THE UNIVERSITY OF SYDNEY

Advanced Machine Learning (COMP 5328)

Loss Functions and Convex Optimisation

Tongliang Liu

THE UNIVERSITY OF SYDNEY

Review

Elements of Machine Learning Algorithms

THE UNIVERSITY OF SYDNEY

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

THE UNIVERSITY OF SYDNEY

Hypothesis

$$H_1 = \{h_1(x), h_2(x) : x \in \mathbb{R}\}.$$

$$H_2 = \{h(x) = w_0 + w_1 x + w_2 x^2 : x, w_0, w_1, w_2 \in \mathbb{R}\}.$$

$$H_3 = \{h(x) = w^\top x : x, w \in \mathbb{R}^d\}.$$

$$H_4 = \{h(x) = \text{sgn}(p(y = 1|x, \theta) - 0.5) : x \in \mathbb{R}^d, \theta \in \Theta\}.$$

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.

The law of large numbers

THE UNIVERSITY OF SYDNEY

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

Let be $x_1, x_2, \ldots, x_n$ iid examples drawn from distribution D. Then

$$\frac{1}{n}\sum_{i=1}^{n} f(x_i) \xrightarrow{n\to\infty} \mathbb{E}_{X\sim D}[f(X)].$$

The law of large numbers

THE UNIVERSITY OF SYDNEY

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

Toss a coin

$$\frac{1}{n}\sum_{i=1}^{n} 1_{\{x_i = \text{“}head\text{”}\}} \xrightarrow{n\to\infty} \mathbb{E}[1_{\{X = \text{“}head\text{”}\}}]$$

$$= \int P(X = \text{“}head\text{”}) 1_{\{x = \text{“}head\text{”}\}} dx = P(X = \text{“}head\text{”}).$$

The law of large numbers

THE UNIVERSITY OF SYDNEY

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.

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

Best classifier

THE UNIVERSITY OF SYDNEY

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

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

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

III. Objective function

- Given a classification task, we want to find a classifier such that the following is minimised:

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

- We don't have the distribution of data. Fortunately, we have some examples (or a training sample) draw from the distribution:

$$S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

- Because of the law of large numbers, we can use

$$\frac{1}{n} \sum_{i=1}^{n} 1_{\{Y \neq \text{sign}(h(X))\}}$$

(unbiased estimator)

to estimate $\mathbb{E}[1_{\{Y \neq \text{sign}(h(X))\}}]$

Objective function

- The empirical estimator is unbiased because

$$\arg\min_h \frac{1}{n}\sum_{i=1}^{n} 1_{\{Y_i \neq h(X_i)\}} \overset{n\to\infty}{\to} \arg\min_h \mathbb{E}\left[1_{\{Y \neq h(X)\}}\right]$$

- This also explains why big data is very helpful.

- The objective function is not convex or smooth, hard to optimise.

THE UNIVERSITY OF SYDNEY

Loss functions

Best classifier

- The best classifier can be mathematically defined as:

$$\arg\min_{h} \mathbb{E}\left[1_{\{Y \neq \text{sign}(h(X))\}}\right]$$

- Some problems: 1, the distribution of data is unknown. We cannot calculate the expectation. 2, the objective function is not convex or smooth, hard to optimise. 3, what kind of hypothesis h should we employ to fit the data?

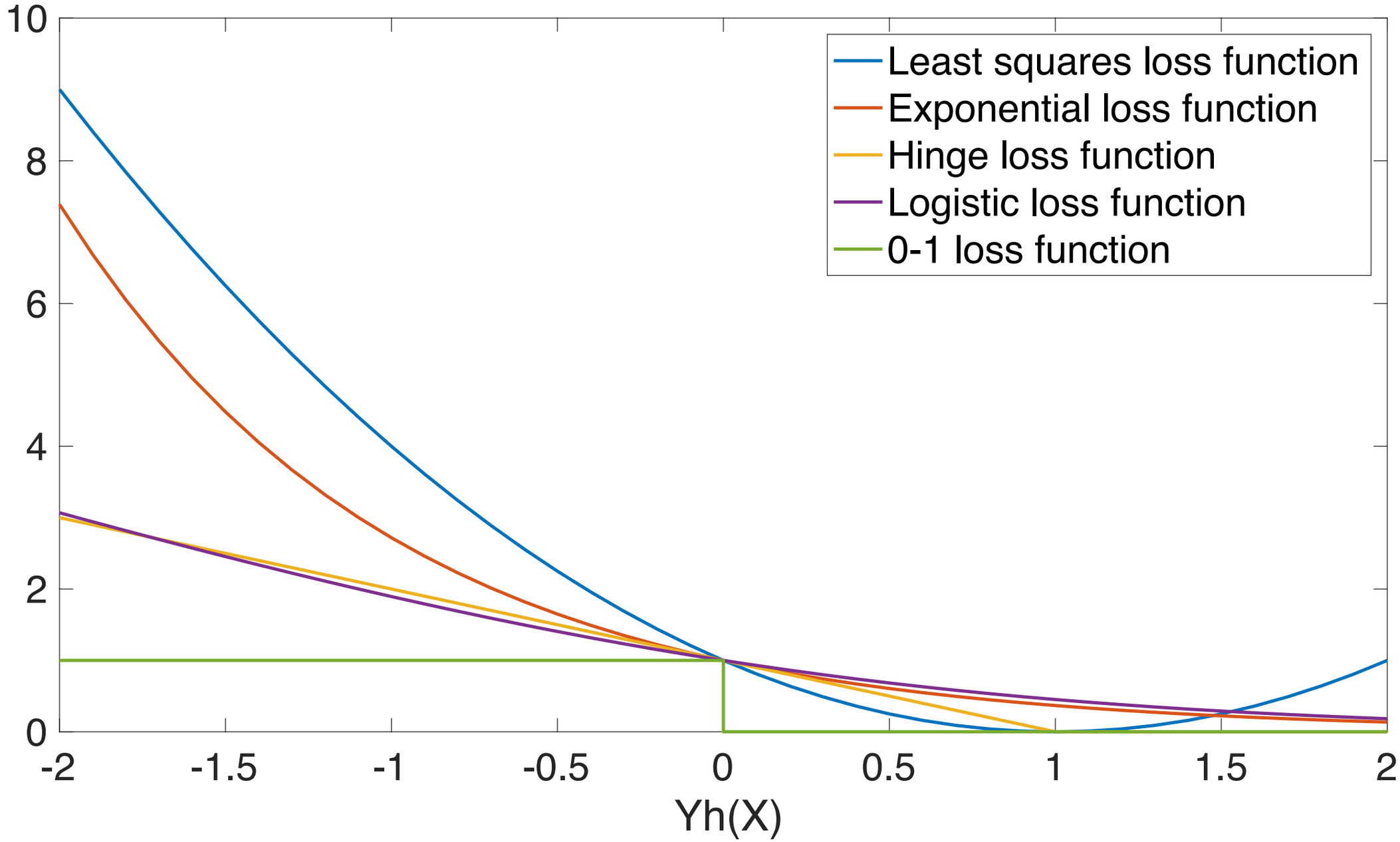
Surrogate loss functions

- 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

- 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



Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- Not all surrogate loss functions are convex
- Cauchy loss:

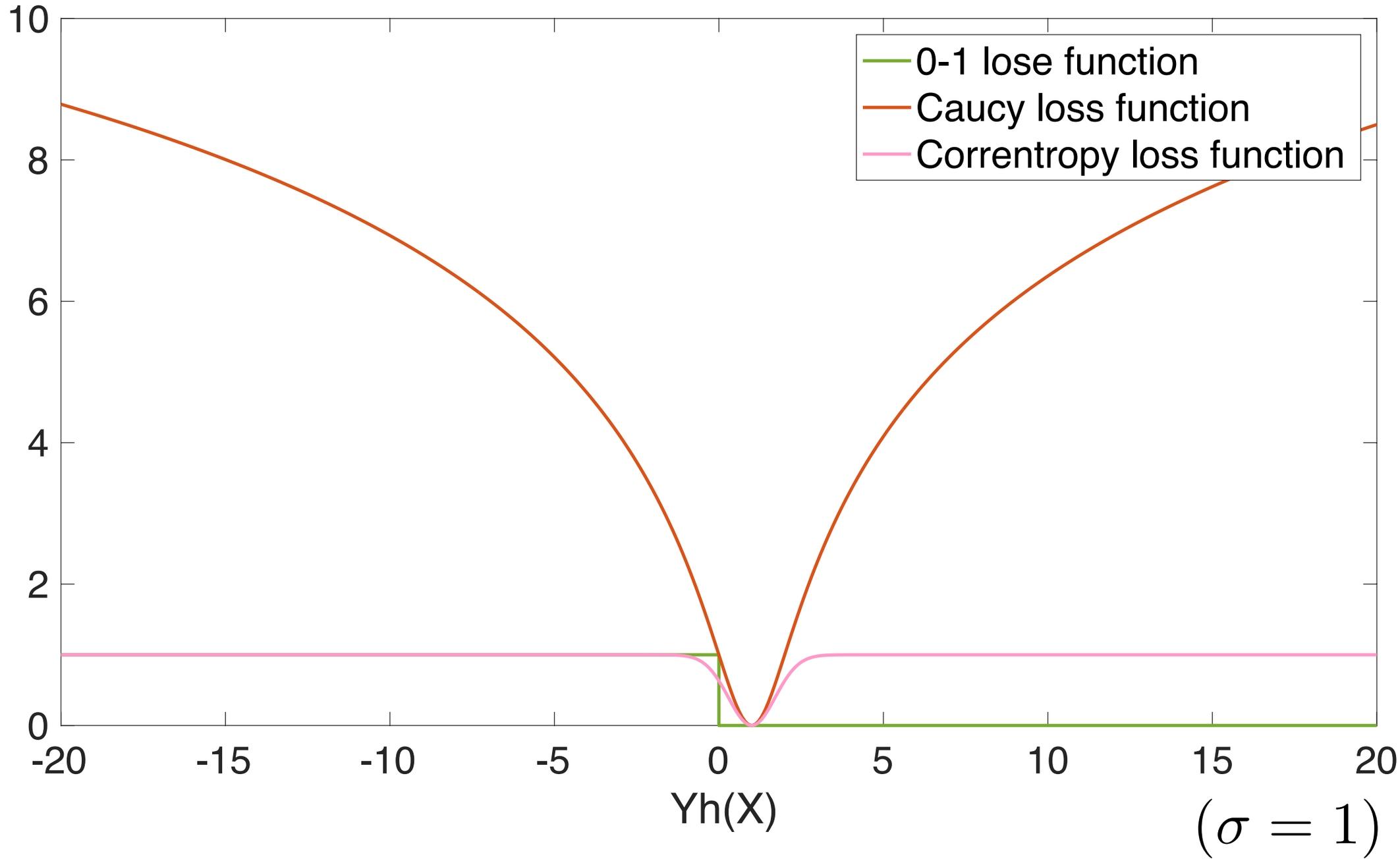
$$\ell(X, Y, h) = \log_2 \left(1 + \left(\frac{1 - Yh(X)}{\sigma}\right)^2\right)$$

- Correntropy loss (Welsch loss):

$$\ell(X, Y, h) = \left(1 - \exp\left(-\left(\frac{1 - Yh(X)}{\sigma}\right)^2\right)\right)$$

Surrogate loss functions

THE UNIVERSITY OF SYDNEY



$(\sigma = 1)$

Surrogate loss functions

- We have two natural questions:
- What are the differences between the 0-1 loss function and the surrogate loss functions?
- What are the differences among those different surrogate loss functions? (We will provide an answer to this question in Week 6.)

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

THE UNIVERSITY OF SYDNEY

- How to check if a given surrogate loss function is a 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

- 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: $\ell(X, Y, h) = (Y - h(X))^2 = (1 - Yh(X))^2$
- Exponential loss: $\ell(X, Y, h) = \exp(-Yh(X))$

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

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

$$\phi'(0) < 0.$$

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).

Objective function

When employing classification-calibrated surrogate loss function, the empirical estimator is unbiased:

$$h_n = \arg\min_h \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

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

$$\mathbb{E}[1_{Y \neq \text{sign}(h_n(X))}] \xrightarrow{n \to \infty} \mathbb{E}[1_{Y \neq \text{sign}(h_c(X))}]$$

Objective function

Recall that a machine learning algorithm is a mapping to find a hypothesis to fit the data

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

The mapping is an optimisation procedure that picks a hypothesis from the predefined hypothesis class to minimise or maximise the objective.

$$\arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

THE UNIVERSITY OF SYDNEY

Convex optimisation

Basics I: Convex combination

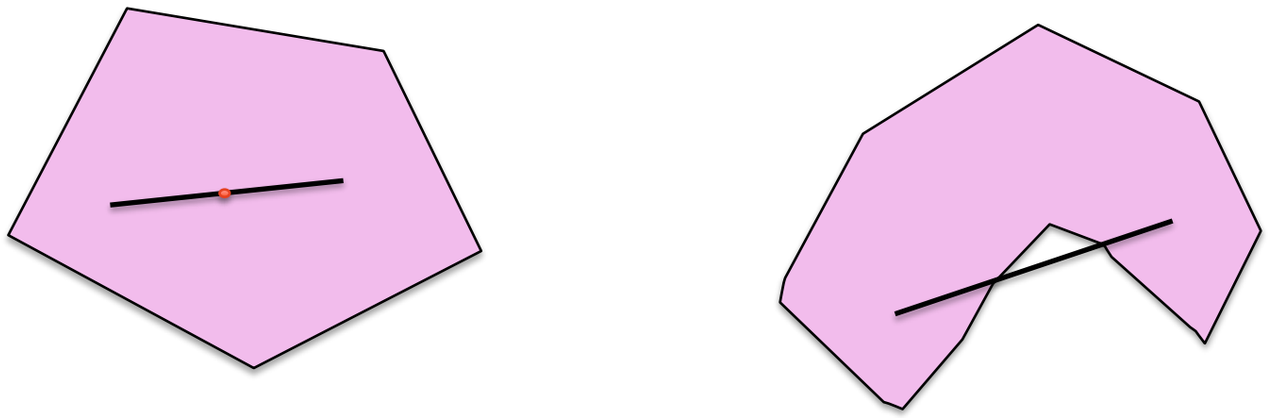
For two points x, y , $\theta x + (1 - \theta)y$ is a convex combination if the scalar $\theta \in [0, 1]$.

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,

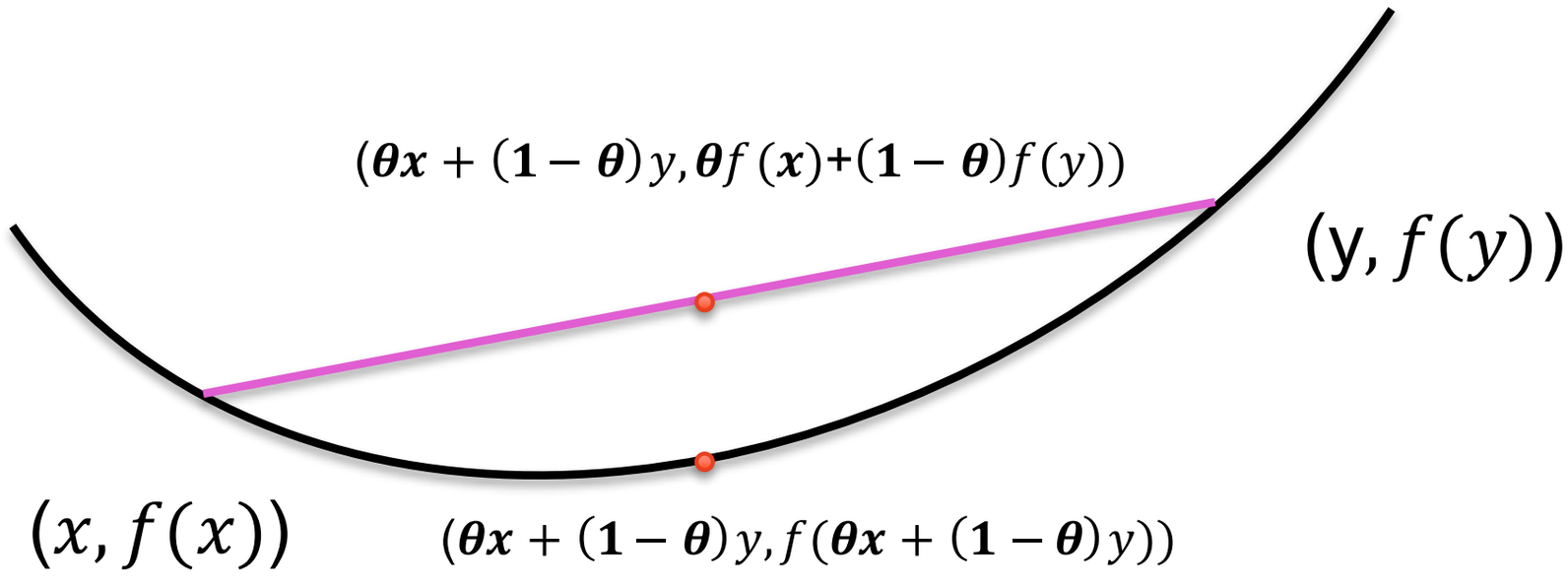


Basics II: Convex functions

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

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

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



Basics II: Convex functions

Function f is differentiable if the gradient

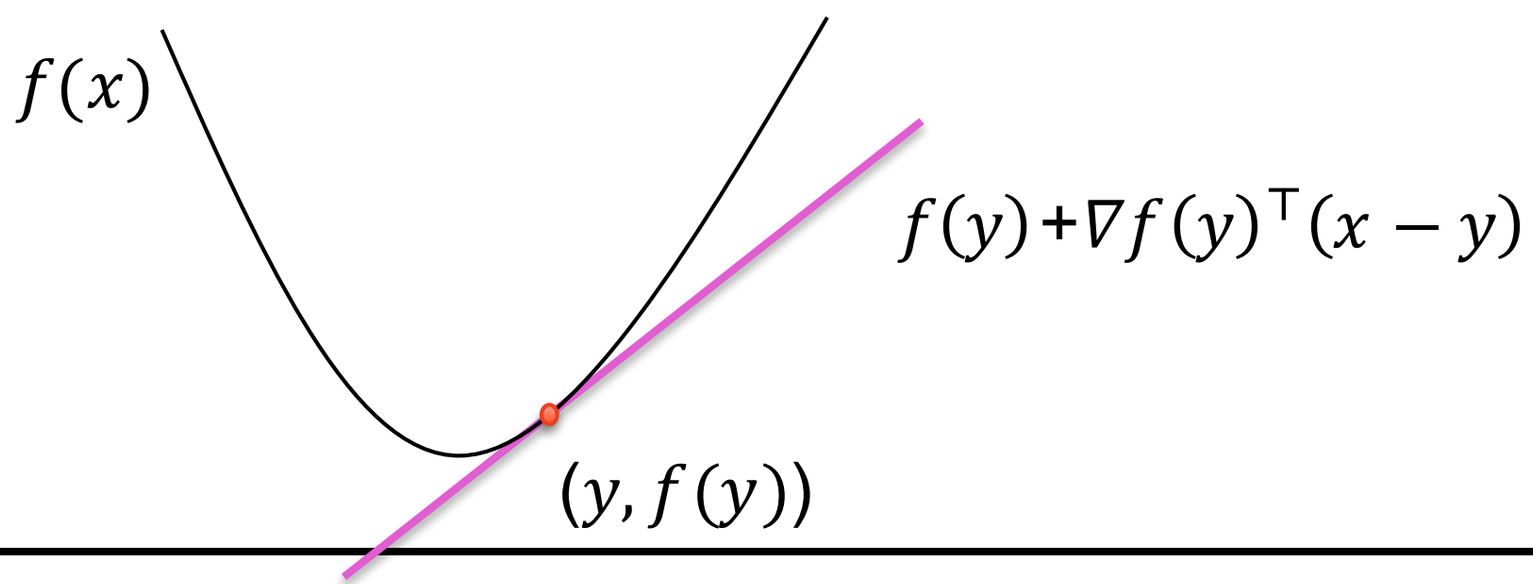
$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_d} \right), \forall x \in \text{domain } f \subseteq \mathbb{R}^d$$

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

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

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.

Unconstrained optimisation

- Unconstrained convex optimisation problem

$$\arg\min_{h} f(h).$$

Pick one from the predefined hypothesis class H to minimise the objective, i.e.,

$$\arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h) = \arg\min_{h \in H} f(h)$$

where the loss function ℓ is a convex surrogate for the 0-1 loss function.

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

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

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

Taylor's Theorem

- Example

Let $f(x) = \frac{1}{6}x^3$, $K = 2$. The Taylor series at the point 1 is as follows

$$f(x) = f(1) + f'(1)(x-1) + \frac{f''(1)}{2}(x-1)^2$$

$$+ \ldots + \frac{f^{(k)}(1)}{K!}(x-1)^k + h_k(x)(x-1)^k$$

$$= \frac{1}{6} + \frac{1}{2}(x-1) + \frac{1}{2}(x-1)^2 + h_2(x)(x-1)^2$$

$$x \to 1$$

$$= \boxed{\frac{1}{6} + \frac{1}{2}(x-1) + \frac{1}{2}(x-1)^2} + o((x-1)^2)$$

Small-o Notation

$f(x) = o(g(x)), x \to 0$ means that $\frac{f(x)}{g(x)} \xrightarrow{x \to 0} 0.$

The notation $f(x-1) = o((x-1)^2), x \to 1$ means that when x approaches 1, $f(x-1)$ converges to 0 faster than $(x-1)^2$.

Example 3 implies $\frac{\frac{1}{6}x^3 - \left(\frac{1}{6} + \frac{1}{2}(x-1) + \frac{1}{2}(x-1)^2\right)}{(x-1)^2} \xrightarrow{x \to 1} 0.$

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,
$f(h_{k+1})$ is smaller than $f(h_k)$,
if the direction d_k is chosen so that

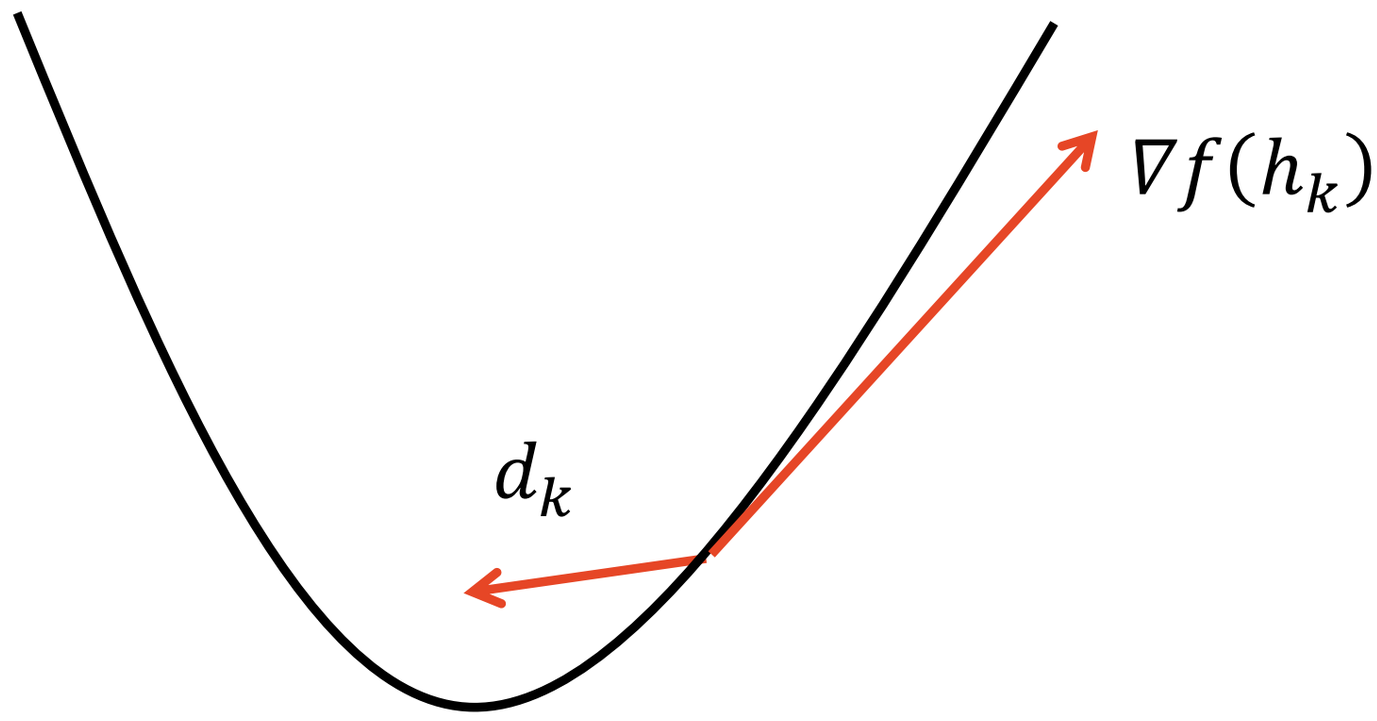
$$\nabla f(h_k)^\top d_k < 0 \quad \text{when} \quad \nabla f(h_k) \neq 0.$$

An iterative updating method

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

Two problems:

- How to find d_k?
- How to choose η?



To find d_k

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

Set $d_k = -D^k \nabla f(h_k)$, many gradient methods are specified in the form

$$h_{k+1} = h_k - \eta D^k \nabla f(h_k).$$

D^k is a positive definite symmetric matrix,

$$\nabla f(h_k)^\top D^k \nabla f(h_k) > 0.$$

η is a positive such that

$$f(h_{k+1}) = f(h_k) - \eta \nabla f(h_k)^\top D^k \nabla f(h_k).$$

To find d_k

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

Set $d_k = -D^k \nabla f(h_k)$, many gradient methods are specified in the form

$$h_{k+1} = h_k - \eta D^k \nabla f(h_k).$$

- Steepest descent

$$D^k = I$$

- Newton's method

$$D^k = [\nabla^2 f(h)]^{-1}$$

Gradient descent method

Basic iteration

$$d_k = -\nabla f(h_k)$$
$$h_{k+1} = h_k - \eta \nabla f(h_k).$$

By Taylor's theorem, we have

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

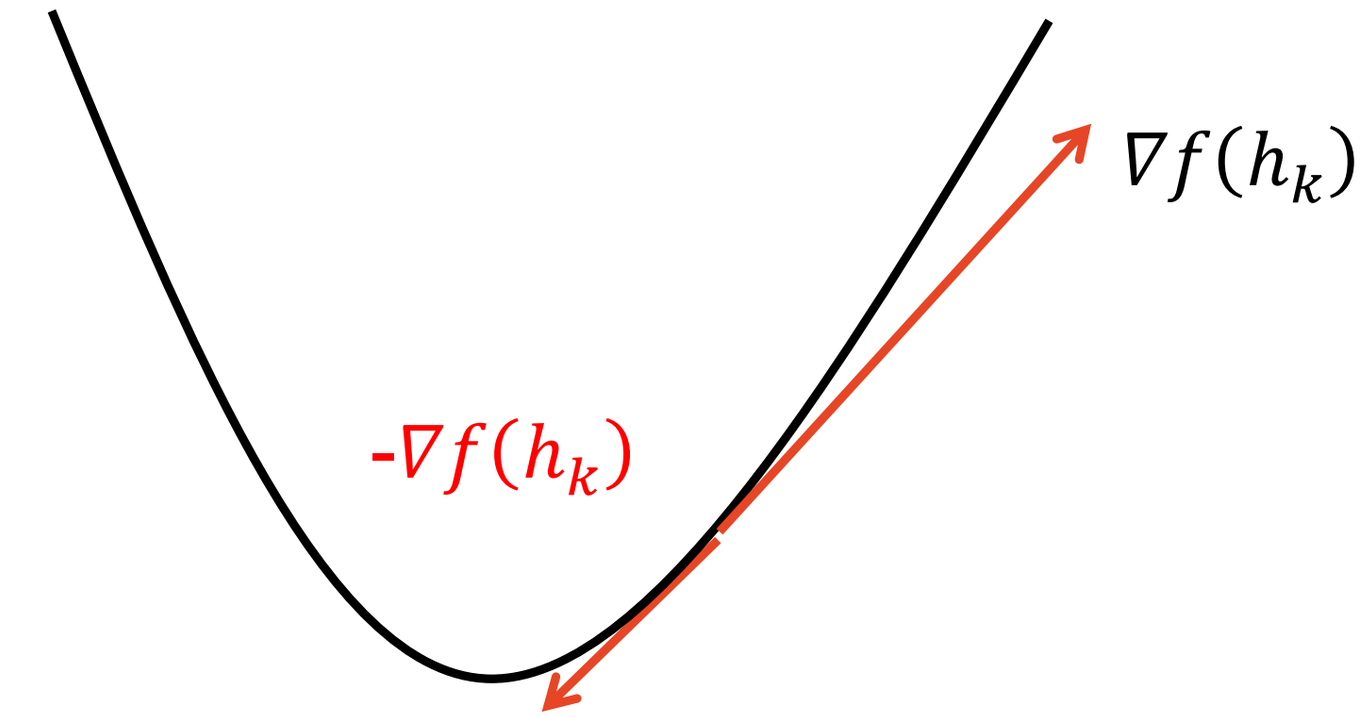
For positive but sufficiently small η,

$f(h_{k+1})$ is smaller than $f(h_k)$,
if the direction d_k is chosen so that

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

Gradient descent method

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



To find η

- Exact line search:

$$\eta = \arg\min_{\eta} f(h_k - \eta \nabla f(h_k))$$

practically expensive.

- Lipschitz smooth constant L exists for the gradient:

$$h_{k+1} = h_k - \frac{1}{L}\nabla f(h_k)$$

$$f(h_{k+1}) \leq f(h_k) - \frac{1}{2L}\|\nabla f(h_k)\|^2.$$

if L is known.

Function f is L-Lipschitz continuous if

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|, \forall x_1, x_2 \in \text{domain } f.$$

Gradient convergence rate

How many iteration steps do we need to achieve the optimal solution ?

$$h_S = \arg\min_{h \in H} f(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

Gradient convergence rate

If the objective function is strongly-convex, and has Lipschitz Gradient, we have a **linear convergence rate**, i.e., defined by

$$f(h_{k+1}) - f(h_S) \leq \left(1 - \frac{\mu}{L}\right)^k (f(h_1) - f(h_S)).$$

Gradient descent method

Algorithm	Assumption	Convergence rate
Gradient	Lipshitz Gradient, Convex	$O(1/k)$
Gradient	Lipshitz Gradient, Strongly-Convex	$O(1-\mu/L)^k$
Newton	Lipshitz Gradient, Strongly-convex	$\prod_{i=1}^{k}\rho_k$, $\rho_k \to 0$

$$f(h_{k+1}) - f(h_S) \leq \left(1 - \frac{\mu}{L}\right)^k (f(h_1) - f(h_S)).$$

Gradient descent method

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

Set $d_k = -D^k \nabla f(h_k)$.

Newton's method sets

$$D^k = [\nabla^2 f(h_k)]^{-1}.$$

Obtaining D^k my be difficult. There are many practical variants of Newton's method:

- Modify the Hessian to be positive-definite
- Only compute the Hessian every m iterations
- Only use the diagonals of the Hessian
- Quasi-Newton: Update a approximate of the Hessian (BFGS, L-BFGS)

THE UNIVERSITY OF SYDNEY

How to deal with constrained optimisation problem?

Constrained optimisation

- Constrained convex optimisation problem

$$\min_h f_0(h)$$
$$\text{s.t. } f_i(h) \leq 0, i = 1, \ldots, k$$
$$g_i(h) = 0, i = 1, \ldots, l,$$

where $f_0(h), f_1(h), \ldots, f_k(h)$ are convex functions, $g_i(h)$ are affine functions, i.e., $g_i(h) = a_i^\top h - b_i$.

THE UNIVERSITY OF SYDNEY

Thank you!

THE UNIVERSITY OF SYDNEY

Appendix Proofs

(not examinable)

To find η

- Exact line search:

$$\eta = \arg\min_{\eta} f(h_k - \eta \nabla f(h_k))$$

practically expensive.

- Lipschitz smooth constant L exists for the gradient:

$$h_{k+1} = h_k - \frac{1}{L}\nabla f(h_k)$$

$$f(h_{k+1}) \leq f(h_k) - \frac{1}{2L}\|\nabla f(h_k)\|^2.$$

if L is known.

Proof I

Function f is L-Lipschitz continuous if

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|, \forall x_1, x_2 \in \text{domain } f.$$

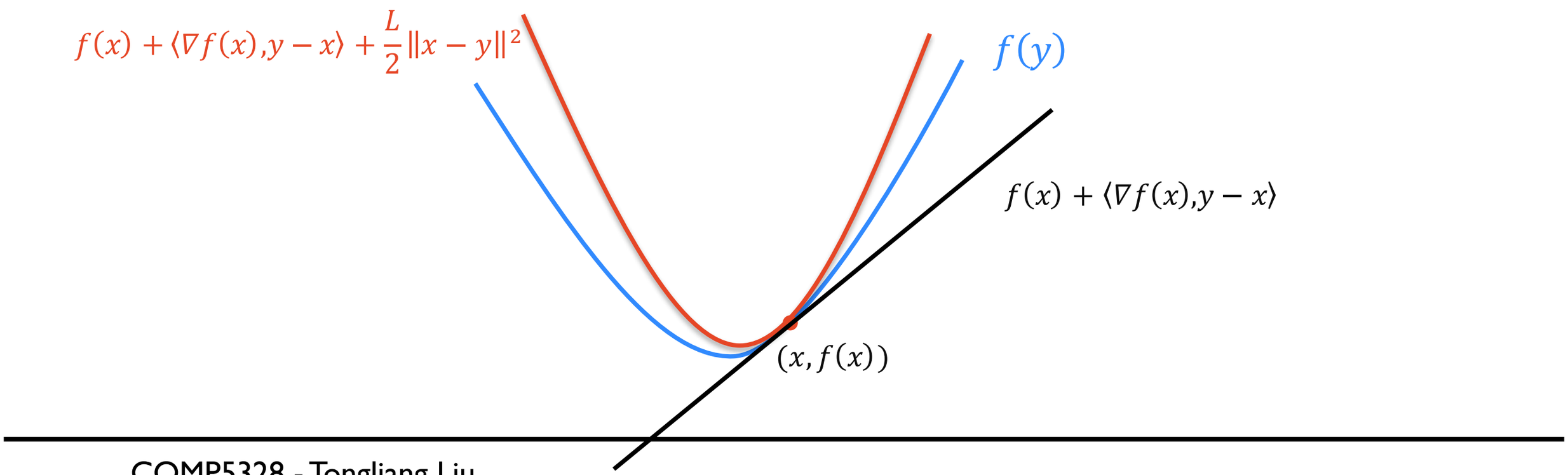
Proof I

Gradient is Lipschitz continuous:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \qquad \forall x, y,$$

$\Leftrightarrow$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|^2, \forall x, y,$$



Proof I

Gradient is Lipschitz continuous:

At x_k, we have

$$f(y) \leq f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{L}{2} \|y - x_k\|^2,$$

Set x_{k+1} to minimise the upper bound in terms of y,

$$x_{k+1} = x_k - \frac{1}{L} \nabla f(x_k)$$

(gradient descent with the step-size of $\frac{1}{L}$)

Plugging into the above inequality:

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L} \|\nabla f(x_k)\|^2$$

(decreasing at least $\frac{1}{2L} \|\nabla f(x_k)\|^2$)

Gradient convergence rate

If the objective function is strongly-convex, and has Lipschitz Gradient, we have a **linear convergence rate**:

$$f(h_{k+1}) - f(h_S) \leq \left(1 - \frac{\mu}{L}\right)^k (f(h_1) - f(h_S)).$$

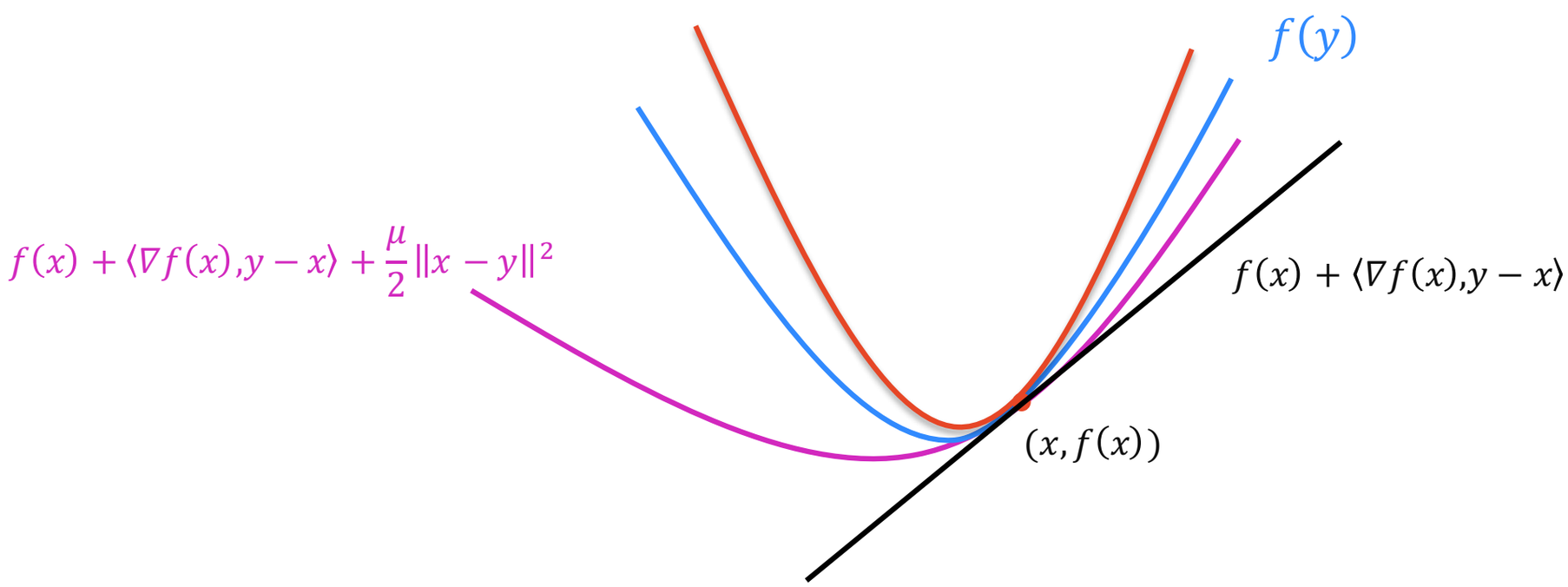
Proof II

A function is μ-strongly convex:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|^2, \forall x, y,$$

$$\Leftrightarrow$$

$$\mu \boldsymbol{I} \preccurlyeq \nabla^2 f(x), \forall x$$



Proof II

A function is μ-strongly convex:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|x - y\|^2, \forall x, y.$$

Let x^* be the minimizer of $f(x)$. We have $\nabla f(x^*) = 0$, then

$$f(\mathrm{y}) \geq f(x^*) + \frac{\mu}{2}\|x^* - y\|^2.$$

Set x_k be the minimizer of $f(\mathrm{y}) - \frac{\mu}{2}\|x^* - y\|^2$, we have

$$x_k = x^* + \frac{\nabla f(x_k)}{\mu}.$$

We have

$$f(x^*) \leq f(x_k) - \frac{1}{2\mu}\|\nabla f(x_k)\|^2.$$

Proof II

Proof:

Because of

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2$$

$$f(x^*) \leq f(x_k) - \frac{1}{2\mu}\|\nabla f(x_k)\|^2.$$

After some rearrangement, we have

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)\left(f(x_k) - f(x^*)\right).$$

This gives a linear convergence rate:

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)^k \left(f(x_1) - f(x^*)\right).$$

Proof II

Proof:

Because of $f(x_{k+1}) - f(x^*) \leq \frac{1}{2\mu}\|\nabla f(x_k)\|^2 - \frac{1}{2L}\|\nabla f(x_k)\|^2 = \left(\frac{1}{2\mu} - \frac{1}{2L}\right)\|\nabla f(x_k)\|^2$

$$f(x_{k+1}) \leq f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|^2$$

$$f(x^*) \leq f(x_k) - \frac{1}{2\mu}\|\nabla f(x_k)\|^2.$$

$$\frac{1}{2\mu}\|\nabla f(x_k)\|^2 \leq f(x_k) - f(x^*)$$

After some rearrangement, we have Or $\|\nabla f(x_k)\|^2 \leq 2\mu\big(f(x_k) - f(x^*)\big)$

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)\big(f(x_k) - f(x^*)\big).$$

This gives a linear convergence rate:

$$f(x_{k+1}) - f(x^*) \leq \left(1 - \frac{\mu}{L}\right)^k \big(f(x_1) - f(x^*)\big).$$