THE UNIVERSITY OF SYDNEY

Advanced Machine Learning

(COMP 5328)

Week 3 Tutorial:
Loss Functions and Convex Optimisation

Anjin Liu
anjin.liu@sydney.edu.au

Tutorial Contents

- Review (30min):
  - Lecture 1: Introduction to Machine Learning Problems
  - Lecture 2: Loss Functions and Convex Optimisation
- Tutorial exercise (30min)

Topics

Week	Lecture	Tutorial
1	Introduction to ML Problems	No tutorial
2	Loss Functions and Convex Optimisation	Tutorial 1 (take home)
3	Hypothesis Complexity and Generalisation	Tutorial 2
4	Dictionary Learning and NMF	Quiz
5	Sparse Coding and Regularisation	Tutorial 3
6	Learning with Noisy Data	Tutorial 4
7	Domain Adaptation and Transfer Learning	Tutorial 5
8	Learning with Noisy Data II: Label Noise	Tutorial 6
9	Reinforcement Learning	Tutorial 7
10	Causal Inference	Tutorial 8
11	Multi-task Learning	Tutorial 9
12	Guest Lecturer (Google)	Tutorial 10
13	Review	Tutorial 11

Assessment overview

- Quiz: 0%
  - Week 4
  - Individual
  - Contents in the first three weeks
  - Lower than 60%
  - The census date is on 1 September 2025

- Assignment 1: 25%
  - Due: Week 9 (9/10), 11:59pm
  - Groups of 3 or 4 students
  - Method comparison and analysis for feature noise

- Assignment 2: 25%
  - Due: Week 13 (6/11), 11:59pm
  - Groups of 3 or 4 students
  - Classification with noisy labels

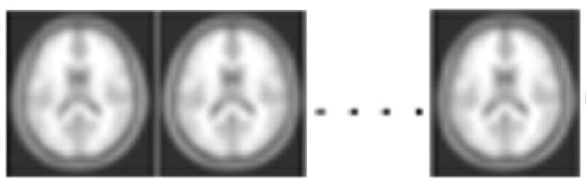
Elements of Machine Learning Algorithms

- I. Input training data
- II. Predefined hypothesis class
- III. Objective function
- IV. Optimisation method
- V. Output hypothesis

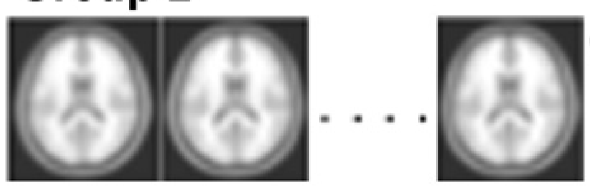
Elements of Machine Learning Algorithms

Input training data

Group 1



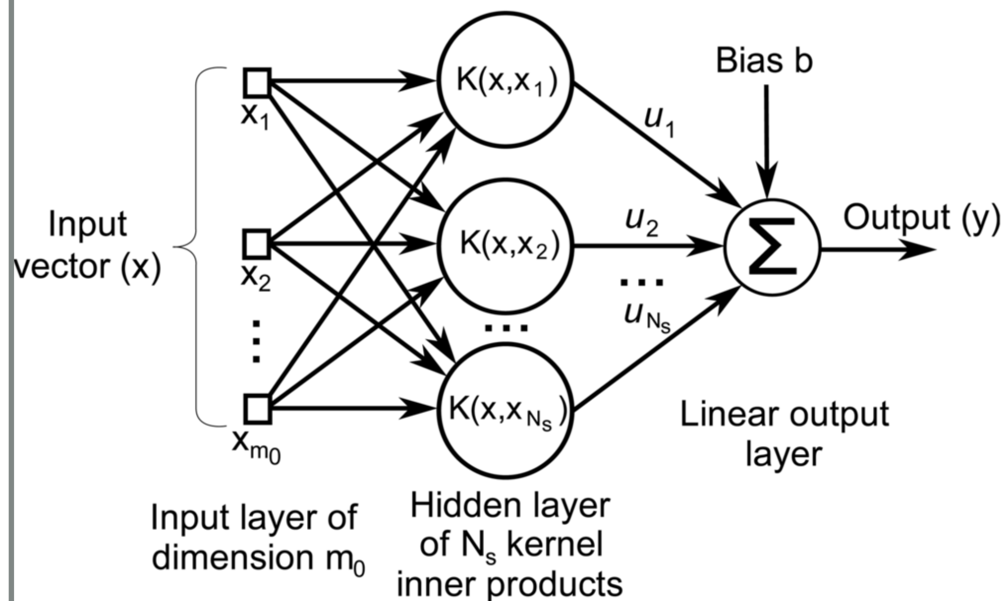
Group 2



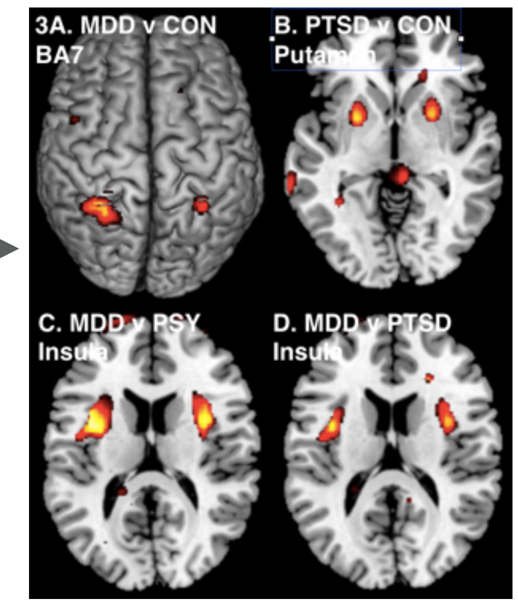
Data

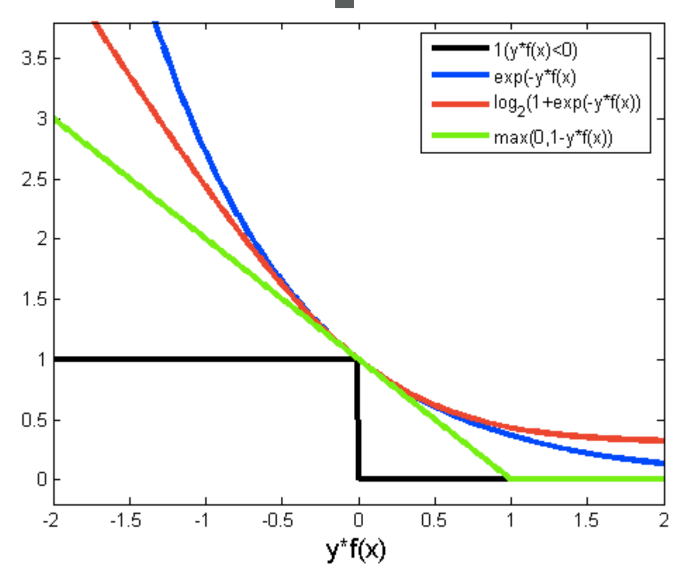
Input predefined hypothesis/function class

Mathematical Model



Output hypothesis (Predictions)





Objective function

Optimisation method

What is Machine Learning? (COMP5328)

THE UNIVERSITY OF SYDNEY

- Input training data: $S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$
- Input predefined hypothesis class: $H = \{h_1, h_2, \ldots\}$
- The objective function and optimisation method together make up a mapping: $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to H$
- Output hypothesis: h_S
- The overall learning algorithm is a mapping:

$$\mathcal{A} : S \in (\mathcal{X} \times \mathcal{Y})^n \mapsto h_S \in H$$

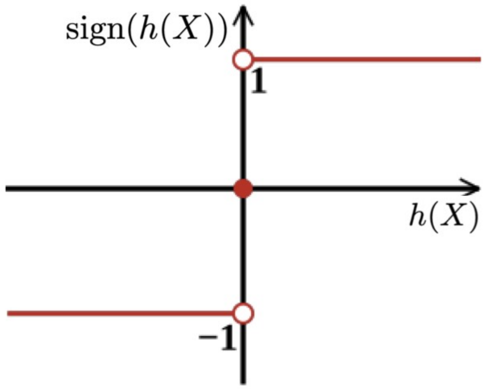
Objective function

- Given a classification task, we should firstly defined which hypothesis or classifier is the best.
- One intuitive way to defined the best classifier: the classifier that has the minimum classification error on the all possible data generated from the task.

Best classifier

- For a given data point (*X*, *Y*), the classification error for a hypothesis h is measured by the 0-1 loss function:

$$1_{\{Y \neq \text{sign}(h(X))\}} = \begin{cases} 0 & Y = \text{sign}(h(X)) \\ 1 & Y \neq \text{sign}(h(X)) \end{cases}$$



- The best classifier can be mathematically defined as:

$$\arg\min_h \frac{1}{|D|} \sum_{i \in D} 1_{\{Y_i \neq \text{sign}(h(X_i))\}}$$

where D is the set of indices of **all possible data** points of the task, and $|D|$ denotes the size of the set D.

Best classifier

- The best classifier (accuracy) can be mathematically defined as:

$$\arg\min_h \mathbb{E}[1_{\{Y \neq \mathrm{sign}(h(X))\}}]$$

- The distribution of data is unknown. We cannot calculate the expectation.

The law of large numbers

LLN describes the result of performing the same experiment a large number of times.

The average of the results obtained from a large number of independent trials should converge to the expected value.

$$\frac{1}{|D|}\sum_{i\in D} 1_{\{Y_i \neq \mathrm{sign}(h(X_i))\}} \xrightarrow{|D|\rightarrow\infty} \mathbb{E}[1_{\{Y \neq \mathrm{sign}(h(X))\}}]$$

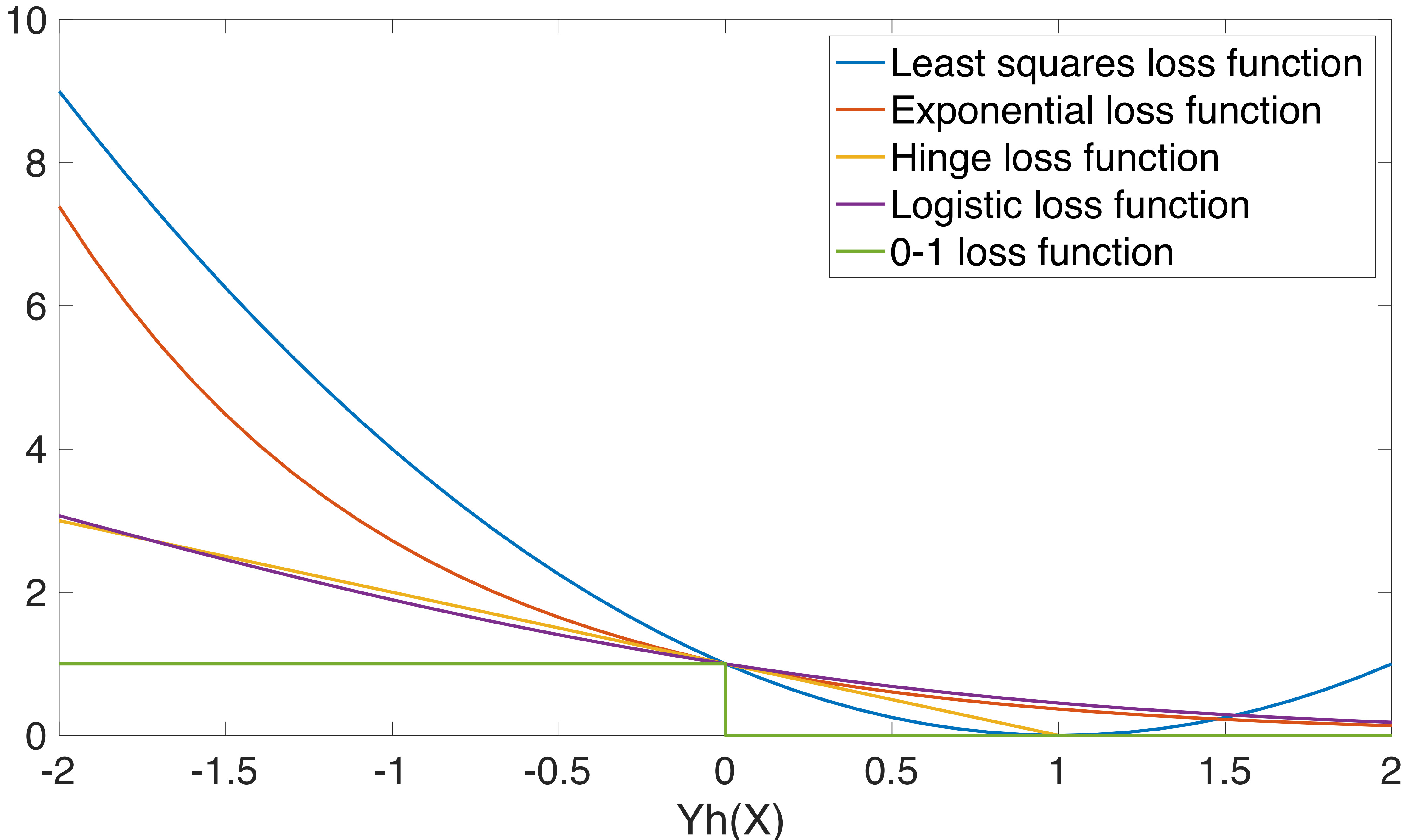
Surrogate loss functions

- Most optimisation methods exploit the derivative information. However, the 0-1 loss function is non-smooth and thus is non-differentiable.

- Convex objective has only one minimum. The convexity makes optimisation easier than the general case since local minimum must be a global minimum.

- Can we find some surrogate loss functions to approximate the 0-1 loss function, which are both smooth and convex?

Surrogate loss functions



Y $\{-1, +1\}$

Surrogate loss functions

- Popular surrogate loss functions:
- Hinge loss: $\ell(X, Y, h) = \max\{0, 1 - Yh(X)\}$
- Logistic loss: $\ell(X, Y, h) = \log_2(1 + \exp(-Yh(X)))$
- Least square loss: $\ell(X, Y, h) = (Y - h(X))^2$
- Exponential loss: $\ell(X, Y, h) = \exp(-Yh(X))$

Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- What are the differences between the 0-1 loss function and the surrogate loss functions?
- Classification-calibrated surrogate loss functions: which will result in the same classifier (same accuracy) as the 0-1 loss function if the training data is sufficiently large (an asymptotical property).
- Most of the popularly used surrogate loss functions are all classification-calibrated surrogate loss functions.

Bartlett, Peter L., Michael I. Jordan, and Jon D. McAuliffe. "Convexity, classification, and risk bounds." Journal of the American Statistical Association 101.473 (2006): 138-156.

Zhang, Jingwei, Tongliang Liu, and Dacheng Tao. "On the Rates of Convergence from Surrogate Risk Minimizers to the Bayes Optimal Classifier." arXiv preprint arXiv:1802.03688 (2018).

Surrogate loss functions

- How to check if a given surrogate loss function is a classification-calibrated surrogate loss functions?

Let $\phi(Yh(X)) = \ell(X, Y, h)$.

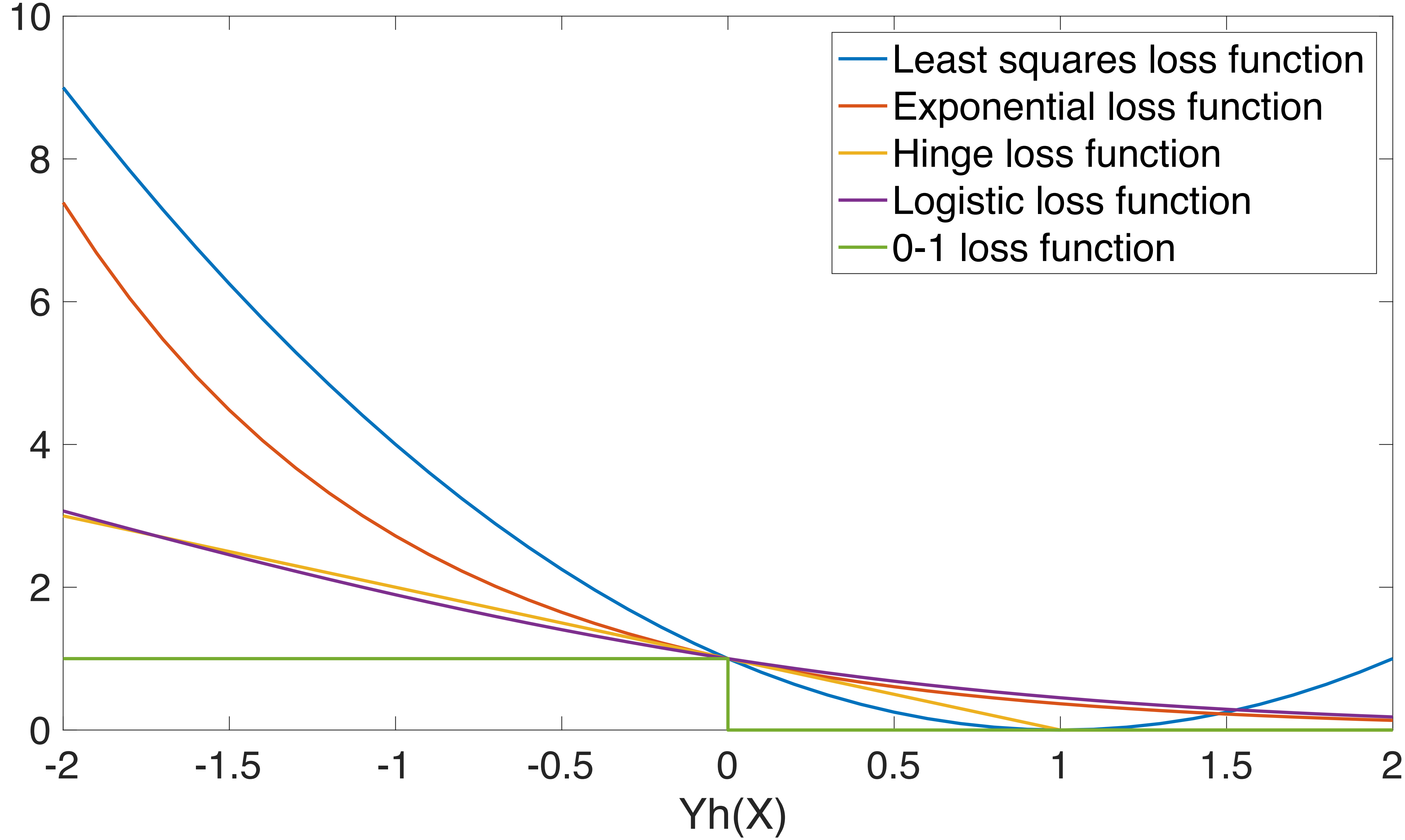
Given ϕ is convex, the loss function is classification-calibrated
if and only if ϕ is differentiable at 0, and

$$\boxed{\phi'(0) < 0.}$$

Bartlett, Peter L., Michael I. Jordan, and Jon D. McAuliffe. "Convexity, classification, and risk bounds." Journal of the American Statistical Association 101.473 (2006): 138-156.

Zhang, Jingwei, Tongliang Liu, and Dacheng Tao. "On the Rates of Convergence from Surrogate Risk Minimizers to the Bayes Optimal Classifier." arXiv preprint arXiv:1802.03688 (2018).

Surrogate loss functions

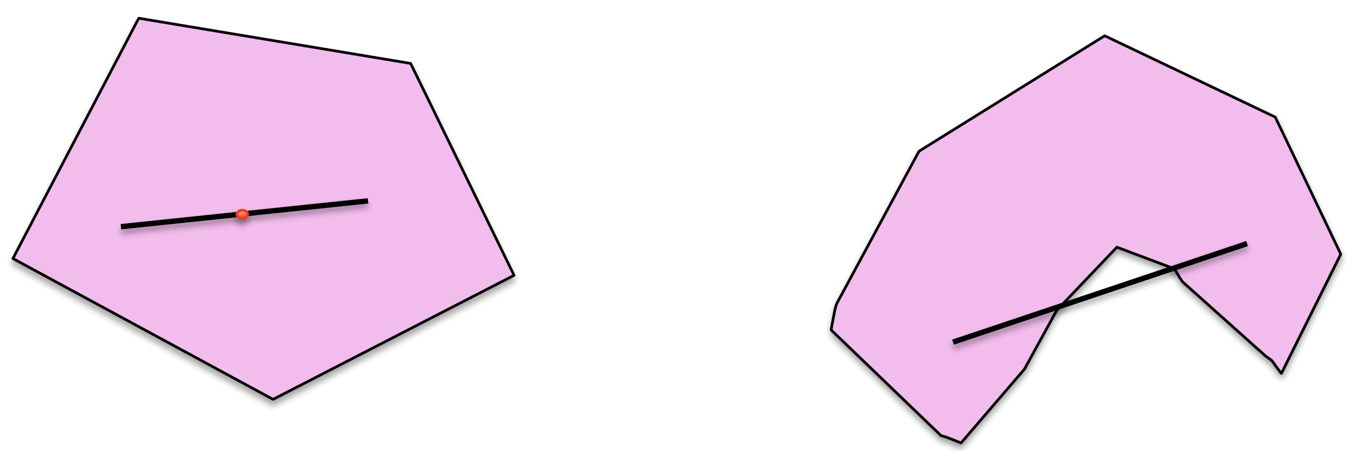


Basics I: Convex set

A set $C \in \mathbb{R}^d$ is convex if $x, y \in C$ and any $\theta \in [0, 1]$

$$\theta x + (1 - \theta) y \in C\,.$$

Examples: convex and non-convex sets, i.e,

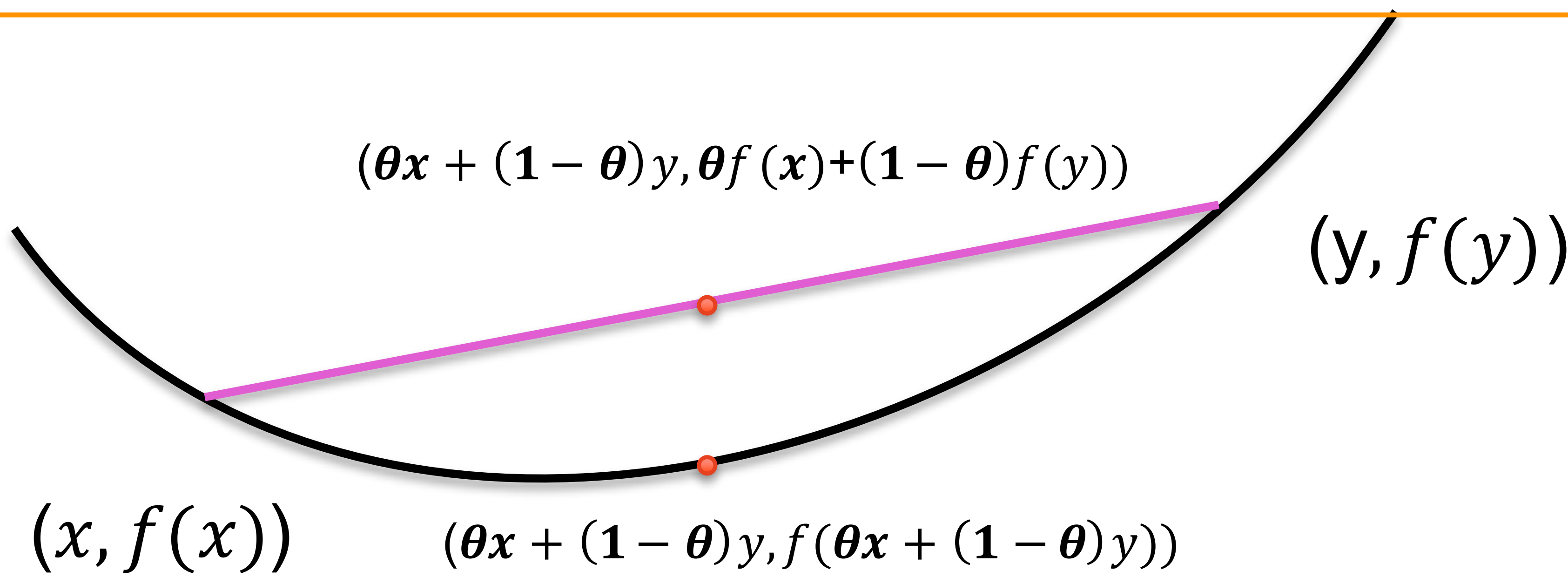


Basics II: Convex functions

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if its domain ($\mathrm{domain}\ f$) is a convex set and

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$

for all $x, y \in \mathrm{domain}\ f$, and $0 \leq \theta \leq 1$.



Basics II: Convex functions

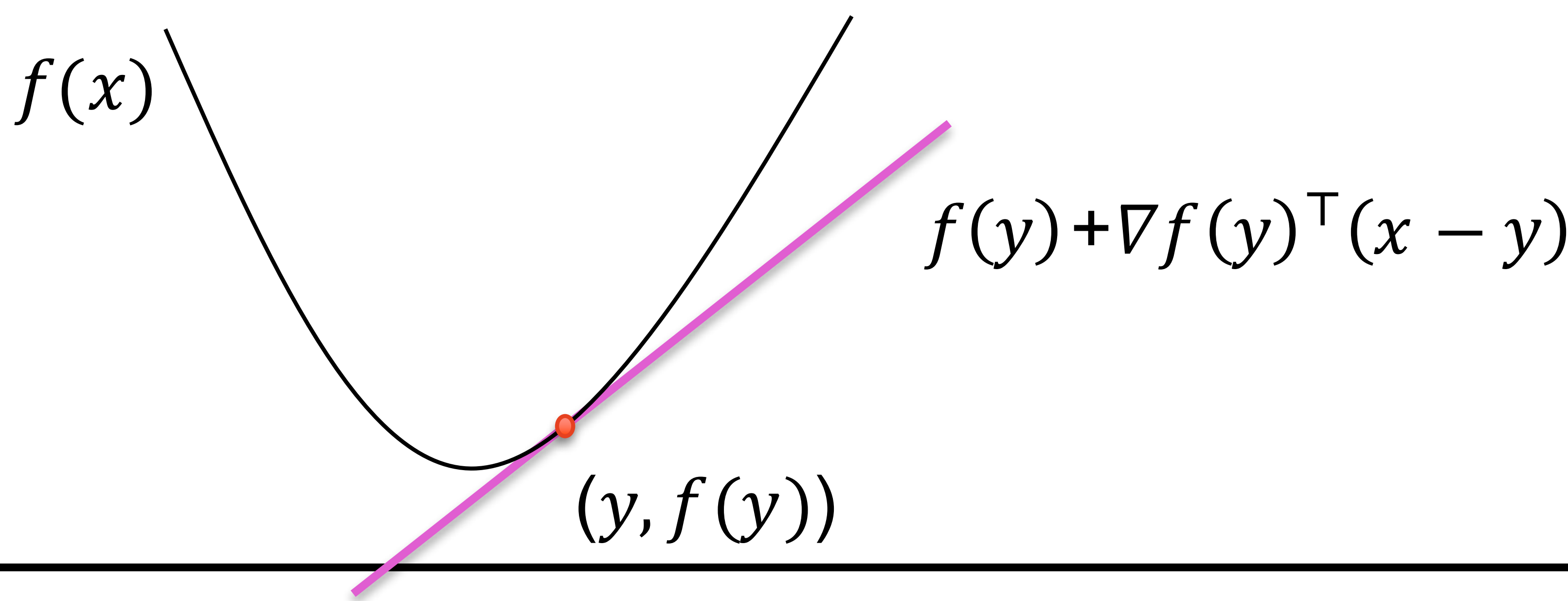
Function f is differentiable if the gradient

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_d}\right), \forall x \in \text{domain } f \subseteq \mathbb{R}^d$$

exists.

Note that differentiable f, with a convex domain, is convex if and only if

$$f(x) \geq f(y) + \nabla f(y)^\top (x-y), \quad \forall x, y \in \text{domain } f$$



Basics II: Convex functions

Function f is twice differentiable if the Hessian matrix

$$H_{ij} = \frac{\partial f(x)}{\partial x_i \partial x_j}, \forall x \in \mathrm{domain}\ f \subseteq \mathbb{R}^d$$

exists.

We now assume that f is twice differentiable, that is, its Hessian matrix exists at each point in the domain of f. Then f is convex if and only if the Hessian matrix is positive semidefinite for all point in the domain.

Basics II: Convex functions

We now assume that f is twice differentiable, that is, its Hessian matrix exists at each point in the domain of f. Then f is convex if and only if the Hessian matrix is positive semidefinite for all point in the domain.

A square matrix $H \in \mathbb{R}^{d \times d}$ is positive semidefinite if and only if

$$\forall x \in \mathbb{R}^d, x^\top H x \geq 0.$$

Or all its eigenvalues are non-negative.

Basics III: Convex functions

If f_1 and f_2 are convex functions then their pointwise maximum f, defined by

$$f(x) = \max\{f_1(x), f_2(x)\}.$$

is also convex. Note that

$$\text{domain } f = \text{domain } f_1 \cap \text{domain } f_2.$$

Basics III: Convex functions

If f_1 and f_2 are convex functions then their pointwise maximum $f,$ defined by

$$f(x) = \max\{f_1(x), f_2(x)\}$$

is also convex. Note that $\text{domain } f = \text{domain } f_1 \cap \text{domain } f_2.$

Proof: if $0 \leq \theta \leq 1$, $x, y \in$ domain f, then

$$\begin{aligned}
& f(\theta x + (1-\theta)y) \\
&= max\{f_1(\theta x + (1-\theta)y), f_2(\theta x + (1-\theta)y)\} \\
&\leq max\{\theta f_1(x) + (1-\theta)f_1(y), \theta f_2(x) + (1-\theta)f_2(y)\} \\
&\leq max\{\theta f_1(x), \theta f_2(x)\} + max\{(1-\theta)f_1(y), (1-\theta)f_2(y)\} \\
&= \theta f(x) + (1-\theta)f(y).
\end{aligned}$$

Basics III: Convex functions

Non-negative weighted sum:

$$f(x) = \theta_1 f_1(x) + \theta_2 f_2(y)$$

Composition with affine mapping:

$$g(x) = f(Ax + b)$$

Pointwise maximum:

$$f(x) = \max_i \{f_i(x)\}$$

The objective of SVM is convex:

$$f(x) = \frac{1}{2}\|x\|^2 + C\sum_{i=1}^{n} max\{0,1 - b_i a_i^\top x\}$$

The first term has Hessian matrix are positive, the second term is the sum of convex functions.

Taylor's Theorem

Let $k \geq 1$ be an integer and let the function $f : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. Then there exists a function $h_k : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = f(a) + f'(a)(x-a) + \ldots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k$$

and $\boxed{\lim_{x \to a} h_k(x) = 0.}$

Gradient descent method

Let $$f(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

$$h_{k+1} = h_k + \eta d_k \, .$$

By Taylor's theorem, we have

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta) \, .$$

For positive but sufficiently small η,

$\boxed{f(h_{k+1}) \text{ is smaller than } f(h_k),}$

if the direction d_k is chosen so that

$$\nabla f(h_k)^\top d_k < 0 \quad \text{when} \quad \nabla f(h_k) \neq 0.$$

Key points

- Elements of Machine Learning Algorithms
- Objective function, Best classifier, The law of large numbers
- Surrogate loss functions (smooth, convex), Classification-calibrated surrogate loss functions

Key points

- Convex optimization
  - Convex set
  - Convex function (definition, properties)
  - Taylor's Theorem
  - Gradient descent (d_k, η)