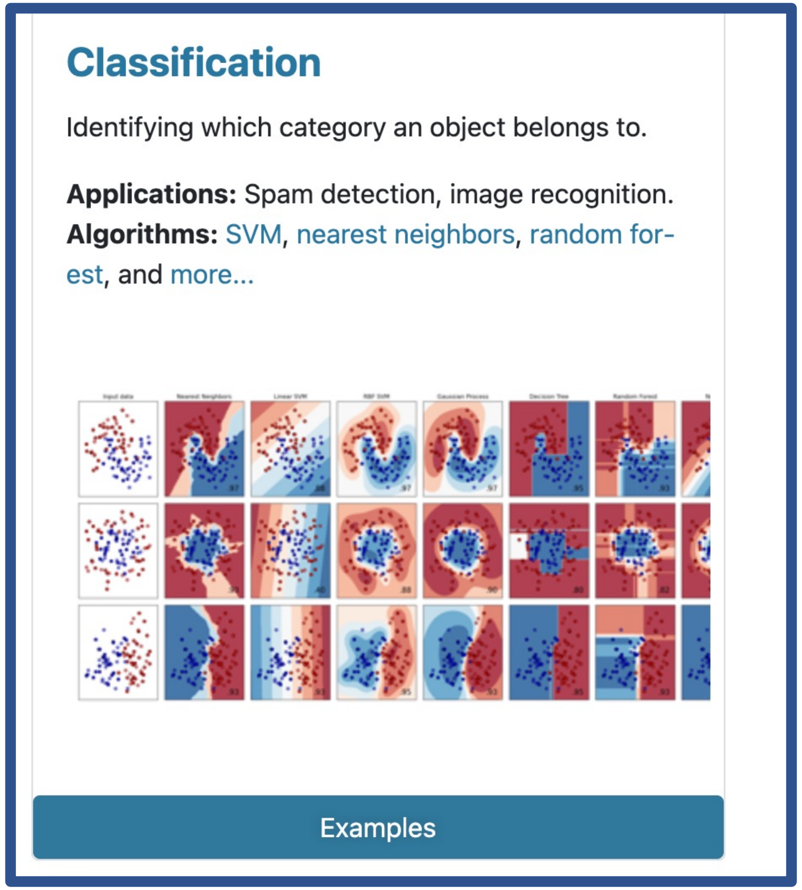
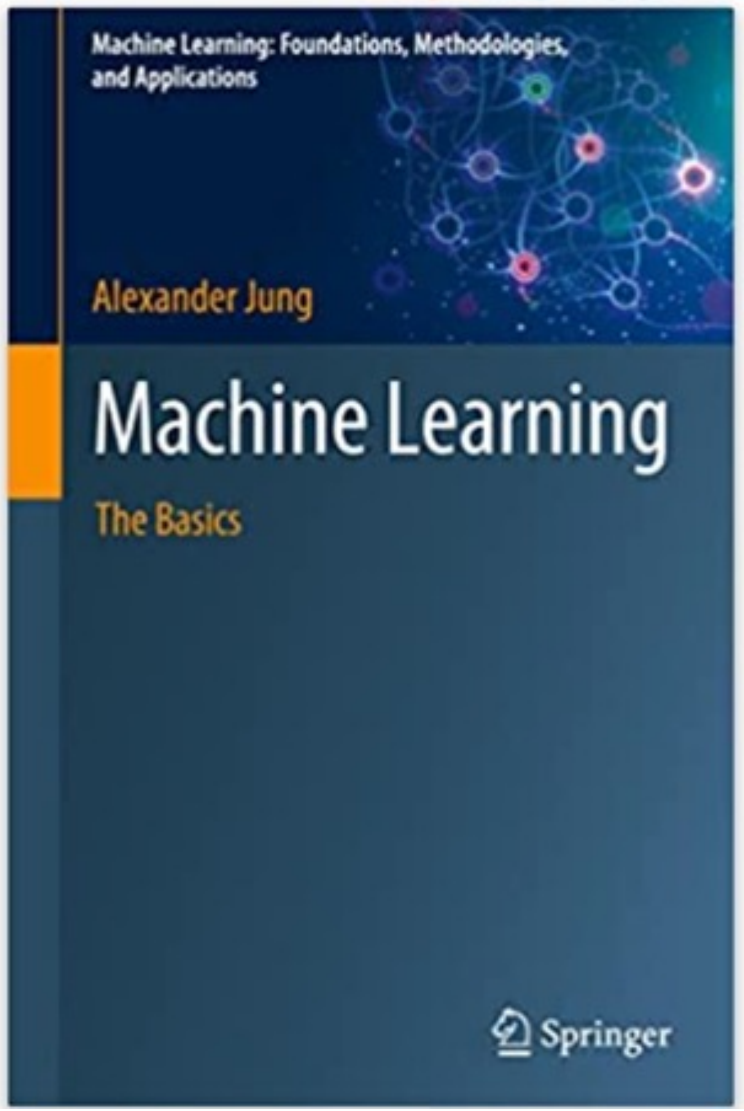
Classification

Alex(ander) Jung
Assistant Professor for Machine Learning
Department of Computer Science
Aalto University

Reading.

- Ch. 2.3, 3.6 of MLBook





https://scikit-learn.org/stable/index.html

Learning Goals:

- be able to recognize classification problems
- know binary, multi-class and multi-label problems
- know design choices of basic classif. methods
- know some stat./comp. aspects of classif. methods

What is ML About ?

fit models to data to make

predictions or forecasts !

Data. Model. Loss.

data: set of datapoints (x,y)

model: set of hypothesis maps h(.)

loss: quality measure L((x,y),h)

Machine Learning.

find hypothesis in model that incurs smallest loss when predicting label of any datapoint

Expected Loss or Risk

$$\mathbb{E}\{L((\mathbf{x},y),h)\} := \int_{\mathbf{x},y} L((\mathbf{x},y),h)dp(\mathbf{x},y). \tag{2.14}$$

note: to compute this expectation
we need to know the probability distribution
p(x,y) of datapoints (x,y)

Empirical Risk

IDEA: approximate expected loss by average loss on some datapoints (training set)

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}.$$

$$\mathbb{E}\{L((\mathbf{x}, y), h)\} \approx (1/m) \sum_{i=1}^{m} L((\mathbf{x}^{(i)}, y^{(i)}), h) \text{ for sufficiently large sample size } m. \quad (2.17)$$

with the average loss or empirical risk

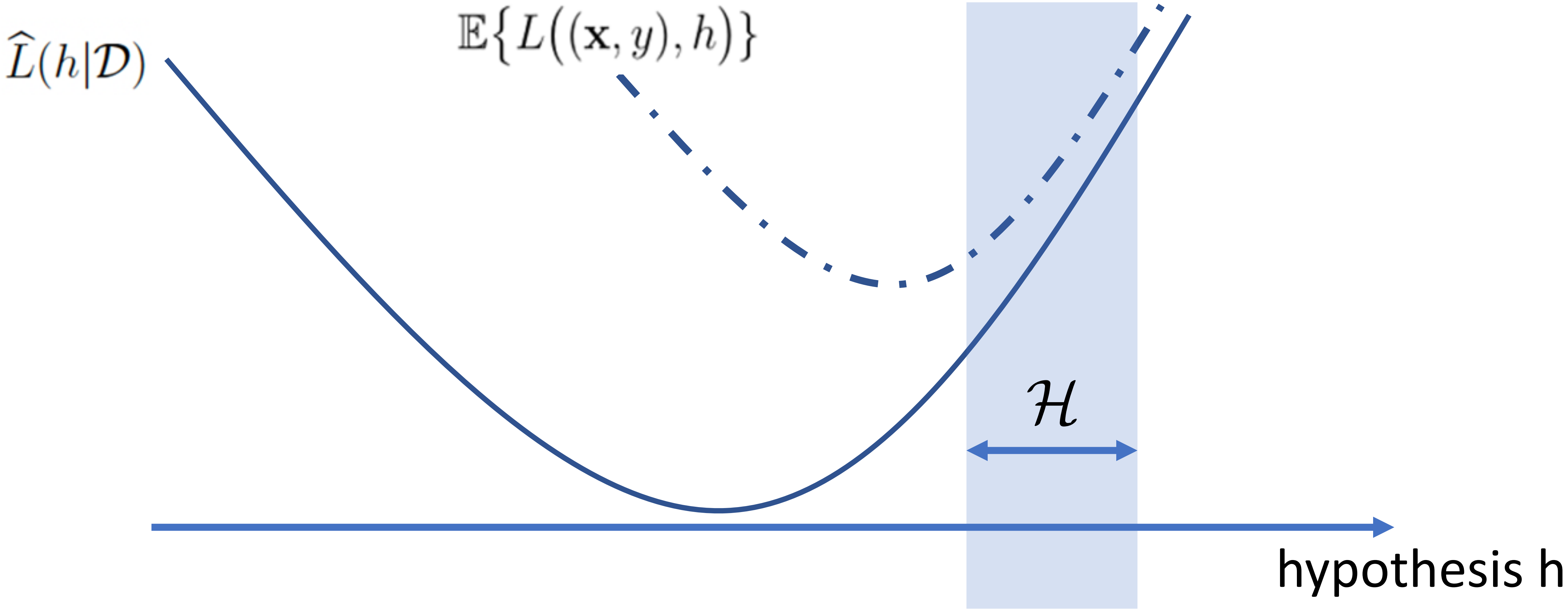
$$\widehat{L}(h|\mathcal{D}) = (1/m) \sum_{i=1}^{m} L((\mathbf{x}^{(i)}, y^{(i)}), h). \quad (2.16)$$

Empirical Risk Minimization

$$\hat{h} \in \underset{h \in \mathcal{H}}{\operatorname{argmin}} \widehat{L}(h|\mathcal{D})$$

$$\stackrel{(\boxed{2.16})}{=} \underset{h \in \mathcal{H}}{\operatorname{argmin}} (1/m) \sum_{i=1}^{m} L\big((\mathbf{x}^{(i)}, y^{(i)}), h\big).$$

Empirical Risk Minimization



ERM for Parametrized Models

learnt (optimal) parameter vector

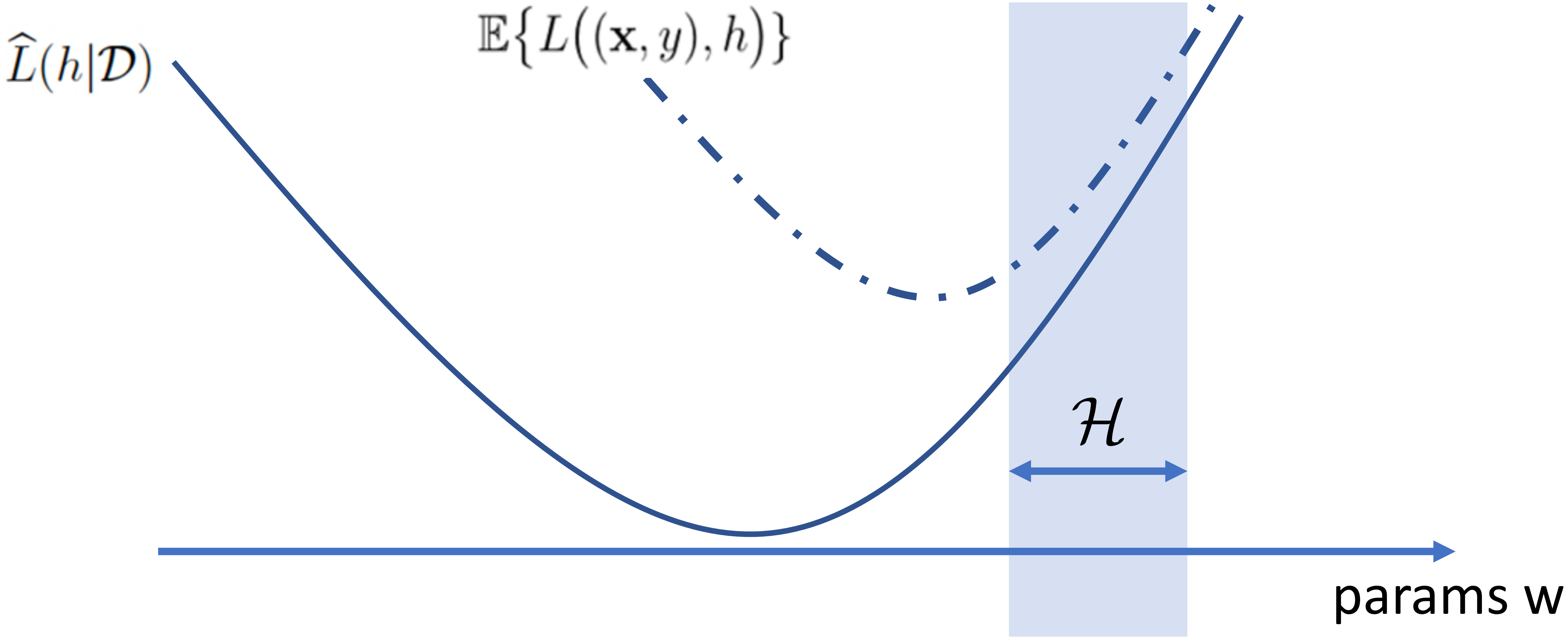
$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w})$$

loss incurred by h(.) for i-th data point

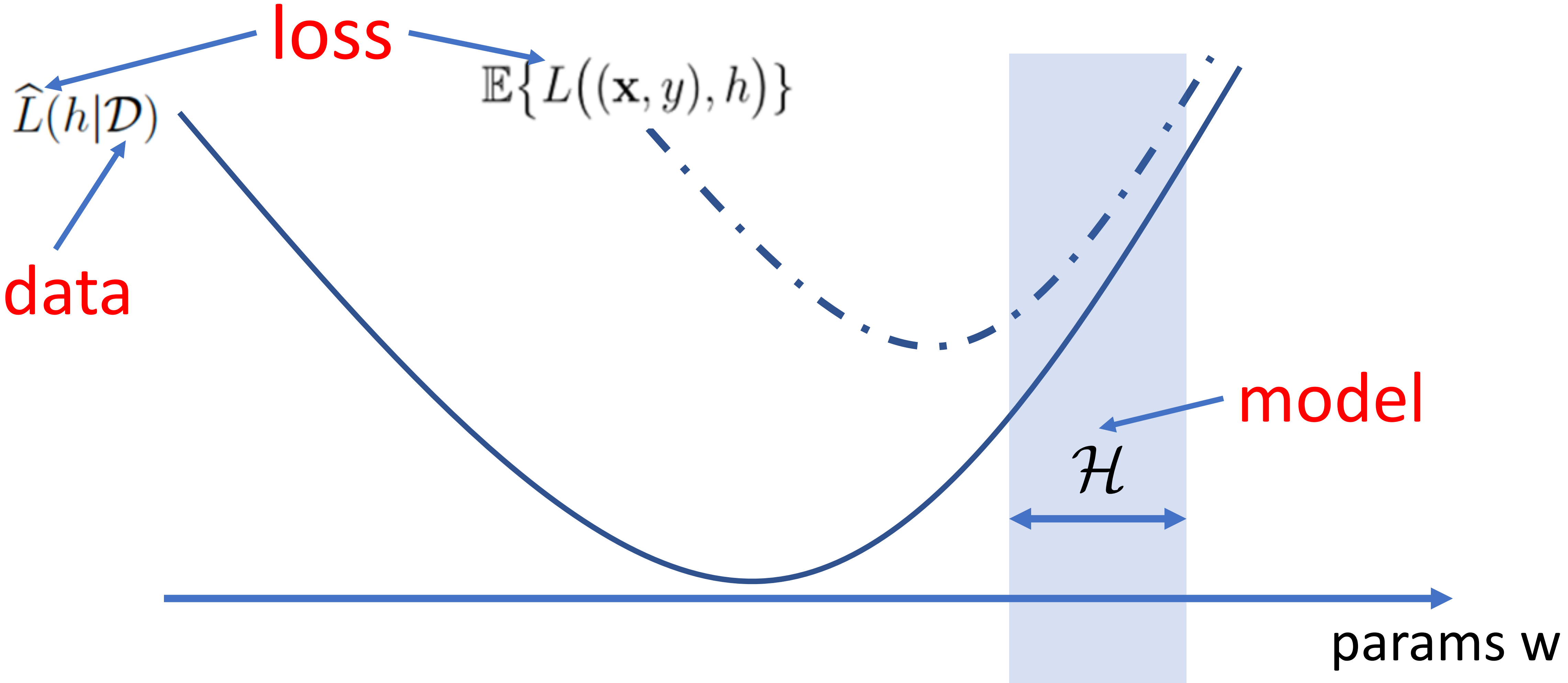
$$\text{with } f(\mathbf{w}) := \underbrace{(1/m) \sum_{i=1}^{m} L\big((\mathbf{x}^{(i)}, y^{(i)}), h^{(\mathbf{w})}\big)}_{\widehat{L}\left(h^{(\mathbf{w})} | \mathcal{D}\right)} .$$

average loss or empirical risk

ERM for Param. Models



Design Choices in ERM



yesterday ("Regression"): numeric labels, loss functions obtained from distance between numbers

today ("Classification"): discrete-valued labels, loss functions obtained from "confidence" measures

Logistic Regression [Sec. 3.6., MLBook]

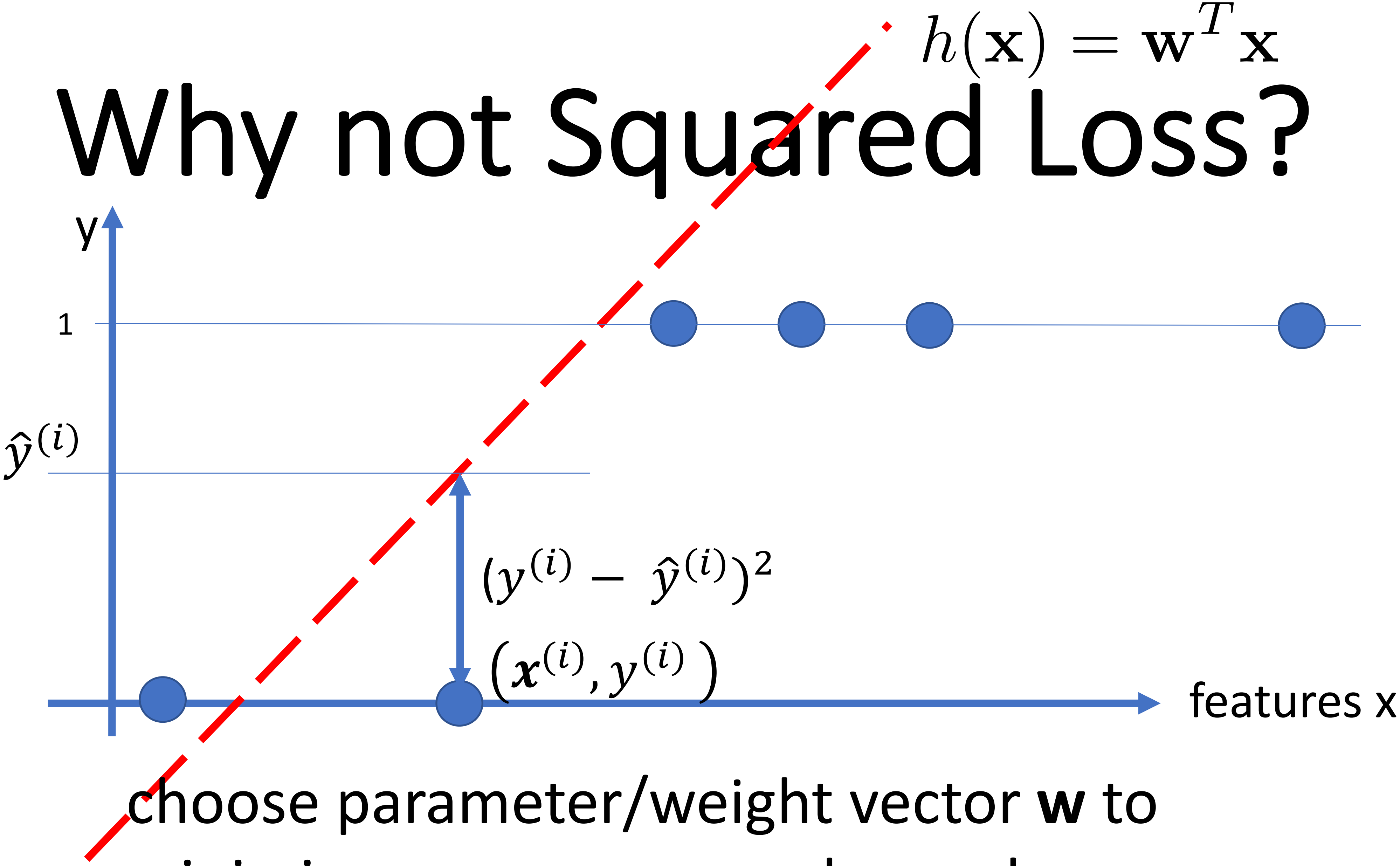
LogReg – Design Choices

- datapoints with numeric features (same as lin.reg.)
- binary label values, e.g., y=0 vs. y=1
- model = space of linear maps (same as lin.reg!)
- logistic loss (different from lin.reg!)

Linear Classifier

- log.reg. uses linear hypothesis h(x) =w'x
- sign of h(x) used for label prediction
- |h(x)| used as confidence measure
- h(x) = 100000 means very confident in \hat{y}=1
- h(x) = -100000 very confident in \hat{y}=0

Why not Squared Loss?



choose parameter/weight vector **w** to minimize average squared error loss

Some Loss Functions [Sec 2.3.3, MLBook]

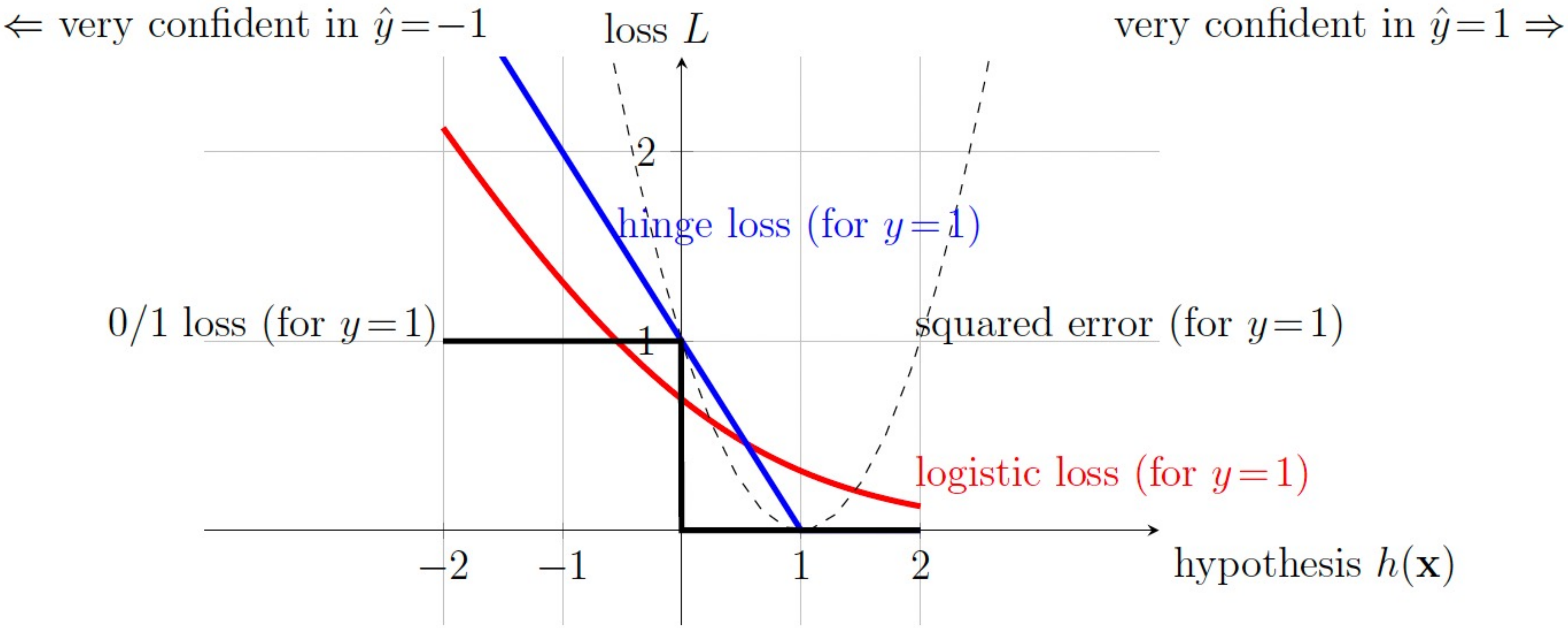
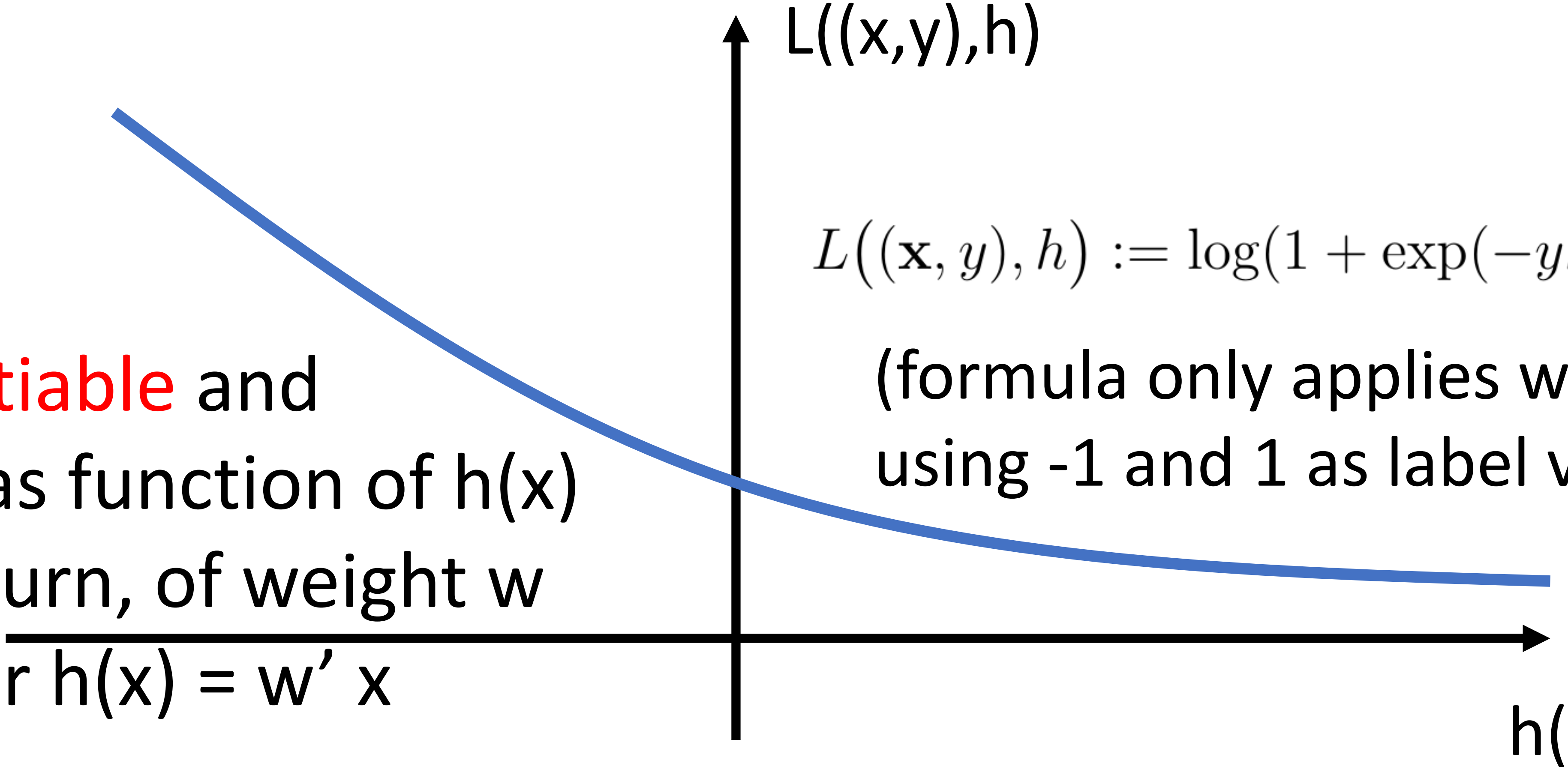


Figure 2.15: The solid curves depict three widely-used loss functions for binary classification.

Logistic Loss



$$L\big((\mathbf{x}, y), h\big) := \log(1 + \exp(-yh(\mathbf{x}))).$$

(formula only applies when using -1 and 1 as label values !)

differentiable and convex as function of h(x) and, in turn, of weight w for linear h(x) = w' x

LogReg. Probabilistic Interpretation

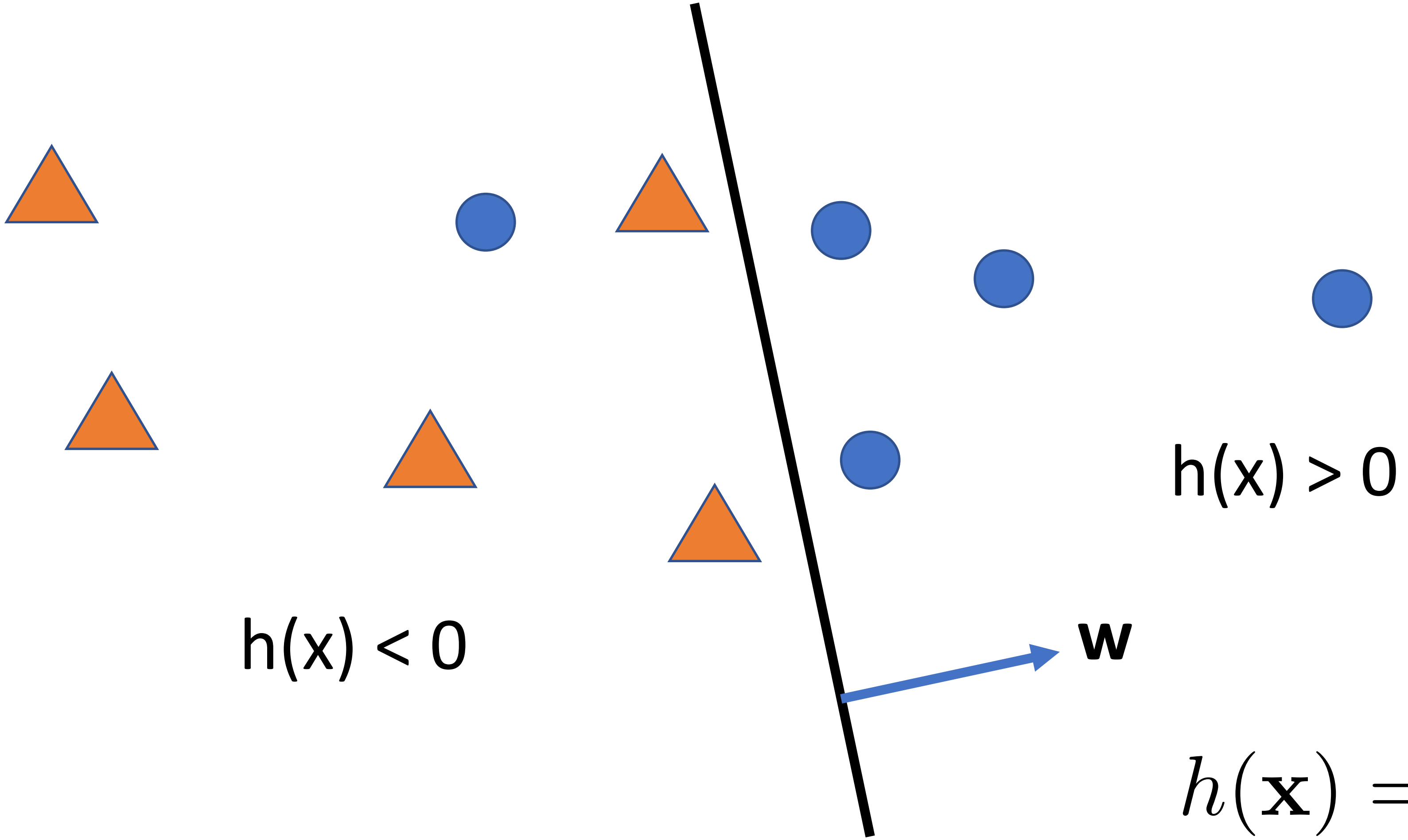
interpret label of data point as realization of binary RV with prob.

$$p(y=1;\mathbf{w}) = 1/(1+\exp(-\mathbf{w}^T\mathbf{x}))$$

$$\overset{h^{(\mathbf{w})}(\mathbf{x})=\mathbf{w}^T\mathbf{x}}{=} 1/(1+\exp(-h^{(\mathbf{w})}(\mathbf{x}))))$$

see Sec. 3.6 of MLBook

Decision Boundary of Log.Reg.



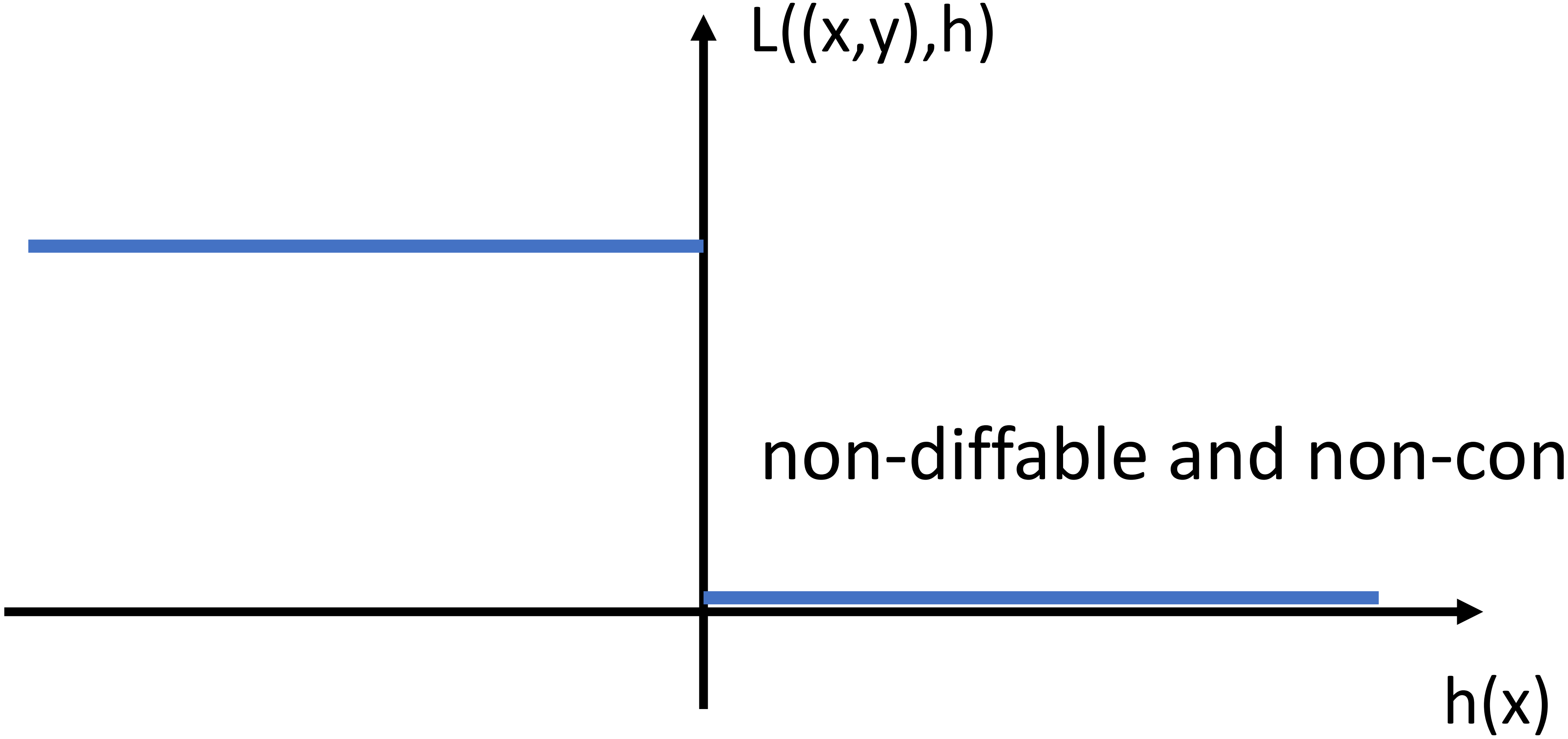
$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

Naïve Bayes’ Classifier (NBClass)

NBClass. – Design Choices

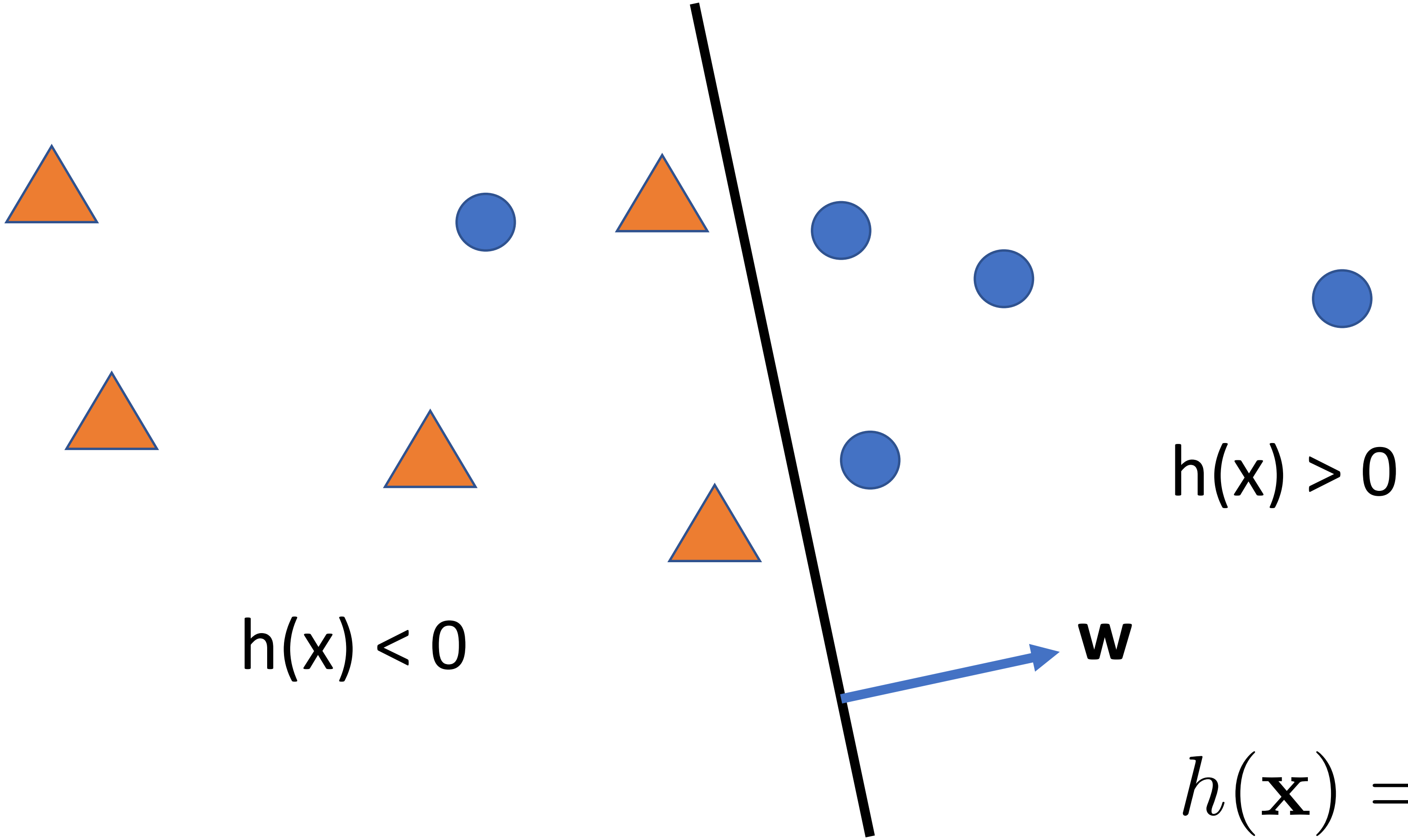
- datapoints with numeric features (same as log.reg.)
- binary label values, e.g., y=0 vs. y=1
- model = space of linear maps (same as log.reg!)
- 0/1 loss (different from log.reg!)

0/1 Loss



non-diffable and non-convex!

Naïve Bayes' Classifier



$$h(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

Logistic Loss vs. 0/1 Loss

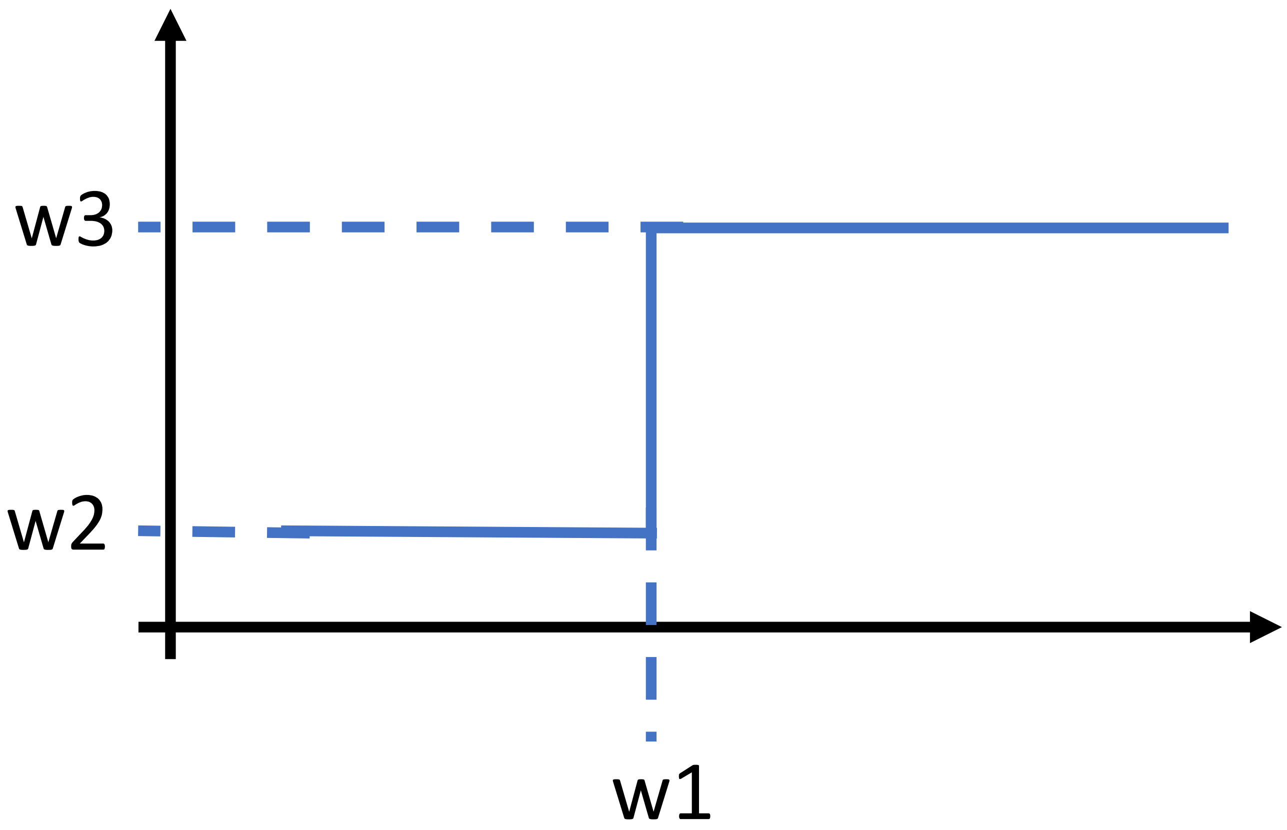
- logistic loss nice for optimization/solving ERM
- log. loss is not very interpretable
- what does log.loss = 0.3 mean ?
- average 0/1 loss (error rate) is more tangible
- accuracy = 1 – average 0/1 loss

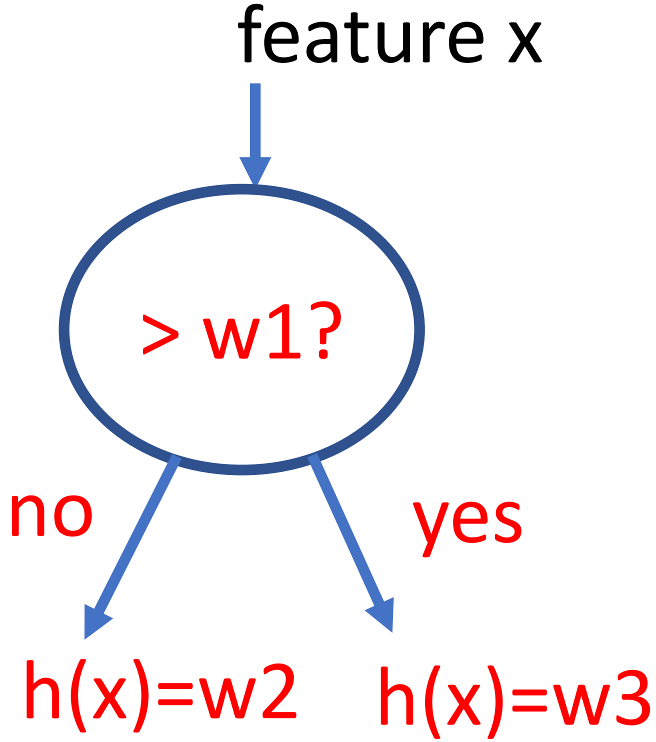
Decision Tree (DT) Classifier

Design Choices

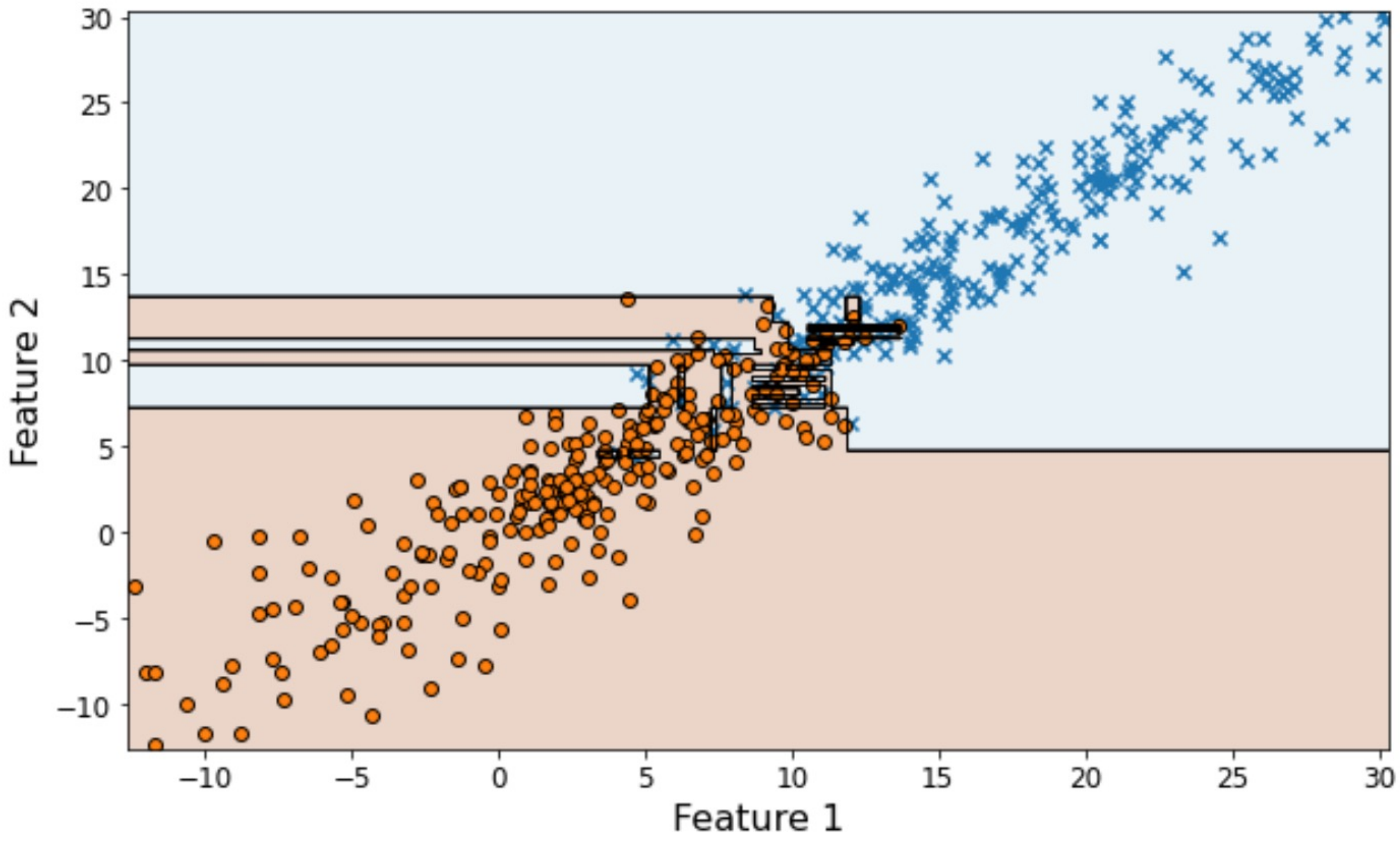
- datapoints with features and binary label
- label values arbitrary, we use y=0 vs. y=1
- model = maps given by flow chart (“decision trees”)
- different options for loss function

Parametrized DT

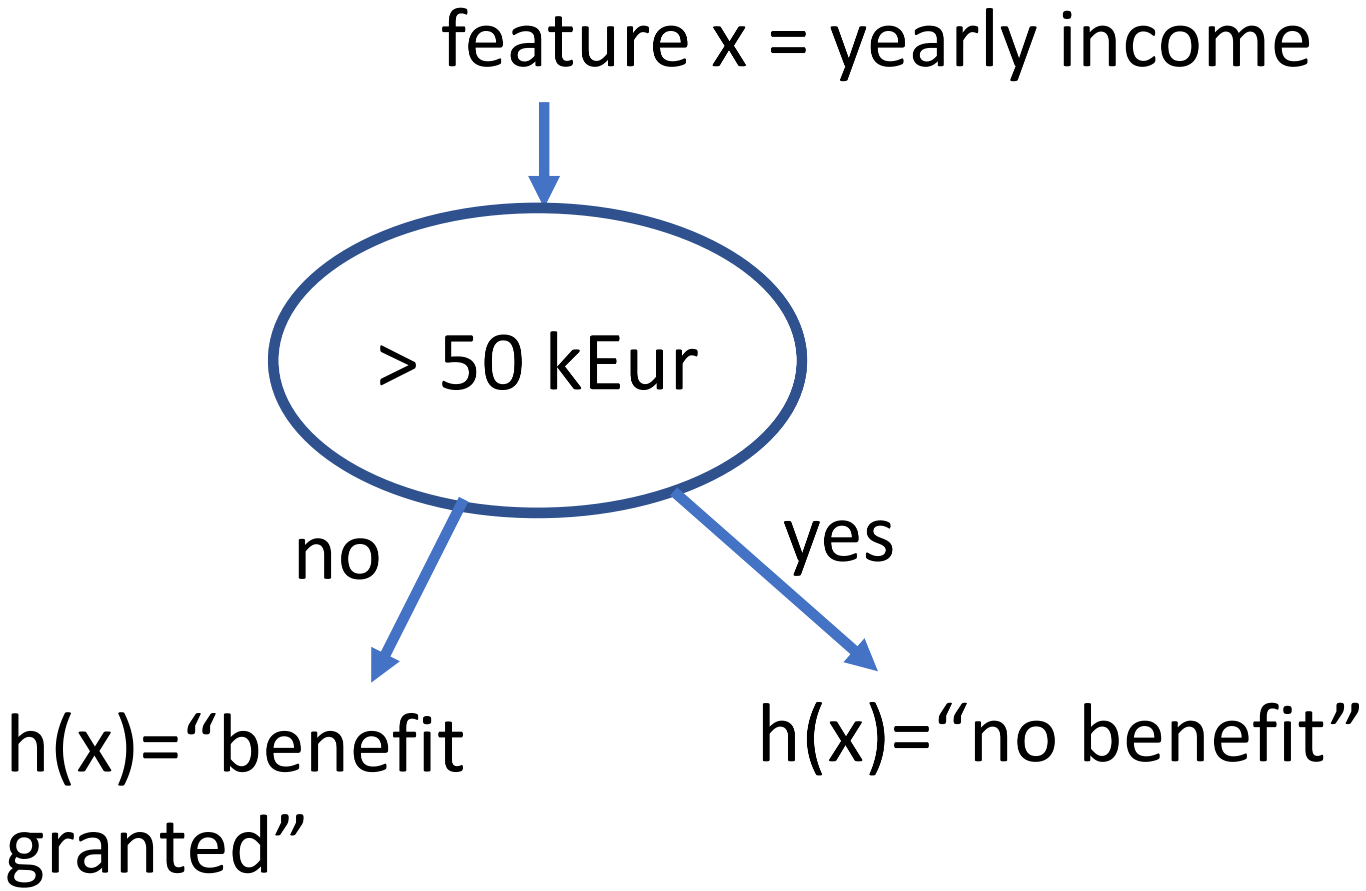




DT - Decision Boundary



DT - Interpretability



DT Pro/Con

- allows for non-linear decision boundary
- computationally expensive
- shallow DT considered interpretable

DT in Python

sklearn.tree.DecisionTreeClassifier

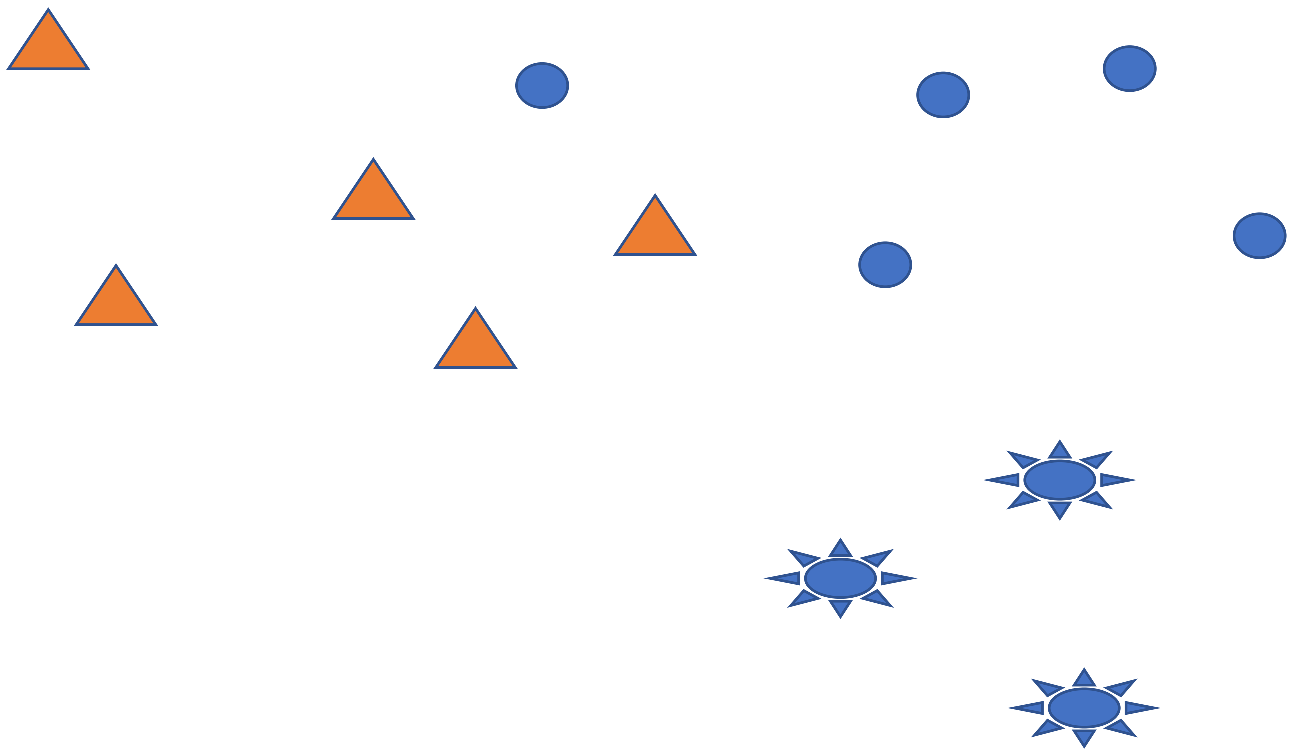
class `sklearn.tree.DecisionTreeClassifier`(**, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0*) [source]

Multi-Class Classification

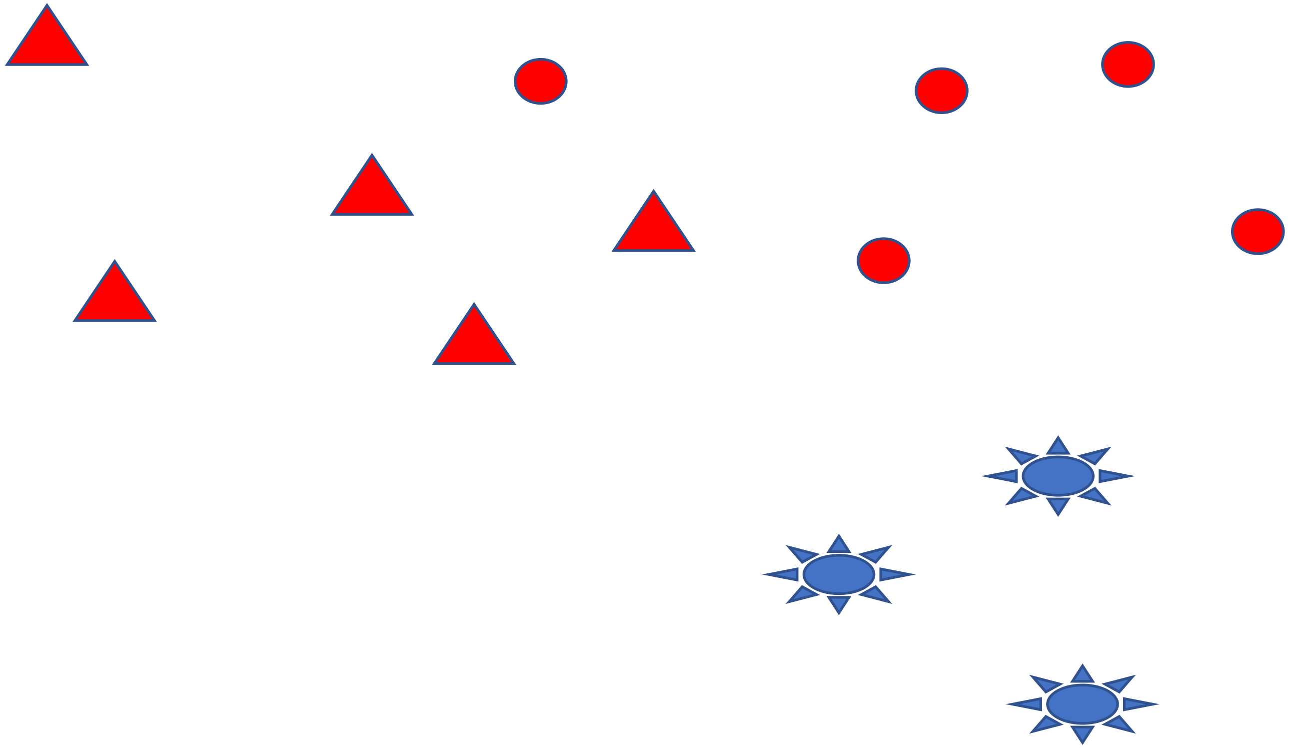
One-vs-Rest Trick

- data points with label values ”1”, “2”, “3”
- break into 3 binary class. problems
  - Problem 1: label values “1” ,“either 2 or 3”
  - Problem 2: label values “2”, “either 1 or 3”
  - Problem 3: label values ”3”, “either 2 or 3”

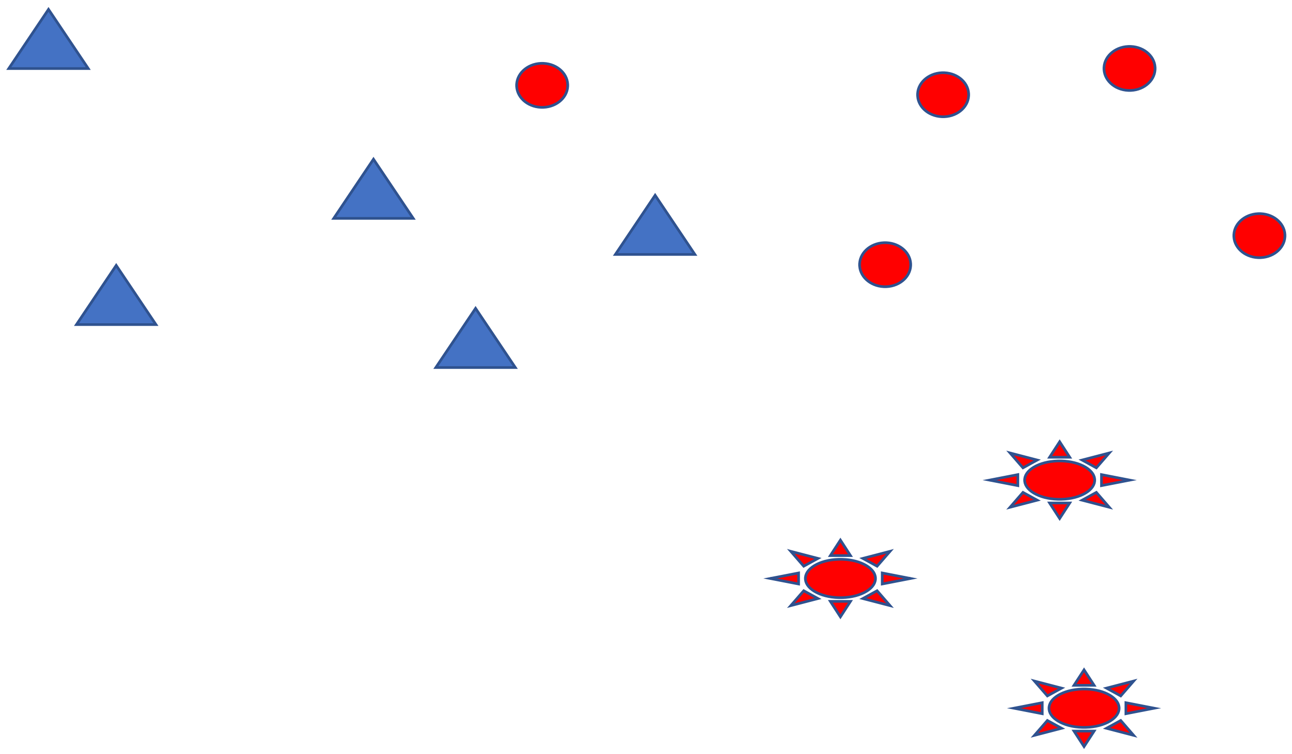
One-vs-Rest Trick



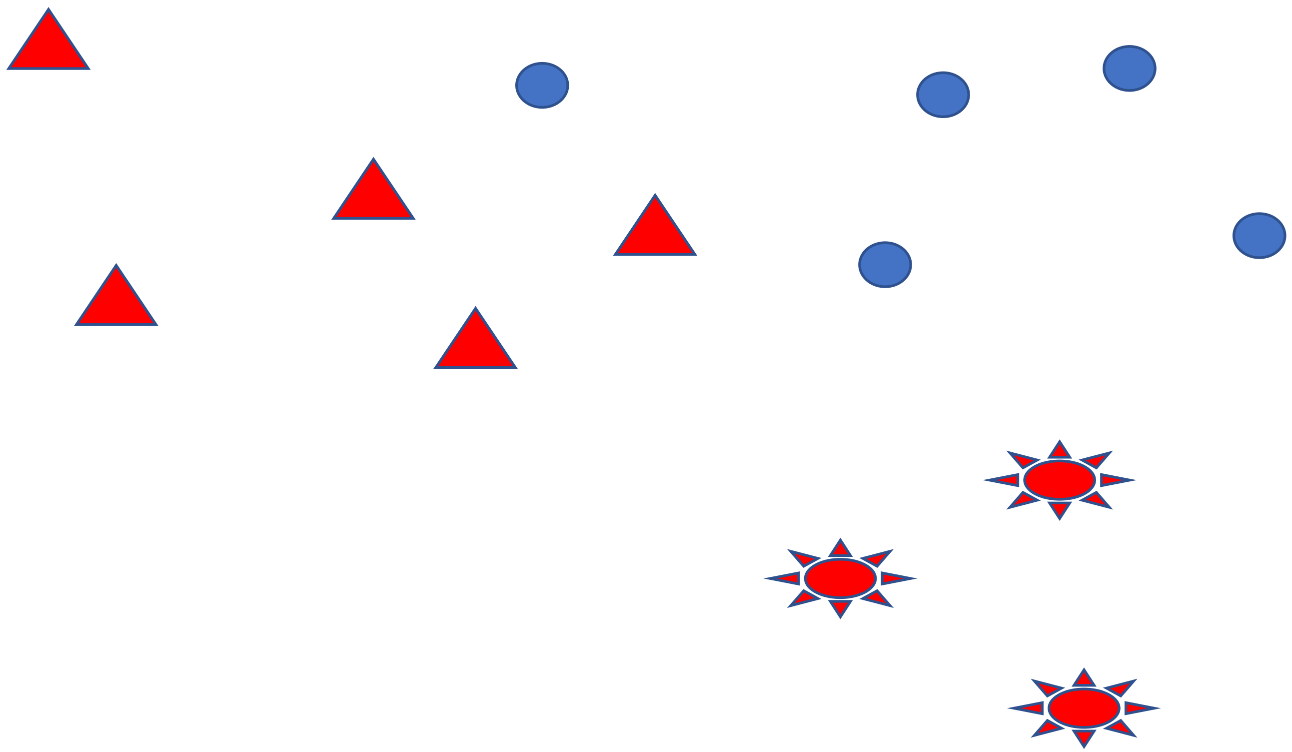
Sub-Problem 1



Sub-Problem 2



Sub-Problem 3



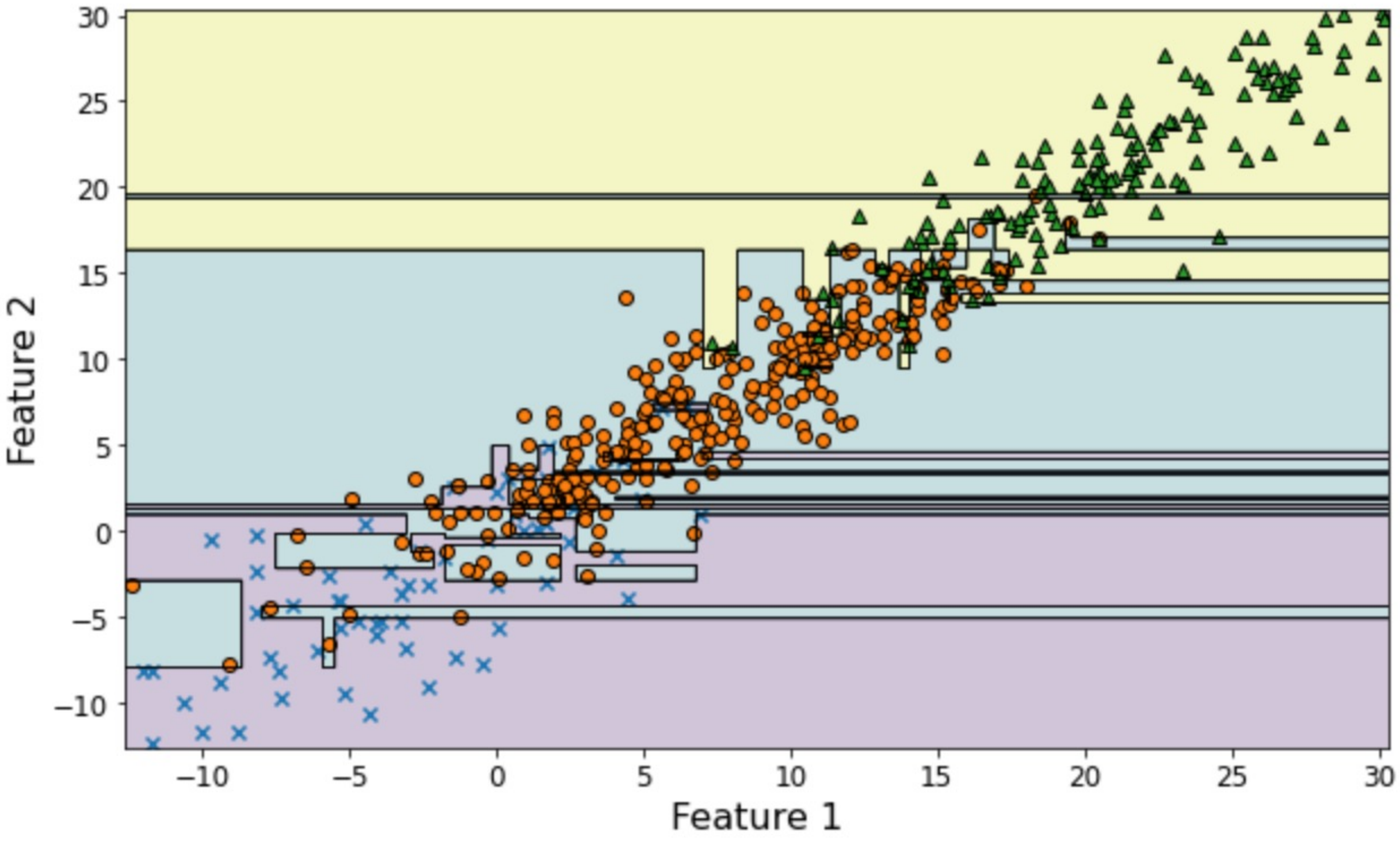
Multi-Class LogReg

- multi-class methods use specific loss functions
- 0/1 loss also works for > 2 label values (classes)
- but how to encode confidence in predictions?
- soft-max:

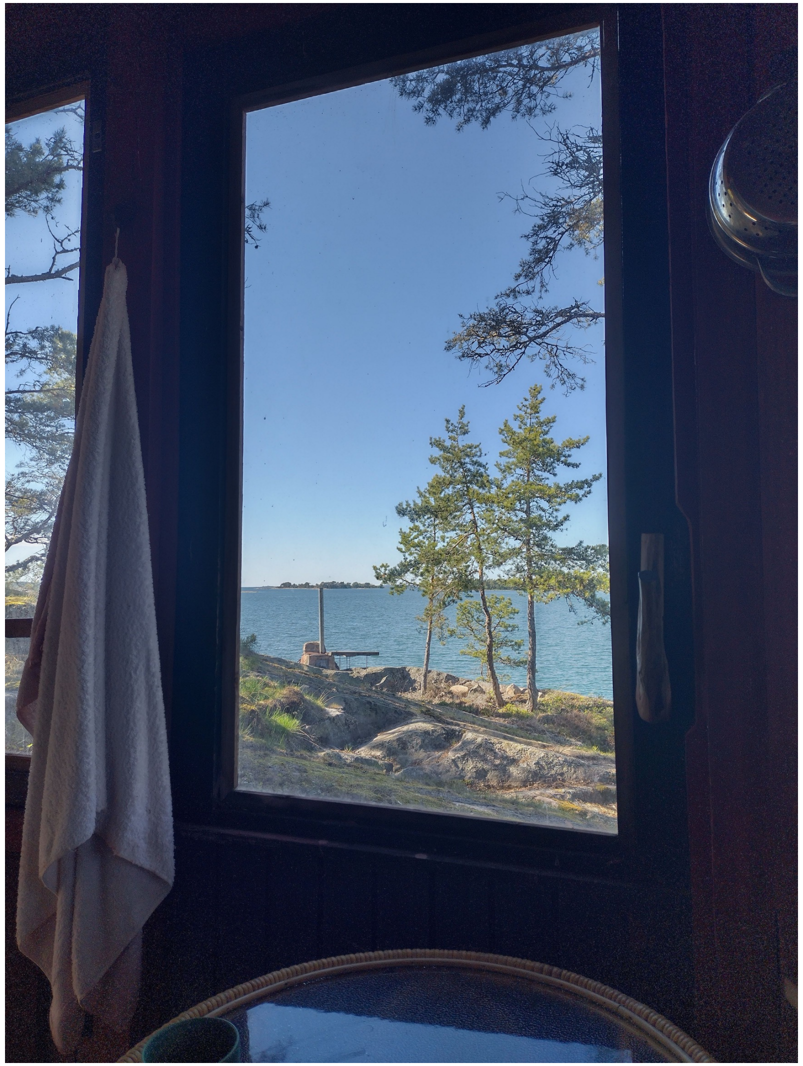
$$P(y=j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^{\mathsf{T}}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^{\mathsf{T}}\mathbf{w}_k}}$$

source: https://en.wikipedia.org/wiki/Softmax_function

DT Multi-Class



Multi-Label Classification



label y1 = contains tree ? yes/no
label y2 = contains house ? yes/no
label y3 = taken during leisure? yes/no
label y4 = taken during office? yes/no
label y5 = location in Finland? yes/no
label y6 = location in Sweden? yes/no

Bonsai - Diverse and Shallow Trees for Extreme Multi-label Classification

Sujay Khandagale[1], Han Xiao[2] and Rohit Babbar[2]

[1]Indian Institute of Technology Mandi, India
[2]Aalto University, Helsinki, Finland

"...benchmark Amazon-3M dataset with 3 million labels,..

Ignorant Approach

- consider each label separately
- solve plain classif. problem for each label
- ignores correlations among different labels

Multi-Class Approach

- each combination of label values defines category
- obtain a multi-class problem with many classes
- huge number of resulting categories

Multi-Task Learning

- each individual label results in separate learning task
- use similarities between learning tasks
- similarities inform regularization techniques
- more in Lecture “Regularization”

Y. Huang, W. Wang, L. Wang and T. Tan, "Multi-task deep neural network for multi-label learning," *2013 IEEE International Conference on Image Processing*, 2013, pp. 2897-2900, doi: 10.1109/ICIP.2013.6738596.

Summary

References

[MLBook] A. Jung, “Machine Learning: The Basics.”,
Springer, 2022,
preprint: mlbook.cs.aalto.fi