ε.δε.μ²
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ & ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Machine Learning (ML) & Artificial Intelligence (AI)
Definitions of Datasets
Discriminative & Generative AI Models
Supervised Learning, Linear & Logistic Regression

Prof. Vasilis Maglaris
maglaris@netmode.ntua.gr
www.netmode.ntua.gr
Room 002, New ECE Building

Tuesday February 11, 2025

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Bibliography

1. Simon Haykin, "**Neural Networks & Learning Machines**", 3rd Ed., Pearson Education, 2009
2. Simon Haykin, "**Νευρωνικά Δίκτυα & Μηχανική Μάθηση**", 3η Έκδοση, Παπασωτηρίου, 2010 (in Greek)
3. Bernhard Mehlig, "**Machine learning with neural networks**", Cambridge Univ. Press 2021 https://arxiv.org/pdf/1901.05639.pdf
4. Μιχάλης Λουλάκης, "**Στοχαστικές Διαδικασίες**", ΣΕΑΒ 2015 http://repfiles.kallipos.gr/html_books/9759/TOC.html (in Greek)
5. Βασίλης Μάγκλαρης, "**Σημειώσεις Μαθήματος Συστήματα Αναμονής**", Συλλογή διαφανειών ΣΗΜΜΥ – ΕΜΠ, 2018 http://www.netmode.ntua.gr/courses/undergraduate/queues/documents/Queuing_Systems_2018.pdf (in Greek)
6. Kevin P. Murphy, "**Machine Learning: A Probabilistic Perspective**", MIT Press, 2012 http://noiselab.ucsd.edu/ECE228/Murphy_Machine_Learning.pdf
7. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "**Deep Learning**", MIT Press, 2016 https://www.deeplearningbook.org/
8. Daniel Jurafsky, James H. Martin, "**Speech & Language Processing: An Introduction to Natural Language Processing, Computational Linguistics & Speech Recognition**", 3rd Ed. (draft), 2025 https://web.stanford.edu/~jurafsky/slp3/
9. Andrew Ng, "**CS229 Lecture Notes**", Stanford University, 2018 https://see.stanford.edu/materials/aimlcs229/cs229-notes1.pdf
10. James Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, "**An Introduction to Statistical Learning**", 2nd Ed., Springer 2021 https://link.springer.com/book/10.1007/978-1-4614-7138-7
11. Richard Sutton, Andrew Barto, "**Reinforcement Learning: An Introduction**", MIT Press, 2018 https://ieeexplore.ieee.org/book/6267343
12. Christopher Bishop, "**Pattern Recognition & Machine Learning**", Springer 2006 https://dl.acm.org/doi/10.5555/1162264
13. Tom Mitchell, "**Machine Learning**", McGraw Hill 1997 http://www.cs.cmu.edu/~tom/mlbook.html
14. Charu C. Aggarwal, "**Outlier Analysis**", Springer 2013 https://link.springer.com/book/10.1007/978-3-319-47578-3
15. Leonida Gianfagna, Antonio Di Cecco, "**Explainable AI with Python**", Springer 2021 https://link.springer.com/book/10.1007/978-3-030-68640-6
16. Christoph Molnar, "**Interpretable Machine Learning**," 2nd Ed., Munich, 2022 https://christophm.github.io/interpretable-ml-book/
17. Frank Kelly, "**Reversibility and Stochastic Networks**", Wiley, 1979 http://www.statslab.cam.ac.uk/~frank/BOOKS/book/whole.pdf
18. Sheldon Ross, "**Applied Probability Models with Optimization Applications**", Dover, 1992
19. Dimitri P. Bertsekas, John Tsitsiklis, "**Neuro-Dynamic Programming**," Athena Scientific, Belmont MA 1996 https://www.researchgate.net/publication/216722122_Neuro-Dynamic_Programming#fullTextFileContent
20. Robert Hogg, Joseph McKean, Allen Craig, "**Introduction to Mathematical Statistics**", 8th Ed., Pearson Education, 2020 https://minerva.it.manchester.ac.uk/~saralees/statbook2.pdf

In the lecture slides Figures from [1], [3] & [4] are reproduced without further notification

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Course Outline (1/2)

1. Optimization Algorithms in Machine Learning: Definitions of Artificial Intelligence (AI) & Machine Learning (ML). Training, Validation & Testing Datasets. Supervised, Unsupervised & Reinforcement Learning. Discriminative & Generative Models, ChatGPT (Chat Generative Pre-trained Transformer) & DeepSeek https://arxiv.org/pdf/2501.12948. Linear & Logistic Regression
2. Neural Networks, Hebb's Rule. Parameter tuning via Supervised Learning, Back-Propagation Algorithm (*Ch. 1* of [1], *Ch. 1* of [7] & [9])
3. Unsupervised Learning: *K*-Means Clustering, Principal Components Analysis - PCA, Self-Organizing Maps (SOM), Autoencoders (*Ch. 5* & *8* of [1], *Ch. 10* of [3])
4. Stochastic Models rooted in Statistical Mechanics: Markov Chains, State Transitions, Chapman – Kolmogorov equations, transient & recurrent states, periodicity, irreducibility, asymptotic behavior, ergodicity & stationarity (*Ch. 11* of [1], & [4], [5])
5. Markov Chain Monte Carlo (MCMC) method, Metropolis – Hastings algorithm. Simulated Annealing, Gibbs sampling. Generative Models, Boltzmann Machine, Restricted Boltzmann Machine (RBM), Deep Belief Nets - DBN) (Ch. *11* of [1] & *Ch. 4* of [3])
6. Reinforcement Learning & Dynamic Programming: Markov Decision Processes, Bellman's Optimality Criterion, Value & Policy Iteration optimization algorithms. Approximate methods in dynamic programming, TD & Q-Learning (*Ch. 12* of [1])
7. Reinforcement Learning for Internet Routing: The Bellman-Ford algorithm, Border Gateway Protocols - BGP (*NTUA ECE Course "Network Management – Intelligent Networks"* https://www.netmode.ntua.gr/wp-content/uploads/2023/01/NetMan_IP_Routing_2022_10_31.pdf)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Course Outline (2/2)

8. Kernel Algorithms & Pattern Separability: Cover's Theorem, applications to Radial-Basis Function (RBF) Networks, Hybrid Learning, Support Vector Machines (SVM) (*Ch. 5* & *6* of [1], *Ch. 6* of [13])
9. Non-parametric classifiers, classification of sample elements in preset classes, *K*-Nearest Neighbors (KNN) algorithm (*Ch. 2* of [10], *Ch. 8* of [13])
10. Statistical evaluation of Binary Classification: Confusion Matrix, Receiver Operating Characteristics (ROC) & Area Under the Curve (AUC). Parametric probabilistic classification - Bayes & Naïve Bayes Classifiers (*Ch. 6* of [13], *Ch. 5* of [7])
11. Decision Trees: CART - Classification And Regression Trees algorithms, Gini Index, Random Forests, Bagging - Bootstrap & aggregating (*Ch. 8* of [10])
12. Recurrent Neural Nets - RNN: Associative Memory & Content Addressable Memory – CAM models, Hopfield Networks, RNNs & time/character series (*Ch. 13* & *15* of [1], *Ch. 2* &*3* of [3])
13. Natural Language Processing (NLP), Large Language Models (LLM), Long-Short Term Memory (LSTM) networks, Transformers (***Ch. 8, 9*** & *10* of [8], *Ch. 10* of [7])
14. eXplainable AI (XAI): Definitions, Intrinsic & Model-Agnostic XAI Methods, PI (Permutation Feature Importance), SHAP (SHapley Additive exPlanations), LIME (Local Interpretable Model Agnostic Explanation) (*Ch. 1, 2* & *4* of [15])

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Founding Fathers of Artificial Intelligence

Thomas Bayes (1701 -1761): Combinatorial Probabilities, Statistical Inference
https://en.wikipedia.org/wiki/Thomas_Bayes

Johann Carl Friedrich Gauss (1777 -1855): Statistical Inference, Distributions of Sample Data
https://en.wikipedia.org/wiki/Carl_Friedrich_Gauss

Josiah Willard Gibbs (1839 -1903): Statistical Mechanics, Thermodynamics
https://en.wikipedia.org/wiki/Josiah_Willard_Gibbs

Ludwig Boltzmann (1844 -1906): Statistical Mechanics, Thermodynamics
https://en.wikipedia.org/wiki/Ludwig_Boltzmann

Andrey Markov (1856 -1922): Probability Theory, Stochastic Processes
https://en.wikipedia.org/wiki/Andrey_Markov

Alan Turing (1912 -1954): Computing Machinery, Codes, Artificial Intelligence, Logic
https://en.wikipedia.org/wiki/Alan_Turing

John von Neumann (1903 -1957): Statistical Modelling, Game Theory, Entropy
https://en.wikipedia.org/wiki/John_von_Neumann

Andrey Kolmogorov (1903 -1987): Probability Theory
https://en.wikipedia.org/wiki/Andrey_Kolmogorov

Richard Bellman (1920 - 1984): Dynamic Programming
https://en.wikipedia.org/wiki/Richard_E._Bellman

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Fathers of Machine Learning

Nicholas Metropolis - Μητρόπουλος (1915 - 1999): Monte Carlo Simulations, Simulated Annealing
https://en.wikipedia.org/wiki/Nicholas_Metropolis

Donald Hebb (1904 - 1985): Neurophysiology, Learning Rules
https://en.wikipedia.org/wiki/Donald_O._Hebb

Frank Rosenblatt (1928 - 1972): Psychology & Artificial Intelligence (AI), Neural Networks - Perceptron
https://en.wikipedia.org/wiki/Frank_Rosenblatt

David Rumelhart (1942 - 2011): Psychology & Artificial Intelligence (AI), Back Propagation Algorithm
https://en.wikipedia.org/wiki/David_Rumelhart

Vladimir Vapnik (1936): Statistical Learning, Support Vector Machines (SVM)
https://en.wikipedia.org/wiki/Vladimir_Vapnik

Teuvo Kohonen (1934 - 2021): Self-Organizing Maps (SOM)
https://en.wikipedia.org/wiki/Teuvo_Kohonen

John Hopfield (1933): Physics, Biology, Recurrent Neural Networks (RNN) (***Nobel Prize in Physics, 2024***)
https://en.wikipedia.org/wiki/John_Hopfield

Geoffrey Hinton (1947): Physics, Boltzmann Machines, Deep Belief Networks (***Nobel Prize in Physics, 2024***)
https://en.wikipedia.org/wiki/Geoffrey_Hinton

Demis Hassabis (1976): AI Research, Protein Structure Prediction (***Nobel Prize in Chemistry, 2024***)
https://en.wikipedia.org/wiki/Demis_Hassabis

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Machine Learning & Artificial Intelligence

A Definition of Artificial Intelligence - AI:
Artificial intelligence leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind
(IBM: https://www.ibm.com/topics/artificial-intelligence)

A Definition of Machine Learning - ML:
Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy
(IBM: https://www.ibm.com/topics/machine-learning)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Discriminative Machine Learning Models

Definition:

Classification or *Regression* (estimation) of *data elements* via *conditional probability* estimates of plausible outputs (*label*) given *input sample* elements, based on what the model learns by iteratively feeding sample elements of a *training dataset* and checking *generalization* by applying elements of a *testing dataset*

Applications:

- Classification of sample elements based on their characteristics (features)
- Pattern recognition based on principal sample element features
- Medical imaging, diagnostics semi-automation tools
- Prediction (regression) of output based on pre-stored pairs of input-output elements

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Generative AI Models

Definition:

Generation of sample elements conforming to joint input-output statistics estimated by iterative input of *training sample* elements from which the system infers *joint probabilities* of the *output* with input *features* (virtual reality output, risk of hallucinations)

Applications:

- Bayes classifiers, a very popular and simple generative classification method
- Current hype, with massive training datasets and lengthy training times (months!), expensive environmentally hazardous datacenters requirements, cloud-hosted multiprocessor GPUs (Graphics Processing Units) and highly specialized staffing
- Generation of text elements and chatboxes based on *Large Language Models* - *LLM*, text translation, voice recognition, production of simulated (virtual reality) images, idealized background screens, animated cartoons…
- Extensive training of *Search-Engines* (*Google*, *MS Bing*…), *OpenAI* - *MS ChatGPT* (*Chat Generative Pre-trained Transformer*), *DeepSeek* chatbox…
- Offered As-a-Service (*AaS*) to customers, stirring fierce competition for supremacy amongst the US, China, Europe (?)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Introduction to Machine Learning Concepts (1/6)

Causes-and-Effects of the Artificial Intelligence Revolution:

- The cataclysmic developments of distributed (*cloud*) computing and storage infrastructures enables extremely complex algorithms of statistical inference and stochastic optimization, based on large historical datasets

- Processing of multi-dimensional huge data (*big data*) with a massive number of characteristics (*features*) triggers novel data mining algorithms to *estimate*, *predict*, *classify* and *generate* new sample elements, statistically close to pre-stored historical data

- The ever-deepening understanding of learning methods in biological systems, leads to emulation of *Human Intelligence* via *Artificial Intelligence* algorithms that characterize or generate new instances, always with some *error probability* (expected in statistical inference decisions) and hopefully minute danger to lead to hazardous *hallucinations*

- The advances in *Natural Language Processing* (*NLP*) and *Text Processing* fields, coupled with technology breakthroughs and the ubiquitous *Internet* availability, lead to generative massive models often referred to as *Large Language Models* (*LLMs*), with human-friendly attributes and tremendous commercial potential and geopolitical might

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Introduction to Machine Learning Concepts (2/6)

Risks of the Artificial Intelligence Revolution:

- Artificial Intelligence exhibits risks associated with all (r)evolutions (e.g. widening of global inequalities, re-alignment of work-force, new employment rules) and new challenges (e.g. how to protect *Individual Privacy Rights* – *IPR* & enforce *Intellectual Property*)
- Use of (Generative) AI to spread fake news, promote plagiarism, infringe Intellectual Property (e.g. unauthorized use of Wikipedia texts by *OpenAI* for *ChatGPT* training)
- Humanity will cast *regulations* (e.g. 2016/6/91 EU *General Data Protection Regulation* - *GDPR*) to harness use of big data and smart algorithms (*perhaps a wishful thinking*…)

Geoffrey Hinton: British – Canadian, Born in London UK 1947

- 1977: Ph.D., University of Edinburgh, Scotland, UK
- Academic Career UK, USA, Canada
- Pioneer in Neural Networks research (Boltzmann Machines, Deep Belief Networks, Generative AI…)
- 2013-2023: Scientific Advisor of Google & Professor, University of Toronto
- May 2023: Resigned from Google to freely speak of uncontrollable AI risks
- **Sept. 2024**: ***Nobel Prize in Physics***

NY Times, May 2023 on *G. Hinton*: *"The Godfather of A.I." Leaves Google and Warns of Danger Ahead: Generative A.I. can already be a tool for misinformation. Soon, it could be a risk to jobs. Somewhere down the line, tech's biggest worriers say, it could be a risk to humanity*
https://www.nytimes.com/2023/05/01/technology/ai-google-chatbot-engineer-quits-hinton.html

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Introduction to Machine Learning Concepts (3/6)

Dataset Definitions

https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets

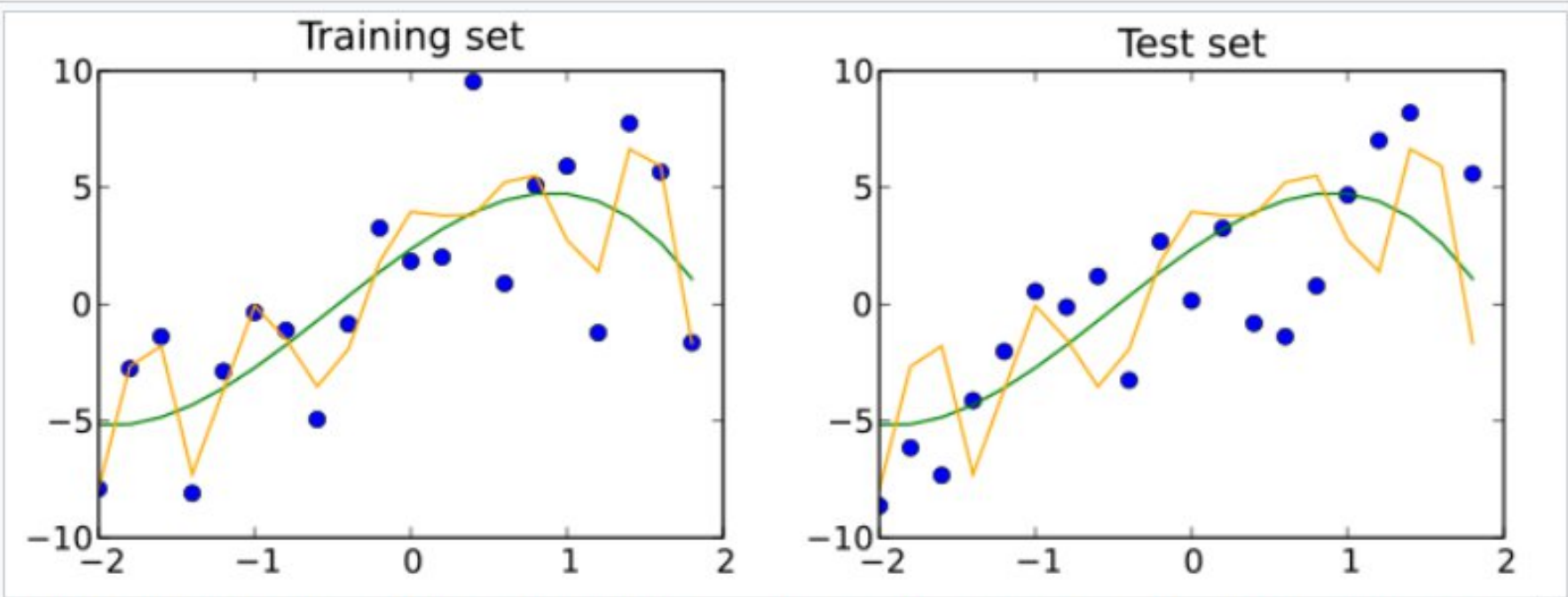
- **Training Set**: *Sample* (set) of examples (sample points or elements) used for tuning the parameters of a specific configuration model during the *training phase* of ML
- **Validation Set**: *Sample* (set) of examples, not used for training, with features similar to the training sample elements, used to validate convergence of the training phase. It unusually leads to the selection of configuration *hyperparameters* of a decision model by comparing various options e.g. in terms of *accuracy* and ability to *generalize* that might suffer from excessive reliance to training examples (*overfitting*). The *validation dataset* could not be defined or (in most cases) it could be a selection of elements, e.g. a 10 - 20% of examples filtered from the *training dataset*
- **Test Set**: *Sample* (set) of examples, not explicitly used for training and validation, inserted as input to a finally selected and tuned ML system. It assesses the *accuracy* and the *generalization* potential of a model fed with unseen data points, prior to its actual production deployment
- In case that no *validation dataset* is defined, performance of a model is deduced directly via the *test dataset*

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Introduction to Machine Learning Concepts (4/6)

Illustration of Overfitting

https://en.wikipedia.org/wiki/Training,_validation,_and_test_sets



A training set (left) and a test set (right) from the same statistical population are shown as blue points. Two predictive models are fit to the training data. Both fitted models are plotted with both the training and test sets. In the training set, the MSE of the fit shown in orange is 4 whereas the MSE for the fit shown in green is 9. In the test set, the MSE for the fit shown in orange is 15 and the MSE for the fit shown in green is 13. The orange curve severely overfits the training data, since its MSE increases by almost a factor of four when comparing the test set to the training set. The green curve overfits the training data much less, as its MSE increases by less than a factor of 2.

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Introduction to Machine Learning Concepts (5/6)

Parameters & Hyperparameters

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

- The *parameters* tuned during training refer to a specific model structure (e.g. determination of a neural network synaptic wrights. They are determined by iterations on a specific input-output configuration that aim to converge in an estimated accurate ML model, subsequently fed by training sample points
- Parameters concerning the model structure (e.g. number of neurons, layers of hidden neurons) and to convergence criteria are referred to as *hyperparameters*
- The *hyperparameters* are selected based on the designer experience and/or with repetitions of parameter tuning (*training*) and *validation* to improve model *accuracy* prior to the *testing* of the ML model
- Search methods for *hyperparameters* selection include Exhaustive search, Grid search, random search… depending on the acceptable number of trials of tuned models
- The selection of *hyperparameters* if there is no *validation dataset* can be performed with repeated trials over the *test dataset*

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Introduction to Machine Learning Concepts (6/6)

General Machine Learning Categories

- **Supervised Learning**
  - Training sample elements (examples) include an output label, appended to input features (*labeled training sample points*). The system parameters are tuned to minimize output – label deviations by iteratively applying training examples
- **Unsupervised Learning**
  - The system infers input sample statistics from the training sample elements, with no guidance from a priori known output values (labels). The system tunes its parameters to fit important (by assumption) statistical properties of the training dataset by discovering *stochastic features, patterns* present in a large number of training examples (*unlabeled training datasets*). This converges to predictions of applicable models and methods e.g. for assigning input elements into *clusters*
- **Reinforcement Learning**
  - This category is closely related to stochastic control systems that affect the *state evolution* of an outside *environment*. The system (a controller) reacts to reinforcement signals from external *critics* related to the environment, who enforce *actions* that may impact the evolution of the environment *state* towards a long-term expected cost objective. Parameter tuning may persist beyond a training period, as the actions can *dynamically* affect the evolution of the environment

Supervised learning results into efficient, fast and reliable convergence for decision making problems. But the requirement for not readily available **labeled datasets** often favors unsupervised methods

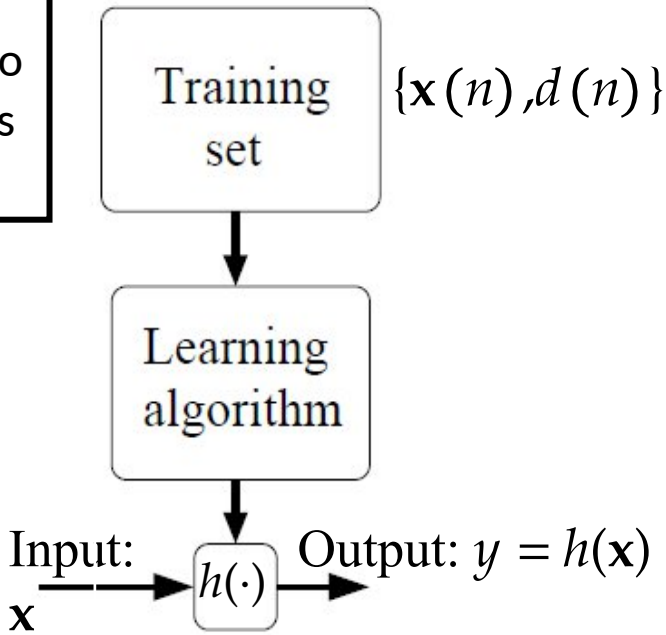
STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Generic Model of Supervised Learning

Based on Andrew Ng, "CS229 Lecture Notes", Stanford University, Fall 2018

- The system goal is to assign input vectors (***input sample points***, ***examples***, ***instances***) $\mathbf{x} = [x_1 \; x_2 ... x_m]^{\mathrm{T}}$ to output values y (***targets***, ***response values***). The coordinates x_i encode m characteristics (***features***) of the input vector $\mathbf{x}$

We seek the input-output function $y = h(\mathbf{x}) \cong d$ that minimizes deviations (errors) between the ***label*** d (known to an external ***supervisor***) and the response y for input vectors in the ***Training Set*** of N pairs $\{\mathbf{x}(n), d(n)\}$, $n = 1,2,...,N$



- The form and parameters of $h(\cdot)$ result from the learning algorithm that converges to the system goal for the N elements of the training sample

$$d(n) \cong y(n) = h(\mathbf{x}(n))$$

- If y is a finite integer we have a ***Classification*** problem (for 2 classes we have binary classification)

- If y assumes continuous real values we have a ***Regression*** problem

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

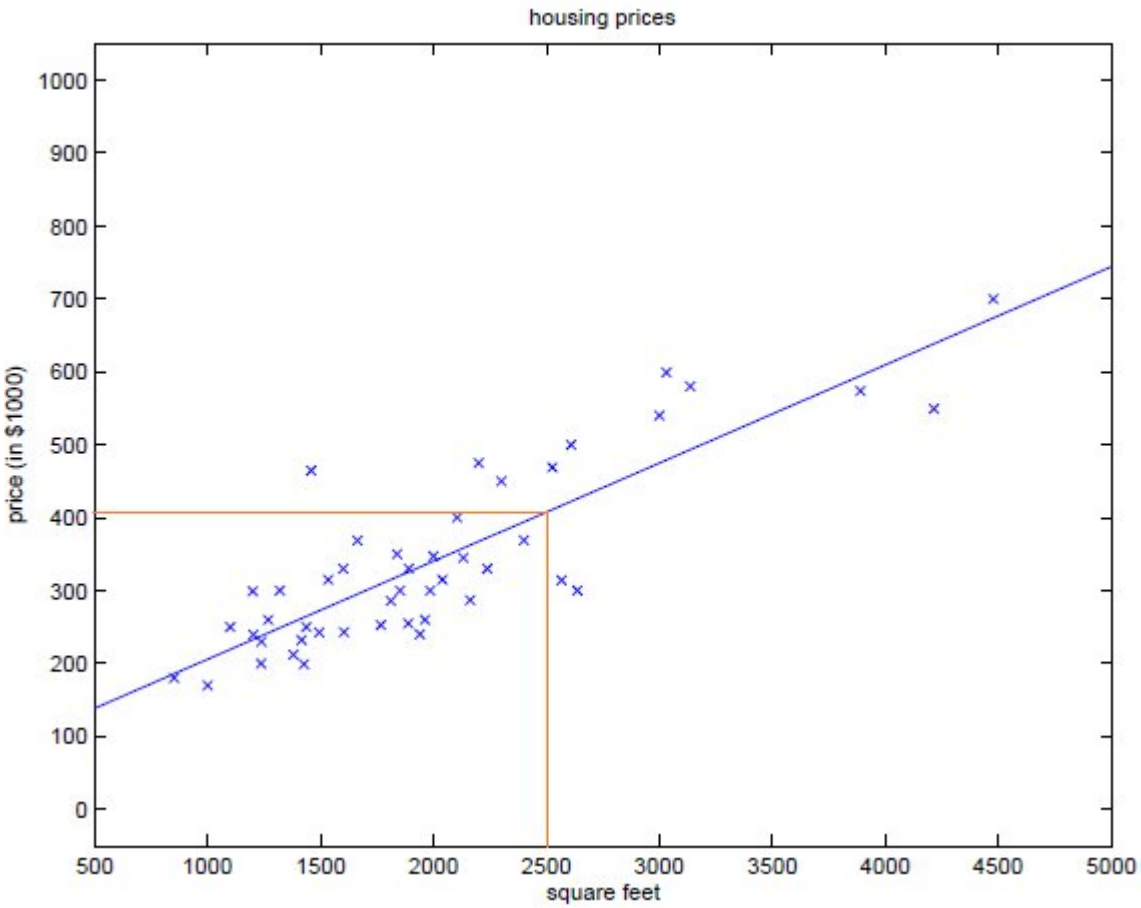
Supervised Learning Example – Linear Regression

Based on Andrew Ng, "CS229 Lecture Notes", Stanford University, Fall 2018

Find the *Linear Function* $h(x)$ that predicts the value $y = h(x)$ based on the property area x the *Labeled Training Sample* $\mathscr{D} = \{(x(1), d(1)), \ldots, (x(N), d(N))\}$ of property sales registered in the area

AREA (square feet)	VALUE (1000$)
2104	400
1600	330
2400	369
1416	232
3000	540
…	…

Sample Elements for $N = 47$ Cases



Linear Regression: $y = h(x) = 0.1392\,x + 89.6$

Predicted Value for a Property of 2500 square feet: $437,000

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Linear Regression Parameter Tuning (1/2)

- The coordinates of the input vector $\mathbf{x} = [x_0\ x_1 ... x_m]^{\mathrm{T}}$ correspond to encoding of its m ***features***: $x_1, x_2, ..., x_m$ with $x_0 \triangleq 1$ (***intercept term***)
- The linear regression system computes parameters $\mathbf{w} = [w_0\ w_1 ... w_m]^{\mathrm{T}}$ of the function $y = h_{\boldsymbol{w}}(\mathbf{x}) = w_0 x_0 + w_1 x_1 + ... + w_m x_m = \mathbf{w}^{\mathbf{T}}\mathbf{x}$ that yield to small deviations for the labeled ***Training Set*** $\mathscr{D} = \{(\mathbf{x}(1), d(1)), ..., (\mathbf{x}(N), d(N))\}$
  - $\mathbf{x}(n)$: Input training vector $n = 1,2,..., N$ (***regressors***)
  - $d(n)$: Known output value (*label*) of training vector n (***regressand***)
  - $y(n) = h_{\boldsymbol{w}}(\mathbf{x}(n))$: Current system response for training vector $\mathbf{x}(n)$
  - $\varepsilon(n) = d(n) - y(n)$: Deviation (***error***) for training input $\{\mathbf{x}(n), d(n)\}$, $n = 1,2,...,N$
  - For large N, $\mathbf{x}(n)$, $d(n)$, $\varepsilon(n)$ can be viewed as sample values of random variables
- Convergence criterion: ***Least Mean Square*** - ***LMS*** yielding the parameters $\mathbf{w}$ of $h_{\mathbf{w}}(\mathbf{x})$ or equivalently determine $\mathbf{w}$ that minimizes the ***Mean Square Error*** (***MSE***) $J(\mathbf{w})$

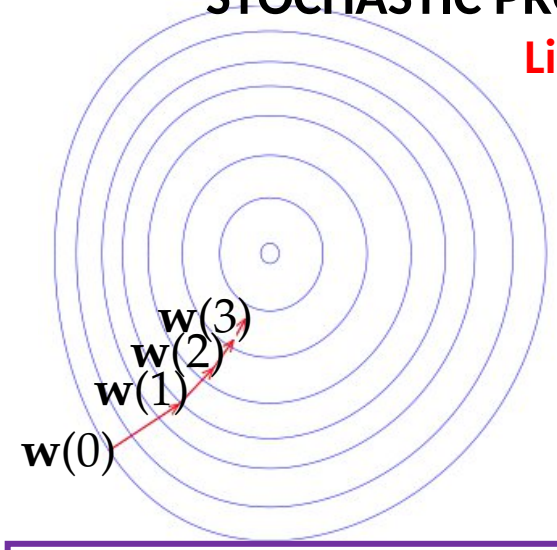
$$J(\mathbf{w}) \triangleq \frac{1}{2}\sum_{n=1}^{N}[\varepsilon(n)]^2 = \frac{1}{2}\sum_{n=1}^{N}[d(n) - h_{\boldsymbol{w}}(\mathbf{x}(n))]^2$$

- With one input variable $x_0 = 1, x_1 = x$ the ***Linear Regression*** is formulated as:

$$\mathbf{x} = [1\ x]^{\mathrm{T}},\ \mathbf{w} = [w_0\ w_1]^{\mathrm{T}},\ \ \mathrm{y} = h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^{\mathbf{T}}\mathbf{x} = w_0 + w_1 x$$

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}[d(n) - (w_0 + w_1 x(i))]^2$$

Linear Regression Parameter Tuning (2/2)



Gradient Descent Minimization Algorithm:

Minimize $J(\mathbf{w})$ in terms of the parameter vector $\mathbf{w}$ by successive descent at the iteration $k \to k+1$ towards the Gradient $\nabla J(\mathbf{w})$, weighted by the *hyperparameter* α:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha \nabla J(\mathbf{w}(k))$$

If the algorithm converges: $\mathbf{w} = \lim_{k \to \infty} \mathbf{w}(k)$

Note: Convergence is guaranteed for linear regression

LMS Convergence Options (Widrow-Hoff)

- **Batch Gradient Descent:** Search for $\mathbf{w} = [w_0 \; w_1 \ldots w_m]^{\mathrm{T}}$ that minimizes $J(\mathbf{w})$ by considering in every iteration the ***whole*** training set $\mathcal{D} = \{(\mathbf{x}(1), d(1)), \ldots, (\mathbf{x}(N), d(N))\}$

$$w_j := w_j - \alpha \frac{\partial J(\mathbf{w})}{\partial w_j} = w_j + \alpha \sum_{n=1}^{N} [d(n) - h_{\mathbf{w}}(\mathbf{x}(n))] x_j(n), \quad j = 0,1,2,\ldots,m \quad \forall i$$

- **Stochastic (Incremental) Gradient Descent, Stochastic Approximations**: Search for $\mathbf{w}$ by considering a random (stochastic) sequence of ***single*** input training elements $(\mathbf{x}(n), d(n))$, $n = 1,2,\ldots,N$ until convergence

$$w_j := w_j + \alpha [d(n) - h_{\mathbf{w}}(\mathbf{x}(n))] x_j(n), \quad j = 0,1,2,\ldots,m \qquad n = 1,2,\ldots,N$$

  - The stochastic method usually yields good results with no considerable use of computational resources, thus it is preferred for ML applications
  - The step α in successive iterations determines the ***learning rate***. It could vary to stabilize convergence, e.g. set α to a large value to start with and fine tune it as we approach convergence

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Polynomial Regression for Sample Elements of a Single Feature

Based on Andrew Ng, "CS229 Lecture Notes", Stanford University, Fall 2018

- Linear Regression for input element $\mathbf{x} = [1\ x]^{\mathrm{T}}$:

$$y = h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x$$

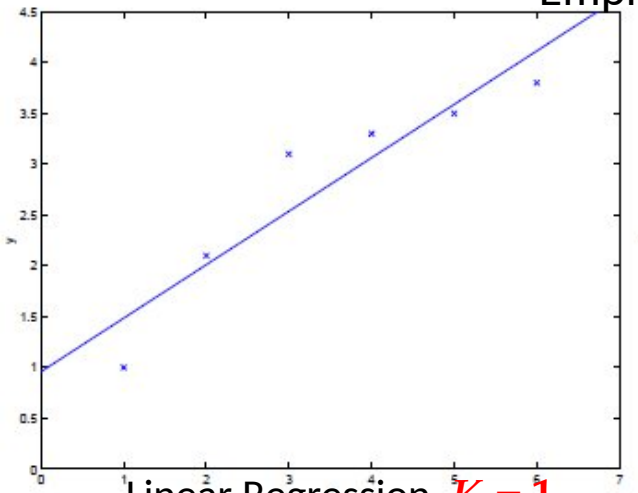
- Regression for $\mathbf{x} = [1\ x]^{\mathrm{T}}$ with 2nd degree polynomial:

$$y = h_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x + w_2 x^2$$

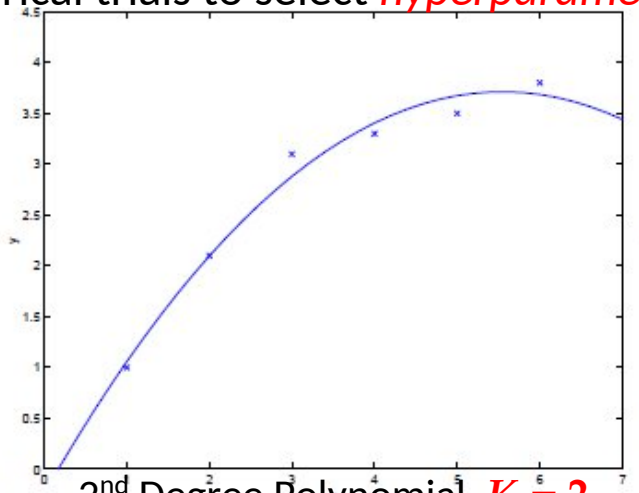
- Regression for $\mathbf{x} = [1\ x]^{\mathrm{T}}$ with K degree polynomial (*hyperparameter* K):

$$y = h_{\mathbf{w}}(\mathbf{x}) = \sum_{j=0}^{K} w_j x^j$$

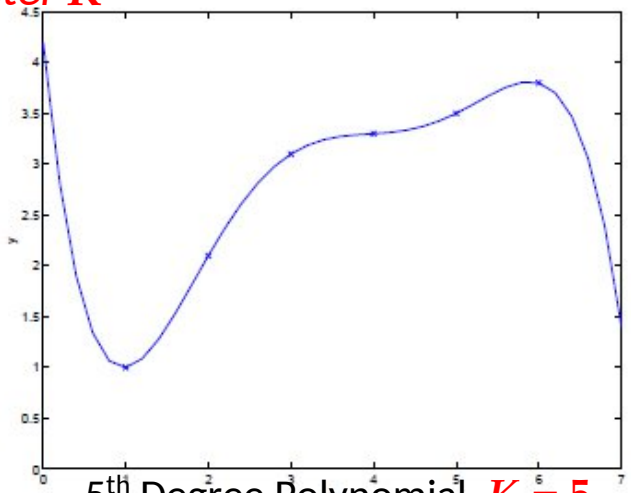
Empirical trials to select *hyperparameter* K



Linear Regression, $K = 1$
(***Underfitting***)



2nd Degree Polynomial, $K = 2$
(***OK***)



5th Degree Polynomial, $K = 5$
(***Overfitting***)

$K = 1$: Oversimplification (***Underfitting***)
$K = 5$: Exaggeration (***Overfitting***)

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Classification (1/2)

Based on Andrew Ng, "CS229 Lecture Notes", Stanford University, Fall 2018

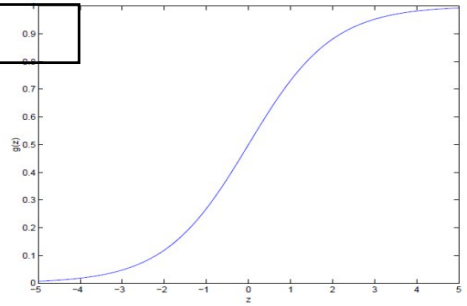
Sample vectors $\mathbf{x}$ of m dimensions (***features***)

Binary Output (***Classes***, ***Labels***): $y \in \{0,1\}$ or $y \in \{-,+\}$

Training Set: $\{(\mathbf{x}(1),d(1)),\ldots,(\mathbf{x}(N),d(N))\}$

- **Logistic Regression Model**: $h_{\mathbf{w}}(\mathbf{x}) = g\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right) = \dfrac{1}{1+e^{-\mathbf{w}^{\mathrm{T}}\mathbf{x}}}$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_{j=0}^{m} w_j x_j = w_0 + \sum_{j=1}^{m} w_j x_j$$



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

Logistic/Sigmoid Function

Assumption: Considering that y and $\mathbf{x}$ are random variables/vectors, the ***Conditional Probabilities*** of $y \in \{0,1\}$ given an input $\mathbf{x}$, provided that the system operates according to the ***Logistic Regression*** $h_{\mathbf{w}}(\mathbf{x}) = \dfrac{1}{1+e^{-\mathbf{w}^{\mathrm{T}}\mathbf{x}}}$ are ***Bernoulli*** distributed. With determined parameters $\mathbf{w}$ these are given by:

$$P(y=1|\mathbf{x};\mathbf{w}) = h_{\mathbf{w}}(\mathbf{x}),\quad P(y=0|\mathbf{x};\mathbf{w}) = 1-h_{\mathbf{w}}(\mathbf{x})$$
$$\text{or}\quad p(y|\mathbf{x};\mathbf{w}) = (h_{\mathbf{w}}(\mathbf{x}))^{y}(1-h_{\mathbf{w}}(\mathbf{x}))^{1-y}$$

Assignment Rule of y :

$y = 1$ αν $h_{\mathbf{w}}(\mathbf{x}) > 1/2$

$y = 0$ αν $h_{\mathbf{w}}(\mathbf{x}) < 1/2$

Due to the non-linear nature of $h_{\mathbf{w}}(\mathbf{x})$ the minimization ***LMS*** objective is replaced by the equivalent maximization ***Likelihood Ratio*** $L(\mathbf{w})$ over the training sample elements $\{\mathbf{x}(n),d(n)\}, n = 1,2,\ldots,N$. Additionally, assuming that labels $d(n)$ in the training sample $\{(\mathbf{x}(1),d(1)),\ldots,(\mathbf{x}(N),d(N))\}$are independent binary random variables we obtain:

$$L(\mathbf{w}) \triangleq p\{(d(1),d(2),\ldots,d(N))|(\mathbf{X};\mathbf{w})\} = \prod_{n=1}^{N} p\{(d(n)|(\mathbf{x}(n);\mathbf{w})\}$$

STOCHASTIC PROCESSES & OPTIMIZATION IN MACHINE LEARNING

Ταξινόμηση – Classification (2/2)

Based on Andrew Ng, "CS229 Lecture Notes", Stanford University, Fall 2018

- **Logistic Regression** (*follow up*) :

$$L(\mathbf{w}) = \prod_{n=1}^{N} p\{(d(n)|(\mathbf{x}(n);\mathbf{w})\} = \prod_{n=1}^{N} \left\{(h_{\mathbf{w}}(\mathbf{x}(n)))^{d(n)} (1-h_{\mathbf{w}}(\mathbf{x}(n)))^{1-d(n)}\right\}$$

Instead of maximizing $L(\mathbf{w})$ we maximize its logarithm $l(\mathbf{w}) = \log L(\mathbf{w})$:

$$l(\mathbf{w}) = \sum_{n=1}^{N} \left\{d(n) \log h_{\mathbf{w}}(\mathbf{x}(n)) + (1-d(n)) \log (1-h_{\mathbf{w}}(\mathbf{x}(n)))\right\}$$

and apply ***Gradient Ascent*** in iteration $k \rightarrow k+1$ with positive *hyperparameter* α :

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha \nabla l(\mathbf{w}(k))$$

To evaluate $\nabla l(\mathbf{w}(k))$ and apply the ***Stochastic Gradient Ascent*** to determine the parameters w_j with successive application to the $n = 1,2,...,N$ vectors of the ***Training Set***, we evaluate the partial derivatives $\frac{\partial}{\partial w_j} l(\mathbf{w}) = ... = [d(n)-h_{\mathbf{w}}(\mathbf{x}(n))]\, x_j(n) \Rightarrow$

$$w_j := w_j + \alpha [d(n)-h_{\mathbf{w}}(\mathbf{x}(n))]\, x_j(n), \quad j = 1,2,...,m \text{ and } n = 1,2,...,N$$

- **The Perceptron Model**: $h_{\mathbf{w}}(\mathbf{x}) = g\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right)$ where:

$$g(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$ **Threshold Function**

Similarly, we obtain an ***Iterative Learning Rule*** to determine the w_j parameters:

$$w_j := w_j + \alpha [d(n)-h_{\mathbf{w}}(\mathbf{x}(n))]\, x_j(n), \quad j = 1,2,...,m \text{ and } n = 1,2,...,N$$