1 Convexity

Cauchy-Schwarz $\|\mathbf{u}^\top \mathbf{u}\| \leq |\mathbf{u}\||\mathbf{v}| \to \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \in [-1,1] \sim \cos\alpha$

Definition of convexity A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if (i) $\mathbf{dom}(f)$ is a convex set and (ii) $\forall \mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$ and $\lambda \in [0,1]$ we have

$$\underbrace{f(\lambda x + (1-\lambda)y)}_{(1)} \leq \underbrace{\lambda f(x) + (1-\lambda) f(y)}_{(2)}$$

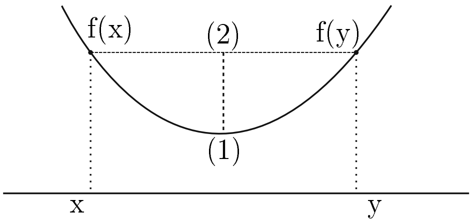


Figure 1: Any point on the line is above f

Epigraph While the *graph* of a function is the line drawn by its expression, the *epigraph* is the set of points that lie above the graph (the "content" of the graph).

Jensen's inequality The above definition is valid for any number of points in $\mathbf{dom}(f)$. Any "middle point" will be in-between them, and always above f. Formally: Let f be convex, and $x_1, \dots, x_m \in \mathbf{dom}(f)$, $\lambda_1, \dots, \lambda_m \in \mathbb{R}_+$ such that $\sum_{i=1}^m \lambda_i = 1$. Then

$$f\left(\sum_{i=1}^m \lambda_i \mathbf{x}_i\right) \leq \sum_{i=1}^m \lambda_i f(\mathbf{x}_i).$$

Convex is continuous Let f be convex, and suppose that $\mathbf{dom}(f)$ is open. Then f is continuous. is continuous.

Differentiable Graph of the affine function $f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y} - \mathbf{x})$ is a tangent hyperplane to the graph of f at $(\mathbf{x}, f(\mathbf{x})$

First-order characterization Suppose $\mathbf{dom}(f)$ is open and $f(\mathbf{x})$ is differentiable, in particular the gradient exists at every point $\mathbf{x} \in \mathbf{dom}(f)$. Then f is convex if and only if $\mathbf{dom}(f)$ is convex and

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$$

This means that f is above all its tangent hyperplanes.

Second-order Characterization Suppose that $\mathbf{dom}(f)$ is open, and that the Hessian (double derivatives) of f exists at every point $\mathbf{x} \in \mathbf{dom}(f)$ and is symmetric. Then f is convex if and only if $\mathbf{dom}(f)$ is convex, and for all $\mathbf{x} \in \mathbf{dom}(f)$ we have

$$\nabla^2 f(\mathbf{x}) \succeq 0^1$$

Operations Multiplication by real constant and addition between convex is convex (not everywhere, only on $\bigcap_{i=1}^m \mathbf{dom}(f_i)$). The composition works the following: Let f be a convex function with $\mathbf{dom}(f) \subseteq \mathbb{R}^d$, $g : \mathbb{R}^m \to \mathbb{R}^d$ an affine function, meaning that $g(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, for some matrix $A \in \mathbb{R}^{d \times m}$ and some vector $\mathbf{b} \in \mathbb{R}^d$. Then the function $f \circ g$ (that maps $\mathbf{x}$ to $f(A\mathbf{x} + \mathbf{b})$ is convex on $\mathbf{dom}(f \circ g) := \{\mathbf{x} \in \mathbb{R}^m : g(\mathbf{x}) \in \mathbf{dom}(f)\}$

Mimima $\mathbf{x}$ is a local minimum if $\exists \epsilon > 0$ with

$$f(\mathbf{x}) \leq f(\mathbf{y})$$

For all $\mathbf{y} \in \mathbf{dom}(f)$ satisfying $\|\mathbf{y} - \mathbf{x}\| < \epsilon$. If $\mathbf{x}^\star$ is the local minimum of a convex function, then it's a global minimum (meaning $f(\mathbf{x}^\star) \leq f(\mathbf{y})$ for all $\mathbf{y} \in \mathbf{dom}(f)$. Similarly, for f convex and differentiable over an open domain, then if $\nabla f(\mathbf{x}) = \mathbf{0}$ then it's a global minimum (it's called a *critical point*).

Strictly convex Same definition as the convexity, with a strict inequality. So the open

[1] positive semidefinite

segment connecting any two points of the graph will be *strictly* above the graph. This leads to the following lemma:

Lemma. *Suppose that* $\mathbf{dom}(f)$ *is open and that* f *is twice continuously differentiable. If the Hessian* $\nabla^2 f(\mathbf{x}) \succeq \mathbf{0}$ *for every* $\mathbf{x} \in \mathbf{dom}(f)$*, then* f *is strictly convex.*

Constrained Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex, and let $X \subseteq \mathbf{dom}(f)$ be a convex set. A point $\mathbf{x} \in X$ is a *minimizer of* f *over* X if $f(\mathbf{x}) \leq f(\mathbf{y})\ \forall \mathbf{y} \in X$. It follows that for $f : \mathbf{dom}(f) \to \mathbb{R}$ convex and differentiable oven an open domain $\mathbf{dom}(f) \subseteq \mathbb{R}^d$, and for $X \subseteq$ a convex set, then the point $\mathbf{x}^\star \in X$ is a minimizer of f over X iff

$$\nabla f(\mathbf{x}^\star)^\top(\mathbf{x} - \mathbf{x}^\star) \geq 0\ \forall \mathbf{x} \in X$$

α-sublevel $f^{\leq\alpha} := \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \leq \alpha$. This represents the values of the domain of f for which the value of f is below a threshold

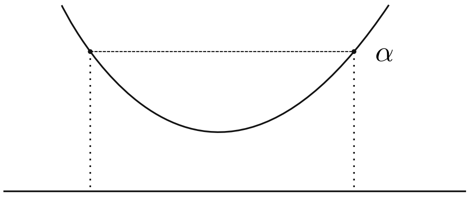


Figure 2: Only what is between vertical bars is in the sublevel

Weierstrass theorem Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function, and suppose there is a nonempty and bounded sublevel set $f^{\leq\alpha}$. Then f has a global minimum. Some function (such as e^x) don't have a minimum (in the exponential case, because the subvlevel is not bounded.

2 Gradient descent

Gradient descent Goal: get near to a minimum $x^\star$ (not necessarily unique), meaning close to the optimal value $f(x^\star)$. For that, we look for $\mathbf{x} \in \mathbb{R}^d$ such that

$$f(\mathbf{x}) - f(\mathbf{x}^\star) \leq \epsilon$$

Iterative algorithm: For that purpose, we start from $\mathbf{x}_0 \in \mathbb{R}^d$, and then iterate:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t)$$

For time steps t=0,1,. . . , and step size $\gamma \geq 0$

Bound to error (vanilla) It's useful to bound the error $(f(\mathbf{x}_t) - f(\mathbf{x}^\star))$. Using the notation $\mathbf{g}_t := \nabla f(\mathbf{x}_t)$ (for gradient descent, $\mathbf{g}_t = \frac{\mathbf{x}_t - \mathbf{x}_{t+1}}{\gamma}$), and applying the following steps:

1. Apply $2\mathbf{v}^\top\mathbf{w} = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 - \|\mathbf{v} - \mathbf{w}\|^2$
2. Sum over for the first T iterations
3. Use first-order characterization with $\mathbf{x} = \mathbf{x}_t, \mathbf{y} = \mathbf{x}^\star$.

We obtain the upper bound for the average error:

$$\sum_{t=0}^{T-1} (f(\mathbf{x}_t) - f(\mathbf{x}^\star)) \leq \frac{\gamma}{2} \sum_{t=0}^{T-1} \|\mathbf{g}_t\|^2 + \frac{1}{2\gamma}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2$$

Lipschitz

Theorem. *A function* $f : \mathbf{dom}(f) \to \mathbb{R}^m$ *is B-Lipschitz if*

$$\|f(\mathbf{x}) - f(\mathbf{y})\| \leq B\|\mathbf{x} - \mathbf{y}\|$$

for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$

Assume all gradients of f bounded in norm (not always cool, e.g. discards x^2). The following theorem holds:

Theorem. *Let* $f : \mathbb{R}^d \to \mathbb{R}$ *be convex and differentiable, and suppose* $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$*, and* $\|\nabla f(\mathbf{x})\| \leq B$ *for all* $\mathbf{x}$*. Choosing the step size*

$$\gamma := \frac{R}{B\sqrt{T}}$$

then gradient descent yields

$$\frac{1}{T}\sum_{t=0}^{T-1} f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \frac{RB}{\sqrt{T}}$$

This relates T and ϵ !

$$T \geq \frac{R^2B^2}{\epsilon^2} \Rightarrow \text{average error } \leq \frac{RB}{\sqrt{T}} \leq \epsilon$$

Note: same for subgradient descent.

Smoothness A function is smooth if it is "Not too curved". That is, if given a point $\mathbf{x} \in X$, all other points smaller than the linearisation + a quadratic term. Formally, let $f : \mathbf{dom}(f) \to \mathbb{R}$ be differentiable, $X \subset \mathbf{dom}(f)$, $L \in \mathbb{R}_+$. f is called smooth (with parameter L over X if

$$f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2$$

$\forall \mathbf{x}, \mathbf{y} \in X$.

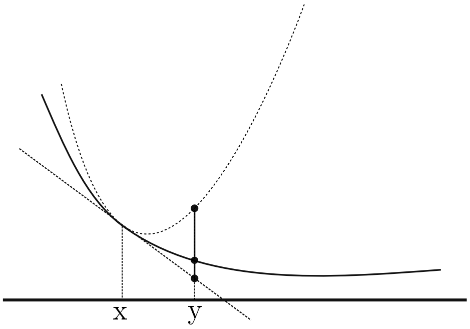


Figure 3: Every $\mathbf{y}$ is below the linearisation+quadratic

(unsure) The smoothness of least squares ($f(x) = \frac{1}{2n}\|A\mathbf{x}+\mathbf{b}\|$) is given by max $\frac{\|A^\top \cdot A\|}{n}$. That is the maximum of the eigenvalues of $A^\top \cdot A$ divided by the number of rows of A.

Operations on Smoothness Addition of smooth function is still smooth: $f_1, f_2, \dots, f_m$ smooth with parameters $L_1, L_2, \dots, L_m$ and let $\lambda_1, \lambda_2, \dots, \lambda_m \in \mathbb{R}_+$. Then $f := \sum_{i=1}^m \lambda_i L_i$ is smooth, with parameter $\sum_{i=1}^m \lambda_i L_i$.

Also, about composition: f L-smooth, and let $g(\mathbf{x}) = A\mathbf{x}+\mathbf{b}$ for $A \in \mathbb{R}^{d\times m}$, $\mathbf{b} \in \mathbb{R}^d$. Then $f \circ g$ is smooth with parameter $L\|A\|^2$ (where $\|A\|$ is the spectral norm of A).

Smooth VS Lipschitz

1. Bounded gradients $\iff$ Lipschitz continuity of f
2. Smoothness $\iff$ Lipschitz continuity of ∇f

For $f : \mathbb{R}^d \to \mathbb{R}$, convex and differentiable, the following are equivalent:

- f is smooth with parameter L
- $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$

Sufficient decrease $f : \mathbb{R}^d \to \mathbb{R}$ differentiable, L-smooth, step size $\gamma := \frac{1}{L}$. Then gradient descent satisfies

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2 \quad t \geq 0$$

This is a strong statement. This implies that the steps always get better (second term is always positive)

Smooth & convex f convex, differentiable, with global minimum, L-smooth, step size $\frac{1}{L}$. Gradient descent yields

$$f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \frac{L}{2T}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2,\ T > 0$$

This relates T and ϵ !

$$T > \frac{R^2L}{2\epsilon} \Rightarrow \text{error} \leq \frac{L}{2T} \leq \epsilon$$

Strongly convex Up until now, error decreased in T or $\sqrt{T}$. Good, but not ideal.

Strongly convex functions converge exponentially. For $f : \mathbf{dom}(f) \to \mathbb{R}$ differentiable function, $X \subset \mathbf{dom}(f)$ convex and $\mu \in \mathbb{R}_+,\ \mu > 0$. Then f is strongly convex (with parameter μ) over X if

$$\begin{aligned} f(\mathbf{y}) \geq\ & \overbrace{f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})}^{\text{Linearisation at } x} \\ & + \underbrace{\frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2}_{\text{Quadratic term}} \quad \forall \mathbf{x}, \mathbf{y} \in X \end{aligned}$$

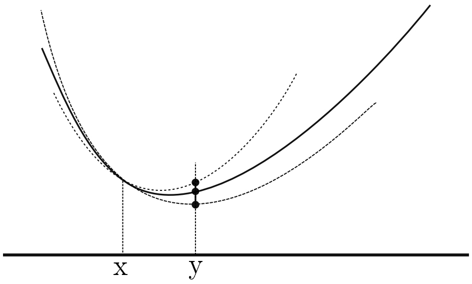


Figure 4: Function is between smooth (above) and strongly convex (below)

Smooth & Strongly convex We start from the vanilla analysis, and use stronger lower bound on left-hand side (coming from strong convexity). That is, we use that

$$\nabla f(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^\star) \geq f(\mathbf{x}_t) - f(\mathbf{x}^\star) + \frac{\mu}{2}\|\mathbf{x}_t - \mathbf{x}^\star\|^2$$

and some rewriting to obtain the final bound:

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq & 2\gamma\,(f(\mathbf{x}^\star) - f(\mathbf{x}_t)) \\ & + \gamma^2 \|\nabla f(\mathbf{x}_t)\| \\ & + (1 - \mu\gamma)\|\mathbf{x}_t - \mathbf{x}^\star\|^2 \end{aligned}$$

In other words: *Squared distance to $\mathbf{x}^\star$ goes down by a constant factor, up to some "noise".* This leads to the following theorem:

Theorem. *Let $f : \mathbb{R}^d \to \mathbb{R}$, differentiable, with a global minimum $x^\star$. Suppose it's smooth with parameter L and strongly convex with parameter $\mu > 0$. With $\gamma := \frac{1}{L}$, gradient descent with arbitrary $\mathbf{x}_0$ satisfies the following 2 properties:*

- *Squared distance to $\mathbf{x}^\star$ are geometrically decreasing:*

  $$\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2 \leq \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^\star\|^2, \ t \geq 0$$

- *The absolute error after T iterations is exponentially small in T :*

  $$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{L}{2}\left(1 - \frac{\mu}{L})\right)^T \|\mathbf{x}_0 - \mathbf{x}^\star\|^2$$

3 Projected Gradient Descent

Constrained Optimization We want to minimize $f(\mathbf{x})$, but being subject to $\mathbf{x} \in X$. 2 ways of solving this: Either use projected gradient descent, or transform it into an *unconstrained* problem.

Projected Gradient Descent Idea: project onto X after every step:

$$\Pi_X(\mathbf{y}) := \underset{\mathbf{x} \in X}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{y}\|$$

Idea: compute first what would be your next step ($\mathbf{y}_{t+1}$) and then project it on your actual next step.

$$\begin{aligned} \mathbf{y}_{t+1} &:= \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t), \\ \mathbf{x}_{t+1} &:= \Pi_X(\mathbf{y}_{t+1}) := \operatorname{argmin}_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{y}_{t+1}\|^2 \end{aligned}$$

Properties Let $X \in \mathbb{R}^d$ be closed and convex, $\mathbf{x} \in X, \mathbf{y} \in \mathbb{R}^d$. Then

- $(\mathbf{x} - \Pi_X(\mathbf{y}))^\top (\mathbf{y} - \Pi_X(\mathbf{y})) \leq 0$
- $\|\mathbf{x} - \Pi_X(\mathbf{y})\|^2 + \|\mathbf{y} - \Pi_X(\mathbf{y})\|^2 \leq \|\mathbf{x} - \mathbf{y}\|^2$

First property means that for any point $\mathbf{y}$, and any point $\mathbf{x}$ in the domain, then the angle formed at the projection of $\mathbf{y}$ (in X) is greater than 90° (negative inner product).

Second property is the triangle inequality: distance x-projection plus projection-$\mathbf{y}$ is bigger than distance $\mathbf{x}-\mathbf{y}$. Inequality is written in the other round because of the squared values. If you remove the square, then inequality reverts.

Last cool property: The expected number of steps in various scenarios (Lipschitz,...), as listed in appendix Table 1, remains unchanged!

Equivalent proofs of projected The last property abode needs to be proved (not here). However, here are some insights: We now only need the function to be continuous over X, and not anymore $\mathbb{R}^d$. Then, in the proofs we replace $\mathbf{x}_{t+1}$ by $\mathbf{y}_{t+1}$, and massage it with the facts above.

Smooth and strongly convex over X

Theorem. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable. Let $D \subseteq \mathbb{R}^d$ be a normal nonempty closed and convex set and suppose that f is smooth over X with parameter L and strongly convex over X with parameter $\mu > 0$. Choosing $\gamma := \frac{1}{L}$,* ***projected*** *gradient descent with arbitrary $\mathbf{x}_0$ satisfies the following two properties:*

1. *Squared distance to $\mathbf{x}^\star$ are geometrically decreasing:*

$$\|\mathbf{x}_{t+1}-\mathbf{x}^\star\|^2 \leq \left(1-\frac{\mu}{L}\right)\|\mathbf{x}_t-\mathbf{x}^\star\|^2,\ t \geq 0$$

2. *The absolute error after T iterations is exponentially small in T:*

$$\begin{aligned}&f(\mathbf{x}_T)-f(\mathbf{x}^\star)\\ &\leq \|\nabla f(\mathbf{x}^\star)\|\left(1-\tfrac{\mu}{L}\right)^{T/2}\|\mathbf{x}_0-\mathbf{x}^\star\|\\ &+\tfrac{L}{2}\left(1-\tfrac{\mu}{L}\right)^{\top}\|\mathbf{x}_0-\mathbf{x}^\star\|\end{aligned}$$

Projection step Computing the projection is far from obvious. But for some relevant cases, it can be efficiently solved:

- Projecting onto an affine subspace (leads to system of linear equations)
- Projecting onto an Euclidean ball with center $\mathbf{c}$ (scale vector $\mathbf{y}-\mathbf{c}$). Compute $y = (\mathbf{y}-\mathbf{c})\cdot\frac{R}{\|\mathbf{x}\|}$.
- Projecting onto ℓ_1-balls (needed in Lasso). Though, we restrict to the ball being centered at $\mathbf{0}$, defined as

$$B_1(R) = \left\{\mathbf{x}\in\mathbb{R}^d : \|\mathbf{x}\|_1 = \sum_{i=1}^d |x_i| \leq R\right\}$$

B_1 is a *cross polytope* ($2d$ facets, 2^d vertices). The projection can be computed in $\mathcal{O}(d\log d)$, and even improved to $\mathcal{O}(d)$

4 Proximal and Subgradient Descent

Composite optimization problems

Consider $f(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x})$, where g is "nice" and h is a "simple" additional term (but not nice). Important case when h is not differentiable.

Proximal gradient Classical gradient step:

$$\mathbf{x}_{t+1} = \underset{\mathbf{y}}{\operatorname{argmin}}\, g(\mathbf{x}_t)^\top(\mathbf{y}-\mathbf{x}_t) + \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{x}_t\|^2$$

For $f = g + h$ keep the same for g and add h unmodified:

$$\begin{aligned}\mathbf{x}_{t+1} &= \overbrace{\underset{\mathbf{y}}{\operatorname{argmin}}\, g(\mathbf{x}_t)^\top(\mathbf{y}-\mathbf{x}_t) + \frac{1}{2\gamma}\|\mathbf{y}-\mathbf{x}_t\|^2 +}^{\text{Quadratic upperbound from smooth } g} \overbrace{h(\mathbf{y})}^{\text{Non smooth}}\\ &= \underset{\mathbf{y}}{\operatorname{argmin}}\, \tfrac{1}{2\gamma}\|\mathbf{y}-(\mathbf{x}_t-\gamma\nabla g(\mathbf{x}_t))\|^2 + h(\mathbf{y})\end{aligned}$$

The name comes because you can rewrite it using the *proximal* operator.

Proximal mapping We define the proximal mapping as follow: for a given h and a $\gamma > 0$:

$$\text{prox}_{h,\gamma}(\mathbf{z}) := \underset{\mathbf{y}}{\text{argmin}} \left\{ \frac{1}{2\gamma} \|\mathbf{y} - \mathbf{z}^2 + h(\mathbf{y})\| \right\}$$

Using the above definition, one iteration of proximal gradient descent can be rewritten as

$$\mathbf{x}_{t+1} := \text{prox}_{h,\gamma}(\mathbf{x} - \gamma \nabla g(\mathbf{x}_t))$$

Generalized gradient The above allows us to describe the generalized gradient as

$$G_{h,\gamma} := \frac{1}{\gamma} \left(\mathbf{x} - \text{prox}_{h,\gamma}(\mathbf{x} - \gamma \nabla g(\mathbf{x}))) \right)$$

and thus rewrite the update step as

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma G_\gamma(\mathbf{x}_t)$$

Generalization Proximal is generalization of gradient descent. If h is 0, it's gradient descent, if it's ι_X, it's the projected gradient descend. With

$$\iota_X(\mathbf{x}) := \begin{cases} 0 & \text{if } \mathbf{x} \in X \\ +\infty & \text{otherwise} \end{cases}$$

Proximal mapping becomes, with ι_X it becomes $\text{prox}_{h,\gamma} = \text{argmin}_{\mathbf{y} \in X} \|\mathbf{y} - \mathbf{z}\|^2$

Convergence $\mathcal{O}(\frac{1}{\epsilon})$ for smooth (same as vanilla), and if also strongly convex $\mathcal{O}(\log(\frac{1}{\epsilon}))$. Same as vanilla case for smooth functions, but now for any *h for which we can compute the proximal mapping*

Subgradient $\mathbf{g} \in \mathbb{R}^d$ is a subgradient of f at $\mathbf{x}$ if

$$f(\mathbf{y}) \geq \mathbf{g}^\top (\mathbf{y} - \mathbf{x}) \ \forall \mathbf{y} \in \mathbf{dom}(f)$$

Subdifferential $\partial f(\mathbf{x}) \subseteq \mathbb{R}^d$ is the set of subgradients of f at $\mathbf{x}$.

Lemma. *If $f : \mathbf{dom}(f) \to \mathbb{R}$ is differentiable at $\mathbf{x} \in \mathbf{dom}(f)$, then $\partial f(\mathbf{x}) \subseteq \{\nabla f(\mathbf{x})\}$*

This means at differentiable points of a function, there is either exactly one subgradient (that is the true gradient) or no subgradient at all (because it's a gradient, but above, in case of non-convex)

Subgradient and convexity "Convex = subgradients everywhere".

Lemma. *Let $f : \mathbf{dom}(f) \to \mathbb{R}$ is convex if and only if $\mathbf{dom}(f)$ is convex and $\partial f(\mathbf{x}) \neq \emptyset$ $\forall \mathbf{x} \in \mathbf{dom}(f)$*

Convex and Lipschitz Convex and Lipschitz = bounded subgradient.

Lemma. *Let $f : \mathbf{dom}(f) \to \mathbb{R}$ convex, $\mathbf{dom}(f)$ open, $B \in \mathbb{R}_+$. Then the following are equivalent:*

- $\|\mathbf{g}\| \leq B \ \forall \mathbf{x} \in \mathbf{dom}(f), \ \forall \mathbf{g} \in \partial f(\mathbf{x})$.
- $|f(\mathbf{x}) - f(\mathbf{y})| \leq B\|\mathbf{x} - \mathbf{y}\| \ \forall \mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$

Subgradient optimality condition

Lemma. *Suppose $f : \mathbf{dom}(f) \to \mathbb{R}$ and $\mathbf{x} \in \mathbf{dom}(f)$. If $\mathbf{0} \in \partial f(x)$, then $\mathbf{x}$ is a global minimum*

Differentiability

Theorem. *A convex function $f : \mathbf{dom}(f) \to \mathbb{R}$ is differentiable almost everywhere.*

Meaning: set of points where f is non-differentiable has measure 0 (no volume), and for all $\mathbf{x} \in \mathbf{dom}(f)$ and all $\epsilon > 0$, there is a point $\mathbf{x}'$ such that $\|\mathbf{x} - \mathbf{x}'\| < \epsilon$ and f is differentiable at $\mathbf{x}'$. This is a problem for gradient descent. Min can be non-differentiable, and non-differentiable points can be requested during descent.

Subgradient descent To solve this: use subgradient descent. Choose starting point $\mathbf{x}_0 \in \mathbb{R}^d$. Then for each times $t = 0, 1, \dots$ and stepsizes $\gamma_t \geq 0$ (not necessarily fixed) :

$$\begin{gathered} \text{Let } \mathbf{g}_t \in \partial f(\mathbf{x}_t) \\ \mathbf{x}_{t+1} := \mathbf{x}_t - \gamma_t \mathbf{g}_t \end{gathered}$$

Optimality Many convergence rates. Can we always improve? No.

Theorem (Nesterov). *For any $T \leq d-1$ and starting point $\mathbf{x}_0$, there is a function f in the problem class of B-Lipschitz function over $\mathbb{R}^d$, such that any (sub)gradient method has an objective error at least*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \geq \frac{RB}{2(1+\sqrt{T+1})}$$

This means that if we only have access to first order methods and the function is only Lipschitz, (no additional assumption), then the subgradient algorithm is optimal, and the convergence rate can't be faster than $\sim \frac{1}{\sqrt{T}}$.

Smooth non-differentiable If function is non-differentiable, it can't be smooth (so we can't use that assumption). Yet, we can still be better than $\mathcal{O}(\frac{1}{\epsilon^2})$ if we require strong convexity.

Strongly convex Definition is similar that for gradient, with the use of subgradient.

$$f(\mathbf{y}) \geq \overbrace{f(\mathbf{x}) + \mathbf{g}^\top(\mathbf{y}-\mathbf{x})}^{\text{Linearisation at } x} + \underbrace{\frac{\mu}{2}\|\mathbf{x}-\mathbf{y}\|^2}_{\text{Quadratic term}}$$

$\forall \mathbf{x}, \mathbf{y} \in \mathbf{dom}(f), \forall \mathbf{g} \in \partial f(\mathbf{x})$. Alternatively:

Lemma. *Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex, $\mathbf{dom}(f)$ open, $\mu \in \mathbb{R}_+^*$. f is strongly convex with parameter μ if and only if $f_\mu : \mathbf{dom}(f) \to \mathbb{R}$ defined by*

$$f_\mu(\mathbf{x}) = f(\mathbf{x}) - \frac{\mu}{2}\|\mathbf{x}\|^2, \ \mathbf{x} \in \mathbf{dom}(f)$$

is convex

Tame strong convexity Over $\mathbb{R}^d$, strong convexity and subgradients contradict each other. With strong convexity, gradients will explode to infinity.

Theorem. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be strongly convex with parameter $\mu > 0$, and let $\mathbf{x}^\star$ be the unique global minimum of f. With decreasing step size*

$$\gamma_t := \frac{2}{\mu(t+1)}, \ t > 0$$

subgradient descent yields

$$\underbrace{f\left(\frac{2}{T(T+1)}\sum_{t=1}^{T} t \cdot \mathbf{x}_t\right)}_{\textit{convex combination of iterates}} - f(\mathbf{x}^\star) \leq \frac{2B^2}{\mu(T+1)}$$

where $B = \max_{t=1}^T \|\mathbf{g}_t\|$

Strong convexity

5 Stochastic Gradient Descent

Idea Many objective functions are sum structured

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

As for example, f_i is the cost function of i-th observation. But doing so on very large datasets (e.g. Imagenet, $n \simeq 14M$). So we only do on a subset.

The algorithm As always, get a starting point $\mathbf{x_0} \in \mathbb{R}^d$, then sample $i \in [n]$ uniformly at random, and compute as before:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma_t \nabla f_i(\mathbf{x}_t)$$

So we only update with the gradient of f_i instead of the full gradient. The vector $\mathbf{g}_t := \nabla f_i(\mathbf{x}_t)$ is called a stochastic gradient.

Analysis We can't use the same as vanilla, because convexity $\big(f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x}^\star)\big)$ doesn't hold anymore. But cool thing! $\mathbf{g}_t$ is an unbiased estimate of the full gradient:

$$\mathbb{E}[\mathbf{g}_t | \mathbf{x}_t = \mathbf{x}] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x}) = \nabla f(\mathbf{x})$$

So the convexity inequality holds in expectation.

Holds in expectation For any fixed $\mathbf{x}$, linearity of conditional expectations yields

$$\begin{aligned} & \mathbb{E}\left[\mathbf{g}_t(\mathbf{x}-\mathbf{x}^\star)|\mathbf{x}_t=\mathbf{x}\right] \\ = \ & \mathbb{E}\left[\mathbf{g}_t|\mathbf{x}_t=\mathbf{x}\right]^\top(\mathbf{x}-\mathbf{x}^\star) \\ = \ & \nabla f(\mathbf{x})^\top(\mathbf{x}-\mathbf{x}^\star) \end{aligned}$$

Bounded stochastic gradient

Theorem. *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be convex and differentiable, $\mathbf{x}^\star$ a global minimum. Furthermore, suppose that $\|\mathbf{x}_0 - \mathbf{x}^\star\| \leq R$ and that $\mathbb{E}\left[\|\mathbf{g}_t\|^2\right] \leq B^2$ for all t. Choosing the constant stepsize*

$$\gamma := \frac{R}{B\sqrt{T}}$$

stochastic gradient descent yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[f(\mathbf{x}_t)\right] - f(\mathbf{x}^\star) \leq \frac{RB}{\sqrt{T}}$$

Note that in this theorem, the step size is constant.

Convergence rate: SGD vs GD For GD: we assumed $\|\nabla\frac{f}{\mathbf{x})}\|^2 \leq B_{GD}^2$, leading to

$$\|\frac{1}{n}\sum_i \nabla f_i(\mathbf{x})\|^2 \leq B_{GD}^2$$

As for SGD, assuming the same for the expected squared norms of our stochastic gradients, now called B_{SGD}^2 :

$$\frac{1}{n}\sum_i \|\nabla f_i(\mathbf{x})\|^2 \leq B_{SGD}^2$$

So GD can me better, but often comparable. Very similar if larger mini-batches are used:

$$\underbrace{\|\frac{1}{n}\sum_i \nabla f_i(\mathbf{x})\|^2}_{\approx B_{GD}^2} \leq \underbrace{\frac{1}{n}\sum_i \|\nabla f_i(\mathbf{x})\|^2}_{\approx B_{SGD}^2}$$

Strong convexity

Theorem. *For a strongly convex f, and with decreasing step size*

$$\gamma_t := \frac{2}{\mu(t+1)}$$

stochastic gradient descent yields

$$\mathbb{E}\left[f\left(\frac{2}{T(T+1)}\sum_{t=1}^{T} t\cdot\mathbf{x}_t\right) - f(\mathbf{x}^\star)\right] \leq \frac{2B^2}{\mu(T+1)}$$

Mini-Batch Instead of using a single element f_i, use an average of several of them

$$\tilde{\mathbf{g}}_t := \frac{1}{m}\sum_{j=1}^{m}\mathbf{g}_t^j$$

At the extremes ($m=1$ or $m=n$), we fall back to full gradient descent or SGD. But now computation can be naively parallelized. Also, by taking an average of many independent random variables reduce the variance. With larger size of the mini-batch m, $\tilde{\mathbf{g}}_t$ will be closer to the true gradient, in expectation:

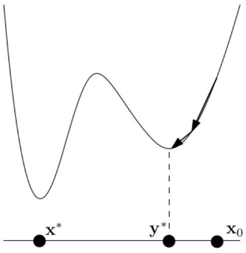
$$Var(\tilde{\mathbf{g}}_t) = \mathbb{E}\left[\|\tilde{\mathbf{g}}_t - \nabla f(\mathbf{x}_t)\|^2\right] \leq \frac{B^2}{m}$$

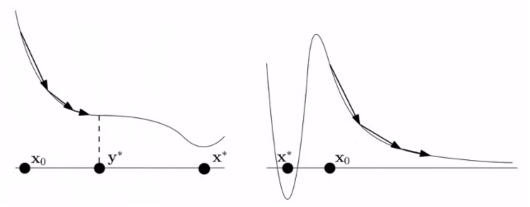
Stochastic Subgradient Descent For problems not necessarily differentiable. Quite the same, but we use a subgradient for one sample. We have $\mathbf{g}_t \in \partial f_i(\mathbf{x}_t)$. The rest is as before. This yields an unbiased estimate of the subgradient. We have a convergence in $\mathcal{O}(\frac{1}{\epsilon^2})$ by using the subgradient property.

Constrained optimization By slapping a projection back to X onto every step, we obtain projected SGD, with the same convergence rate as above., we obtain projected SGD, with the same convergence rate as above.

6 Non-convex Optimization

Gradient descent Main problem, is that





we may get stuck in a local minimum, or in a saddle point, or run off to infinity,...

Smoothness Interestingly, smoothness doesn't require convexity, so we can use it even with non-convex functions.

Convave f is concave is $-f$ is convex (for all $\mathbf{x}$, f is below the tangent at x).

Bounded Hessians

Lemma. *Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be twice differentiable, with $X \subset \mathbf{dom}(f)$ a convex set, and $\|\nabla^2 f(\mathbf{x})\| \leq L$ for all $\mathbf{x} \in X$, where $\|\cdot\|$ is the spectral norm. Then f is smooth with parameter L over X.*

So bounded Hessians $\Rightarrow$ smooth. The opposite (smooth $\Rightarrow$ bounded Hessians) is true over any open convex set X.

Convergence We sadly can't prove that we will converge to $x^\star$. But we can show that we will converge to a null gradient, at the same rate that we converge to $x^\star$ in the convex case (for $t \to \infty$). This implies that we may converge to a saddle point, or we are following an asymptote (e^{-x} will converge after infinite steps).

Smooth (but not necessarily convex). For f smooth with parameter L, and with stepsize $\gamma := \frac{1}{L}$, GD yields

$$\frac{1}{T}\sum_{t=0}^{T-1}\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2L}{T}\left(f(\mathbf{x}_0) - f(\mathbf{x}^\star)\right)$$

Note that we now have information on the average of the squared gradients, and not the actual values.

No overshooting If the function is smooth, and with the stepsize 1/L, gradient cannot overshoot (i.e. pass a critical point).

Trajectory analysis For non-convex functions, it's nice to have a good starting point and what happens when we start there.

Linear models with many outputs We have n data points, meaning n inputs $x_i \in \mathbb{R}^d$ and n outputs $y_n \in \mathbb{R}$. Hypothesis: $y_i \approx \mathbf{w}^\top x_i$ for some weight vector $\mathbf{w} \in \mathbb{R}^d$. We can generalize to more than one output value for each data point: n outputs $\mathbf{x}_n \in \mathbb{R}^m$. Hypothesis: $\mathbf{y}_i \approx W\mathbf{x}_i$ for a weight *matrix* $W \in \mathbb{R}^{m\times d}$.

Minimize LS To find that matrix $W^\star$ we pick the one that minimizes the least-squares error when taking into account all data points:

$$W^\star = \underset{W\in\mathbb{R}^{m\times d}}{\operatorname{argmin}} \sum_{i=1}^{n}\|W\mathbf{x}_i - \mathbf{y}_i\|^2$$

Notation:

- $X \in \mathbb{R}^{d\times n}$: columns are the $\mathbf{x}_i$
- $Y \in \mathbb{R}^{m\times n}$: columns are the $\mathbf{y}_i$

This makes equivalent to compute

$$W^\star = \underset{W\in\mathbb{R}^{m\times d}}{\operatorname{argmin}} \|WX - Y\|_F^2$$

This is cool for us: this argmin is a linear transformation $f(W)$. With that, the optimal is when the gradient is 0: $\nabla f(W^\star) = \mathbf{0}$. Equivalent to training a linear neural network with one layer under LS error.

Deep Linear NN Even with several layers (W_1, W_2, W_3), as long as the composition is linear, we can concatenate to $\mathbf{y} = W\mathbf{x}$, $W := W_3 W_2 w_1$.

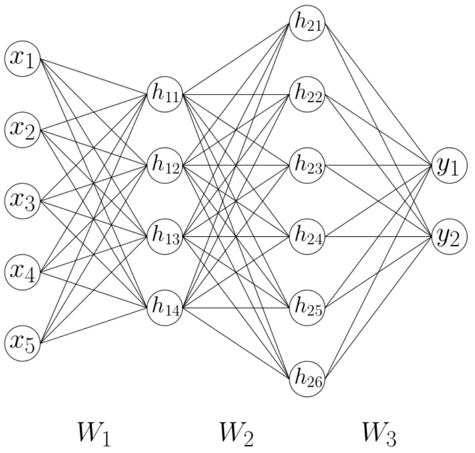


Figure 5: Multi-layers linear NN

Training with ℓ layers:

$$W^\star = \underset{W_1, W_2, \dots, W_\ell}{\text{argmin}} \| \underbrace{W_\ell W_{\ell-1} \cdots W_1 X}_{\text{Prediction}} - \overbrace{Y}^{\text{Output}} \|_F^2$$

From here, we use a toy example: all matrices are 1×1, $W_i = x_i, X = 1, Y = 1, \ell = d$, so $f : \mathbb{R}^d \to \mathbb{R}$.

$$f(\mathbf{x}) := \frac{1}{2}\left(\prod_{k=1}^{d} x_k - 1\right)^2$$

Gradient With this toy f, the gradient is the following:

$$\nabla f(\mathbf{x} = \left(\prod_k x_k - 1\right)\left(\prod_{k\neq 1}, \dots, \prod_{k\neq d} x_k\right).$$

Convergence analysis We want to show that for any $d > 1$ and from anywhere in $X : \{\mathbf{x} : \mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \leq 1\}$, gradient descent will converge to a global minimum. f is not smooth over X, but we can show that it is smooth along the trajectory of the gradient for suitable L, so that we get sufficient decrease

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2, \ t \geq 0$$

So we can't converge to a saddle point. For $\mathbf{x} > \mathbf{0}, \prod_k \mathbf{x}_k \geq 1$, we also get convergence. But there are also starting points from which gradient descent will not converge to a global minimum.

Balanced iterates Let $\mathbf{x} > \mathbf{0}$ (component wise) and let $c \geq 1$ be a real number. $\mathbf{x}$ is called *$c -$ balanced* if $x_i \leq c x_j$ for all $1 \leq i, j \leq d$.

Lemma. *Let $\mathbf{x} \geq \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$. Then for any stepsize $\gamma > 0$, $\mathbf{x}' := \mathbf{x} - \gamma \nabla f(\mathbf{x})$ satisfies $\mathbf{x}' \geq \mathbf{x}$ (component-wise) and is also c-balanced.*

Bounded Hessians We start by computing the Hessian:

$$\nabla^2 f(\mathbf{x})_{ij} = \left\{ \begin{array}{ll} \left(\prod_{k\neq i} x_k\right)^2 & j = i \\ 2\prod_{k\neq i} x_k \prod_{k\neq j} x_k - \prod_{k\neq i,j} x_k & j \neq i \end{array} \right\}$$

Lemma. *Suppose that $\mathbf{x} > \mathbf{0}$ is c-balanced. Then for any $I \subseteq \{1, \dots, d\}$, we have*

$$\left(\frac{1}{c}\right)^{|I|}\left(\prod_k x_k\right)^{1-\frac{|I|}{d}} \leq \prod_{k\notin I} x_k \leq c^{|I|}\left(\prod_k x_k\right)^{1-\frac{|I|}{d}}$$

Lemma. *Let $\mathbf{x} > \mathbf{0}$ be c-balanced with $\prod_k x_k \leq 1$. Then*

$$\|\nabla^2 f(\mathbf{x})\| \leq \|\nabla^2 f(\mathbf{x})\|_F \leq 3dc^2$$

Where $\|\cdot\|$ is the spectral norm, and $\|\cdot\|_F$ is the Frobenius norm.

Smoothness along the trajectory

Lemma. *Let $\mathbf{x} > \mathbf{0}$ be c-balanced, with $\prod_k x_k < 1$, $L = 3dc^2$. Let $\gamma := \frac{1}{L}$. Then for all $0 \geq \nu \geq \gamma$,*

$$\mathbf{x}' := \mathbf{x} - \nu \nabla f(\mathbf{x}) \geq \mathbf{x}$$

is c-balanced with $\prod_k x'_k \leq 1$, and f is smooth with parameter L over the line segment connecting $\mathbf{x}$ and $\mathbf{x} - \gamma \nabla f(\mathbf{x})$.

Convergence

Theorem. *Let $c \geq 1$ and $\delta > 0$ such that $\mathbf{x}_0 > \mathbf{0}$ is c-balanced with delta $\leq \prod_k (\mathbf{x}_0)_k < 1$. Choosing stepsize $\gamma = \frac{1}{3dc^2}$, gradient descent satisfies*

$$f(\mathbf{x}_T) \leq \left(1 - \frac{\delta^2}{3c^4}\right)^T f(\mathbf{x}_0),\ T \geq 0$$

7 Accelerated Gradient Descent

Idea On smooth functions, gradient descent is good but perhaps not as fast as it could be. Don't forget it's just some algorithm that uses gradient information, perhaps there is a better first-order method for smooth convex functions with a smaller upper-bound on the number of gradients computed.

Theorem. *Every first-order method needs in the worst case $\Gamma/\sqrt{\epsilon}$ steps (gradients evaluations) in order to achieve an additive error of ϵ on smooth functions.*

There is a gap between $\mathcal{O}(1/\epsilon)$ (gradient descent) and the lower bound!

Nesterov's accelerated GD Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex, differentiable and smooth with parameter L, Choose $\mathbf{z}_z = \mathbf{y}_0 = \mathbf{x}_0$ arbitrary. For $t \geq 0$, set

$$\begin{array}{lcl} \mathbf{y}_{t+1} & := & \mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t) \\ \mathbf{z}_{t+} & := & \mathbf{z}_z - \frac{t+1}{2L}\nabla f(\mathbf{x}_t) \\ \mathbf{x}_{t+1} & := & \frac{t+1}{t+3}\mathbf{y}_{t+1} + \frac{2}{t+3}\mathbf{z}_{t+1} \end{array}$$

Perform a "smooth step" from $\mathbf{x}_t$ to $\mathbf{y}_{t+1}$, then perform a more aggressive step from $\mathbf{z}_t$ to $\mathbf{z}_{t+1}$. Next iterate, $\mathbf{x}_{t+1}$ is a weighted average of $\mathbf{y}_{t+1}$ and $\mathbf{z}_{t+1}$, where we compensate for the more aggressive step by giving $\mathbf{z}_{t+1}$ a relatively low weight.

Error bound

Theorem. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^\star$; furthermore, suppose that f is smooth with parameter L. Accelerated gradient descent yields*

$$f(\mathbf{y}_T) - f(\mathbf{x}^\star) \leq \frac{2L\|\mathbf{z}_0 - \mathbf{x}^\star\|^2}{T(T+1)},\ T > 0$$

To reach error at most ϵ, accelerated gradient descent therefore only needs $\mathcal{O}(1/\sqrt{\epsilon})$ steps instead of $\mathcal{O}(1/\epsilon)$. Recall the bound for gradient descent:

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{L}{2T}\|\mathbf{x}_0 - \mathbf{x}^\star\|^2, T > 0$$

Potential function Idea: assign a potential $\Phi(T)$ to each time t and show that $\Phi(t+1) \leq \Phi(t)$. We define the potential as

$$\Phi(t) := t(t+1)(f(\mathbf{y}_t) - f(\mathbf{x}^\star)) + 2L\|\mathbf{z}_t - \mathbf{x}^\star\|^2$$

If we can show that the potential always decrease, we get $\Phi(T) \leq \Phi(0)$, and replacing + rewriting we get the claimed error bound.

8 Newton's Method

1D Case Also called the *Newton-Raphson Method.* The goal is to find a zero of a differentiable $f : \mathbb{R} \to \mathbb{R}$. Method:

$$x_{t+1} := x_t - \frac{f(x_t)}{f'(x_t)}, t \geq 0$$
$$f(x_t) + f'(x_t)(x - x_t) = 0$$

Illustration is at figure 6

Newton's Method for optim 1D Case: find a global minimum $x^\star$ of a differentiable function $f : \mathbb{R} \to \mathbb{R}$. Can equivalently

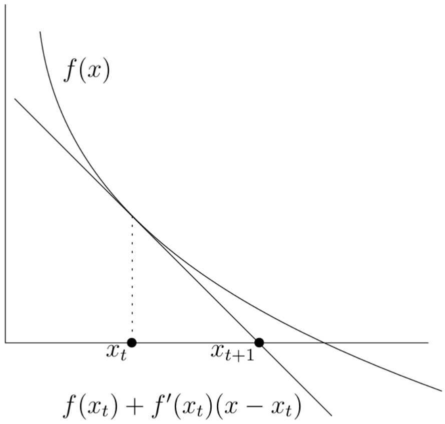


Figure 6: Newton-Raphson Method

search for a zero of the derivative f' : apply the Newton-Raphson method to f'. Note that requires f twice differentiable. Update step:

$$x_{t+1} := x_t - \frac{f'(x_t)}{f''(x_t)} = x_t - f''(x_t)^{-1} f'(x_t)$$

d-dimensional case: Newton's method for minimizing a convex function $f : \mathbb{R}^d \to \mathbb{R}$:

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \nabla^2 f(\mathbf{x}_t)^{-1} \nabla f(\mathbf{x}_t)$$

Newton as adaptive GD In general, the update for first order method is as follow: $\mathbf{x}_{t+1} = \mathbf{x}_t - H(\mathbf{x}_t)\nabla f(\mathbf{x}_t)$, with $H(\mathbf{x}) \in \mathbb{R}^{d\times d}$ is some matrix. For regular GD, $H = \gamma I$, while for newton we have $H = \nabla^2 f(\mathbf{x}_t)^{-1}$. Thus Newton's method is an "adaptive gradient descent", adaptation is w.r.t the local geometry of the function at $\mathbf{x}_t$.

Nondegenerate quadratic function A nondegenerate quadratic function is of the form

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top M\mathbf{x} - \mathbf{q}^\top\mathbf{x} + c$$

Where $M \in \mathbb{R}^{d\times d}$ is an invertible symmetric matrix, $\mathbf{q} \in \mathbb{R}^d$, $c \in \mathbb{R}$. Let $\mathbf{x}^\star = M^{-1}\mathbf{q}$ be the unique solution of $\nabla f(\mathbf{x}) = \mathbf{0}$. $\mathbf{x}^\star$ is the unique global minimum if f is convex.

Lemma. *On nondegenerate quadratic functions, any starting point $\mathbf{x}_0 \in \mathbb{R}^d$, Newton's method yields $\mathbf{x}_1 = \mathbf{x}^\star$.*

Affine Invariance Newton's method is affine invariant (invariant under any invertible affine transformation).

$$\begin{tikzcd}
\mathbf{x}_t \arrow[r, "N_f"'] & \mathbf{x}_{t+1} \arrow[d, "g^{-1}"'] \\
\mathbf{y}_t \arrow[u, "g"'] \arrow[r, "N_{f\circ g}"] & \mathbf{y}_{t+1}
\end{tikzcd}$$

Figure 7: Affine Invariance

Lemma. *let $f : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable, $A \in \mathbb{R}^{d\times d}$ an invertible matrix, $\mathbf{b} \in \mathbb{R}^d$. Let $g : \mathbb{R}^d \to \mathbb{R}$ be the (bijective) affine function $g(\mathbf{y}) = A\mathbf{y} + \mathbf{b}$, $\mathbf{y} \in \mathbb{R}^d$. Finally, for a twice differentiable function $h : \mathbb{R}^d \to \mathbb{R}$, let $N_h : \mathbb{R}^d \to \mathbb{R}^d$ denote the Newton step for h, i.e.*

$$N_h(\mathbf{x}) = \mathbf{x} - \nabla^2 h(\mathbf{x})^{-1}\nabla h(\mathbf{x})$$

whenever this is defined. Then we have $N_{f\circ g} = g^{-1} \circ N_f \circ g$.

Newton as Taylor Alternatively, we can interpret each step as minimizing the local second-order Taylor approximation

Lemma. *Let f be convex and twice differentiable at $\mathbf{x}_t \in \mathbf{dom}(f)$, with $\nabla^2 f(\mathbf{x}_t) \succeq 0$ being invertible. The vector $\mathbf{x}_{t+1}$ resulting from the Newton step satisfies*

$$\begin{aligned} \mathbf{x}_{t+1} = \mathrm{argmin}_{\mathbf{x}\in\mathbb{R}^d} \quad & f(\mathbf{x}_t) \\ & + \nabla f(\mathbf{x}_t)^\top(\mathbf{x} - \mathbf{x}_t) \\ & + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_t)^\top \nabla^2 f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t) \end{aligned}$$

Local convergence Under suitable conditions, and starting close to the global minimum, Newton's method will reach distance at most ϵ to the minimum within $\log\log(1/\epsilon)$ steps. Much faster, but we need to start close to the minimum already. This is a local convergence result.

[. . . missing]

Downsides We have a computational bottleneck in each step: we need to either compute and invert the Hessian matrix, or solve the linear system $\nabla^2 f(\mathbf{x}_t)\Delta\mathbf{x} = -\nabla f(\mathbf{x}_t)$ for the next step $\Delta\mathbf{x}$. Matrix/system has size $d \times d$, taking up to $\mathcal{O}(d^3)$ time to invert/solve, while d may be large.

The secant method Yet another iterative method for finding zeros in dimension 1. Now, use finite difference approximation: $f'(x_t) \simeq \frac{f(x_t)-f(x_{t-1})}{x_t - x_{t-1}}$ [2], and replace in Newton-Raphson step to have the secant method.

This is nice, because this is a derivative-free version of the Newton-Raphson method. We can also optimize a differentiable univariate function f, by applying the secant method to f'.

$$x_{t+1} := x_t - f'(x_t)\frac{x_t - x_{t-1}}{f'(x_t) - f'(x_{t-1})}$$

Secant in high dim Possible if we match the **the secant condition**

Quasi-newton methods

9 Frank-Wolfe

Intro An alternative algorithm to optimize smooth convex function over compact set. The main difference with the projected gradient descent is that it doesn't require a projection step, which can be quite costly.

Algorithm

$$\begin{aligned}\mathbf{s} &:= LMO(\nabla f(\mathbf{x}_t)),\\ \mathbf{x}_{t+1} &:= (1-\gamma)\mathbf{x}_t + \gamma\mathbf{s}\end{aligned}$$

For time steps $t = 0, 1, \ldots$, and stepsize $\gamma := \frac{2}{t+2}$. We also define LMO (Linear Minimization Oracle) as a scalar product:

$$LMO(\mathbf{g}) := \underset{\mathbf{s}\in X}{\operatorname{argmin}} \langle \mathbf{s}, \mathbf{g}\rangle$$

Properties

- Always feasible (implies $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t \in X$). Because $\mathbf{x}_{t+1}$ is on line segment $[\mathbf{s}, \mathbf{x}_t]$ for $\gamma \in [0, 1]$.
- Reduces non-linear to linear optimization.
- Projection-free
- Sparse iterates (in terms of corners $\mathbf{s}$ used). At most, one extreme point is added at each step!

Frank-Wolfe and Lasso The each step has complexity $\mathcal{O}(d)$, much better than projected gradient descent.

Duality Gap

$$g(\mathbf{x}) := \langle \mathbf{x} - \mathbf{s}, \nabla f(\mathbf{x})\rangle$$

This gives a certificate for optimization quality:

$$\begin{aligned} g(\mathbf{x}) &= \max_{\mathbf{s}\in X} \langle \mathbf{x} - \mathbf{s}, \nabla f(\mathbf{x})\rangle\\ &\geq \langle \mathbf{x} - \mathbf{a}^\star, \nabla f(\mathbf{x})\rangle\\ &\geq f(\mathbf{x}) - f(\mathbf{x}^\star)\end{aligned}$$

Stepsize variants

$$\begin{aligned} \gamma &:= \tfrac{2}{t+2} & \\ \gamma_t &:= \underset{\gamma\in[0,1]}{\operatorname{argmin}} f\big((1-\gamma)\mathbf{x}_t + \gamma\mathbf{s}\big) & \text{line search}\\ \gamma_t &:= \min\left\{\tfrac{g(\mathbf{x}_t)}{L\|\mathbf{s}-\mathbf{x}\|^2}, 1\right\} & \text{gap-based}\end{aligned}$$

Convergence

[2] for $|x_t - x_{t-1}|$ small

Theorem. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and smooth with parameter L, and $\mathbf{x}_0 \in X$. Then choosing any of the above stepsizes, the Frank-Wolfe algorithm yields*

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{2L diam(X)^2}{T+1}$$

Where $diam(X) := \max_{\mathbf{x},\mathbf{y}\in X} \|\mathbf{x}-\mathbf{y}\|$ is the diameter of X.

We then have a convergence of rate $\mathcal{O}(\frac{L}{T})$, which is the same as PGD. It's also suboptimal, compared to accelerated GD (with rate of $\mathcal{O}(\frac{L}{T^2})$. The proof uses the following lemma:

Lemma. *For a step $\mathbf{x}_{t+1} := \mathbf{x}_t + \gamma(\mathbf{s}-\mathbf{x}_t)$ with an arbitrary step-size $\gamma \in [0,1]$, it holds that*

$$f(\mathbf{x}_{t+1} \leq f(\mathbf{x}_t) - \gamma g(\mathbf{x}_t) + \frac{\gamma^2}{2} L diam(X)^2$$

if $\mathbf{s} = LMO(\nabla f(\mathbf{x}_t))$.

Curvature constant We define the constant C_f as

$$\sup_{\substack{\mathbf{x},\mathbf{s}\in X,\gamma\in[0,1] \\ \mathbf{y}=\mathbf{x}+\gamma(\mathbf{s}-\mathbf{x})}} \frac{2}{\gamma^2}(f(\mathbf{y}) - f(\mathbf{x}) - \langle \mathbf{y}-\mathbf{x}, \nabla f(\mathbf{x})\rangle)$$

Affine invariance The algorithm is invariant to scaling (affine transformations) of the input problem. And so is C_f.

Convergence in Duality Gap

Theorem. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and smooth with parameter L, and $\mathbf{x}_0 \in X$, $T \geq 2$. Then choosing any of the above stepsizes, the Fran-Wolfe algorithm yields a t, $1 \leq t \leq T$ such that*

$$g(\mathbf{x}_t) \leq \frac{27/4C_f}{T+1}$$

10 Coordinate Descent

Idea This is yet another first-order optimization method. As before, we look for $x^\star \in \mathbb{R}^d$, by minimizing $f(\mathbf{x})$. We do so by updating one coordinate at a time, while keeping the other fixed. By definition, the value will decrease at each step.

Algorithm At time t, select a coordinate $i_t \in [d]$, then compute $\mathbf{x}_{t+1} := \mathbf{x}_t + \gamma \mathbf{e}_{i_t}$. From that, you have 2 variants: either a gradient-based step-size $\left(\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{1}{L}\nabla_{i_t} f(\mathbf{x}_t)\mathbf{e}_{i_t}\right)$, or by exact coordinate minimization (solve the single-variable minimization $\text{argmin}_{\gamma\in\mathbb{R}} f(\mathbf{x}_t + \gamma\mathbf{e}_{i_t})$). The latter is easy to solve, can be approximately solved, and has no hyperparameter to tune.

Randomized Coordinate Descent

We first select the index $i_t \in [d]$ uniformly at random, then compute the next step

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \frac{1}{L}\nabla_{i_t} f(\mathbf{x}_t)\mathbf{e}_{i_t}$$

Convergence We assume coordinate-wise smoothness. It is equivalent to coordinate-wise Lipschitz-gradient. Additionally, we assume strong convexity ($\mu > 0$).

Theorem. *Let f be coordinate-wise smooth with constant L, and strongly convex with parameter $\mu > 0$. Then, coordinate descent with a step-size of $\frac{1}{L}$ (as above) when choosing the active coordinate i_t uniformly at random, has an expected linear convergence rate of*

$$\mathbb{E}\left[f(\mathbf{x}_t) - f^\star\right] \leq \left(1 - \frac{\mu}{dL}\right)^t \left[f(\mathbf{x}_0) - f^\star\right].$$

The Polyak-Lojasiewicz Condition

Definition. *f satisfies the PL inequality if the following holds for some $\mu > 0$:*

$$\frac{1}{2}\|\nabla f(\mathbf{x})\}\|^2 \geq \mu(f(\mathbf{x}) - f^\star),\ \forall \mathbf{x}$$

Lemma. *Strong convexity → PL. Let f be strongly convex, with parameter $\mu > 0$. Then f satisfies PL for the same μ.*

Importance Sampling Uniformly random selection is not always the best. For that, we define individual smoothness constants L_i for each coordinate i :

$$f(\mathbf{x}+\gamma\mathbf{e}_i) \leq f(\mathbf{x}) + \gamma\nabla_i f(\mathbf{x}) + \frac{L_i}{2}\gamma^2$$

With these constants, we can define a biased sampling: $P[i_t = i] = \frac{L_i}{\sum_i L_i}$, and using a step-size of $\frac{1}{L_{i_t}}$, this converges with a faster rate of

$$\mathbb{E}[f(\mathbf{x}_t) - f^\star] \leq \left(1 - \frac{\mu}{d\bar{L}}\right)^t [f(\mathbf{x}_0) - f^\star]$$

With $\overline{L}$ the mean of L_i

Steepest coordinate descent Other coordinate selection rule: $i_t := \text{argmax}_{i\in[d]} |\nabla_i f(\mathbf{x}_t)|$. This is a greedy algorithm. It now becomes deterministic, and not random anymore. Same convergence rate as for random coordinate descent.

Faster convergence with steepest A faster convergence can be obtained when the strong convexity f f is measured with respect to the ℓ_1-norm instead of the normal Euclidean norm, i.e.

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle\nabla f(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle + \frac{\mu_1}{2}\|\mathbf{y}-\mathbf{x}\|_1^2.$$

If f is coordinate-wise L-smooth, strongly convex w.r.t the ℓ_1-norm with parameter $\mu_1 > 0$, and step-size of $\frac{1}{L}$, we have a linear convergence rate of

$$[f(\mathbf{x}_t) - f^\star] \leq \left(1 - \frac{\mu_1}{L}\right)^t [f(\mathbf{x}_0) - f^\star]$$

Non-Smooth For general non-smooth f, coordinate descent fails (gets permanently stuck). But if the non-smooth part is separable over the coordinates, we have global convergence. This means

$$f(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x}),\ h(\mathbf{x}) = \sum_i h_i(x_i)$$

(e.g. $f(\mathbf{x}) := \|\mathbf{x}\|^2 + \|\mathbf{x}\|_1$)

10.1 Duality

Conjugate Given a function $f : \mathbb{R}^d \to \mathbb{R}^+$, the conjugate $f^* : \mathbb{R}^d \to \mathbb{R}^+$ is defined as

$$f^*(\mathbf{y}) := \max_{\mathbf{x}}\ \mathbf{x}^\top\mathbf{y} - f(\mathbf{x})$$

Also called *Legendre transform* or *Fenchel conjugate function.* Note that $\mathbb{R}^+ := \mathbb{R} \cup \{+\infty\}$. Here is another way of looking at it: given $\mathbf{y}$, draw the slope line (green). Then look at the point of $f(\mathbf{x})$ that is the *furthest below* (it's not an absolutes difference). This can also be achieved by taking the parallel to the slope, and evaluating at $\mathbf{x} = \mathbf{0}$.

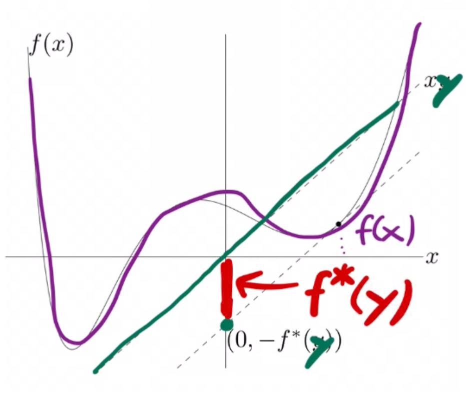


Figure 8: Maximum gap

Properties of conjugate

- f^* is always convex, even if f is not.
- *Fenchel's inequality*:

$$f(\mathbf{x}) + f^*(\mathbf{y}) \geq \mathbf{x}^\top\mathbf{y},\ \forall\mathbf{x},\mathbf{y}$$

- Hence, the conjugate of the conjugate f^{**} satisfies $f^{**} \leq f$. The second conjugate is also convex, and can be interpreted as the "envelope" of the original function.

- If f is closed an convex, then $f^{**} = f$

- if if closed and convex, then for any $\mathbf{x}, \mathbf{y}$:

$$\begin{aligned}\mathbf{y} \in \partial f(\mathbf{x}) \quad &\Leftrightarrow \mathbf{x} \in \partial f^*(\mathbf{y}) \\ &\Leftrightarrow f(\mathbf{x}) + f^*(\mathbf{y}) = \mathbf{x}^\top \mathbf{y}\end{aligned}$$

- Separable functions: if $f(\mathbf{u}, \mathbf{v}) = f_1(\mathbf{u}) + f_2(\mathbf{v})$, then $f^*(\mathbf{w}, \mathbf{z}) = f_1^*(\mathbf{w}) + f_2^*(\mathbf{z})$

Some known conjugates

- **Indicator function** of a set $C \subseteq \mathbb{R}^d$

$$\iota_C(\mathbf{x}) := \begin{cases} 0 & \mathbf{x} \in C, \\ +\infty & \text{otherwise.} \end{cases}$$

  If $f(\mathbf{x}) = \iota_C(\mathbf{x})$, then the conjugate is

$$f^*(\mathbf{y}) = \max_{\mathbf{x} \in C} \mathbf{y}^\top \mathbf{x}$$

- **Norm**: if $f(\mathbf{x}) = \|\mathbf{x}\|$, then its conjugate is

$$f^*(\mathbf{y}) = \iota_{\{\mathbf{z}:\|\mathbf{z}\|_* \leq 1\}}(\mathbf{y})$$

  Which corresponds to the indicator of the dual norm ball. Note: the dual norm of $\|.\|$ is defined as $\|\mathbf{y}\|_* := \max_{\|\mathbf{x}\| \leq 1} \mathbf{y}^\top - \mathbf{x}$. E.g. $\|.\|_1 \leftrightarrow \|.\|_\infty$

- **Generalized linear models** $\min_{\mathbf{x} \in \mathbb{R}^d} f(A\mathbf{x}) + g(\mathbf{x})$. We reformulate as $\min_{\mathbf{x} \in \mathbb{R}^d, \mathbf{w} \in \mathbb{R}^n} f(\mathbf{w}) + g(\mathbf{x})$, such that $\mathbf{w} = A\mathbf{x}$. We define the Lagrange dual function:

$$\begin{aligned}\mathcal{L}(\mathbf{u}) \quad &:= \min_{\substack{\mathbf{x} \in \mathbb{R}^d \\ \mathbf{w} \in \mathbb{R}^n}} f(\mathbf{w}) + g(\mathbf{x}) + \mathbf{u}^\top(\mathbf{w} - A\mathbf{x}) \\ &= -f^*(-\mathbf{u}) - g^*(A^\top \mathbf{u})\end{aligned}$$

  And thus, gives rise to the **dual problem**: minimizing the above function is the same as maximizing the conjugate :

$$\max_{\mathbf{u} \in \mathbb{R}^n} \left[\mathcal{L}(\mathbf{u}) = -f^*(-\mathbf{u}) - g^*(A^\top \mathbf{u})\right]$$

- **Lasso** $\min_{x \in \mathbb{R}^d} \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda\|\mathbf{x}\|_1$. We use separability to split the two as $f(\mathbf{w})$ and $g(\mathbf{x})$. This yields

$$\begin{aligned}f^*(\mathbf{u}) &= \tfrac{1}{2}\|\mathbf{b}\|^2 - \tfrac{1}{2}\|\mathbf{b} - \mathbf{u}\|^2 \\ g^*(\mathbf{v}) &= \iota_{\mathbf{z}:\|\mathbf{z}\|_\infty \leq 1}(\mathbf{v}/\lambda)\end{aligned}$$

  We then define the dual problem using the above function (GLM)

Why duality Duality gap gives a certificate of current optimization quality. The minimization of the former function is greater or equal to the maximization of the conjugate. The optimum value is then between those two. If they are close, we have a certificate of quality. You can then use the gap as a **stopping criterion**. And finally, in some cases, the dual can be easier to solve.

10.2 Zero-order Optimization

Goal Can we optimize without access to the gradient ?

Random search Pick a random direction $\mathbf{d}_t \in \mathbb{R}^d$. Define $\gamma := \text{argmin}_{\gamma \in \mathbb{R}} f(\mathbf{x}_t + \gamma \mathbf{d}_t)$ (line-search). Finally, do a step: $\mathbf{x}_{t+1} := \mathbf{x}_t + \gamma \mathbf{d}_t$. *Convergence* is the same as gradient descent, up to a slow-down factor d. For convex functions, we get a rate $\mathcal{O}(dL/\epsilon)$, and for strongly convex we have $\mathcal{O}\left(dL \log(\frac{1}{\epsilon})\right)$

Reinforcement Learning State changes based on some inputs: $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t, \mathbf{e}_t)$, where $\mathbf{s}_t$ is the state of the system, a_t is the control action and $\mathbf{e}_t$ is some random noise. We assume that f is fixed, but unknown. We want to establish a policy based on previous actions and states:

$$\mathbf{a}_t := \pi(\mathbf{a}_1, \ldots, \mathbf{a}_{t-1}, \mathbf{s}_0, \ldots, \mathbf{s}_t)$$

Which takes a trajectory of the dynamical system and outputs a new control action. The goal is to maximize the overall reward

$$\max_{\mathbf{a}_t} \mathbb{E}_{\mathbf{e}_t}\left[\sum_{t=0}^{N} R_t(\mathbf{s}_t, \mathbf{a}_t)\right]$$

using the definition of the next state ($\mathbf{s}_0$ given)

10.3 Other SGD Methods

Adagrad Adaptive variant of SGD:

1. Pick a stochastic gradient $\mathbf{g}_t$
2. update $[G_t]_i := \sum_{s=0}^{t}\left([\mathbf{g}_s]_i\right)^2$
3. $[\mathbf{x}_{t+1}]_i := [\mathbf{x}_t]_i - \frac{\gamma}{\sqrt{[G_t]_i}}[\mathbf{g}_t]_i$

Standard choice for $\mathbf{g}_t := \nabla f_j(\mathbf{x}_t)$ four sum-structured objective functions $f = \sum_j f_j$. Normal SGD would skip step 2, and not divide γ on step 3.

Adam Adam is a momentum variant of Adagram:

1. pick a stochastic gradient $\mathbf{g}_t$
2. $\mathbf{m}_t := \beta_1\mathbf{m}_{t-1} + (1-\beta_1)\mathbf{g}_t$ (momentum term)
3. $[\mathbf{v}_t]_i := \beta_2[\mathbf{v}_{t-1}]_i + (1-\beta_2)([\mathbf{g}_s]_i)^2\ \forall i$ (2nd order statistics
4. $[\mathbf{x}_{t+1}]_i := [\mathbf{x}_t]_i - \frac{\gamma}{\sqrt{[\mathbf{v}_t]_i}}[\mathbf{m}_t]_i\ \forall i$

We now have an exponential decay. It's then faster at forgetting older weights. We have a momentum from previous gradients. There exists a simplified version without correction for initialization of $\mathbf{m}_0, \mathbf{v}_0$. In practice, we see strong performances e.g. for self-attention networks.

SignSGD Only use the sign (one bit) of each gradient entry. This is a communication efficient variant:

1. Pick a stochastic gradient $\mathbf{g}_t$
2. $[\mathbf{x}_{t+1}]_i := [\mathbf{x}_t]_i - \gamma_t sign([\mathbf{g}_t]_i)\ \forall i$

With possible rescaling of γ_t with $\|\mathbf{g}_t\|_1$. This is communication efficient, for distributed training. But we face convergence issues.

11 Optim for ML in practice

11.1 Parallelization

Why We need more power, and machines are not getting more powerful. So we distribute. But it's not obvious.

One-shot averaging We distribute the dataset across machine, then average the weights at the end:

$$\mathbf{w} := \frac{1}{K}\sum_k \mathbf{w}^{(k)}$$

Sadly this doesn't work, because $\operatorname{argmin}_w \operatorname{avg} f_i(\mathbf{w}) \neq \operatorname{avg}\operatorname{argmin}_w f_i(\mathbf{w})$

Communication So communicating once at the end, doesn't work. The opposite (communicate at each step) does work, but require a lot of communication. Take a minibatch of your share of the dataset (e.g. 1 sample), then compute the difference $\Delta\mathbf{w}^{(i)}$, and centralize, with the reduce function

$$\mathbf{w} := \mathbf{w} + \sum_k \Delta\mathbf{w}^{(k)}$$

Obviously, sending between machine is excruciatingly slow. We would rather do some computation locally and reduce the number of communications.

Batch size Increasing batch size seems like a good idea. It is, but only up to a certain point. Figure 9 shows the limits.

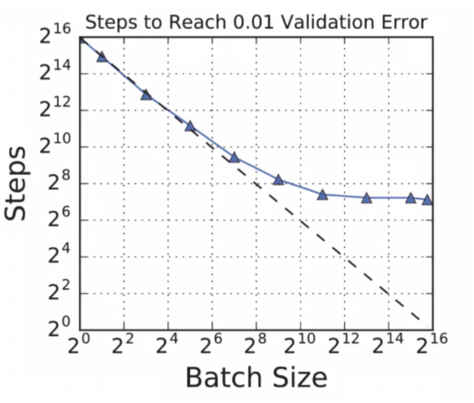


Figure 9: Increasing batch size.

A Tables and summaries

	GD	SGD	Sub-GD
Lipschitz convex	$\mathcal{O}(\frac{1}{\epsilon^2})$		$\mathcal{O}(\frac{1}{\epsilon^2})$
Smooth & convex	$\mathcal{O}(\frac{1}{\epsilon})$		
Strongly convex		$\mathcal{O}(\frac{1}{\epsilon})$	
Smooth & strongly convex	$\mathcal{O}(\log\frac{1}{\epsilon})$		

Table 1: Convergence depending on function properties

B Useful functions

Euclidean Norm $\|X\|^2 = X^\top X = \sum_{n=1}^d x_i^2 \in \mathbb{R}_+$

Triangle inequality $\|\mathbf{x}+\mathbf{y}\| \le \|\mathbf{x}\| + \|\mathbf{y}\|$

Gradient

$$\nabla f(\mathbf{x}) := \left(\frac{\partial f}{\partial x_1}(\mathbf{x}), \dots, \frac{\partial f}{\partial x_d}(\mathbf{x})\right)$$

Hessian

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(\mathbf{x}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d}(\mathbf{x}) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\mathbf{x}) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d}(\mathbf{x}) \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1}(\mathbf{x}) & \frac{\partial^2 f}{\partial x_d \partial x_2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_d \partial x_d}(\mathbf{x}) \end{pmatrix}$$

Positive semidefinite A symmetric matrix M is positive semidefinite if $\mathbf{x}^\top M\mathbf{x} \ge 0$ for all $\mathbf{x}$ and positive definite if $\mathbf{x}^\top M\mathbf{x} > 0$ for all $\mathbf{x} \ne \mathbf{0}$

Spectral norm Let A be an $(m \times d)$-matrix. Then

$$\|A\| := \max_{\mathbf{v}\in\mathbb{R}^d, \mathbf{v}\ne 0} \frac{\|A\mathbf{v}\|}{\|\mathbf{v}\|} = \max_{\|\mathbf{v}\|=1} \|A\mathbf{v}\|$$

is the 2-norm (or spectral norm) of A.

Frobenius Norm

$$\|A\|_F = \sqrt{\sum\nolimits_{i,j} a_{ij}^2}$$

Equivalent to the euclidean norm of $vec(A)$, the “flattening” of A.

Scalar product

$$\langle \mathbf{s}, \mathbf{g} \rangle = \sum_i s_i g_i$$