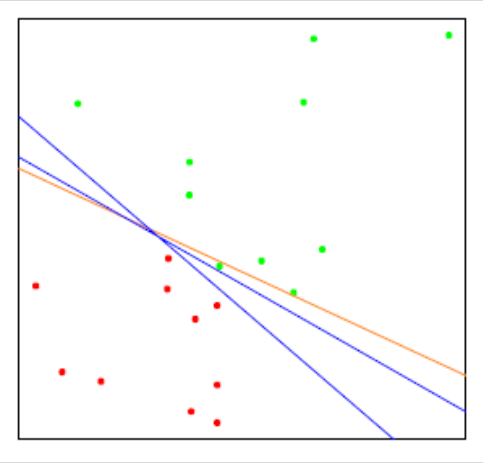


Parallel & Scalable Data Analysis

Introduction to Machine Learning Algorithms

Dr. – Ing. Morris Riedel
Adjunct Associated Professor
School of Engineering and Natural Sciences, University of Iceland
Research Group Leader, Juelich Supercomputing Centre, Germany

LECTURE 3

Supervised Classification & Applications

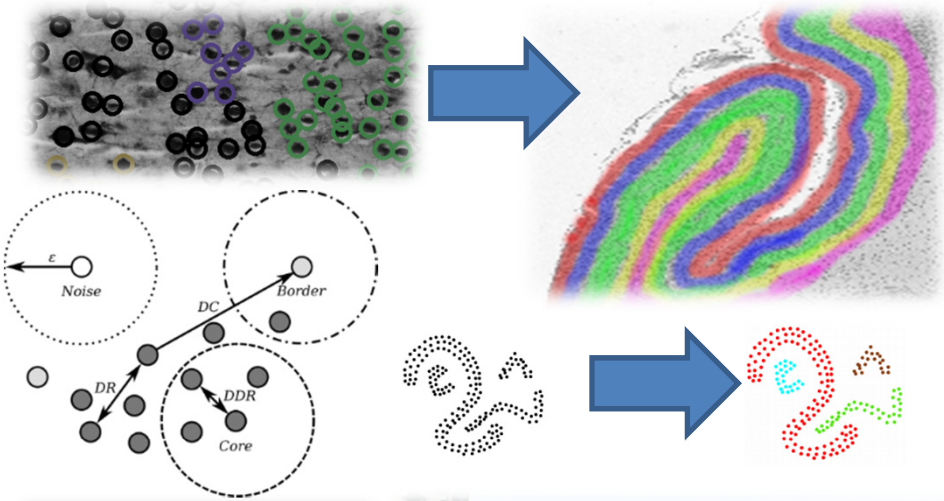
November 23th, 2017
Ghent, Belgium

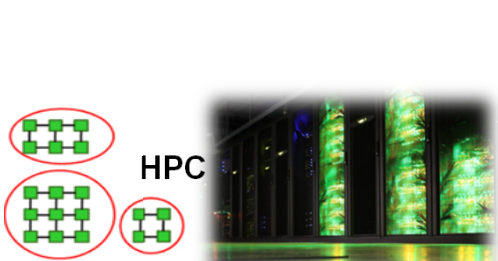
UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

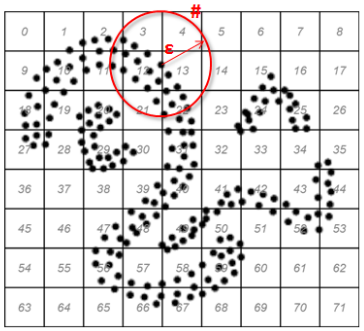
JÜLICH
FORSCHUNGSZENTRUM

Review of Lecture 2

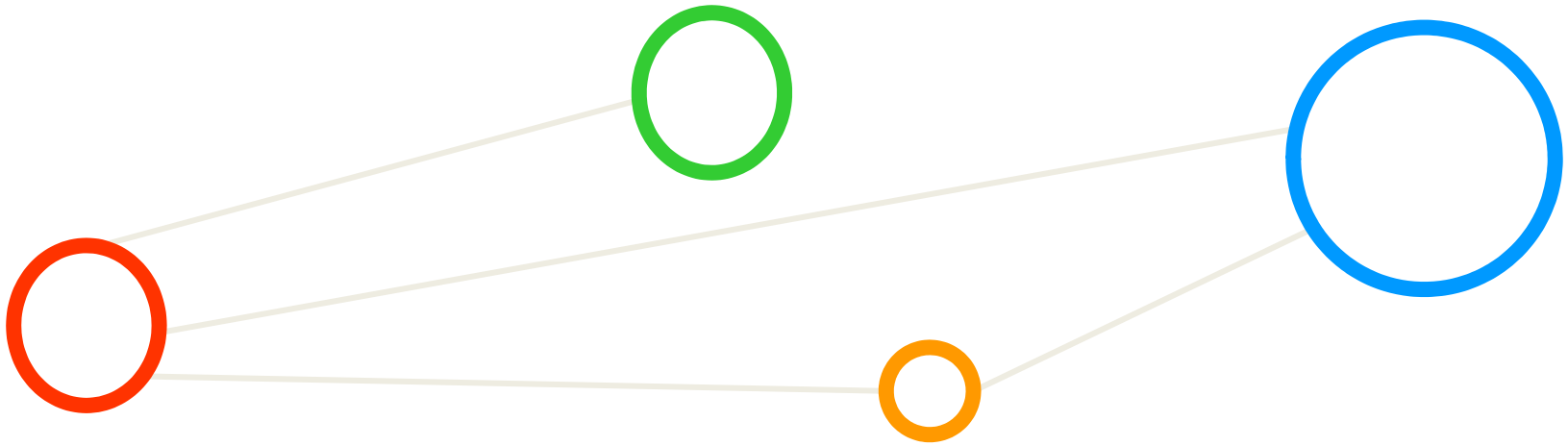
- Unsupervised Clustering
  - K-Means & K-Median
  - DBSCAN very effective
  - Applications in Context
- Point Cloud Datasets
  - 3D/4D laser scans
  - Cities, Buildings, etc.
  - Big Data: Whole Countries
- Parallel Computing
  - HPC and Cluster Environments
  - Massively parallel HPDBSCAN
  - Applied to Point Cloud Datasets







Outline



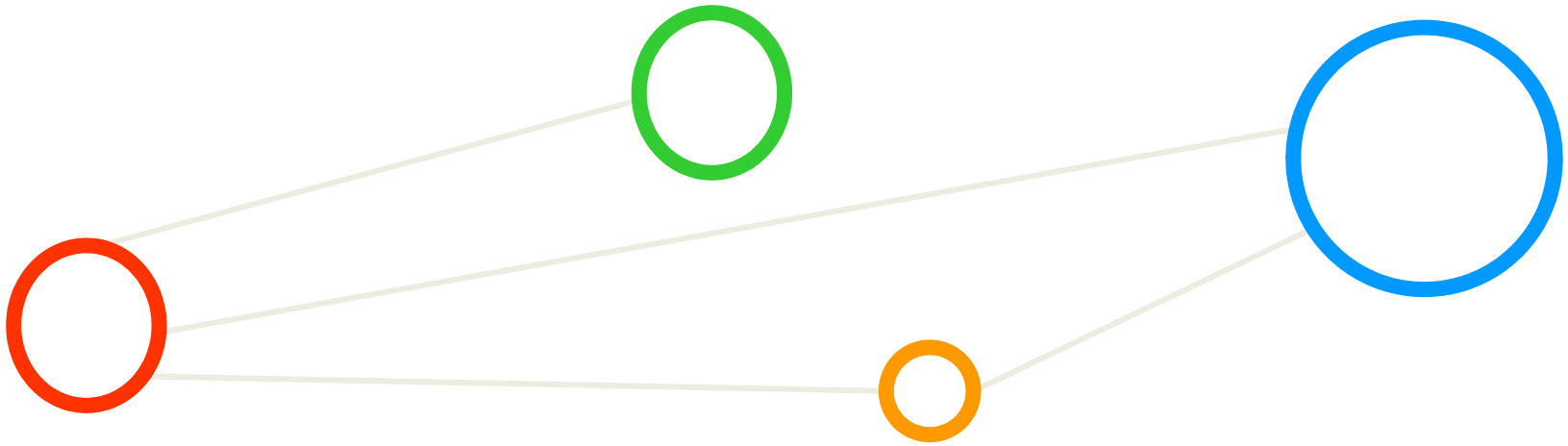
Outline of the Course

1. Machine Learning Fundamentals
2. Unsupervised Clustering and Applications
3. Supervised Classification and Applications
4. Classification Challenges and Solutions
5. Regularization and Support Vector Machines
6. Validation and Parallelization Benefits

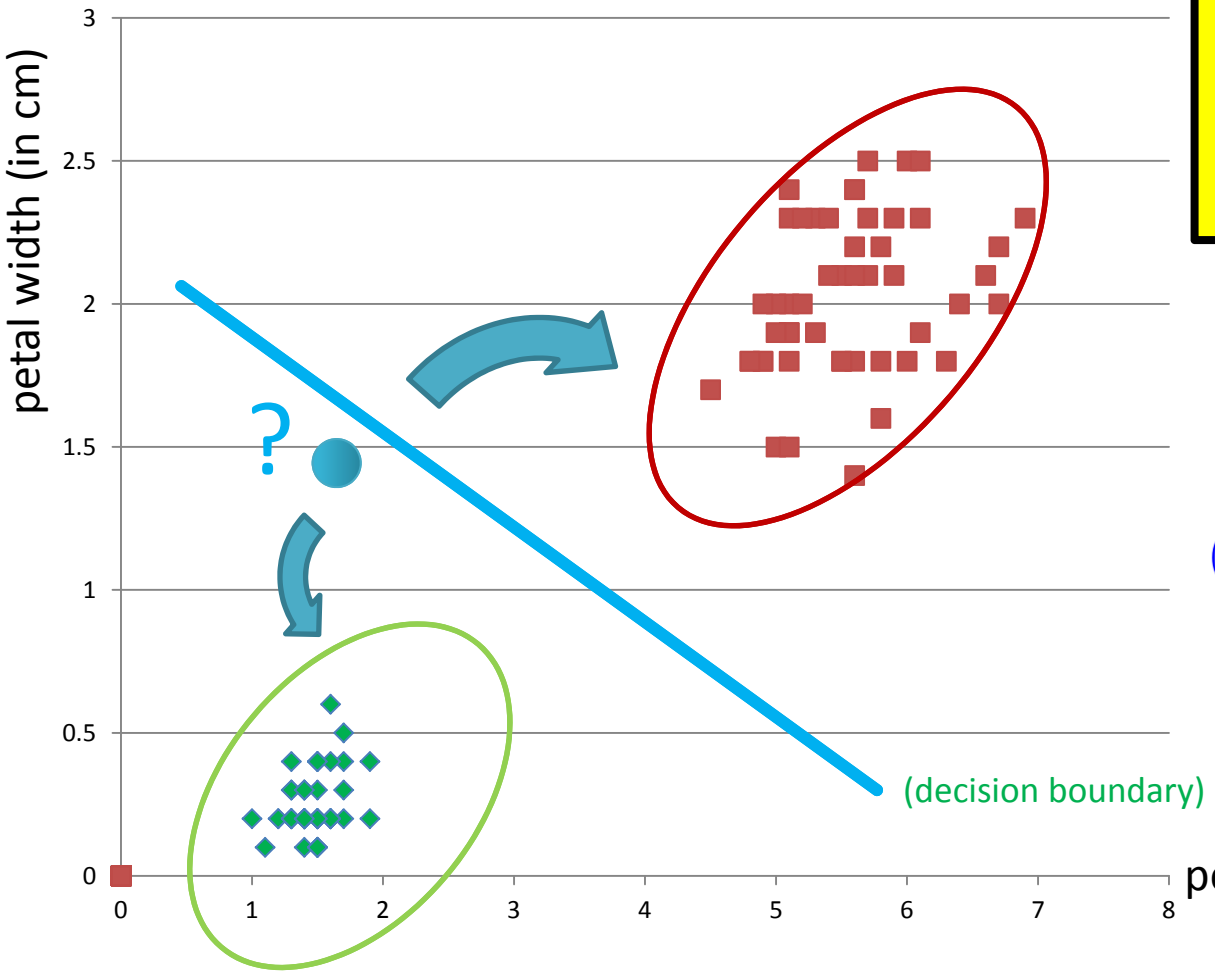
Outline

- Supervised Classification Approach
  - Formalization of Machine Learning
  - Mathematical Building Blocks
  - Feasibility of Learning
  - Statistical Learning Theory in Short
  - Theory of Generalization
  - Linear Perceptron Example in Context
  - Problem of Overfitting
- Remote Sensing Applications
  - Introduction to Application Domain
  - Rome Dataset
  - Indian Pines Dataset
  - Explore need for Parallelization

Supervised Classification Approach



Learning Approaches – Supervised Learning Revisited



- **Example of a very simple linear supervised learning model: The Perceptron**

Iris-setosa

Iris-virginica

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(N = 100 samples)

Learning Approaches – Supervised Learning – Formalization

- Each observation of the predictor measurement(s) has an associated response measurement:
  - Input $\mathbf{x} = x_1, ..., x_d$
  - Output $y_i, i = 1, .., n$
  - Data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$

Training Examples

$$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$$

(historical records, groundtruth data, examples)

- Goal: Fit a model that relates the response to the predictors
  - Prediction: Aims of accurately predicting the response for future observations
  - Inference: Aims to better understanding the relationship between the response and the predictors

- **Supervised learning approaches fits a model that related the response to the predictors**
- **Supervised learning approaches are used in classification algorithms such as SVMs**
- **Supervised learning works with data = [input, correct output]**

[1] An Introduction to Statistical Learning

Feasibility of Learning

- **Statistical Learning Theory deals with the problem of finding a predictive function based on data**

[2] Wikipedia on 'statistical learning theory'

- Theoretical framework underlying practical learning algorithms
  - E.g. Support Vector Machines (SVMs)
  - Best understood for 'Supervised Learning'

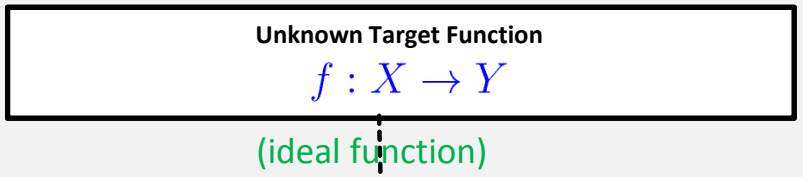
- Theoretical background used to solve 'A learning problem'
  - Inferring one 'target function' that maps between input and output
  - Learned function can be used to predict output from future input (fitting existing data is not enough)

Unknown Target Function

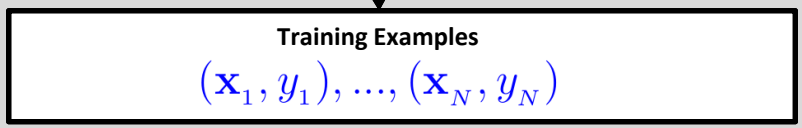
$$f : X \rightarrow Y$$

(ideal function)

Mathematical Building Blocks (1)



(ideal function)



(historical records, groundtruth data, examples)

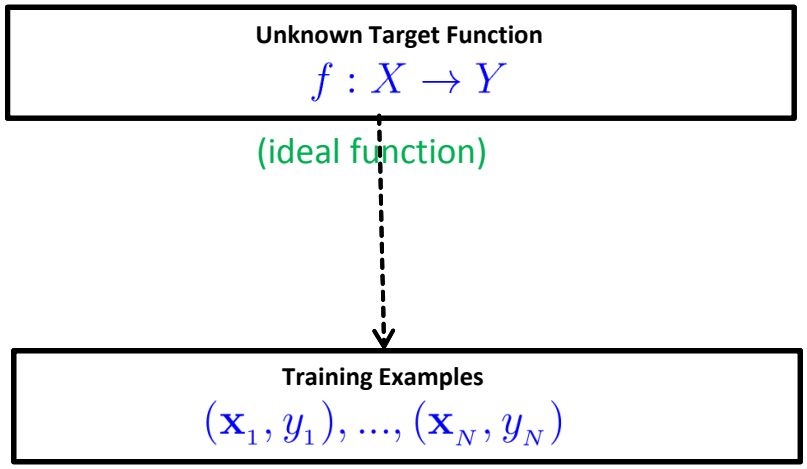
Elements we not exactly (need to) know

Elements we must and/or should have and that might raise huge demands for storage

Elements that we derive from our skillset and that can be computationally intensive

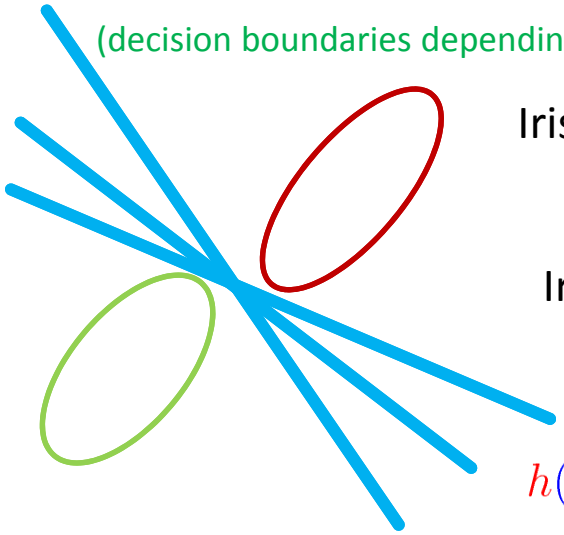
Elements that we derive from our skillset

Mathematical Building Blocks (1) – Our Linear Example



1. Some pattern exists
2. No exact mathematical formula (i.e. target function)
3. Data exists

(if we would know the exact target function we dont need machine learning, it would not make sense)



Iris-virginica if $\sum_{i=1}^{d} w_i x_i > threshold$

Iris-setosa if $\sum_{i=1}^{d} w_i x_i < threshold$

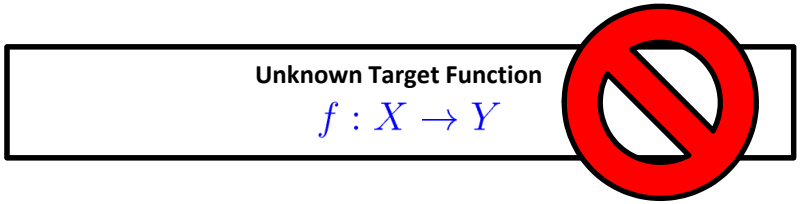
(w_i and threshold are still unknown to us)

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right); h \in \mathcal{H}$$

(we search a function similiar like a target function)

Feasibility of Learning – Hypothesis Set & Final Hypothesis

- The ‘ideal function‘ will remain unknown in learning

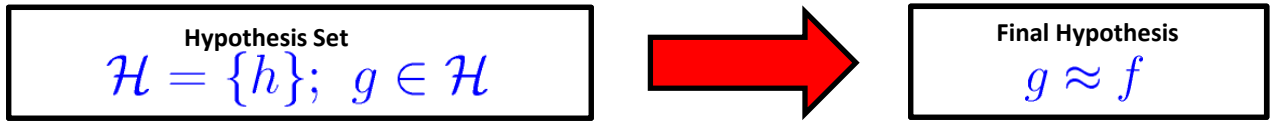


  - Impossible to know and learn from data
  - If known a straightforward implementation would be better than learning
  - E.g. hidden features/attributes of data not known or not part of data

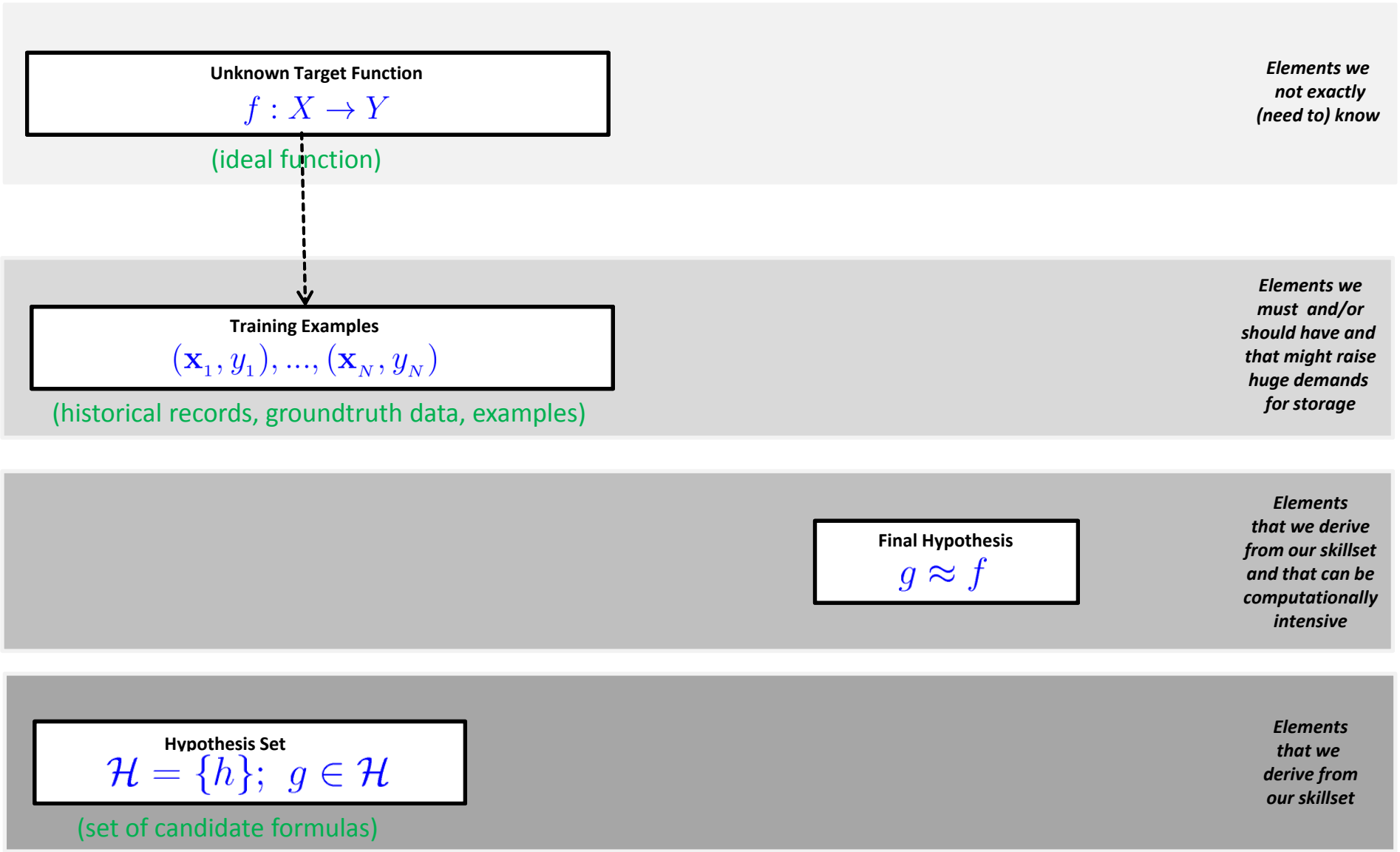
- But ‘(function) approximation‘ of the target function is possible
  - Use training examples to learn and approximate it
  - Hypothesis set $\mathcal{H}$ consists of m different hypothesis (candidate functions)

$$\mathcal{H} = \{h_1, \ldots, h_m\};$$

‘select one function‘ that best approximates $g : X \rightarrow Y$

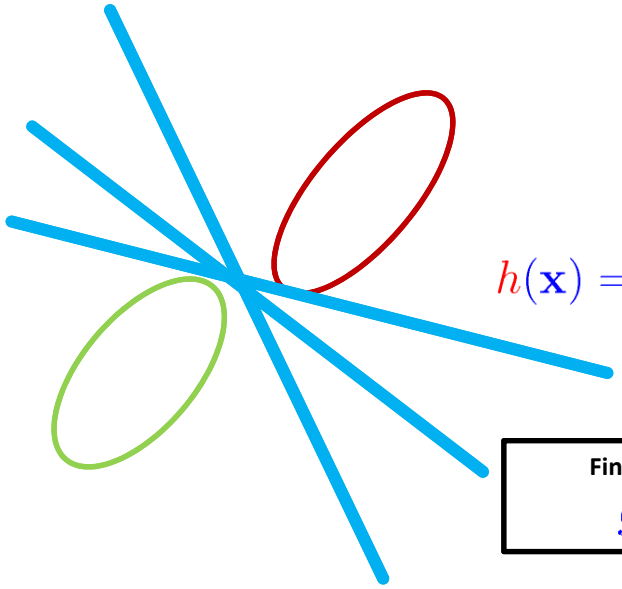


Mathematical Building Blocks (2)



Mathematical Building Blocks (2) – Our Linear Example

(decision boundaries depending on f)



$$\mathcal{H} = \{h_1, \ldots, h_m\};$$

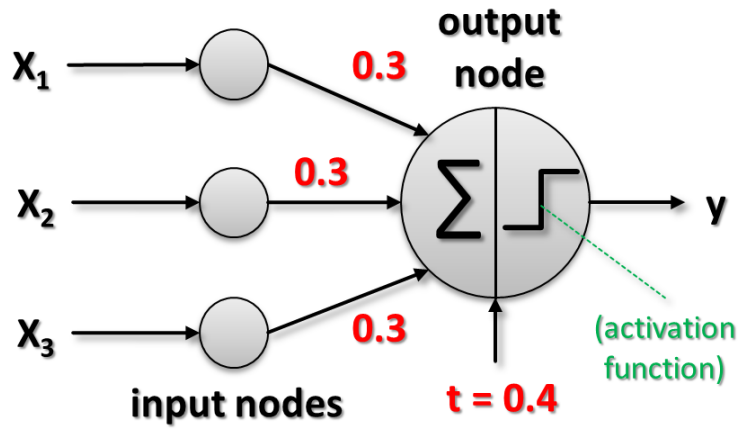
(we search a function similiar like a target function)

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right); h \in \mathcal{H}$$

Final Hypothesis

$$g \approx f$$

$$h(\mathbf{x}) = sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right); h \in \mathcal{H}$$



Hypothesis Set

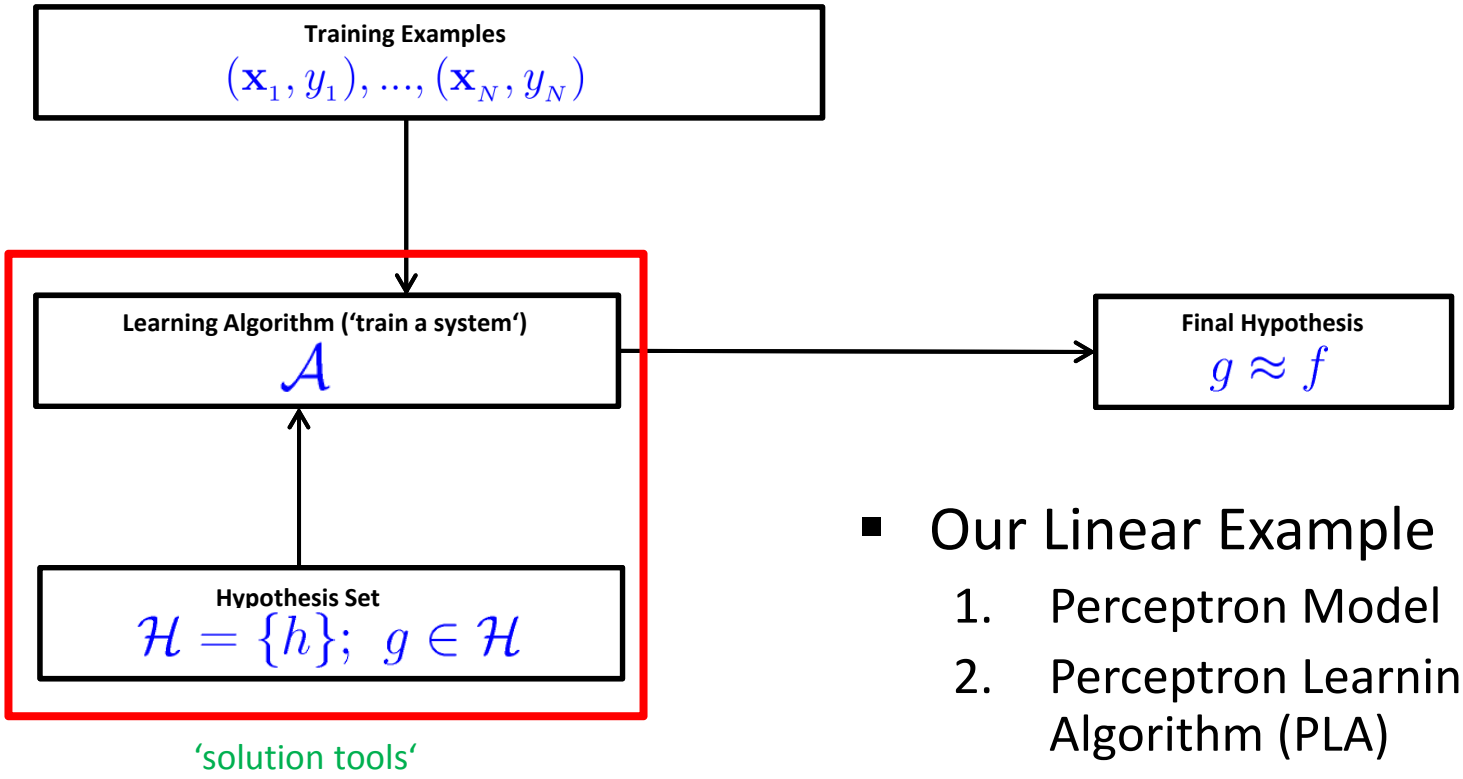
$$\mathcal{H} = \{h\};\ g \in \mathcal{H}$$

(Perceptron model – linear model)

(trained perceptron model and our selected final hypothesis)

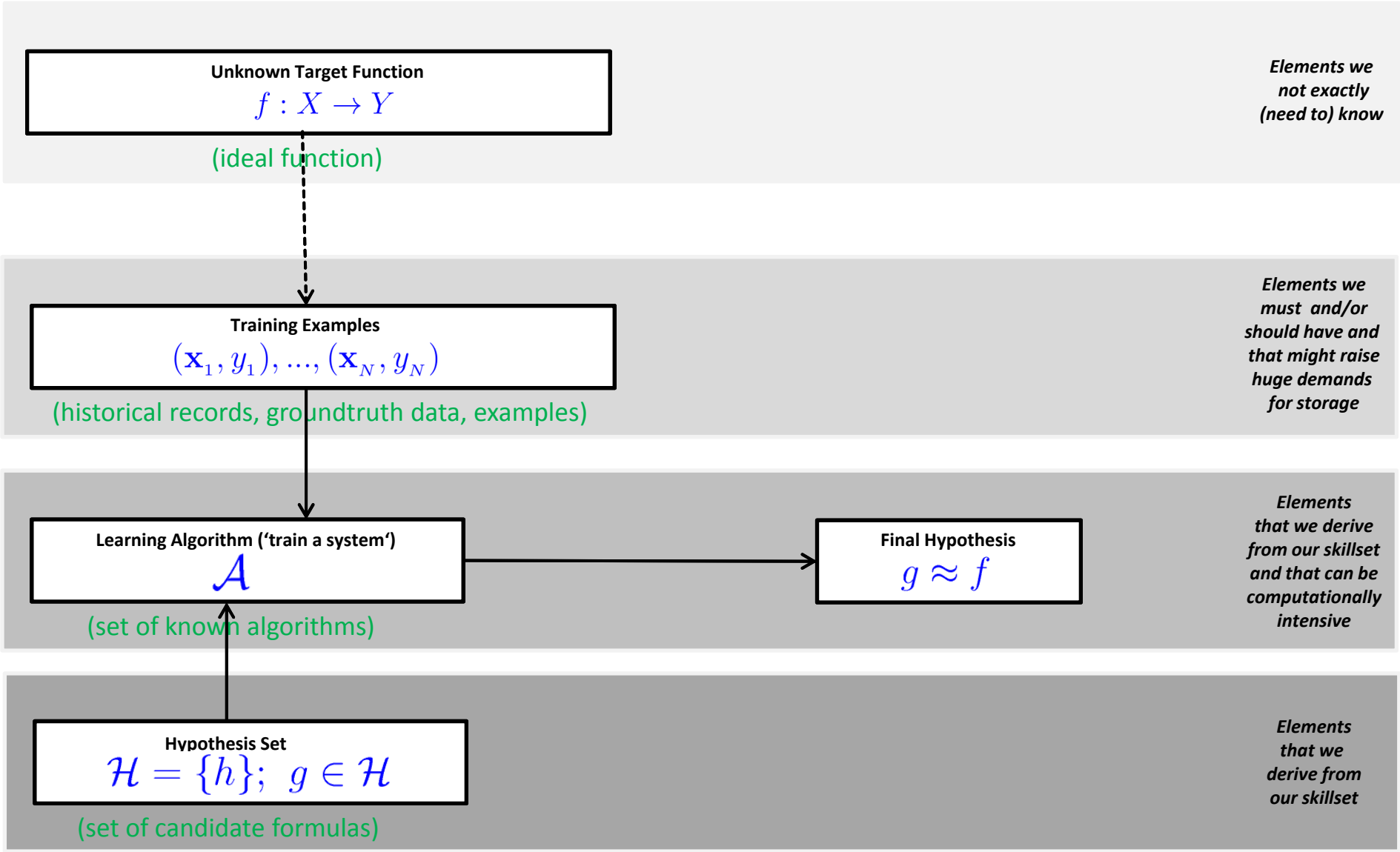
The Learning Model: Hypothesis Set & Learning Algorithm

- The solution tools – the learning model:
  1. Hypothesis set $\mathcal{H}$ - a set of candidate formulas /models
  2. Learning Algorithm $\mathcal{A}$ - 'train a system' with known algorithms

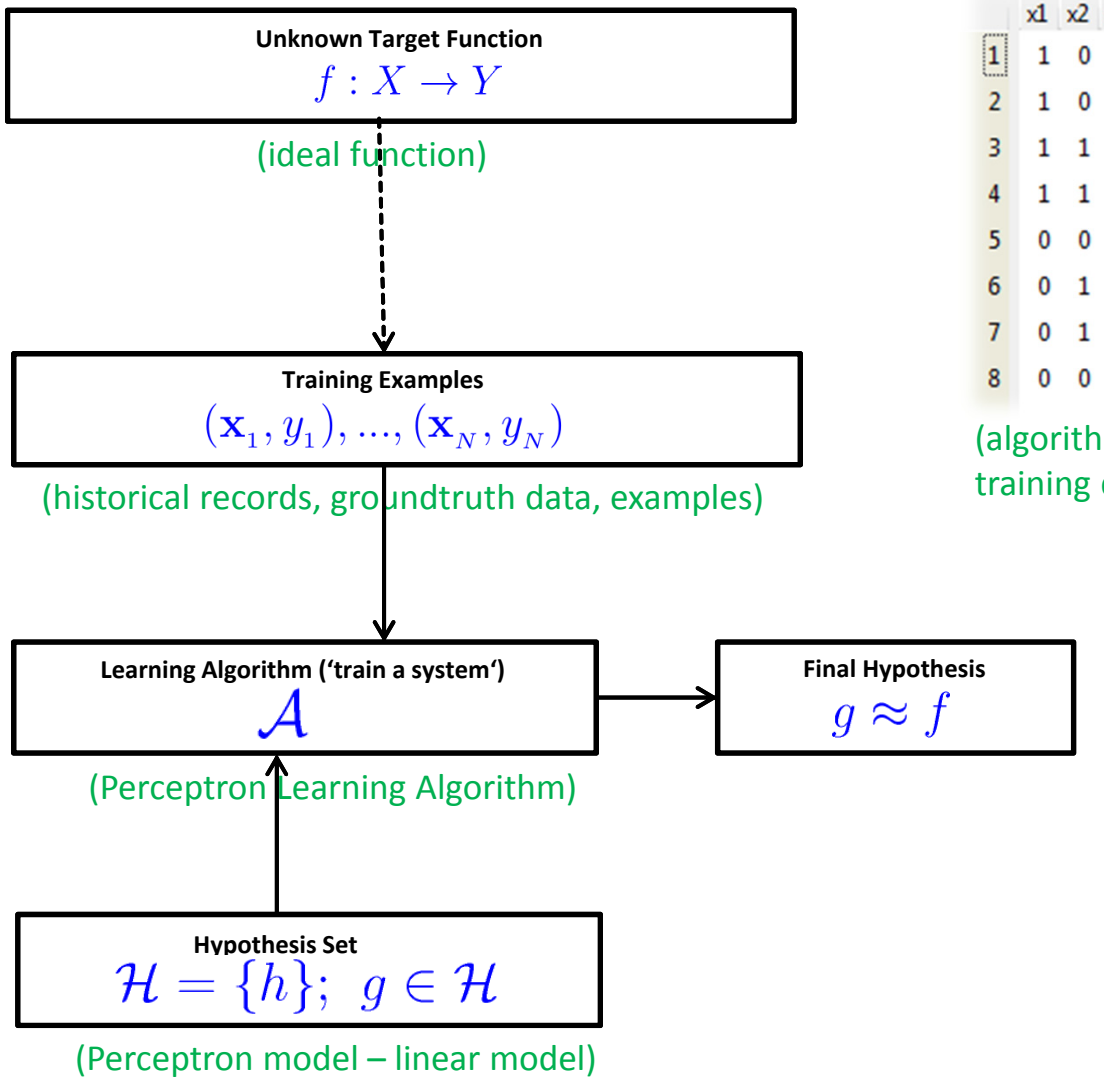


- Our Linear Example
  1. Perceptron Model
  2. Perceptron Learning Algorithm (PLA)

Mathematical Building Blocks (3)



Mathematical Building Blocks (3) – Our Linear Example



	x1	x2	x3	y
1	1	0	0	-1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	1
5	0	0	1	-1
6	0	1	0	-1
7	0	1	1	1
8	0	0	0	-1

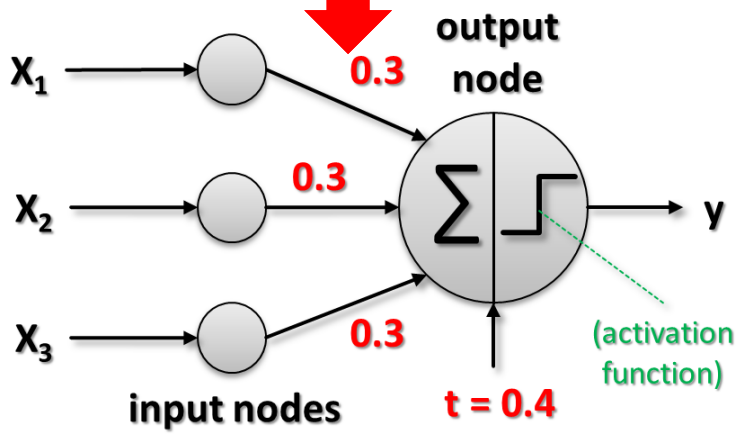
(algorithm uses training dataset)

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(training data)

$$sign\left(\left(\sum_{i=1}^{d} w_i x_i\right) - threshold\right)$$

(training phase; Find w_i and threshold that fit the data)

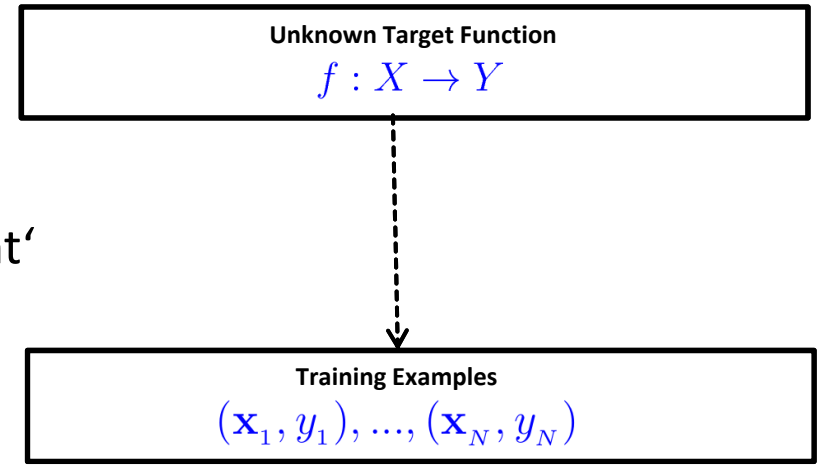


(trained perceptron model and our selected final hypothesis)

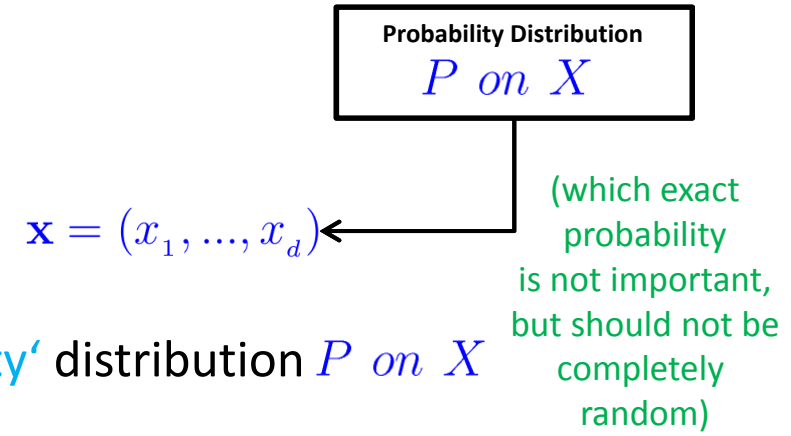
Exercises

Feasibility of Learning – Probability Distribution

- Predict output from future input
  (fitting existing data is not enough)
  - In-sample ‘1000 points‘ fit well
  - Possible: Out-of-sample >= ‘1001 point‘ doesn‘t fit very well
  - Learning ‘any target function‘ is not feasible (can be anything)



- Assumptions about ‘future input‘
  - Statement is possible to define about the data outside the in-sample data $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
  - All samples (also future ones) are derived from same ‘unknown probability‘ distribution $P \ on \ X$

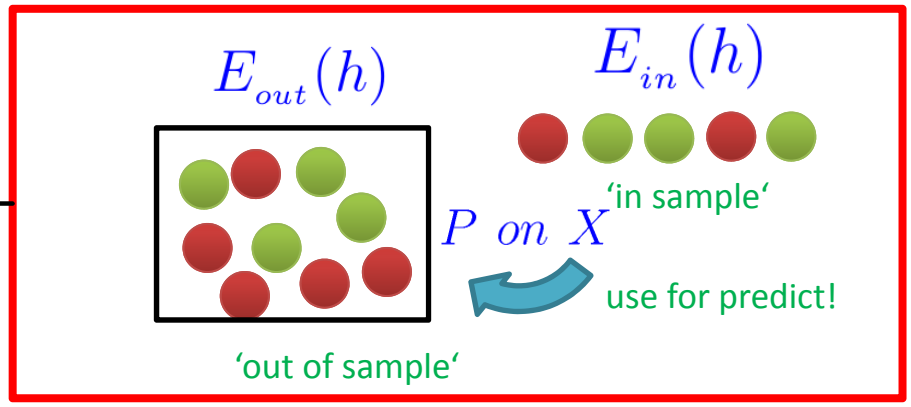


- **Statistical Learning Theory assumes an unknown probability distribution over the input space X**

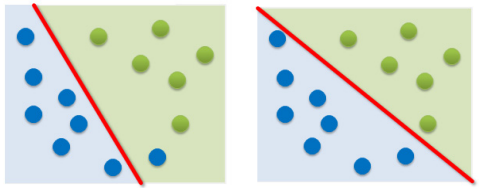
Feasibility of Learning – In Sample vs. Out of Sample

- Given ‘unknown‘ probability $P\ on\ X$
  - Given large sample N for $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$
  - There is a probability of ‘picking one point or another‘
  - ‘Error on in sample‘ is known quantity (using labelled data): $E_{in}(h)$
  - ‘Error on out of sample‘ is unknown quantity: $E_{out}(h)$
  - In-sample frequency is likely close to out-of-sample frequency E_{in} tracks E_{out}

depend on which hypothesis h out of M different ones



$\mathcal{H} = \{h_1, ..., h_m\};$



$E_{in}(h) \approx E_{out}(h)$

use $E_{in}(h)$ as a proxy thus the other way around in learning

$E_{out}(h) \approx E_{in}(h)$

- **Statistical Learning Theory part that enables that learning is feasible in a probabilistic sense (P on X)**

Feasibility of Learning – Union Bound & Factor M

- **The union bound means that (for any countable set of m 'events') the probability that at least one of the events happens is not greater that the sum of the probabilities of the m individual 'events'**

- Assuming no overlaps in hypothesis set
  - Apply mathematical rule 'union bound'
  - (Note the usage of g instead of h, we need to visit all)

Final Hypothesis

$$g \approx f$$

Think if E_{in} deviates from E_{out} with more than tolerance Є it is a 'bad event' in order to apply union bound

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <= \Pr\ [\ |\ E_{in}(h_1) - E_{out}(h_1)\ |\ > \epsilon$$

$$\text{or}\ \ |\ E_{in}(h_2) - E_{out}(h_2)\ |\ > \epsilon\ \ \ldots$$

$$\text{or}\ \ |\ E_{in}(h_M) - E_{out}(h_M)\ |\ > \epsilon\ \ ]$$

'visiting M different hypothesis'

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <= \sum_{m=1}^{M} \Pr\ [\ |\ E_{in}(h_m) - E_{out}(h_m)\ |\ > \epsilon\ ]$$

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <= \sum_{m=1}^{M} 2e^{-2\epsilon^2 N}$$

fixed quantity for each hypothesis obtained from Hoeffdings Inequality

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <= 2Me^{-2\epsilon^2 N}$$

problematic: if M is too big we loose the link between the in-sample and out-of-sample

Feasibility of Learning – Modified Hoeffding's Inequality

- Errors in-sample $E_{in}(g)$ track errors out-of-sample $E_{out}(g)$
  - Statement is made being 'Probably Approximately Correct (PAC)'
  - Given M as number of hypothesis of hypothesis set $\mathcal{H}$
  - 'Tolerance parameter' in learning ϵ
  - Mathematically established via 'modified Hoeffdings Inequality':
    (original Hoeffdings Inequality doesn't apply to multiple hypothesis)

[3] Valiant, 'A Theory of the Learnable', 1984

'Approximately' 'Probably'

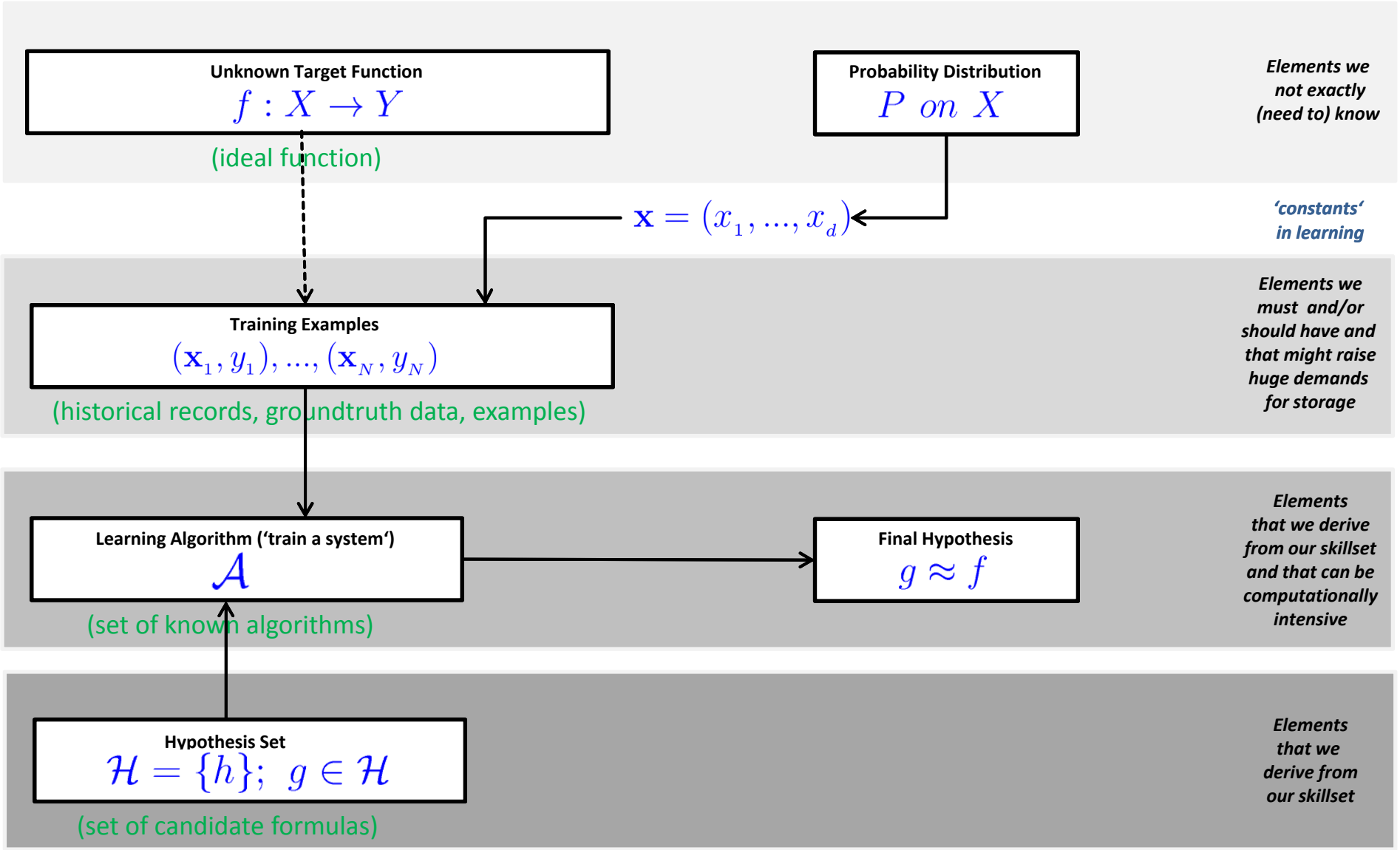
$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

'Probability that E_{in} deviates from E_{out} by more than the tolerance Є is a small quantity depending on M and N'

- Theoretical 'Big Data' Impact → more N → better learning
  - The more samples N the more reliable will track $E_{in}(g)\ E_{out}(g)$ well
  - (But: the 'quality of samples' also matter, not only the number of samples)

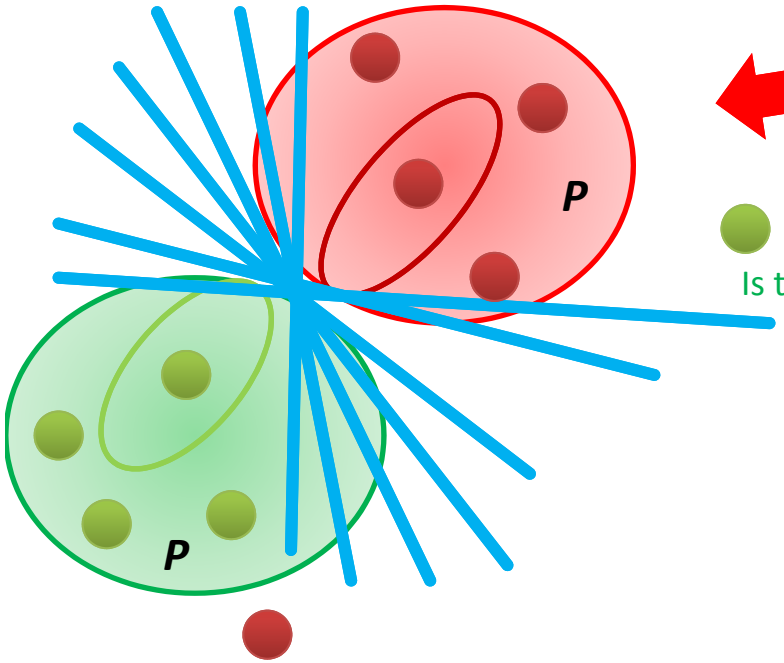
- **Statistical Learning Theory part describing the Probably Approximately Correct (PAC) learning**

Mathematical Building Blocks (4)



Mathematical Building Blocks (4) – Our Linear Example

(infinite M decision boundaries depending on f)



Probability Distribution

$P \ on \ X$

$\mathbf{x} = (x_1, ..., x_d)$

Is this point very likely from the same distribution or just noise?

We assumme future points are taken from the same probability distribution as those that we have in our training examples

Training Examples

$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)$

Is this point very likely from the same distribution or just noise?

(we help here with the assumption for the samples)

(we do not solve the M problem here)

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

(counter example would be for instance a random number generator, impossible to learn this!)

Exercises

Statistical Learning Theory – Error Measure & Noisy Targets

- Question: How can we learn a function from (noisy) data?
- ‘Error measures‘ to quantify our progress, the goal is: $h \approx f$
  - Often user-defined, if not often ‘squared error‘:

$$e(h(\mathbf{x}), f(\mathbf{x})) = (h(\mathbf{x}) - f(\mathbf{x}))^2$$

Error Measure α

  - E.g. ‘point-wise error measure‘ (e.g. think movie rated now and in 10 years from now)
- ‘(Noisy) Target function‘ is not a (deterministic) function
  - Getting with ‘same x in‘ the ‘same y out‘ is not always given in practice
  - Problem: ‘Noise‘ in the data that hinders us from learning
  - Idea: Use a ‘target distribution‘ instead of ‘target function‘
  - E.g. credit approval (yes/no)

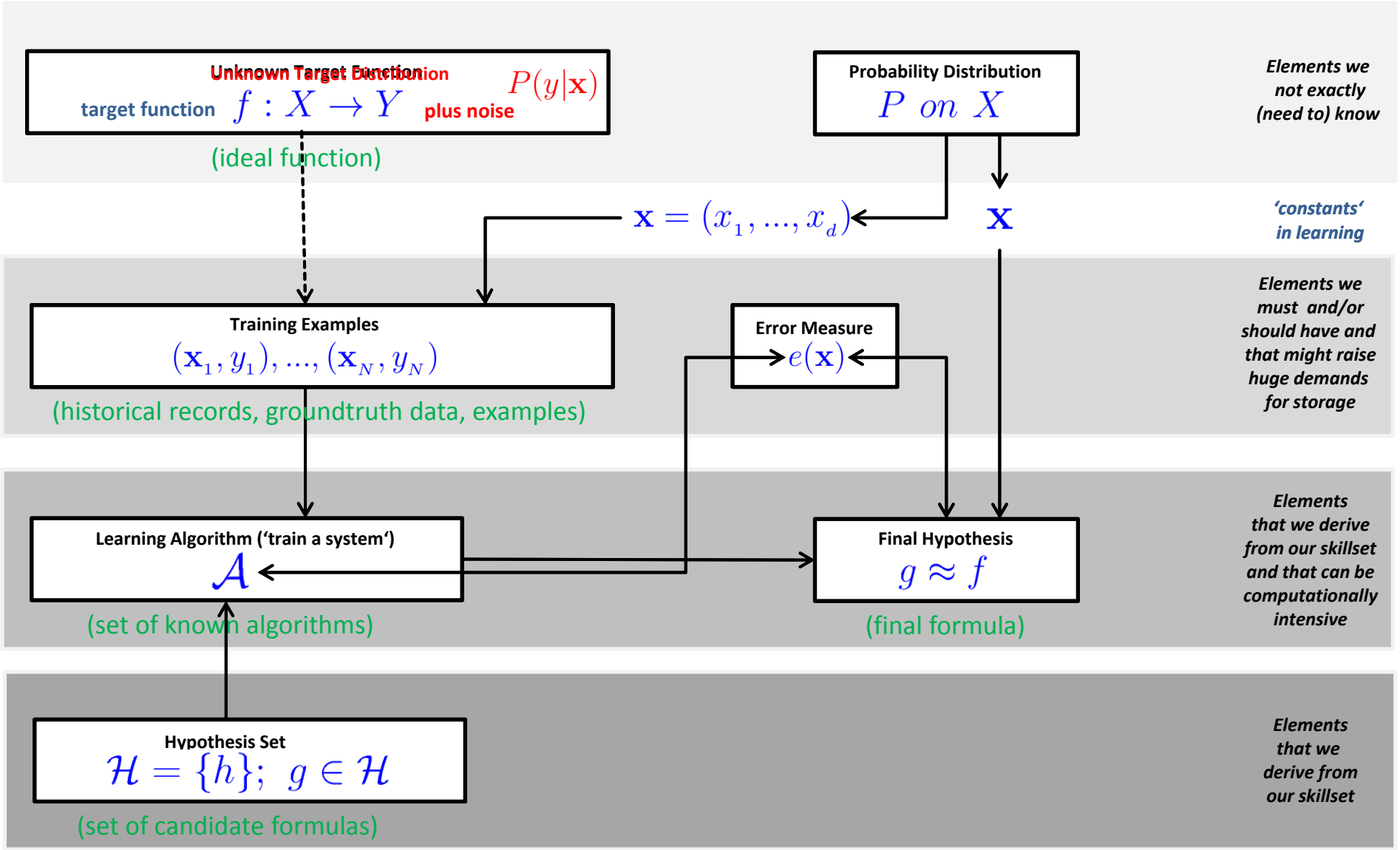
Unknown Target Distribution $P(y|\mathbf{x})$

target function $f : X \rightarrow Y$ plus noise

(ideal function)

- **Statistical Learning Theory refines the learning problem of learning an unknown target distribution**

Mathematical Building Blocks (5)



Mathematical Building Blocks (5) – Our Linear Example

- Iterative Method using (labelled) training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$

(one point at a time is picked)

1. Pick one misclassified training point where:

$$sign(\mathbf{w}^T \mathbf{x}_n) \neq y_n$$

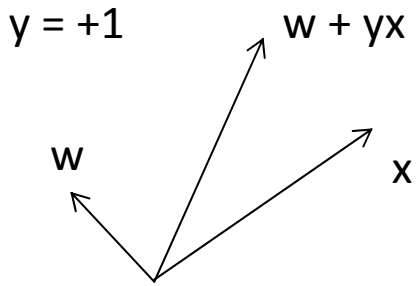
Error Measure α

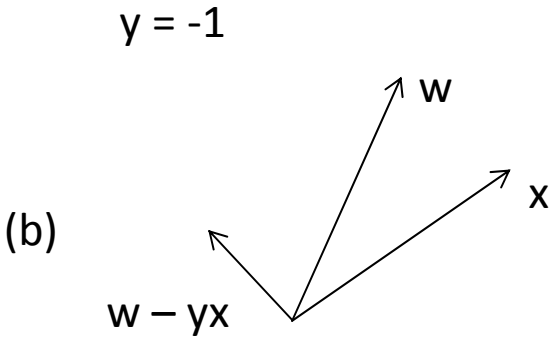
2. Update the weight vector:

$$\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n$$

(y_n is either +1 or -1)

(a) adding a vector or
(b) subtracting a vector





- Terminates when there are no misclassified points

Error Measure α

(converges only with linearly seperable data)

Training and Testing – Influence on Learning

- Mathematical notations
  - Testing follows: (hypothesis clear)
    $$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\epsilon\ ]\ <=\ 2\quad e^{-2\epsilon^2 N}$$
  - Training follows: (hypothesis search)
    $$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$
    (e.g. student exam training on examples to get E_{in} ‚down', then test via exam)
- Practice on 'training examples'
  - Create two disjoint datasets
  - One used for training only (aka training set)
  - Another used for testing only (aka test set)

Training Examples

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$

(historical records, groundtruth data, examples)

- Training & Testing are different phases in the learning process
  - Concrete number of samples in each set often influences learning

Exercises

Theory of Generalization – Initial Generalization & Limits

- Learning is feasible in a probabilistic sense
  - Reported final hypothesis – using a 'generalization window' on $E_{out}(g)$
  - Expecting 'out of sample performance' tracks 'in sample performance'
  - Approach: $E_{in}(g)$ acts as a 'proxy' for $E_{out}(g)$

$$E_{out}(g) \approx E_{in}(g)$$

This is not full learning – rather 'good generalization' since the quantity $E_{out}(g)$ is an unknown quantity

- Reasoning
  - Above condition is not the final hypothesis condition:
  - More similiar like $E_{out}(g)$ approximates 0
    (out of sample error is close to 0 if approximating f)
  - $E_{out}(g)$ measures how far away the value is from the 'target function'
  - Problematic because $E_{out}(g)$ is an unknown quantity (cannot be used…)
  - The learning process thus requires 'two general core building blocks'

Final Hypothesis

$$g \approx f$$

Theory of Generalization – Learning Process Reviewed

- ‘Learning Well‘
  - Two core building blocks that achieve $E_{out}(g)$ approximates 0
- First core building block
  - Theoretical result using Hoeffdings Inequality $E_{out}(g) \approx E_{in}(g)$
  - Using $E_{out}(g)$ directly is not possible – it is an unknown quantity
- Second core building block (try to get the ‘in-sample‘ error lower)
  - Practical result using tools & techniques to get $E_{in}(g) \approx 0$
  - e.g. linear models with the Perceptron Learning Algorithm (PLA)
  - Using $E_{in}(g)$ is possible – it is a known quantity – ‘so lets get it small‘
  - Lessons learned from practice: in many situations ‘close to 0‘ impossible
  - E.g. remote sensing images use case of land cover classification

- **Full learning means that we can make sure that $E_{out}(g)$ is close enough to $E_{in}(g)$ [from theory]**
- **Full learning means that we can make sure that $E_{in}(g)$ is small enough [from practical techniques]**

Complexity of the Hypothesis Set – Infinite Spaces Problem

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

theory helps to find a way to deal with infinite M hypothesis spaces

- Tradeoff & Review
  - Tradeoff between Є, M, and the 'complexity of the hypothesis space H'
  - Contribution of detailed learning theory is to 'understand factor M'
- M Elements of the hypothesis set $\mathcal{H}$ M elements in H here
  - Ok if N gets big, but problematic if M gets big → bound gets meaningless
  - E.g. classification models like perceptron, support vector machines, etc.
  - Challenge: those classification models have continous parameters
  - Consequence: those classification models have infinite hypothesis spaces
  - Aproach: despite their size, the models still have limited expressive power

- **Many elements of the hypothesis set H have continous parameter with infinite M hypothesis spaces**

Factor M from the Union Bound & Hypothesis Overlaps

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <= \Pr\ [\ |\ E_{in}(h_1) - E_{out}(h_1)\ |\ > \epsilon$$
$$\text{or}\ \ |\ E_{in}(h_2) - E_{out}(h_2)\ |\ > \epsilon\ \ \ldots$$
$$\text{or}\ \ |\ E_{in}(h_M) - E_{out}(h_M)\ |\ > \epsilon\ \ ]$$

assumes no overlaps, all probabilities happen disjointly

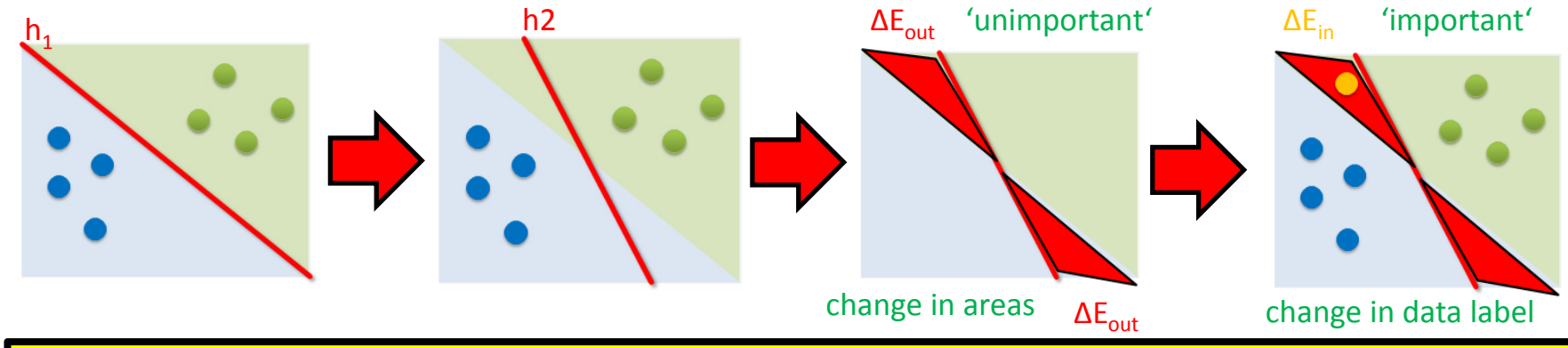
$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ > \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

takes no overlaps of M hypothesis into account

- Union bound is a 'poor bound', ignores correlation between h
  - Overlaps are common: the interest is shifted to data points changing label

$$|\ E_{in}(h_1) - E_{out}(h_1)\ | \approx\ \ |\ E_{in}(h_2) - E_{out}(h_2)\ |$$

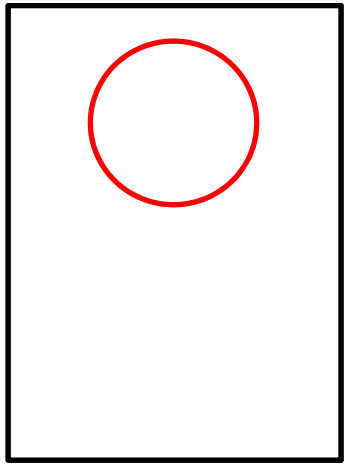
(at least very often, indicator to reduce M)



- **Statistical Learning Theory provides a quantity able to characterize the overlaps for a better bound**

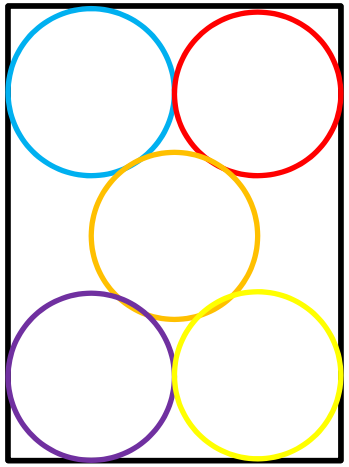
Replacing M & Large Overlaps

(Hoeffding Inequality)



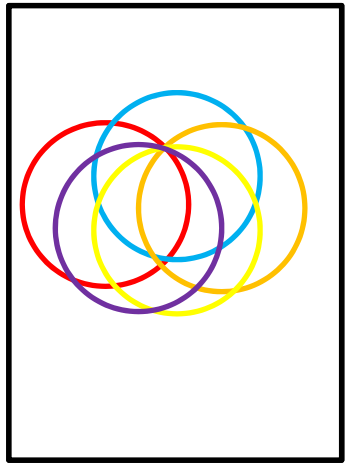
(valid for 1 hypothesis)

(Union Bound)



(valid for M hypothesis, worst case)

(towards Vapnik Chervonenkis Bound)



(valid for m (N) as growth function)

- Characterizing the overlaps is the idea of a 'growth function'
  - Number of dichotomies: $\mathbf{m}_{\mathcal{H}}(N) = \max_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N} |\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$
    Number of hypothesis but
    on finite number N of points
  - Much redundancy: Many hypothesis will reports the same dichotomies

- **The mathematical proofs that $m_H(N)$ can replace M is a key part of the theory of generalization**

Complexity of the Hypothesis Set – VC Inequality

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\ \epsilon\ ]\ <=\ 2Me^{-2\epsilon^2 N}$$

$$\mathbf{m}_{\mathcal{H}}(N) = \max_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N} |\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)|$$

- Vapnik-Chervonenkis (VC) Inequality
  - Result of mathematical proof when replacing M with growth function m
  - 2N of growth function to have another sample (2 x $E_{in}(h)$, no $E_{out}(h)$)

$$\Pr\ [\ |\ E_{in}(g) - E_{out}(g)\ |\ >\ \epsilon\ ]\ <=\ 4m_{\mathcal{H}}(2N)e^{-1/8\epsilon^2 N}$$

(characterization of generalization)

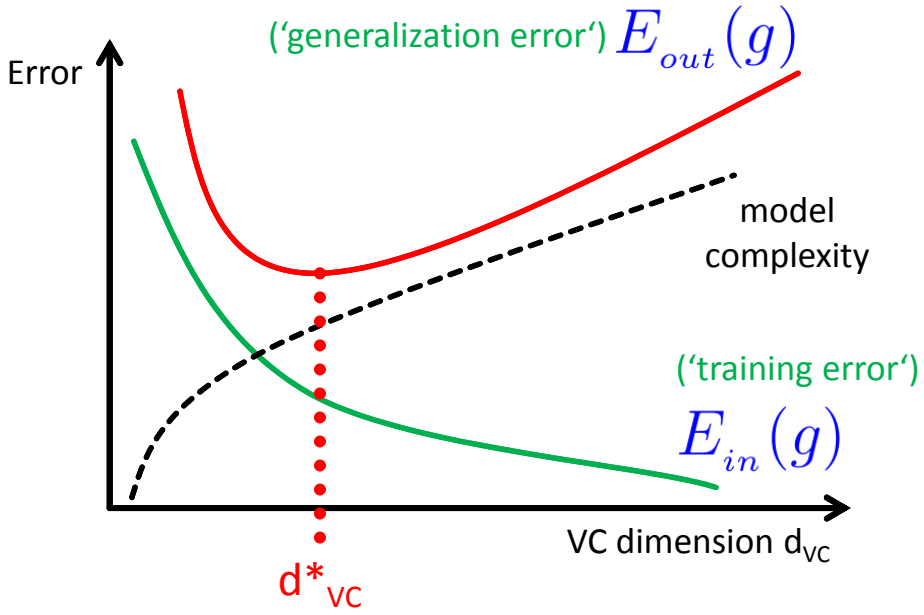
  - In Short – finally : We are able to learn and can generalize ‘ouf-of-sample‘

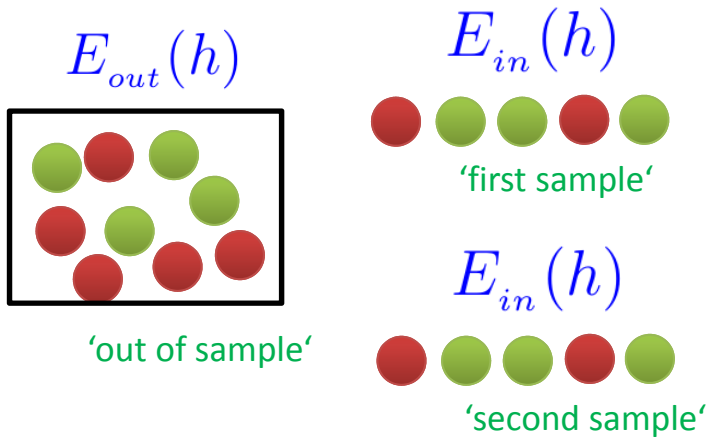
- **The Vapnik-Chervonenkis Inequality is the most important result in machine learning theory**
- **The mathematial proof brings us that M can be replaced by growth function (no infinity anymore)**

Complexity of the Hypothesis Set – VC Dimension

- Vapnik-Chervonenkis (VC) Dimension over instance space X
  - VC dimension gets a 'generalization bound' on all possible target functions

Issue: unknown to 'compute' – VC solved this using the growth function on different samples





idea: 'first sample' frequency close to 'second sample' frequency

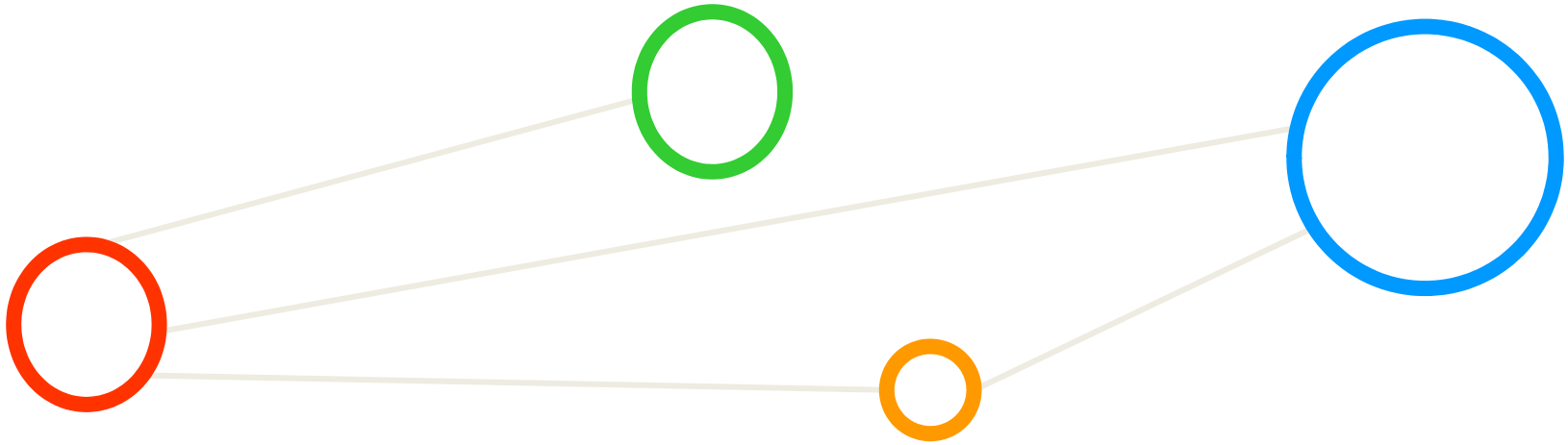
- **Complexity of Hypothesis set H can be measured by the Vapnik-Chervonenkis (VC) Dimension d_{VC}**
- **Ignoring the model complexity d_{VC} leads to situations where $E_{in}(g)$ gets down and $E_{out}(g)$ gets up**

Prevent Overfitting for better ‘ouf-of-sample‘ generalization

[4] Stop Overfitting, YouTube

Exercises

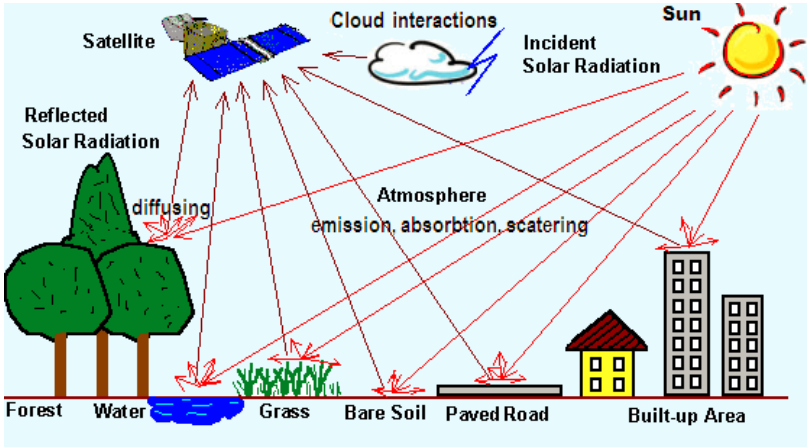
Remote Sensing Applications

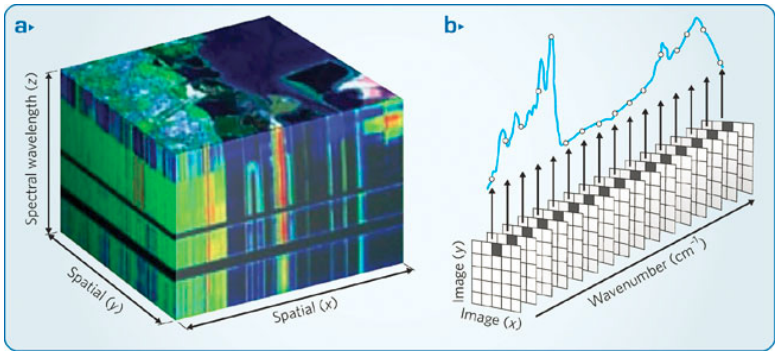


Introduction to Application Field

- **Remote sensing is the acquisition of information about an object or phenomenon without making physical contact with an object**

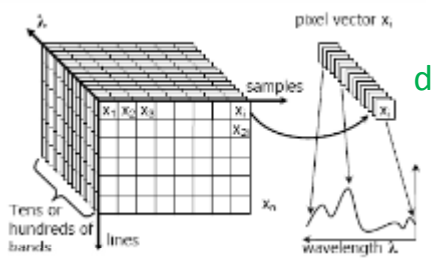
[5] Wikipedia on 'Remote Sensing'





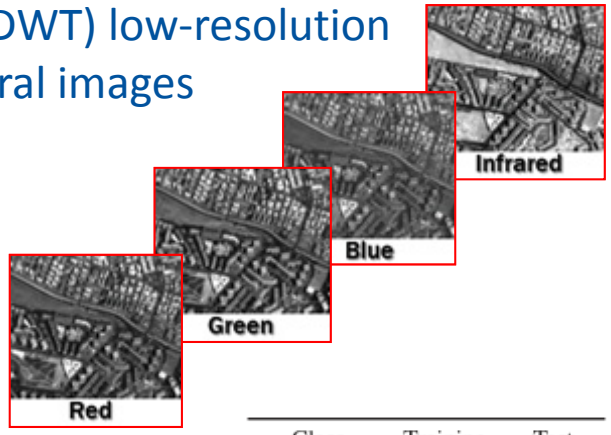
- The overall system is complex:
  - Scattering or emission of energy from the earth's surface
  - Transmission through the atmosphere to instruments mounted on the remote sensing platform
  - Sending data back to the earth's surface
  - Processing into images products ready for application by the user

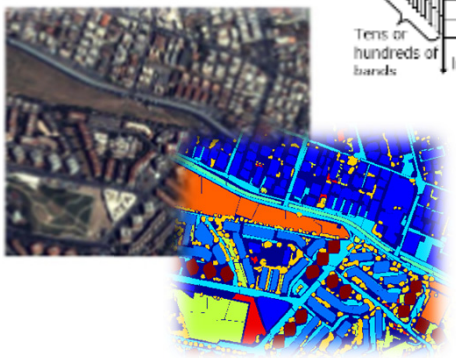
Supervised Learning Application – Labelled Data



(high dimension)

Pansharpened (UDWT) low-resolution (2.4m) multispectral images





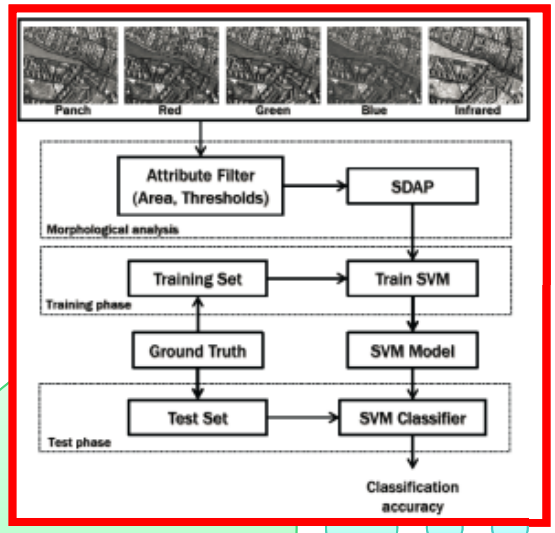
(Quickbird)

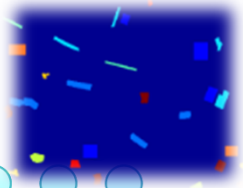
Sattelite Data

Groundtruth

Model & Algorithm

Classification Study of Land Cover Types





Class	Training	Test
Buildings	18126	163129
Blocks	10982	98834
Roads	16353	147176
Light Train	1606	14454
Vegetation	6962	62655
Trees	9088	81792
Bare Soil	8127	73144
Soil	1506	13551
Tower	4792	43124
Total	77542	697859

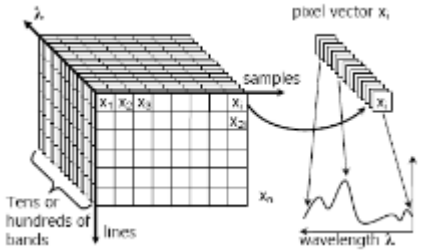
[6] G. Cavallaro & M. Riedel et al., 2014

Remote Sensing Application – The Dataset

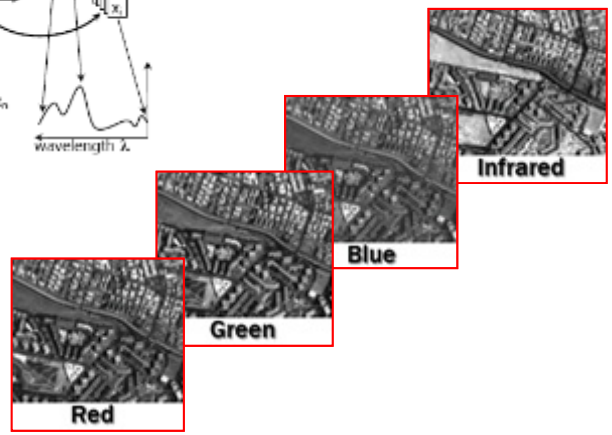
- Example dataset: Geographical location: Image of Rome, Italy
  - Remote sensor data obtained by Quickbird satellite

- High-resolution (0.6m) panchromatic image

Pansharpened (UDWT) low-resolution (2.4m) multispectral images



(high dimension)



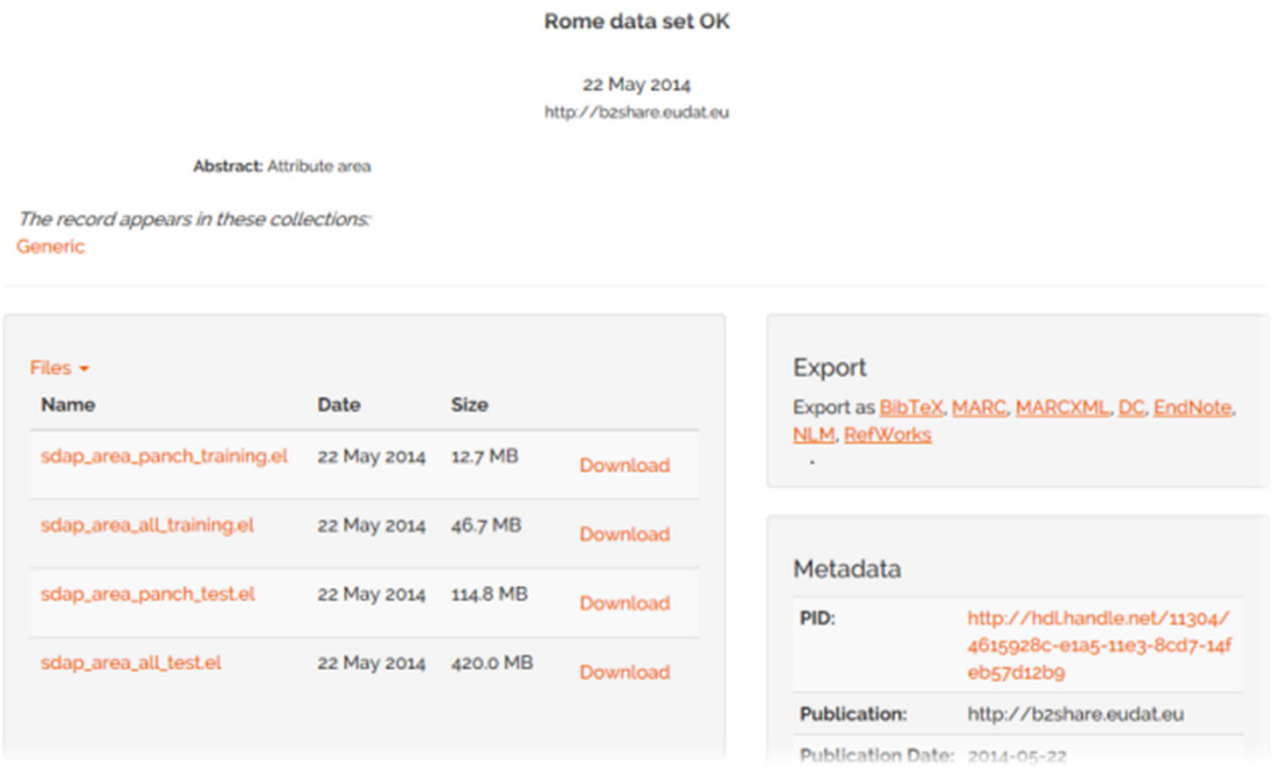
(Reasoning for picking SVM: Good classification accuracies on high dimensional datasets, even with a small ‚rare' number of training samples)

[7] Rome Image dataset

Inspect and Understanding the Data – Rome, Italy

- Data is publicly available in EUDAT B2SHARE tool

[7] Rome Image dataset



(persistent handle link for publication into papers)

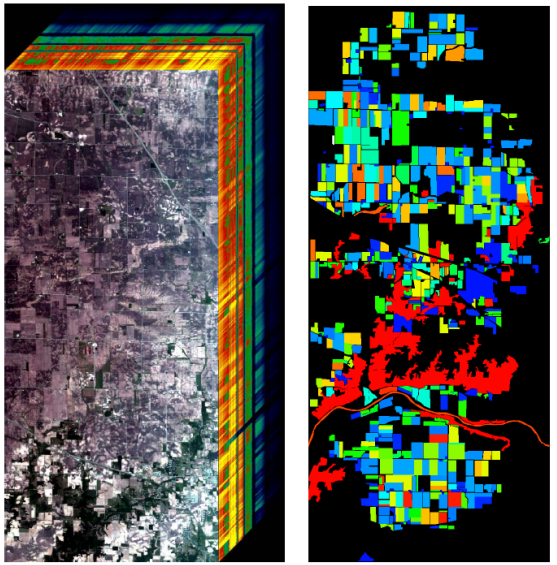
Exercises

Advanced Supervised Learning Application – Indian Pines

- Challenging (non-linearly separable) dataset
  - 52 classes; less groundtruth samples; mixed pixels; high dimensions

Class		Number of samples		Class		Number of samples	
Number	Name	Training	Test	Number	Name	Training	Test
1	Buildings	1720	15 475	27	Pasture	1039	9347
2	Corn	1778	16 005	28	pond	10	92
3	Corn?	16	142	29	Soybeans	939	8452
4	Corn-EW	51	463	30	Soybeans?	89	805
5	Corn-NS	236	2120	31	Soybeans-NS	111	999
6	Corn-CleanTill	1240	11 164	32	Soybeans-CleanTill	507	4567
7	Corn-CleanTill-EW	2649	23 837	33	Soybeans-CleanTill?	273	2453
8	Corn-CleanTill-NS	3968	35 710	34	Soybeans-CleanTill-EW	1180	10 622
9	Corn-CleanTill-NS-Irrigated	80	720	35	Soybeans-CleanTill-NS	1039	9348
10	Corn-CleanTilled-NS?	173	1555	36	Soybeans-CleanTill-Drilled	224	2018
11	Corn-MinTill	105	944	37	Soybeans-CleanTill-Weedy	54	489
12	Corn-MinTill-EW	563	5066	38	Soybeans-Drilled	1512	13 606
13	Corn-MinTill-NS	886	7976	39	Soybeans-MinTill	267	2400
14	Corn-NoTill	438	3943	40	Soybeans-MinTill-EW	183	1649
15	Corn-NoTill-EW	121	1085	41	Soybeans-MinTill-Drilled	810	7288
16	Corn-NoTill-NS	569	5116	42	Soybeans-MinTill-NS	495	4458
17	Fescue	11	103	43	Soybeans-NoTill	216	1941
18	Grass	115	1032	44	Soybeans-NoTill-EW	253	2280
19	Grass/Trees	233	2098	45	Soybeans-NoTill-NS	93	836
20	Hay	113	1015	46	Soybeans-NoTill-Drilled	873	7858
21	Hay?	219	1966	47	Swampy Area	58	525
22	Hay-Alfalfa	226	2032	48	River	311	2799
23	Lake	22	202	49	Trees?	58	522
24	NotCropped	194	1746	50	Wheat	498	4481
25	Oats	174	1568	51	Woods	6356	57 206
26	Oats?	34	301	52	Woods?	14	130

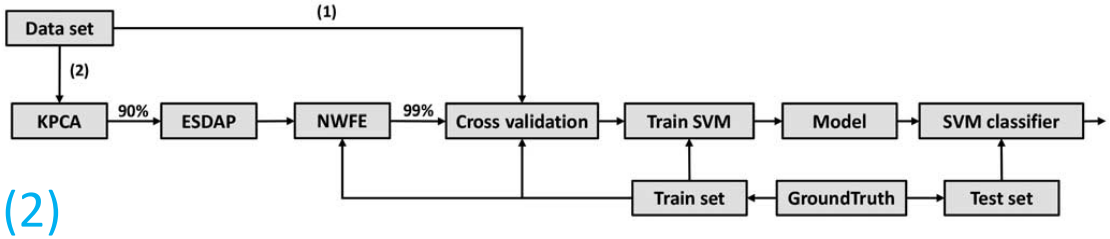
remote sensing cube & ground reference

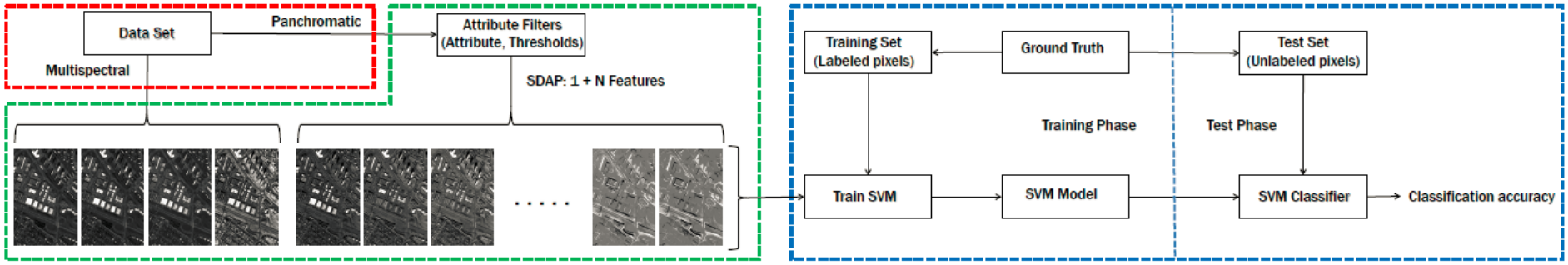


[8] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

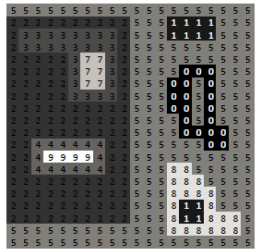
Importance of Features Engineering & Process Overview

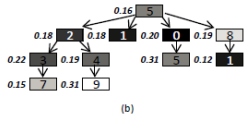
- Application
  - Using dataset raw (1)
  - Using dataset processed (2)





- Feature Enhancement & Selection
  - Kernel Principle Component Analysis (KPCA)
  - Extended Self-Dual Attribute Profile (ESDAP)
  - Nonparametric weighted feature extraction (NWFE)





[8] G. Cavallaro, M. Riedel, J.A. Benediktsson et al.,
Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015

Publicly Available Datasets – Location

- *Indian Pines Dataset Raw and Processed*

Abstract: 1) Indian raw: 1417x614x200 (training 10% and test)
2) Indian processed:1417x614x30 (training 10% and test)

[9] Indian Pine Image dataset

B2SHARE
Store and Share Research Data

Files

Name	Date	Size	
indian_processed_training.el	05 Feb 2015	11.7 MB	Download
indian_raw_test.el	05 Feb 2015	747.1 MB	Download
indian_raw_training.el	05 Feb 2015	83.0 MB	Download
indian_processed_test.el	05 Feb 2015	105.6 MB	Download

```
[morris@login-0-1 172-IndianPines]$ ls -al
insgesamt 925280
drwxrwxr-x  2 morris morris      4096 19. Nov 14:04 .
drwxrwxr-x 11 morris morris      4096 19. Nov 13:51 ..
-rw-rw-r--  1 morris morris 105594346  5. Feb 2015  indian_processed_test.el
-rw-rw-r--  1 morris morris  11732509  5. Feb 2015  indian_processed_training.el
-rw-rw-r--  1 morris morris 747125597  5. Feb 2015  indian_raw_test.el
-rw-rw-r--  1 morris morris  83014311  5. Feb 2015  indian_raw_training.el
```

Available Datasets – Training Data Example

- *Indian Pines Dataset Processed – Training*
  - Indian_processed_training.el
  - LibSVM data format: class feature1:value1 feature2:value2

[9] Indian Pine Image dataset

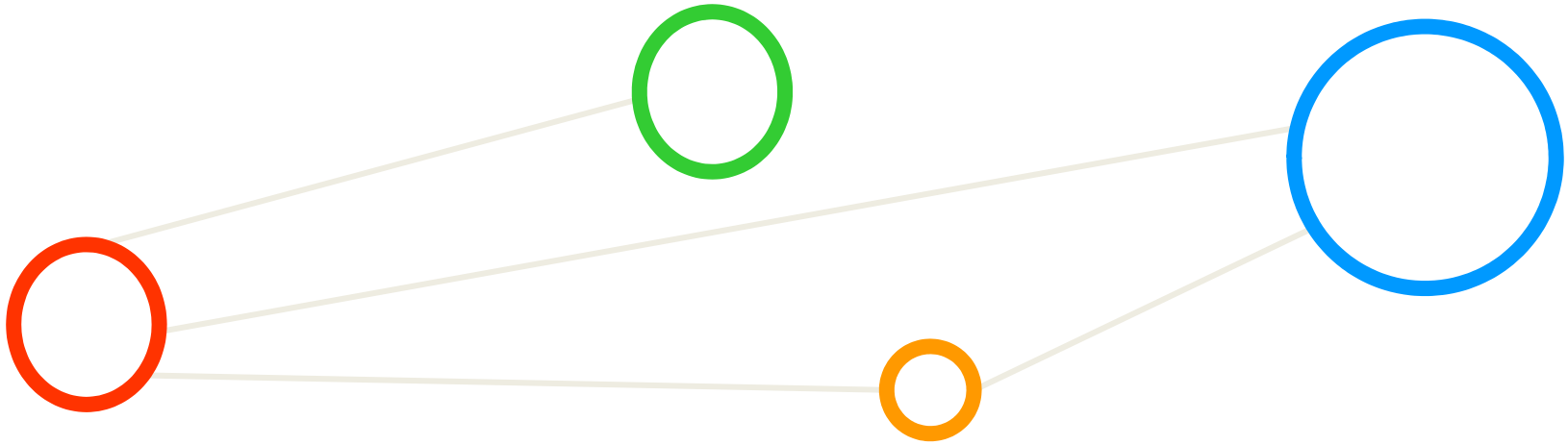
```
[morris@login-0-1 pisvmexamples]$ head /home/morris/BIGDATA/172-IndianPines/indian_processed_training.el
48 1:0.69021 2:0.599122 3:0.864571 4:0.265246 5:0.566572 6:0.561181 7:0.665698 8:0.430511 9:0.717402 10:0.514414 11:0.335391 12:0.486979 13:
0.58382 14:0.503849 15:0.420948 16:0.624377 17:0.43594 18:0.324333 19:0.614007 20:0.431993 21:0.600198 22:0.577696 23:0.528974 24:0.458141 2
5:0.543731 26:0.382421 27:0.33752 28:0.543062 29:0.5737 30:0.467784
48 1:0.675361 2:0.585579 3:0.85587 4:0.249317 5:0.579717 6:0.5633 7:0.634459 8:0.434777 9:0.719665 10:0.510279 11:0.330765 12:0.478097 13:0.
559455 14:0.505097 15:0.396293 16:0.627475 17:0.425701 18:0.32044 19:0.637625 20:0.437285 21:0.620149 22:0.535474 23:0.525151 24:0.511439 25
:0.504738 26:0.385925 27:0.303322 28:0.572963 29:0.570833 30:0.491508
51 1:0.554975 2:0.499039 3:0.578284 4:0.648481 5:0.867561 6:0.17041 7:0.841141 8:0.54466 9:0.746417 10:0.459481 11:0.356237 12:0.420116 13:0
.248721 14:0.541081 15:0.444207 16:0.286447 17:0.364712 18:0.522284 19:0.647555 20:0.538008 21:0.51632 22:0.437087 23:0.402312 24:0.574872 2
5:0.633603 26:0.426417 27:0.397177 28:0.400566 29:0.470729 30:0.51207
51 1:0.561296 2:0.495371 3:0.330829 4:0.513766 5:0.556074 6:0.387238 7:0.932418 8:0.643317 9:0.735262 10:0.216349 11:0.284832 12:0.499805 13
:0.272644 14:0.289776 15:0.699631 16:0.142032 17:0.161307 18:0.657442 19:0.72633 20:0.446231 21:0.542728 22:0.526951 23:0.433363 24:0.647875
 25:0.679122 26:0.760654 27:0.302971 28:0.515992 29:0.602717 30:0.34428
46 1:0.870891 2:0.768472 3:0.783699 4:0.28838 5:0.632466 6:0.418282 7:0.508729 8:0.437482 9:0.610721 10:0.480019 11:0.291032 12:0.427472 13:
0.428984 14:0.447632 15:0.452797 16:0.527423 17:0.583162 18:0.42837 19:0.557041 20:0.290409 21:0.547233 22:0.642817 23:0.535524 24:0.4478 25
:0.601117 26:0.405341 27:0.315288 28:0.488306 29:0.659466 30:0.469213
46 1:0.87119 2:0.779909 3:0.769342 4:0.292747 5:0.643116 6:0.416394 7:0.530679 8:0.4283 9:0.61224 10:0.474311 11:0.325682 12:0.468692 13:0.4
2888 14:0.461495 15:0.471023 16:0.524454 17:0.527418 18:0.398571 19:0.555335 20:0.35517 21:0.557823 22:0.588179 23:0.564881 24:0.444141 25:0
.576903 26:0.495859 27:0.344496 28:0.541081 29:0.539253 30:0.494238
15 1:0.869845 2:0.781471 3:0.412471 4:0.318165 5:0.414446 6:0.320591 7:0.605434 8:0.726664 9:0.590541 10:0.457315 11:0.355967 12:0.429357 13
:0.509808 14:0.421272 15:0.5406 16:0.532659 17:0.568507 18:0.404705 19:0.523209 20:0.52817 21:0.575417 22:0.570611 23:0.578825 24:0.452228 2
5:0.484941 26:0.542836 27:0.335545 28:0.491274 29:0.656853 30:0.546783
15 1:0.892436 2:0.790123 3:0.485313 4:0.318438 5:0.478488 6:0.329743 7:0.602466 8:0.581215 9:0.597853 10:0.442717 11:0.409894 12:0.479275 13
:0.51758 14:0.445422 15:0.483192 16:0.549861 17:0.49607 18:0.469129 19:0.475024 20:0.638295 21:0.578851 22:0.586814 23:0.579959 24:0.463105
25:0.550809 26:0.490199 27:0.376347 28:0.468157 29:0.55987 30:0.42689
15 1:0.809988 2:0.508887 3:0.782026 4:0.284838 5:0.682891 6:0.361665 7:0.243857 8:0.330213 9:0.456187 10:0.385119 11:0.230994 12:0.403049 13
:0.333648 14:0.473923 15:0.570014 16:0.45283 17:0.314596 18:0.574783 19:0.708582 20:0.292825 21:0.531681 22:0.616414 23:0.506682 24:0.493297
 25:0.396451 26:0.599563 27:0.329084 28:0.541152 29:0.56897 30:0.466168
8 1:0.843723 2:0.734137 3:0.686274 4:0.335633 5:0.6219 6:0.250049 7:0.575193 8:0.50452 9:0.623444 10:0.496143 11:0.263468 12:0.482361 13:0.5
46293 14:0.402207 15:0.730217 16:0.378631 17:0.433787 18:0.654169 19:0.513311 20:0.202298 21:0.530649 22:0.696446 23:0.631684 24:0.447328 25
:0.425982 26:0.40432 27:0.568027 28:0.461795 29:0.411903 30:0.428683
```

Exercises

[Video] Remote Sensing

[10] YouTube Video, 'Remote Sensing'

Lecture Bibliography



Lecture Bibliography

- [1] An Introduction to Statistical Learning with Applications in R, Online: http://www-bcf.usc.edu/~gareth/ISL/index.html
- [2] Wikipedia on 'Statistical Learning Theory', Online: http://en.wikipedia.org/wiki/Statistical_learning_theory
- [3] Leslie G. Valiant, *'A Theory of the Learnable'*, Communications of the ACM 27(11):1134–1142, 1984, Online: https://people.mpi-inf.mpg.de/~mehlhorn/SeminarEvolvability/ValiantLearnable.pdf
- [4] Udacity, 'Overfitting', Online: https://www.youtube.com/watch?v=CxAxRCv9WoA
- [5] Wikipedia on 'Remote Sensing', Online: http://en.wikipedia.org/wiki/Remote_sensing
- [6] G. Cavallaro and M. Riedel, 'Smart Data Analytics Methods for Remote Sensing Applications', 35th Canadian Symposium on Remote Sensing (IGARSS), 2014, Quebec, Canada
- [7] Rome Dataset, B2SHARE, Online: http://hdl.handle.net/11304/4615928c-e1a5-11e3-8cd7-14feb57d12b9
- [8] G. Cavallaro, M. Riedel, J.A. Benediktsson et al., 'On Understanding Big Data Impacts in Remotely Sensed Image Classification using Support Vector Machine Methods', IEEE Journal of Selected Topics in Applied Earth Observation and Remote Sensing, 2015
- [9] Indian Pine Image Dataset, B2SHARE, Online: http://hdl.handle.net/11304/7e8eec8e-ad61-11e4-ac7e-860aa0063d1f
- [10] YouTube Video, 'What is Remote Sensing', Online: https://www.youtube.com/watch?v=nU-CjAKry5c

- Acknowledgements and more Information: Yaser Abu-Mostafa, Caltech Lecture series, YouTube


services
measurement
within
fusion
new
funding
Policy-based
concepts
forms
cross-disciplinary
resources
device
Analysis
Centre
climate
Computer
expertise
year
like
Computational
Test
E.g
Computing
PB
community
disciplines
results
Approaches
construction
Fusion
Enable
Cross-Disciplinary
EU
methods
High
big
nodes
centers
Environment
Modelling
different
Science
Key
scientific
important
storage
basis
Infrastructure
files
etc
DKRZ
Use
SMAQ
international
model
engineering
References
formats
simulations
ENES
technologies
physical
brain
increasing
via
EUDAT
IPCC
Big
Project
Data
processing
Common
TB
Methods
MapReduce
create
computing
Using
European
databases
GB
Supercomputing
Work
images
codes
analysis
stored
Resources
analyze
Large
Summary
data-intensive
Service
Scientific
levels
often
use
DLCL
Level
Research
Matters
Concrete
external
performance
JSC
computational
research
e.g
science
HPC
Juelich
system
datasets
acquisition
compute
Landscape
management
System
Climate
modelling
AR5
Hardware
distributed
large-scale
cores
access
current
hundreds
Nutshell
Services
network
DLCLs
Understanding
structures
Technologies
Earth
Structure
General
Germany
Simulation
directory
project
Provide
NASA
Energy
systems
day
Processing
Health