5.2 Gradient descent

We now turn to numerical optimization methods for the computation of

$$\mathbf{w}_s \in \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} F_s(\mathbf{w}), \qquad F_s(\mathbf{w}) := \lambda R(\mathbf{w}) + \mathcal{R}_s(\mathbf{w})$$

where

- $\mathcal{W} \subseteq \mathbb{R}^p$ is the parameter set to the hypothesis class $\mathcal{H}$,
- $\lambda \geq 0$ and $R\colon \mathcal{W} \to [0,\infty)$ regularization parameter or functional and
- $\mathcal{R}_s\colon \mathcal{W} \to [0,\infty)$ be the empirical risk w. r. t a loss function $\ell\colon \mathcal{W}\times\mathcal{X}\times\mathcal{Y}$ and a given sample $s \in (\mathcal{X}\times\mathcal{Y})^m$.

Convention

For the sake of clarity we simply consider the task

$$\mathbf{w}^* \in \operatorname*{argmin}_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w}), \qquad F\colon \mathcal{W} \to [0,\infty), \quad \mathcal{W} \subseteq \mathbb{R}^p,$$

without including the sample s or the specific learning rule in the notation.

Iterative methods

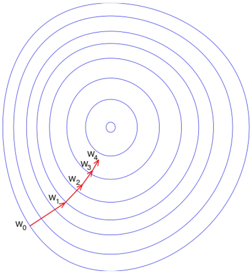
- Here we consider iterative optimization methods which compute a sequence of $\mathbf{w}_k \in \mathbb{R}^p$, $k \in \mathbb{N}$, such that under suitable assumptions

$$\|\mathbf{w}_k - \mathbf{w}^*\| \xrightarrow{k\to\infty} 0 \quad \text{or} \quad F(\mathbf{w}_k) \xrightarrow{k\to\infty} F(\mathbf{w}^*).$$

- The iterates $\mathbf{w}_k$ are calculated recursively

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \eta_k \mathbf{v}_k, \qquad k \geq 0,$$

  where
  - $\mathbf{v}_k \in \mathbb{R}^p$ is a suitable search resp. descent direction
  - and $\eta_k > 0$ is a corresponding step size.



- We also consider the application in machine learning and discuss bounds on the optimization error at the end.

The gradient descent method

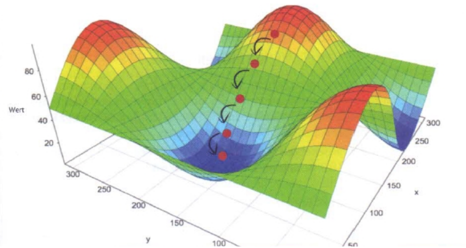
- The idea of gradient descent is to go in the steepest descent direction $\mathbf{v}_k = -\nabla F(\mathbf{w}_k)$:

Gradient descent method

Given a starting vector $\mathbf{w}_0 \in \mathbb{R}^p$, calculate for $k = 0, 1, 2, \ldots$

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \nabla F(\mathbf{w}_k)$$

with corresponding step sizes $\eta_k > 0$.



Source: biteye.at

- The step size η_k can be calculated adaptively to guarantee a maximum decay of the objective function.
- We will focus on a-priori choices of step size such as

$$\eta_k \equiv \eta_0 \qquad \text{oder} \qquad \eta_k = \eta_0 k^{-r}, \quad r > 0.$$

Differentiable log loss

- To apply the gradient method to (regularized) ERM rules such as

$$F(\mathbf{w}) = \lambda \|\mathbf{w}\|^2 + \mathcal{R}_S(\mathbf{w}),$$

we require a differentiable loss function $\quad \mathbf{w} \mapsto \ell(\mathbf{w}, \mathbf{x}, y)$.

Being differentiable means being smooth as well

- For instance, the log loss function:

$$\ell_{\text{log}}(\mathbf{w}, \mathbf{x}, y) = \ln\left(1 + \mathrm{e}^{-y(\mathbf{w}\cdot\mathbf{x})}\right), \qquad \nabla_{\mathbf{w}} \ell_{\text{log}}(\mathbf{w}, \mathbf{x}, y) = -\frac{y\mathrm{e}^{-y(\mathbf{w}\cdot\mathbf{x})}}{1 + \mathrm{e}^{-y(\mathbf{w}\cdot\mathbf{x})}}\mathbf{x}$$

- For search direction $\mathbf{v}_k = -\nabla F(\mathbf{w}_k)$ for $F(\mathbf{w}) = \lambda \|\mathbf{w}\|^2 + \mathcal{R}_S^{\text{log}}(\mathbf{w})$ we have

$$\mathbf{v}_k = -\nabla F(\mathbf{w}_k) = -2\lambda \mathbf{w}_k + \frac{1}{m} \sum_{i=1}^{m} y_i \frac{\mathrm{e}^{-y_i(\mathbf{w}\cdot\mathbf{x}_i)}}{1 + \mathrm{e}^{-y_i(\mathbf{w}\cdot\mathbf{x}_i)}} \mathbf{x}_i$$

Note: To calculate $\mathbf{v}_k$ all training data is needed!

Symbol / Notation	Meaning
$F : \mathbb{R}^p \to \mathbb{R}$	A real-valued function with input from $\mathbb{R}^p$ (e.g., loss function with p parameters)
$\nabla F(\mathbf{w})$	The gradient (vector of partial derivatives) of function F at point $\mathbf{w}$
$\|\cdot\|$	Norm (usually Euclidean) — measures distance or length of a vector
$L > 0$	A constant that controls how "fast" the gradient can change — called the **Lipschitz constant**
$\lambda > 0$	A constant that shows how "strongly curved upward" the function is — used in **strong convexity**
$\mathbf{w}, \mathbf{v} \in \mathcal{W}$	Two points (vectors) in the domain of the function
$\nabla^2 F(\mathbf{w})$	The **Hessian** — a matrix of second derivatives of F at point $\mathbf{w}$
$\lambda_{\min}(\nabla^2 F(\mathbf{w}))$	The smallest eigenvalue of the Hessian — shows curvature at its weakest point

✅ 1. L-smooth (Lipschitz smooth)

Mathematical condition:

$$\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{v})\| \leq L\|\mathbf{w} - \mathbf{v}\|$$

Plain English:

> The **slope (gradient)** of the function doesn't change too quickly.
> If you move a little from $\mathbf{w}$ to $\mathbf{v}$, the change in gradient is not too big.
> This makes the function **smooth and well-behaved** — no sudden cliffs or spikes.

✅ This helps Gradient Descent know how big or small its step should be without jumping too far.

$$F(\mathbf{v}) \geq F(\mathbf{w}) + \nabla F(\mathbf{w})^\top(\mathbf{v} - \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{v} - \mathbf{w}\|^2$$

🔍 Plain English Translation

- The right side is a **quadratic function** that opens upward.
- This inequality says:

  > The function $F(\mathbf{v})$ is always **above this curved lower bound** (the red curve in your diagram).

- The **larger λ is**, the **steeper that bowl** is.

So strong convexity means:

- The function **doesn't get too flat**
- It **pulls you harder toward the minimum**
- You **don't waste time** in flat or slow areas

Strongly convex and Lipschitz-smooth

- The log risk has further desirable properties for numerical optimization.

Definition 5.2:

A differentiable function $F\colon \mathbb{R}^p \to \mathbb{R}$

- is called *L-smooth* if for $L > 0$ we have

we have gradients

$$\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{v})\| \;\leq\; L\|\mathbf{w} - \mathbf{v}\| \qquad \forall \mathbf{v}, \mathbf{w} \in \mathcal{W}$$

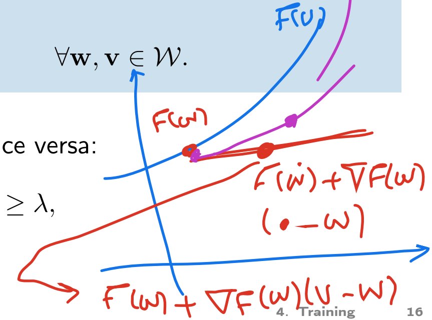
- is called *λ-strongly convex* if for $\lambda > 0$ we have

$$F(\mathbf{v}) \geq F(\mathbf{w}) + \nabla F(\mathbf{w})^\top(\mathbf{v} - \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{v} - \mathbf{w}\|^2 \qquad \forall \mathbf{w}, \mathbf{v} \in \mathcal{W}.$$

- Both properties bound the Hessian $\nabla^2 F \in \mathbb{R}^{p \times p}$ (if existing) and vice versa:

$$\|\nabla^2 F(\mathbf{w})\| \leq L, \qquad \lambda_{\min}(\nabla^2 F(\mathbf{w})) \geq \lambda,$$

with $\lambda_{\min}(\nabla^2 F(\mathbf{w}))$ denoting the smallest eigenvalue of $\nabla^2 F(\mathbf{w})$.



Theorem 5.3:

Sum of [illegible] norm of training data^2

1. The empirical log risk $\mathcal{R}_s^{\log}$ is L-smooth, i.e.,

$$\|\nabla \mathcal{R}_S^{\log}(\mathbf{w}) - \nabla \mathcal{R}_S^{\log}(\mathbf{v})\| \leq L\, \|\mathbf{w} - \mathbf{v}\|, \qquad L \leq \frac{1}{4m} \sum_{i=1}^{m} \|\mathbf{x}_i\|^2$$

2. The regularized empirical log risk $F(\mathbf{w}) = \lambda\|\mathbf{w}\|^2 + \mathcal{R}_S(\mathbf{w})$ with $\lambda > 0$ is 2λ-strongly convex.

3. If the sample size m is sufficiently large such that there are p linearly independent data vectors $\mathbf{x}_i \in \mathbb{R}^p$, then the empirical log risk $\mathcal{R}_s^{\log}$ on restricted areas $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^p : \|\mathbf{w}\| \leq r\}$, $r > 0$, is already λ_r-strongly convex, where λ_r depends on r and the feature vectors $\mathbf{x}_i$.

What are the benefits of the strong convexity and L-smoothness for numerical optimization?

Proof of Th 5.3:

we have $\nabla^2 R_S(w) = \frac{1}{m}\sum_{i=1}^{m} G(y_i(wx_i+b)) + \underbrace{x_i x_i^T}$ (transposes)

with $G(t) = \ln(1+e^{-t})$.

Hence $G''(t) = \frac{e^{-t}}{(1+e^{-t})^2} \le \frac{1}{4}$ & $\|\nabla^2 R_S(w)\| \le \frac{1}{4m}\sum_{i=1}^{m} \underbrace{\|x_i x_i^T\|^2}_{\le \|x_i\|^2}$

$\Rightarrow$ $L \le \frac{1}{4m}\sum_{i=1}^{m} \|x_i\|^2$ (the empirical log risk)

moreover, for $F(w) = \lambda\|w\|^2 + R_S(w)$, we have:

$$\nabla^2 F(w) = 2\lambda I_{p\times p} + \underbrace{\nabla^2 R_S(w)}_{\text{it is positiv semi definite}} \longrightarrow \bar{a}^T \nabla^2 R_S(w)\, a$$

$$\le \underbrace{\sum a^T (x_i x_i^T)\, a}$$

$\therefore \lambda_i$ the values of $\nabla^2 F(m)$, are all died below by 2λ.

$\to A_\epsilon = \epsilon I + A \to \underbrace{A_\epsilon v}_{(\epsilon I + A)v} - \lambda_\epsilon v = \epsilon v + \lambda v = v(\epsilon + \lambda)$

The benefit of L-smoothness

If the differentiable funtion $F\colon \mathbb{R}^p \to \mathbb{R}$ is L-smooth, then

Helps you pick a proper step size η in gradient descent. If the function is L-smooth, you know how far you can move safely

$$F(\mathbf{v}) \le F(\mathbf{w}) + \nabla F(\mathbf{w})^\top (\mathbf{v} - \mathbf{w}) + \frac{L}{2}\|\mathbf{v} - \mathbf{w}\|^2, \qquad \mathbf{v}, \mathbf{w} \in \mathbb{R}^p.$$

Also smoothness keeps the training numerically stable.

This guarantees a decrease in the objective function value for gradient descent:

Proposition 5.4:

Let $F\colon \mathbb{R}^p \to \mathbb{R}$ be differentiable and L-smooth. Then, for the iterates of the gradient descent method

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta_k \nabla F(\mathbf{w}_k)$$

for sufficiently small step sizes $\eta_k \le \frac{1}{L}$ we have

$$F(\mathbf{w}_{k+1}) \le F(\mathbf{w}_k) - \frac{\eta_k}{2}\|\nabla F(\mathbf{w}_k)\|^2, \qquad k \in \mathbb{N}_0.$$

Proof of Th 5,4: $\overbrace{\text{Teta}}$

setting $v = w_{k+1} = w_k - \eta_k \nabla F(w_k)$ & using inequality from Top of the slide we get $F(w_{k+1}) \leq F(w_k) + \underbrace{\nabla F(w_k)^T(\eta_k \nabla F(w_k))}_{= \eta_k \|\nabla F(w_k)\|^2}$

$+ L/2 \|-\eta_k \nabla F(w_k)\|^2 = \eta_k^2 \|\nabla F(w_k)\|^2$

$= F(w_k) - \underbrace{\left(\eta_k - \frac{L}{2}\eta_k^2\right)}_{\geq \eta_k - L/2 \, \frac{1}{L} \eta_k = \frac{1}{2}\eta_k} \|\nabla F(w_k)\|^2$. for $\eta_k \leq \frac{1}{L}$. which

yields statement.

$F(w_{k+1}) \leq F(w_k) - \frac{1}{2}\eta_k \|\nabla F(w_k)\|^2$ #

The benefit of strong convexity

We collect some useful properties of strongly convex functions $f, g\colon \mathbb{R}^p \to \mathbb{R}$.

- If f is λ-strongly convex and g convex, then $f + g$ is λ-strongly convex.
- If f is λ-strongly convex, then αf is $\alpha\lambda$-strongly convex for $\alpha > 0$.
- The function $f(\mathbf{w}) = \lambda\|\mathbf{w}\|^2$, $\lambda > 0$, is 2λ-strongly convex.
- Strongly convex functions possess at most one minimum.
- If $\mathbf{w}^*$ is a minimizer of a λ-strongly convex, differentiable f then

$$\frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}^*\|^2 \leq f(\mathbf{w}) - f(\mathbf{w}^*) \qquad \forall \mathbf{w} \in \mathbb{R}^p$$

and, moreover, the Polyak–Łojasiewicz condition holds

$$f(\mathbf{w}) - f(\mathbf{w}^*) \leq \frac{1}{2\lambda}\|\nabla f(\mathbf{w})\|^2 \qquad \forall \mathbf{w} \in \mathbb{R}^p.$$

there's only one best solution.
That’s very important in
learning and optimization.

GD, reaches the minimum
faster.
Strong convexity gives you
linear convergence

Strong convexity makes
learning more robust —
less sensitive to small
disturbances or noise.

We are now able to proof the linear convergence of gradient descent:

Theorem 5.5:

If $F\colon \mathbb{R}^p \to \mathbb{R}$ is λ-strongly convex and L-smooth, then for the iterates of the gradient descent method

$$\mathbf{w}_{k+1} = \mathbf{w}_k \underbrace{- \eta \nabla F(\mathbf{w}_k),}_{\text{constant step size / learning rate}} \qquad \mathbf{w}_0 \in \mathbb{R}^p,$$

with $\eta \leq \min(L^{-1}, \lambda^{-1})$ we have

$$*** \quad \underline{|F(\mathbf{w}_k) - F(\mathbf{w}^*)| \leq (1 - \eta\lambda)^k (F(\mathbf{w}_0) - F(\mathbf{w}^*)),}$$

global

and

$$\|\mathbf{w}_k - \mathbf{w}^*\|^2 \leq (1 - \eta\lambda)^k \, \|\mathbf{w}_0 - \mathbf{w}^*\|^2.$$

If F is only convex and L-smooth then for $\eta \leq \frac{1}{L}$ Ieta ($\overline{w}$))

$$F(\mathbf{w}_k) - F(\mathbf{w}^*) \leq \frac{2L\|\mathbf{w}_0 - \mathbf{w}^*\|^2}{k+4}$$

Proof of Th 5.5:

By prop 5.4, we have $\eta_k = \eta \leq \frac{1}{L} \frac{1}{\lambda}$, we have: $F(w_{k+1}) \leq F(w_k) - \frac{\eta}{2} \|\nabla F(w_k)\|^2$

using the PLC condition, we get the objective function:

$F(w_{k+1}) \leq F(w_k) - \frac{\eta}{2} 2\lambda (F(w_k) - F(w^*))$. Denoting $F^* = F(w^*)$

$= \min_{w \in \mathbb{R}^d} F(w)$ & subtracting F^* on both sides on inequality above

yields $F(w_{k+1}) - F^* \leq F(w_k) - F^* - \frac{\eta}{2} \cdot 2\lambda (F(w_k) - F^*)$

$= (1 - \lambda\eta)(F(w_k) - F^*) \xrightarrow{k=1,\dots,k+1} (1-\lambda\eta)^{2}(F(w_k) - F^*) \Rightarrow$

$F(w_{k+1}) - F^* \leq (1 - \eta\lambda)^{k+1} (F(w_0) - F^*)$. moreover by λ-strong

complexity we have: $\|w_k - w^*\|^2 \leq 2\lambda (F(w_k) - F^*)$

$\leq 2\lambda(1-\eta\lambda)^k(F(w_0)-F^*)$. The statement for convex & L-smooth F is proven in lit.inter of section 5,4 #

Property	Meaning	Key Benefits
L-smooth	Gradient doesn't change too quickly	Helps choose learning rate, ensures stability, proves convergence
λ-strongly convex	Function curves upward at least quadratically	Guarantees unique solution, fast & stable convergence

Example

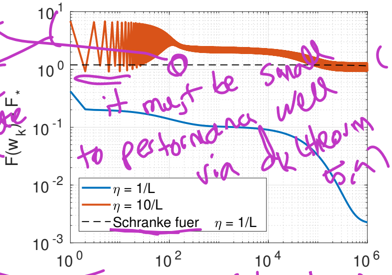
we say in order to apply logistic Regression Gradient decents, convergence is vital

- We apply gradient descent to minimize $\mathcal{R}_S^{\log}$ for the heart dataset from the exercise.
- The Lipschitz constant L of the gradient of $\mathcal{R}_S^{\log}$ is estimated as described and here we have $\frac{1}{\lambda} \geq \left(\frac{1}{m}\sum_{i=1}^m \|\mathbf{x}_i\|^2\right)^{-1} \geq \frac{1}{L}$.
- We also add a regularization $\frac{1}{m}\|\mathbf{w}\|^2$ which yields $\lambda \geq \frac{2}{m}$.

if we apply L2 to logistics & more it make it convex

second formula of Th 5.5 *

$F(\mathbf{w}) = \mathcal{R}_S^{\log}(\mathbf{w})$ → log loss function



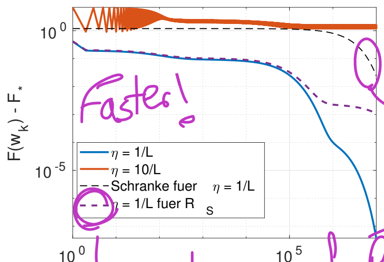
k (iteration)

due to bad choice of learning rate bound

it must be small well to performance via dk theorem 5.5

(highest convergence point)

$F(\mathbf{w}) = \frac{1}{m}\|\mathbf{w}\|^2 + \mathcal{R}_S^{\log}(\mathbf{w})$ (Gradient decent log loss)



Faster!

η is the same reason for going down is

↳ performance of $R_S(h)$

The optimization error

- The optimization error describes the error w.r.t. the expected risk $\mathcal{R}_\mu$ caused by using iterates $\mathbf{w}_k \approx \mathbf{w}_s$ of numerical optimization procedures instead of the actual ERM hypothesis or ERM parameter $\mathbf{w}_s$

$$\varepsilon_{\text{opt}}(k) = \varepsilon_{\text{opt}}(k, s, \mathcal{W}) = \mathcal{R}_\mu(\mathbf{w}_k) - \mathcal{R}_\mu(\mathbf{w}_s).$$

- Of course $\varepsilon_{\text{opt}}(k) < 0$ can occur, but this is generally not to be expected.

- Our convergence analysis concerning numerical optimization, especially bounds on the difference

$$\underbrace{\mathcal{R}_s(\mathbf{w}_k)}_{\text{iterate}} - \underbrace{\mathcal{R}_s(\mathbf{w}_s)}_{\text{optimizer}}$$

empirical risk

now help to control the optimization error ε_{opt}.

- For this purpose, we consider again the mean optimization error

$$\mathbb{E}_{\mu^m}[\mathcal{R}_\mu(\mathbf{w}_k) - \mathcal{R}_\mu(\mathbf{w}_S)]$$

and obtain by triangle inequality

VC controls it

Control optimization error

$$\underbrace{\mathbb{E}_{\mu^m}[\mathcal{R}_\mu(\mathbf{w}_k) - \mathcal{R}_\mu(\mathbf{w}_S)]}_{\text{mean } E_{opt}} \le \underbrace{\mathbb{E}_{\mu^m}[\mathcal{R}_S(\mathbf{w}_k) - \mathcal{R}_S(\mathbf{w}_S)]}_{\text{convergence theory of gradient descent}} + 2\mathbb{E}_{\mu^m}\left[\sup_{\mathbf{w}\in\mathcal{W}} |\mathcal{R}_\mu(\mathbf{w}) - \mathcal{R}_S(\mathbf{w})|\right]$$

- The term $\sup_{\mathbf{w}\in\mathcal{W}} |\mathcal{R}_\mu(\mathbf{w}) - \mathcal{R}_S(\mathbf{w})|$ reminds us of the uniform convergence condition.

- Indeed, similar to uniform convergence theorem one can show that for a $c \in (0,\infty)$

$$\mathbb{E}_{\mu^m}\left[\sup_{\mathbf{w}\in\mathcal{W}} |\mathcal{R}_\mu(\mathbf{w}) - \mathcal{R}_S(\mathbf{w})|\right] \le c\sqrt{\frac{\mathrm{VCD}(\mathcal{H}_\mathcal{W})\ \log(2m)}{2m}}$$

see Section 6.5.2 in "Understanding Machine Learning" (2014), where $\mathcal{H}_\mathcal{W} \subseteq \mathcal{Y}^\mathcal{X}$ is the hypothesis class parameterized by $\mathcal{W} \subseteq \mathbb{R}^p$.

- If $d = \mathrm{VCD}(\mathcal{H}_{\mathcal{W}}) < \infty$ then we obtain for the mean optimization error

$$\mathbb{E}_{\mu^m}[\mathcal{R}_\mu(\mathbf{w}_k) - \mathcal{R}_\mu(\mathbf{w}_S)] \leq c_d \sqrt{\frac{\log(2m)}{2m}} + \mathbb{E}_{\mu^m}[\mathcal{R}_S(\mathbf{w}_k) - \mathcal{R}_S(\mathbf{w}_S)]$$

decays exponentially fast based on iteration

- We can now use the convergence result for the gradient method with constant step size and obtain for the (Tichonow regularized) log loss

$F(w_0) - F^*$

$$\mathbb{E}_{\mu^m}[\mathcal{R}_\mu(\mathbf{w}_k) - \mathcal{R}_\mu(\mathbf{w}_S)] \leq c_d \sqrt{\frac{\log(2m)}{2m}} + c_0\, r^k$$

dependent of data

exp fast decay independent of data

where $r \in (0, 1)$ and c_0 depends on $\mathbf{w}_0$

(Here we technically need to assume that $\mathbb{E}[\|\mathbf{w}_S\|] < \infty$)

- What can you tell from this estimate?