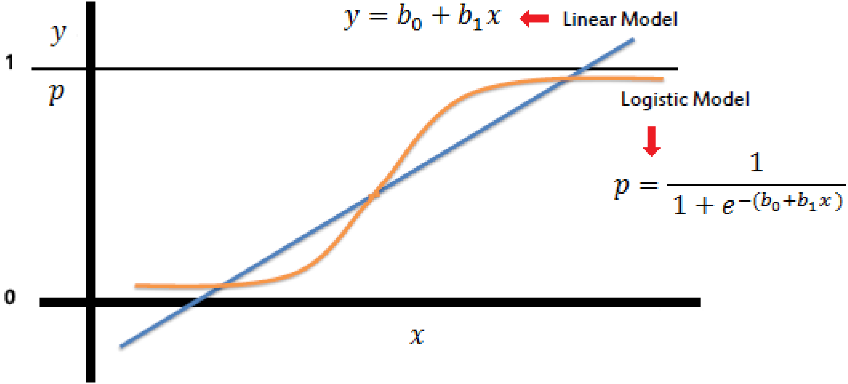
- Working with a set of n data points $\{y_i, x_{i1}, \ldots, x_{ip}\}_{i=1}^{n}$, where y_i is the output (dependent) variable and the vector $\mathbf{x}$ of p regressors represents the input (independent) variables
- The difference between the real y_i observations and their assumed linear dependence on the $\mathbf{x}_i$ vectors is modelled using error variables ϵ_i:
  - $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i = \mathbf{x}_\mathbf{i}^\mathbf{T} \boldsymbol{\beta} + \epsilon_i$ for $i = 1, \ldots, n$ (vector notation)
  - $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ (the same in a concise matrix notation)
- The training then consists of minimising the error term ϵ while learning the corresponding values of the $\boldsymbol{\beta}$ parameter vector

- Graphical illustration of the basic principle



[1] Author of the image: Krishna Vedala. Downloaded from Wikimedia Commons. License: CC BY-SA 3.0.

A Sample Method – Logistic Regression

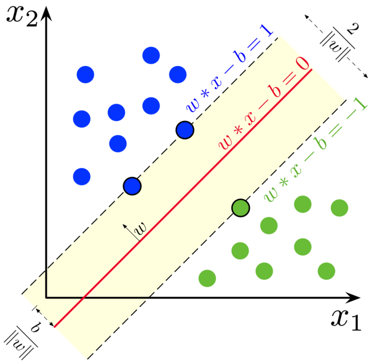


[2] Image source: Sayad, Saed. An Introduction to Data Science. Blog available at https://saedsayad.com/.

A Sample Method – Support Vector Machines

- Originally a supervised linear model for binary classification
- Later extended to regression, multi-class problems, semi-supervised settings, etc.
- Works by learning a maximum-margin hyperplane that separates the data points belonging to different classes

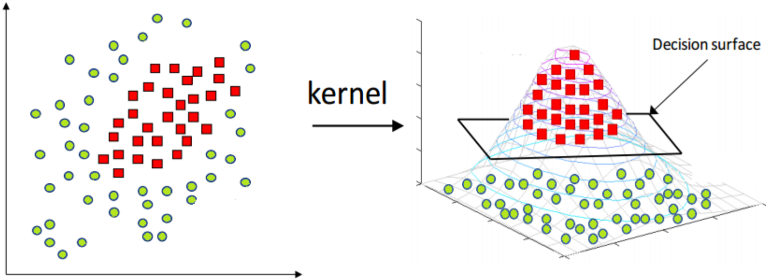
- Graphical illustration of the basic principle:



[3] Author of the image: Lahrmam via Wikimedia Commons. License: CC BY-SA 4.0.

Support Vector Machines – Dealing with Non-Linearities

- Done using so called kernel trick
  - For two vectors $\mathbf{x}, \mathbf{x}'$ in a space $\mathcal{X}$...
  - kernel $k(\mathbf{x}, \mathbf{x}')$ is a function that can be expressed...
  - as an inner product in another space $\mathcal{V}$.
- Convenient to compute using a feature map $\phi : \mathcal{X} \rightarrow \mathcal{V}$ such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{V}}$
- Graphical illustration of the basic principle:

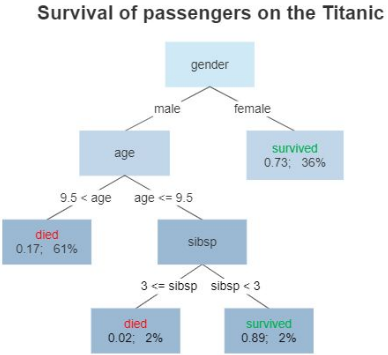


[4] Source of the image: `https://miro.medium.co` (license unknown).

A Sample Method – Decision Trees

- Classifying data based on their characteristic features
- Constructed in a top-down manner
  - Recursively splitting the data set using the most discriminative feature (at the moment)
  - To determine such features, various homogeneity metrics used, such as Gini impurity or information gain
- Naturally describes the structure of the problem (inherent explainability)
- Can be extended to regression trees, random forests, etc.

- A sample decision tree:

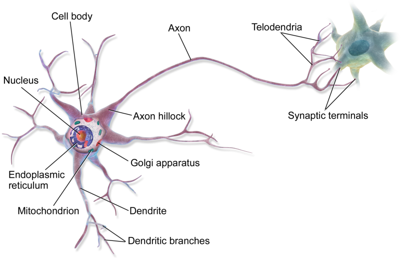


[5] Author of the image: Gilgoldm via Wikimedia Commons.

A Sample Method – Artificial Neuron (Motivation)

- Motivated by neuroscience (how it works in biological brains)
- In a nutshell, neuron is a:
  - unit recieving input signals via synapses. . .
  - modulating them. . .
  - and passing the resulting output signal on via an axon.
- Neurons are basic building blocks of complex signal processing pathways that make up nervous systems of living organisms

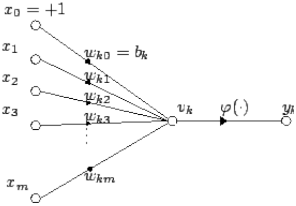
- An actual biological neuron:



[6] Author of the image: Bruce Blaus via Wikimedia Commons. License: CC BY 3.0.

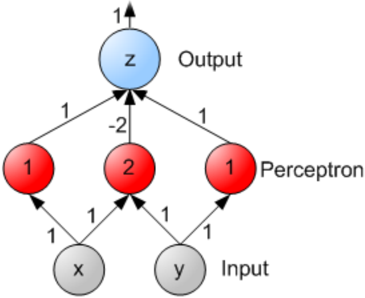
A Sample Method – Artificial Neuron (Implementation)

- Various ways of reverse-engineering the biology proposed
- Most boil down to the equation $y_k = \varphi(\sum_{j=0}^{m} w_{kj} x_j)$, where:
  - y_k is the output,
  - x_j are the inputs,
  - w_{kj} are the weights associated with each input,
  - φ is an activation function that modulates the aggregated, weighted input signal (typically via thresholding, mapping it to the $< 0, 1 >$ interval, etc.)
- Graphical illustration of the general equation:



Stacking the Neurons Up – Artificial Neural Networks

- Where it all started – feed-forward neural network
  - Stacking up the neurons into a network with input and output layers. . .
  - that have at least one hidden layer in between.
  - The trainable parameters are the weights of the connections between the neurons
  - Typically, non-linear activation functions are used (e.g., ReLU or Softmax)
  - The parameter values are learned iteratively by gradient descent
- Graphical illustration of the basic principle:



[8] Author of the image: Robert Ensor via Wikimedia Commons. License: Public Domain.

Outline

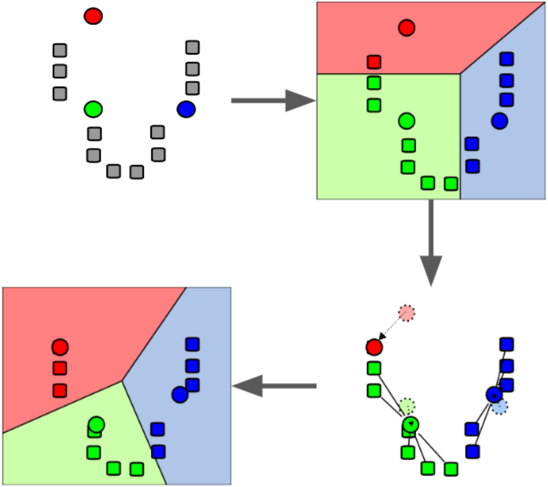
Unsupervised Learning – the Gist

- No output values known
- Learning patterns (e.g., clusters or distribution) characterising the data
- Some popular methods:
  - Unsupervised neural networks (Boltzmann and Helmholtz machines, autoencoders)
  - Probabilistic methods (PCA, cluster analysis)

A Sample Method – k-means (Description)

- Partitioning n observations (data points) into k clusters
- The data points belonging to the same cluster are assumed to share some characteristic properties
- The observations represented as feature vectors (similarly to the supervised ML approaches)
- First, a set of k random mean vectors is generated
- The algorithm then repeatedly (until convergence) executes two steps:
  1. Assign each observation to the cluster with the nearest mean.
  2. Recalculate the means for clustered observations (as centroids of the clusters).
- Non-deterministic and not guaranteed to converge, but that can be mitigated by repeated runs and heuristics

A Sample Method – k-means (Example)



[9] The image based on the grpahics of Weston Pace (via Wikimedia Commons). License: CC BY-SA 3.0.

Outline

Deep Learning

- Good old artificial neural networks
- Only deeper (many hidden layers) and bigger (up to trillions of parameters, based on some GPT-4 rumours)
- Also, with increasingly sophisticated architectures and training methods

Reinforcement Learning

- Agents learning how to take actions in an environment. . .
- to maximise the cumulative reward function.
- Crucial in fine-tuning self-supervised neural models (such as LLMs)
- Some popular methods:
  - Monte Carlo
  - Q-learning
  - Deep refinforcement learning

Dimensionality Reduction

- Techniques to reduce the number of features in the data set
- Either by elimination or transformation into a new, smaller and more discriminative feature space
- Sometimes viewed as a preprocessing method
- Uses some pretty sophisticated models, though
- Examples of popular methods:
  - Principal compoment analysis (PCA)
  - Manifold learning

Semi-supervised Learning

- Falls between supervised and unsupervised learning
- Only some labels available for the data
- Generally works by either propagating the labels across the data or estimating a joint distribution over both labelled and unlabelled data
- Examples of general approaches:
  - Generative statistical models
  - Density-based models
  - Graph-based models
  - Heuristic approaches

Feature/Representation Learning

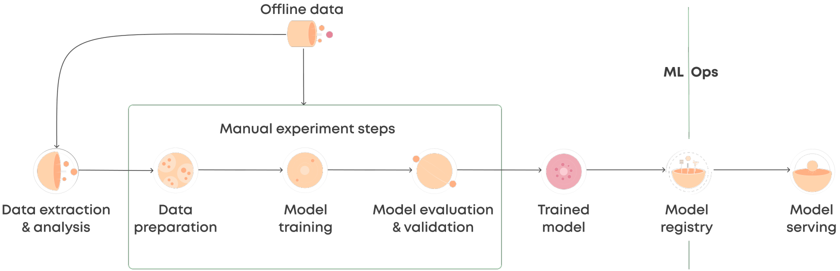
- Skips the feature engineering step of a typical ML pipeline
- Extracts the features directly from the data
- Examples of general approaches:
  - Supervised: (deep) neural networks
  - Unsupervised: autoencoders, matrix factorisation

Rule-Based ML

- Attempts to derive patterns from the data in the form of rules
- The extracted rules collectively represent the knowledge implied by the data
- Another inherently explainable ML paradigm
- Examples of general approaches:
  - Association rule mining
  - Inductive logic programming

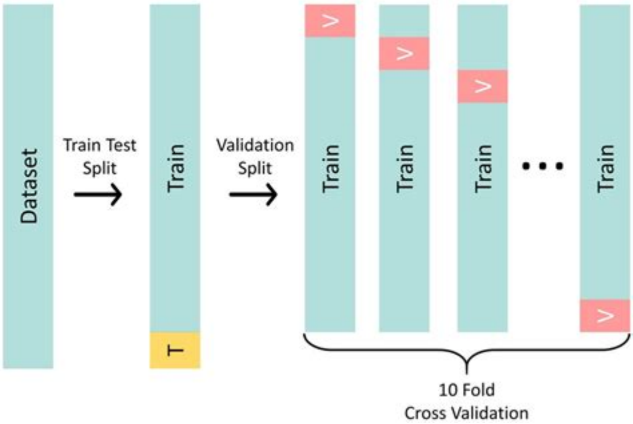
Outline

Typical ML Pipeline



10 The image source: `https://valohai.com/machine-learning-pipeline/`. License: unknown.

Train / Test / Validation Split



[11] The image source: Silveira Kupssinskü, Lucas, et al. "A method for chlorophyll-a and suspended solids prediction through remote sensing and machine learning." Sensors 20.7 (2020): 2125. License: CC BY 3.0.

Evaluating ML Models (Supervised Scenario)

- Typically done using various more or less standard quantitative metrics
- Based on true-positive, false-positive, etc. rates (c.f. confusion matrix)
- Some examples:
  - Precision, specificity, sensitivity/recall, accuracy, F1-score, ...

- A sample confusion matrix:

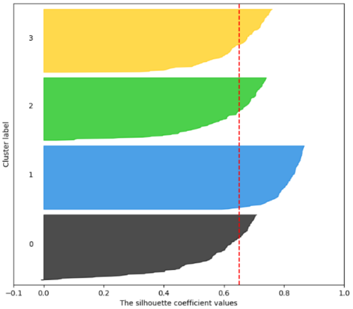
		Predicted condition	
	Total 8 + 4 = 12	Cancer 7	Non-cancer 5
Actual condition	Cancer 8	6	2
	Non-cancer 4	1	3

[12] The image source: https://en.wikipedia.org/wiki/Confusion_matrix. License: unknown.

Evaluating ML Models (Unsupervised Scenario)

- The metrics are slightly more arbitrary or fuzzy
- Typically based on formalising notions like the distinstiveness, density or informativeness of the clusters
- Some metric examples:
  - Silhouette coefficient, Dunn index, purity, Rand index, . . .

- A sample silhouette score plot:



[13] The image source: `https://scikit-learn.org/` (section on cluster analysis). License: BSD.

Evaluating ML Models (Other Metrics)

- Used in learning-to-rank, machine translation, information retrieval and other specific scenarios
- Some examples:
  - Area under the precision-recall or ROC curve, mean reciprocal rank, hits@k, BLEU, ...
- One hot area of research: metrics for evaluating the explainability of ML models
- Final remark
  - No matter what metric you use, qualitative analysis may be crucial, too!

Outline

1 Supervised Learning

2 Unsupervised Learning

3 Other ML Paradigms

4 Notes on the ML Methodology

5 Useful References

Useful References

- Publications
  - Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.
  - Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science 349.6245 (2015): 255-260.
  - LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.
  - Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
  - Zhou, Zhi-Hua. Machine learning. Springer Nature, 2021.
- Courses at FI MU
  - IB031: Introduction to ML (spring)
  - PV021: Neural Networks (fall)
  - PV056: Machine Learning and Data Mining (spring)