Optimization for Data Science

by Nils Jensen in FS23 - No guarantee of completeness

1 Introduction

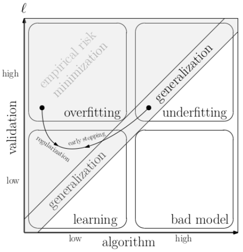
Expected risk minimization

We have a **data source** $\mathcal{X}$ from which we draw **samples** $X_1, \ldots, X_n$. We can see $\mathcal{X}$ as a probability distribution. Define $\mathcal{H}$ as a class of **hypotheses** (possible explanations of $\mathcal{X}$). Want to select the one that best explains $\mathcal{X}$.
A **risk/loss function** $l : \mathcal{H} \times \mathcal{X} \to \mathbb{R}$ quantifies how well we think that a given hypothesis $H \in \mathcal{H}$ explains given data $X \in \mathcal{X}$.
The **expected risk** is $l(H) := \mathbb{E}_{\mathcal{X}}[l(H, X)]$. Goal: Find $H \in \mathcal{H}$ with the smallest expected risk: $H^* = \text{argmin}_{H\in\mathcal{H}} l(H)$. Problem: We do not know the distribution: we can only work with finitely many samples. We try to be **probably approximately correct (PAC)**: take tolerances $\delta, \epsilon > 0$, we want to produce a hypothesis $\tilde{H} \in \mathcal{H}$ such that $l(\tilde{H}) \le \inf_{H\in\mathcal{H}} l(H) + \epsilon$.

Empirical risk minimization

We have **training data** $X_1, \ldots, X_n$ from which we compute the **empirical risk**: $l_n(H) = \frac{1}{n}\sum_{i=1}^n l(H, X_i)$ of a hypothesis H. For $n \to \infty$, this converges to $l(H)$. Formally the **weak law of large numbers** states, that for $H \in \mathcal{H}$ and $\delta, \epsilon > 0$ we have n_0 such that for $n_0 \ge n$ we have $|l_n(H) - l(H)| \le \epsilon$ with probability at least $1 - \delta$.
We use **empirical risk minimization** as a proxy for risk minimization: For $n \in \mathbb{N}$ and $X_1, \ldots, X_n \sim \mathcal{X}$ produce a hypothesis $\tilde{H}_n$ such that $l_n(\tilde{H}_n) \le \inf_{H\in\mathcal{H}} l_n(H) + \epsilon$.
Careful: Empirical risk does *not* always converge to expected risk(!).

The map of learning



Algorithm: computational procedure according to which a hypothesis H_n is obtained from training data $X_1, \ldots, X_n$.
Validation: Once a hypothesis H_n has been obtained through training, one also needs to assess its expected risk, i.e. locate H_n in the l-dimension. Using the weak law of large numbers, $l(H_n)$ can be estimated via its empirical risk on test data—fresh samples from $\mathcal{X}$ that the algorithm has not seen.
Overfitting: If our learning algorithm returns a hypothesis with low empirical risk but high expected risk, we have a case of overfitting. The main cause of overfitting is that our theory (hypothesis class $\mathcal{H}$ and loss function l) is so complex that it allows us to almost perfectly explain any training data.
Underfitting: If the learning algorithm returns a hypothesis with high empirical risk, we cannot even explain the training data. In this case, there is no justified hope to be able to explain unseen data. The main cause of underfittting is that our theory is too simple to capture the nature of the data.
Learning: If both empirical and expected risk are low, we can make a case that we have learned something.
Generalization: Ideally, the expected risk is close to the empirical risk, and if this happens, we have generalization. This means that the hypothesis explains unseen data equally well as the training data. But it does *not* mean that the explanation is good.
Regularization: In the case that overfitting is observed, a possible remedy is to add a regularization term r to the loss function l with the goal of *punishing* complex hypotheses. Empirically minimizing $l' = l + \lambda r$ for a real number $\lambda > 0$ therefore has the effect that we introduce a **bias**, meaning that we deviate more and more from our theory, with the effect that the empirical risk increases. But as the intended consequence, the **variance** (sensitivity to the training data) decreases, and this may reduce the expected risk.

Worst-case versus average-case complexity

The classical measure of algorithm performance is its **worst-case complexity**, the function that maps n to the maximum runtime of the algorithm over all possible inputs of size n. The **average case complexity** is the function that maps n to the expected runtime of the algorithm, taken over its input distribution.

The estimation-optimization tradeoff

As we inevitably lose precision in going from empirical to expected risk, it doesn't help to optimize the empirical risk to a significantly higher precision. Let us call the precision that we lose in going from empirical to expected risk the **estimation error**; the precision we lose in finding only an almost best explanation of the training data is the **optimization error**. In small-scale learning, it doesn't hurt to go for as small an optimization error as we can. But in large-scale learning, we may need to give up on some optimization precision in order to be able to stay within the optimization time budget. The **estimation-optimization tradeoff** consists in finding the most efficient way of spending the resources under the given constraints.

2 Theory of Convex Functions

Mathematical Background

Cauchy-Schwarz Inequality: $\left|\mathbf{u}^T\mathbf{v}\right| \le \|\mathbf{u}\|\|\mathbf{v}\|$. We have equality if and only if $\mathbf{u}$ and $\mathbf{v}$ are colinear.
Cosine Theorem: $2\mathbf{v}^T\mathbf{w} = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 - \|\mathbf{v} - \mathbf{w}\|^2$
The **spectral norm** of a matrix A is:
$\|A\| := \max_{\mathbf{v}\in\mathbb{R}^d, \mathbf{v}\neq 0} \frac{\|A\mathbf{v}\|_2}{\mathbf{v}}_2 = \max_{\|\mathbf{v}\|_2=1} \|A\mathbf{v}\|_2$.
It follows that $\|A\mathbf{v}\|_2 \le \|A\|\|\mathbf{v}\|_2$ for all $\mathbf{v}$.
Mean value theorem: Let $a < b$ and $h : [a, b] \to \mathbb{R}$ be a continuous function which is differentiable on (a, b). Then there exists a $c \in (a, b)$ such that: $h'(c) = \frac{h(b)-h(a)}{b-a}$.

Let $f : \mathbf{dom}(f) \to \mathbb{R}^m$ where $\mathbf{dom}(f) \subset \mathbb{R}^d$. The function f is called **differentiable** at $\mathbf{x}$ if there exists a $(m \times d)$-matrix A and an error function $r : \mathbb{R}^d \to \mathbb{R}^m$ defined in some neighborhood of $\mathbf{0} \in \mathbb{R}^d$ such that for all $\mathbf{y}$ in a neighborhood of $\mathbf{x}$:
$f(\mathbf{y}) = f(\mathbf{x}) + A(\mathbf{y} - \mathbf{x}) + r(\mathbf{y} - \mathbf{x})$ where $\lim_{\mathbf{v}\to 0} \frac{\|r(\mathbf{v})\|}{\|v\|} = 0$. A is unique and called the **jacobian** of f at $\mathbf{x}$. We denote it $Df(x)$ and have $Df(\mathbf{x})_{i,j} = \frac{\partial f_i}{\partial x_j}(\mathbf{x})$.
For $m = 1$ (i.e $f : \mathbb{R}^d \to \mathbb{R}$) we call the jacobian the **gradient** of f and denote it ∇f^T.
Geometrically, this means that the graph of the affine function $f(x) + \nabla f(\mathbf{x})^T$ is a tangent hyperplane to the graph of f at $(\mathbf{x}, f(\mathbf{x}))$.
Chain Rule: $D(f \circ g)(\mathbf{x}) = Df(g(\mathbf{x}))Dg(\mathbf{x})$.

Convex Sets

A set $C \subset \mathbb{R}^d$ is **convex** if for any two points $\mathbf{x}, \mathbf{y} \in C$ the connecting line segment is in C. In formula this means for all $\lambda \in [0, 1]$: $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in C$.
Intersection of convex sets: Let C_i, $i \in I$ be convex sets, where I is a (possibly infinite) index set. Then $C = \cap_{i\in I} C_i$ is a convex set.

Mean value inequality: Let $f : \mathbf{dom}(f) \to \mathbb{R}^m$ be differentiable, $X \subset \mathbf{dom}(f)$ a convex, nonempty and open set, $B > 0$. The following are equivalent:
(i) f is B-Lipschitz: $\forall \mathbf{x}, \mathbf{y} \in X : \|f(\mathbf{x}) - f(\mathbf{y})\| \le B\|\mathbf{x} - \mathbf{y}\|$
(ii) f has differentials bounded by B (in spectral norm): $\forall \mathbf{x} \in X : \|Df(\mathbf{x})\| \le B$.
(ii) $\implies$ (i) even if X is not open.

Convex functions

A function f as above is said to be **convex** if $\mathbf{dom}(f)$ is convex and for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$ and $\lambda \in [0, 1]$ we have: $f(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) \le \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$.

While the graph of f is the set $\{(\mathbf{x}, f(\mathbf{x})) \in \mathbb{R}^{d+1} : \mathbf{x} \in \mathbf{dom}(f)\}$, the **epigraph** is the set of points above the graph, $\mathbf{epi}(f) := \{(\mathbf{x}, \alpha) \in \mathbb{R}^{d+1} : \mathbf{x} \in \mathbf{dom}(f), \alpha \ge f(\mathbf{x})\}$. f is a convex function if and only if $\mathbf{epi}(f)$ is a convex set.
Jensen's inequality Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function, $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbf{dom}(f)$ and $\lambda_1, \ldots, \lambda_m \in \mathbb{R}_+$ such that $\sum_{i=1}^n \lambda_i = 1$, then: $f\left(\sum_{i=1}^m \lambda_i x\mathbf{x}_i\right) \le \sum_{i=1}^n \lambda_i f(\mathbf{x}_i)$.
If f is convex and $\mathbf{dom}(f)$ is open then f is continuous.

First-order characterization of convexity Let $\mathbf{dom}(f)$ be open and f differentiable then f is convex if and only if $\mathbf{dom}(f)$ is convex and $f(\mathbf{y}) \ge f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$. Geometrically, this means that for all $\mathbf{x} \in \mathbf{dom}(f)$, the graph of f lies above its tangent hyperplane at the point $(\mathbf{x}, f(\mathbf{x}))$.

Monotonicity of the gradient: Suppose that $\mathbf{dom}(f)$ is open and that f is differentiable. Then f is convex if and only if $\mathbf{dom}(f)$ is convex and $(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))^T(\mathbf{y} - \mathbf{x}) \ge 0$ holds for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$.

Second-order characterization of convexity Suppose that $\mathbf{dom}(f)$ is open and that f is twice differentiable; in particular, the Hessian (matrix of second partial derivatives) 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 $\nabla^2 f(\mathbf{x}) \succcurlyeq 0$ (psd).

Operations that preserve convexity: Let $f_1, \ldots, f_n$ be convex functions and $\lambda_1, \ldots, \lambda_n \in \mathbb{R}_+$. Then $f := \max_{i=1}^m f_i$ and $f := \sum_{i=1}^n \lambda_i f_i$ are convex on $\mathbf{dom}(f) = \cap_{i=1}^n \mathbf{dom}(f_i)$. Furthermore let f be convex with $\mathbf{dom}(f) \subset \mathbb{R}^d$ and $g : \mathbb{R}^m \to \mathbb{R}^d$ be an affine function (i.e. $g(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$. Then $f \circ g$ is convex on $\mathbf{dom}(f \circ g) = \{\mathbf{x} \in \mathbb{R}^m : g(\mathbf{x}) \in \mathbf{dom}(f)\}$.

Minimizing convex functions

A **local minimum** of $f : \mathbf{dom}(f) \to \mathbb{R}$ is a point $\mathbf{x}$ such that there exists $\epsilon > 0$ with: $f(\mathbf{x}) \le f(\mathbf{y})$ for all $\mathbf{y} \in \mathbf{dom}(f)$ with $\|\mathbf{y} - \mathbf{x}\| < \epsilon$.
Let x^* be a local minimum of a convex function $f : \mathbf{dom}(f) \to \mathbb{R}$. Then x^* is a global minimum, meaning that $f(x^*) \le f(y) \forall y \in \mathbf{dom}(f)$.
Suppose that $f : \mathbf{dom}(f) \to \mathbb{R}$ is convex and differentiable over an open domain $\mathbf{dom}(f) \subset \mathbb{R}^d$. Let $x \in \mathbf{dom}(f)$. If $\nabla f(\mathbf{x}) = 0$, then $\mathbf{x}$ is a global minimum.
Suppose that $f : \mathbf{dom}(f) \to \mathbb{R}$ is differentiable over an open domain $\mathbf{dom}(f) \subset \mathbb{R}^d$. Let $\mathbf{x} \in \mathbf{dom}(f)$. If $\mathbf{x}$ is a global minimum then $\nabla f(\mathbf{x}) = 0$.

Strictly convex functions

A function $f : \mathbf{dom}(f) \to \mathbb{R}$ is **strictly convex** if $\mathbf{dom}(f)$ is convex and for all $\mathbf{x} \neq \mathbf{y} \in \mathbf{dom}(f)$ and all $\lambda \in (0, 1)$, we have $f(\lambda\mathbf{x} + (1 - \lambda)\mathbf{y}) < \lambda f(\mathbf{x}) + (1 - \lambda)f(\mathbf{y})$.
Suppose that $\mathbf{dom}(f)$ is open and that f is twice continuously differentiable. If the Hessian $\nabla^2 f(\mathbf{x}) \succ 0$ for every $\mathbf{x} \in \mathbf{dom}(f)$, then f is strictly convex.
Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be strictly convex then f has at most one global minimum.

Constrained Minimization

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex and $X \subset \mathbf{dom}(f)$ be a convex set. A point $\mathbf{x}$ is a **minimizer** of f over X if $f(\mathbf{x}) \le f(\mathbf{y}), \forall \mathbf{y} \in X$.
Suppose that $f : \mathbf{dom}(f) \to \mathbb{R}$ is convex and differentiable over an open domain $\mathbf{dom}(f) \subset \mathbb{R}^d$, and let $X \subset \mathbf{dom}(f)$ be a convex set. A point $\mathbf{x}^* \in X$ is a minimizer if and only if $\forall \mathbf{x} \in X : \nabla f(\mathbf{x}^*)^T(\mathbf{x} - \mathbf{x}^*) \ge 0$.

Existence of a minimizer

Let $f : \mathbb{R}^d \to \mathbb{R}$ and $\alpha \in \mathbb{R}$. The set $f^{\le\alpha} := \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) \le \alpha\}$ is the α**-sublevel set** of f.
Weierstrass Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function and suppose there is a non-empty and bounded sublevel set $f^{\le\alpha}$. Then f has a global minimum.

Convex programming

An **optimization problem** in standard form is given by:

$$\begin{aligned} &\text{minimize } f_0(\mathbf{x}) \\ &\text{subject to } f_i(\mathbf{x}) \le 0, i = 1, \ldots m \\ &\qquad\qquad h_i(\mathbf{x}) = 0 i = 1, \ldots, p \end{aligned}$$

A **convex program** arises when the f_i are convex functions and the h_i are affine functions with domain $\mathbb{R}^d$. We call the region
$X = \{\mathbf{x} \in \mathbb{R}^d : f_i(\mathbf{x}) \le 0, i = 1, \ldots, m; h_i(\mathbf{x}) = 0, i = 1, \ldots p\}$ the **feasible region**, in this case it is a convex set.
The **Lagrangian** is the functional $L : \mathcal{D} \times \mathbb{R}^m \to \mathbb{R}$ given by:
$L(\mathbf{x}, \lambda, \nu) = f_0(\mathbf{x}) + \sum_{i=1}^m \lambda_i f_i(\mathbf{x}) + \sum_{i=1}^p \nu_i h_i(\mathbf{x})$. We call the λ_i, ν_i are called **Lagrange multipliers**. The Lagrange **dual function** is $g : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R} \cup \{-\infty\}$ defined by $g(\lambda, \nu) = \inf_{\mathbf{x}\in D} L(\mathbf{x}, \lambda, \nu)$.
Weak duality If $\mathbf{x}$ is a feasible solution then $g(\lambda, \nu) \le f_0(\mathbf{x})$, for all $\lambda \in \mathbb{R}^m, \nu \in \mathbb{R}^p$ with $\lambda \ge 0$.
The **Lagrangian dual problem** is given by:

$$\text{maximize } g(\lambda, \nu) \qquad\qquad \text{subject to } \lambda \ge 0$$

The equivalent minimization is always a convex program (even if the original program was not).
Slaters Condition: If there is a Slater's point $\tilde{\mathbf{x}}$ (a point which satisfies all inequality constraints of the original program strictly), then the infimum value of the primal equals the supremum value of its Lagrange dual. Moreover, if this value is finite, it is attained by a feasible solution of the dual. This is called **strong duality**.
Karush-Kuhn-Tucker necessary conditions Let $\tilde{\mathbf{x}}$ and $(\tilde{\lambda}, \tilde{\nu})$ be feasible solutions of the primal optimization problem and its Lagrangian dual repectively with zero duality gap. If all f_i and h_i are differentiable then: $\tilde{\lambda}_i f_i(\tilde{\mathbf{x}}) = 0$ for $i = 1, \ldots, m$ and $\nabla f_0(\tilde{\mathbf{x}}) + \sum_{i=1}^m \tilde{\lambda}_i \nabla f_i(\tilde{\mathbf{x}}) + \sum_{i=1}^p \tilde{\nu}_i \nabla h_i(\tilde{\mathbf{x}}) = 0$.
Karush-Kuhn-Tucker sufficient conditions The KKT necessary conditions are sufficient to ensure strong duality if all f_i, h_i are differentiable and the f_i are convex and the h_i affine.

3 Gradient Descent

Overview

Number of steps is given which the respective variant needs on the respective function class to achieve additive approximation error at most ϵ.

	Lipschitz convex functions	smooth convex functions	strongly convex functions	smooth and strongly convex functions
gradient descent	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon)$		$\mathcal{O}(\log(1/\epsilon))$
accelerated gradient descent		$\mathcal{O}(1/\sqrt{\epsilon})$		
projected gradient descent	$\mathcal{O}(1/\epsilon^2)$	$\mathcal{O}(1/\epsilon)$		$\mathcal{O}(\log(1/\epsilon)$
subgradient descent	$\mathcal{O}(1/\epsilon^2)$		$\mathcal{O}(1/\epsilon)$	
stochastic gradient descent	$\mathcal{O}(1/\epsilon^2)$		$\mathcal{O}(1/\epsilon)$	

Vanilla Gradient Descent

Each step of **gradient descent** is defined as: $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma\nabla f(\mathbf{x}_t)$. Where γ is a fixed **stepsize**.

The **vanilla analysis** of gradient descent yields with $\mathbf{g}_t = \nabla f(\mathbf{x}_t)$:
$\sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) \le \frac{\gamma}{2}\sum_{t=0}^{T-1}\|\mathbf{g}_t\|^2 + \frac{1}{2\gamma}\|\mathbf{x}_0 - \mathbf{x}^*\|^2$. Trick: use cosine theorem and first order characterisation of convexity.

Lipschitz convex functions

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^*$. Suppose that $\|\mathbf{x}_0 - \mathbf{x}^*\| \le R$ and $\|\nabla f(\mathbf{x})\| \le B$ for all $\mathbf{x}$.
With stepsize $\gamma = \frac{R}{B\sqrt{T}}$ we get: $\frac{1}{T}\sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}^*)) \le \frac{RB}{\sqrt{T}}$.

Smooth convex functions

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be a differentiable function and $X \subset \mathbf{dom}(f)$ be convex and $L \in \mathbb{R}_+$. f is called **smooth** over X if:

$$f(\mathbf{y}) \le f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2 \qquad \forall \mathbf{x}, \mathbf{y} \in X$$

Alternative characterisation of smoothness: Suppose that $\mathbf{dom}(f)$ is open and convex and that f is differentiable then the following are equivalent:
(i) f is smooth with parameter L.
(ii) $g(\mathbf{x}) = \frac{L}{2}\mathbf{x}^T\mathbf{x} - f(\mathbf{x})$ is convex over $\mathbf{dom}(g) = \mathbf{dom}(f)$.

Suppose that $f : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable, then the following two statements are equivalent:
(i) f is smooth with parameter L.
(ii) $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.
Operations that preserve convexity Let $f_1, \ldots, f_n$ be smooth with parameters $L_1, \ldots, L_n$ and let $\lambda_1, \ldots, \lambda_n \in \mathbb{R}_+$. Then $f := \sum_{i=1}^n \lambda_i f_i$ is smooth with parameter $\sum_{i=1}^n \lambda_i L_i$. Furthermore if $f : \mathbf{dom}(f) \to \mathbb{R}$ is smooth with parameter L and $g : \mathbb{R}^m \to \mathbb{R}^d$ is an affine function $g(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$, then $f \circ g$ is smooth with parameter $L\|A\|^2$.
Sufficient decrease Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and smooth with parameter L, and $\gamma = \frac{1}{L}$, then gradient descent satisfies: $f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2$.
Theorem: The above yields $f(\mathbf{x}_T) - f(\mathbf{x}^*) \le \frac{L}{2T}\|\mathbf{x}_0 - \mathbf{x}^*\|^2$. Trick use vanilla analysis to bound the sum of g_t.

Accelerated Gradient Descent

Choose $\mathbf{z}_0 = \mathbf{y}_0 = \mathbf{x}_0$ arbitrary and: $\mathbf{y}_{t+1} = \mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t)$, $\mathbf{z}_{t+1} = z_t - \frac{t+1}{2L}\nabla f(\mathbf{x}_t)$ and $\mathbf{x}_{t+1} = \frac{t+1}{t+3}\mathbf{y}_{t+1} + \frac{2}{t+3}\mathbf{z}_{t+1}$. Idea: y_t is a normal "smooth" step and z_t is a more aggressive step. We perform a weighted average of these two steps.

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^*$; furthermore suppose that f is smooth with parameter L. Accelerated gradient descent yields: $f(\mathbf{y}_T) - f(\mathbf{x}^*) \le \frac{2L}{T(T+1)}\|\mathbf{z}_0 - \mathbf{x}^*\|$. Trick: define a potential function $\Phi(t) = t(t + 1)(f(\mathbf{y}_t) - f(\mathbf{x}^*)) + 2L\|\mathbf{z}_t - \mathbf{x}^*\|^2$ and show that it is decreasing.

Strongly convex functions

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be a convex and differentiable function, $X \subset \mathbf{dom}(f)$ convex and $\mu > 0$. f is called **strongly convex** with parameter μ over X if:

$$f(\mathbf{y}) \ge f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2$$

Alternative characterisation of strong convexity: Suppose that $\mathbf{dom}(f)$ is open and convex and that f is differentiable then the following are equivalent:
(i) f is strongly convex with parameter μ.
(ii) $g(\mathbf{x}) = f(\mathbf{x}) - \frac{\mu}{2}\mathbf{x}^T\mathbf{x}$ is convex over $\mathbf{dom}(g) = \mathbf{dom}(f)$.

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable. Suppose that f is smooth with parameter L and strongly convex with parameter μ. Choose $\gamma = \frac{1}{L}$, then gradient descent satisfies:

(i) Squared distances to $\mathbf{x}^*$ are geometrically decreasing
$\left\|\mathbf{x}_{t+1} - \mathbf{x}^*\right\|^2 \leq \left(1 - \frac{\mu}{L}\right) \|\mathbf{x}_t - \mathbf{x}^*\|^2$

(ii) The absolute error after T iterations is exponentially small in T:
$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \frac{L}{2} \left(1 - \frac{\mu}{L}\right)^T \|x_0 - x^*\|^2$

Trick you can show using the vanilla analysis and the lower bound for g_t from strong convexity that: $\left\|\mathbf{x}_{t+1} - \mathbf{x}^*\right\|^2 \leq 2\gamma(f(\mathbf{x}^*) - f(\mathbf{x}_t)) + \gamma^2 \|\nabla f(\mathbf{x}_t)\|^2 + (1 - \mu\gamma)\|\mathbf{x}_t - \mathbf{x}^*\|^2$, then use sufficient decrease.

4 Projected Gradient Descent

Algorithm

Goal: Minimize a function f over a *closed convex* subset $X \subset \mathbb{R}^d$.

Projected gradient descent: Choose $\mathbf{x}_0 \in X$ arbitrary and define: $\mathbf{y}_{t+1} = \mathbf{x}_t - \gamma \nabla f(\mathbf{x}_t)$ and $\mathbf{x}_{t+1} := \Pi_X(\mathbf{y}_{t+1}) := \text{argmin}_{\mathbf{x} \in X} \left\|\mathbf{x} - \mathbf{y}_{t+1}\right\|^2$.

Projected gradient descent requires the same number of steps as gradient descent but projected gradient descent requires a nontrivial primitive to be solved in each step (projection onto the feasible region)

Useful fact: Let $X \subset \mathbb{R}^d$ be closed and convex and $\mathbf{x} \in X$, $\mathbf{y} \in \mathbb{R}^d$, then:

(i) $(\mathbf{x} - \Pi_X(\mathbf{y}))^T(\mathbf{y} - \Pi_X(\mathbf{y})) \leq 0$

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

All the results from which we proved in the previous chapter still hold as long as the function is smooth/strongly convex over X.

Projecting onto l_1-balls

Theorem: Let $\mathbf{v} \in \mathbb{R}^d$ and $R \in \mathbb{R}_+$, $X = B_1(R)$ the l_1-ball around 0 of radius R. The projection $\Pi_X(\mathbf{v}) = \text{argmin}_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{v}\|^2$ of $\mathbf{v}$ onto $B_1(R)$ can be computed in time $\mathcal{O}(d \log d)$.

5 Coordinate Descent

In large-scale learning, an issue with the gradient descent algorithms is that in every iteration, we need to compute the full gradient $\nabla f(\mathbf{x}_t)$ in order to obtain the next iterate $\mathbf{x}_{t+1}$. If the number of variables d is large, this can be very costly. The idea of coordinate descent is to update only one coordinate of $\mathbf{x}_t$ at a time, and to do this, we only need to compute one coordinate of $\nabla f(\mathbf{x}_t)$ (one partial derivative). We expect this to be by a factor of d faster than computation of the full gradient and update of the full iterate.

Polyak-Łojasiewicz inequality (PL-Inequality)

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable function with global minimum $\mathbf{x}^*$. We say that f satisfies the **Polyak-Łojasiewicz inequality** if the following holds for some $\mu > 0$: $\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \geq \mu(f(\mathbf{x}) - f(\mathbf{x}^*))$ for all $\mathbf{x} \in \mathbb{R}^d$.

Strong Convexity $\implies$ PL inequality: Let $f : \mathbb{R}^d \to \mathbb{R}$ be a differentiable and strongly convex with parameter $\mu > 0$, then f satisfies the PL-Inequality for the same μ. The opposite is *not* true.

We can use the PL-Inequality to repeat the analysis of gradient descent. We get:

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\mathbf{x}^*$. Suppose that f is smooth with parameter L and satisfies the PL-Inequality with parameter $\mu > 0$, then choosing stepsize $\gamma = \frac{1}{L}$, gradient descent satisfies:

$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \left(1 - \frac{\mu}{L}\right)^T (f(\mathbf{x}_0) - f(\mathbf{x}^*))$. Trick: Start with sufficient decrease.

Coordinate descent algorithms

Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and $\mathcal{L} = (L_1, \ldots, L_d) \in \mathbb{R}^d_+$. f is called **coordinate-wise smooth** (with parameter $\mathcal{L}$) if for every coordinate $i = 1, \ldots, d$: $f(\mathbf{x} + \lambda\mathbf{e}_i) \leq f(\mathbf{x}) + \lambda\nabla_i f(\mathbf{x}) + \frac{L_i}{2}\lambda^2$, $\forall \mathbf{x} \in \mathbb{R}^d, \lambda \in \mathbb{R}_+$.

Coordinate descent algorithms first choose an active coordinate $i \in [d]$, and then do the following: $\mathbf{x}_{t+1} = \mathbf{x}_t - \lambda_i \mathbf{e}_i$. We often use a gradient based stepsize: $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_i \nabla_i f(\mathbf{x}_t)\mathbf{e}_i$.

Lemma: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and coordinate-wise smooth with parameter $\mathcal{L}$. With active coordinate i in iteration t and stepsize $\gamma_i = \frac{1}{L_i}$ coordinate descent satisfies:

$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \frac{1}{2L_i}|\nabla_i f(\mathbf{x}_t)|^2$. **Randomized coordinate descent**: The active coordinate is choosen uniformely at random from the set $[d]$. It is at least as fast as gradient descent on smooth functions, and if we assume the PL-inequality we get:

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\mathbf{x}^*$. Suppose that f is coordinate-wise smooth with parameter L and satisfies the PL-Inequality with parameter $\mu > 0$. Choosing stepsize $\gamma_i = \frac{1}{L}$ randomized coordinate descent satisfies:

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

Importance Sampling We choose the active coordinate as follows: sample $i \in [d]$ with probability $\frac{L_i}{\sum_{j=1}^d L_j}$.

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\mathbf{x}^*$. Suppose that f is coordinate-wise smooth with parameter $\mathcal{L}$ and satisfies the PL-Inequality with parameter $\mu > 0$. Let $\bar{L} = \frac{1}{d}\sum_{i=1}^d L_i$, be the average of the smoothness constants. Then importance sampling coordinate descent with $\gamma_i = \frac{1}{L_i}$ satisfies:

$\mathbb{E}[f(\mathbf{x}_T) - f(\mathbf{x}^*)] \leq \left(1 - \frac{\mu}{d\bar{L}}\right)^T (f(\mathbf{x}_0) - f(\mathbf{x}^*))$.

Steepest coordinate descent: We choose the active coordinate: $i = \text{argmax}_{i \in [d]}|\nabla_i f(\mathbf{x}_t)|$.

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\mathbf{x}^*$. Suppose that f is coordinate-wise smooth with parameter L and satisfies the PL-Inequality with parameter $\mu > 0$. Choosing stepsize $\gamma_i = \frac{1}{L}$ steepest coordinate descent satisfies:

$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \left(1 - \frac{\mu}{dL}\right)^T (f(\mathbf{x}_0) - f(\mathbf{x}^*))$.

Strong convexity with respect to l_1-norm A function is strongly convex with respect to the l_1-norm if: $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \frac{\mu_1}{2}\|\mathbf{y} - \mathbf{x}\|_1^2$. Note that: $\sqrt{d} \cdot \|\mathbf{y} - \mathbf{x}\|_2 \geq \|\mathbf{y} - \mathbf{x}\|_1 \geq \|\mathbf{y} - \mathbf{x}\|_2$. Hence the function is also strongly convex in the classical sense.

Lemma: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable and strongly convex with parameter μ_1 w.r.t. l_1-norm. Then f satisfies the PL-Inequality w.r.t. l_∞-norm with the same μ_1: $\frac{1}{2}\|\nabla f(\mathbf{x})\|_\infty^2 \geq \mu_1(f(\mathbf{x}) - f(\mathbf{x}^*))$.

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with a global minimum $\mathbf{x}^*$. Suppose that f is coordinate-wise smooth with parameter L and satisfies the l_1 PL-Inequality with parameter $\mu_1 > 0$. Choosing stepsize $\gamma_i = \frac{1}{L}$ steepest coordinate descent satisfies:

$f(\mathbf{x}_T) - f(\mathbf{x}^*) \leq \left(1 - \frac{\mu_1}{L}\right)^T (f(\mathbf{x}_0) - f(\mathbf{x}^*))$.

Greedy coordinate descent: We do not require f to be differentiable. In each iteration, we make the step that maximizes the progress in the chosen coordinate. This requires to perform a line search by solving a 1-dimensional optimization problem: choose $i \in [d]$ and set $\mathbf{x}_{t+1} = \text{argmin}_{\lambda \in \mathbb{R}} f(\mathbf{x}_t + \lambda\mathbf{e}_i)$. There are cases where the line search can exactly be done analytically, or approximately by some other means. In the differentiable case, we can take any of the previously studied coordinate descent variants and replace some of its steps by greedy steps if it turns out that we can perform line search along the selected coordinate.

Let $f : \mathbb{R}^d \to \mathbb{R}$ be of the form $f(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x})$, with $h(\mathbf{x})) \sum_i h_i(\mathbf{x})$, g convex and differentiable and all h_i convex. We call such a function **separable**. Greedy coordinate descent will always make progress for such a function. This is relevant for the LASSO Regression.

Summary

Algorithm	PL norm	Smoothness	Bound
Randomized	l_2	L	$1 - \frac{\mu}{dL}$
Importance sampling	l_2	$(L_1, \ldots, L_d)$	$1 - \frac{\mu}{dL}$
Steepest	l_2	L	$1 - \frac{\mu}{dL}$
Steeper (than Steepest)	l_1	L	$1 - \frac{\mu_1}{L}$

6 Nonconvex Functions

Smooth functions

A function f is called **concave** if $-f$ is convex. Every concave function is smooth with parameter $L = 0$.

Alternative caracterisation of smoothness: Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be twice differentiable, with $X \subset \mathbb{R}^d$ a convex set and $\left\|\nabla^2 f(\mathbf{x})\right\| \leq L$ for all $\mathbf{x} \in X$, then f is smooth with parameter L over X. **Converse**: If f is smooth over an *open* convex subset $X \subset \mathbf{dom}(f)$, it has bounded Hessians over X.

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be differentiable with global minimum $\mathbf{x}^*$, furthermore suppose that f is smooth with parameter L. Choosing stepsize $\gamma = \frac{1}{L}$ gradient descent will yield: $\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2L}{T}(f(\mathbf{x}_0) - f(\mathbf{x}^*))$. In particular $\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2L}{T}(f(\mathbf{x}_0) - f(\mathbf{x}^*))$ for some $t \in \{0, \ldots, T - 1\}$, and $\lim_{t\to\infty} \|\nabla f(\mathbf{x}_t)\|^2 = 0$. Trick: Use sufficient decrease.

7 Frank-Wolfe Algorithm

Goal: Solve problems of the form: minimize $f(\mathbf{x})$, subject to $\mathbf{x} \in X$.

Problem: projections onto a set X can sometimes be very complex to compute, even in cases when the set is convex. Would it still be possible to solve constrained optimization problems using a gradient-based algorithm, but without any projection steps?

The Algorithm

Linear minimization oracle: For the feasible region $X \subset \mathbb{R}^d$ and an arbitrary vector $\mathbf{g} \in \mathbb{R}^d$ (which we can think of as an optimization direction, we define $\text{LMO}_X(\mathbf{g}) := \text{argmin}_{\mathbf{z} \in X} \mathbf{g}^T\mathbf{z}$.

The **Frank-Wolfe algorithm** proceeds iteratively, starting from an initial feasible point $\mathbf{x}_0 \in X$, using a (time-dependent) stepsize $\gamma_t \in [0, 1]$.

$$\mathbf{s} := \text{LMO}_X(\nabla f(\mathbf{x}_t)) \qquad\qquad \mathbf{x}_{t+1} := (1 - \gamma_t)\mathbf{x}_t + \gamma_t\mathbf{s}$$

The algorithm reduces non-linear constrained optimization to linear optimization over the same set X: It is able to solve general non-linear constrained optimization problems, by only solving a simpler linear constrained optimization over the same set X in each iteration — that is the call to the linear minimization oracle LMO_X.

Nice properties: *(i)* Iterates are always feasible, if the constraint set X is convex. In other words, $\mathbf{x}_0, \ldots, \mathbf{x}_t \in X$. *(ii)* The algorithm is projection-free. Depending on the geometry of the constraint set X, the subproblem LMO_X is often easier to solve than a projection onto the same set X. Intuitively, this the case because LMO_X is only a linear problem, while a projection operation is a quadratic optimization problem. *(iii)* The iterates always have a simple sparse representation: $\mathbf{x}_t$ is always a convex combination of the initial iterate and the minimizers $\mathbf{s}$ used so far.

Linear minimization oracles

The algorithm is particularly useful for cases when the constraint set X can be described as a convex hull of a finite or otherwise “nice” set of points $\mathcal{A}$, formally $\text{conv}(A) = X$. We call $\mathcal{A}$ the **atoms** describing the constraint set. In this case a solution to the linear subproblem LMO_X is always attained by an atom $\mathbf{a} \in \mathcal{A}$. This is because every $\mathbf{s} \in \text{conv}(X)$ is a convex combination $\mathbf{s} = \sum_{i=1}^n \lambda_i\mathbf{a}_i$ of finitely many atoms ($\sum_{i=1}^n \lambda_i = 1$, all λ_i non-negative). It follows that for every $\mathbf{g}$ there is an atom such that $\mathbf{g}^t\mathbf{s} \geq \mathbf{g}^T\mathbf{a}_i$. Hence, if $\mathbf{s}$ minimizes $\mathbf{g}^T\mathbf{z}$, then there is also an atomic minimizer.

The “optimal” set of atoms is the set of **extreme points**. A point $\mathbf{x} \in X$ is extreme if $\mathbf{x} \notin \text{conv}(X \setminus \{\mathbf{x}\})$. Such an extreme point must be in every set of atoms, but not every atom must be extreme. All that we require for A to be a set of atoms is that $\text{conv}(\mathcal{A}) = X$.

We define the **LASSO-Problem** in its standard (primal) form as: $\min_{\mathbf{x} \in \mathbb{R}^d} \|A\mathbf{x} - \mathbf{b}\|^2$ subject to $\|\mathbf{x}\|_1 \leq 1$. Here we observe that the constraint set $X = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_1 \leq 1\}$ is the unit l_1-ball, the convex hull of the unit basis vectors: $X = \text{conv}(\{\pm\mathbf{e}_1, \ldots, \pm\mathbf{e}_d\})$. Linear problems over the unit l_1-ball are easy to solve: For any direction $\mathbf{g}$, the minimizer can be chosen as one of the atoms (the unit basis vectors and their negatives): $\text{LMO}_X(\mathbf{g}) = -\text{sgn}(g_i)\mathbf{e}_i$ with $i := \text{argmax}_{i \in [d]}|g_i|$.

Duality gap

Given $\mathbf{x} \in X$ we define the **duality gap** (also known as Hearn Gap) at $\mathbf{x}$ as:

$$g(\mathbf{x}) := \nabla f(\mathbf{x})^T(\mathbf{x} - \mathbf{s}) \qquad \text{for} \qquad \mathbf{s} := \text{LMO}_X(\nabla f(\mathbf{x}))$$

Suppose that the constrained minimization problem has a minimizer $\mathbf{x}^*$. Let $\mathbf{x} \in X$ then $g(\mathbf{x}) \geq f(\mathbf{x}) - f(\mathbf{x}^*)$ meaning that the duality gap is an upper bound for the optimality gap.

Note that we always have $g(\mathbf{x}) \geq 0$.

Convergence

Assumptions: We need to assume that the function f is smooth, but unlike for gradient descent, the stepsize can be chosen independently from the smoothness parameter.

For a closed and bounded set X we define the **diameter** of X as $\text{diam}(X) = \max_{\mathbf{x},\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|$.

Convergence result: Consider the constrained minimization problem where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and smooth with parameter L, and X is convex, closed and bounded (in particular, a minimizer $\mathbf{x}^*$ of f over X exists, and all linear minimization oracles have minimizers). With any $\mathbf{x}_0 \in X$, and with stepsizes $\gamma_t = \frac{2}{t+2}$, the Frank-Wolfe algorithm yields:

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

The proof uses that for a step $\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t(\mathbf{s} - \mathbf{x}_t)$ with stepsize $\gamma_t \in [0, 1]$ it holds that: $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \gamma_t g(\mathbf{x}_t) + \gamma_t^2 \frac{L}{2}\|\mathbf{s} - \mathbf{x}_t\|^2$, where $\mathbf{s} = \text{LMO}_X(\nabla f(\mathbf{x}_t))$. Then use duality gap and induction.

The same proof idea also holds for other stepsizes:

Line search stepsize: Here, $\gamma_t \in [0, 1]$ is chosen such that the progress in f-value (and hence also in h-value) is maximized: $\gamma_t := \text{argmin}_{\gamma \in [0,1]} f((1 - \gamma)\mathbf{x}_t + \gamma\mathbf{s})$. If $\mathbf{y}_{t+1}$ is the iterate obtained with standard stepsize μ_t then we get: $h(\mathbf{x}_{t+1}) \leq h(\mathbf{y}_{t+1}) \leq (1 - \mu_t)h(\mathbf{x}_t) + \mu_t^2 \frac{L}{2}\text{diam}(X)$.

Gap-based stepsize: We choose $\gamma_t = \min\left(\frac{g(\mathbf{x}_t)}{L\|\mathbf{s}-\mathbf{x}_t\|^2}, 1\right)$, this yields: $h(\mathbf{x}_{t+1}) \leq (1 - \mu_t)h(\mathbf{x}_t) + \mu_t^2 \frac{L}{2}\text{diam}(X)$.

Affine invariance

We call two problems (f, X) and (f', X') **affinely equivalent** if $f'(\mathbf{x}) = f(A\mathbf{x} + \mathbf{b})$ for some invertable matrix A and some vector $\mathbf{b}$ and $X' = \{A^{-1}(\mathbf{x} - \mathbf{b}) : \mathbf{x} \in X\}$.

The Frank-Wolfe Algorithm will incure the same optimization error on two affinely equivalent functions. Hence a good analysis of the Frank-Wolfe algorithm should provide a bound that is invariant under affine transformations,

Curvature Constant

We define the **curvature constant** of the constrained optimization problem as:

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

The curvature constant is affine invariant.

Note that $d(\mathbf{y}) := f(\mathbf{y}) - f(\mathbf{x}) - \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x})$ is the pointwise vertical distance between the graph of f and its linear approximation at $\mathbf{x}$. By convexity, $d(\mathbf{y}) \geq 0$ for all $\mathbf{y} \in X$. For $\mathbf{y}$ resulting from $\mathbf{x}$ by a Frank-Wolfe step with stepsize γ, we normalize the vertical distance with γ^2 (a natural choice if we think of f as being smooth), and take the supremum over all possible such normalized vertical distances.

The convergence rate of the Frank-Wolfe algorithm can be described purely in terms of this quantity, without resorting to any smoothness constants L or diameters $\text{diam}(X)$, which are not smooth.

Theorem: Consider the constrained minimization problem where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and X is convex, closed and bounded. Let $C_{(f,X)}$ be the curvature constant of f over X. With any $\mathbf{x}_0 \in X$ and stepsizes $\gamma_t = \frac{2}{t+2}$ the Frank-Wolfe Algorithm yields:

$f(\mathbf{x}_t) - f(\mathbf{x}^* = \leq \frac{4C_{(f,X)}}{T+1}$. Trick: we proceed as before but we show $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)^T\gamma_t(\mathbf{x}_t - \mathbf{s}) + \gamma_t^2 C_{(f,X)}$.

Lemma: Let f be convex and smooth with parameter L over X, then: $C_{(f,X)} \leq \frac{L}{2}\text{diam}(X)^2$.

Convergence in duality gap

Theorem: Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex and smooth with parameter L and $\mathbf{x}_0 \in X$, $T \geq 2$, then choosing any of the previously discussed stepsizes, the Frank-Wolfe algorithm yields at t, $1 \leq t \leq T$ such that: $g(\mathbf{x}_t) \leq \frac{27/2 \cdot C_{(f,X)}}{T+1}$.

Sparsity

: The previous results means that $\mathcal{O}(\frac{1}{\epsilon})$ many iterations are sufficent to obtain optimality gap at most ϵ. At this time, the current solution is a convex combination of $\mathbf{x}_0$ and $\mathcal{O}(\frac{1}{\epsilon})$ many atoms of the constraint set X. Thinking of ϵ as a constant (such as 0.01), this means that constantly many atoms are sufficient in order to get an almost optimal solution.

Example	$\mathcal{A}$	$\lvert\mathcal{A}\rvert$	dim	$\text{LMO}_X(\mathbf{g})$
l_1-Ball	$\{\pm\mathbf{e}_i\}$	$2d$	d	$\pm\mathbf{e}_i$ with $\text{argmax}_i\lvert g_i\rvert$
Simplex	$\{\mathbf{e}_i\}$	d	d	$\mathbf{e}_i$ with $\text{argmin}_i g_i$
Spectahedron	$\{\mathbf{x}\mathbf{x}^T, \lVert x\rVert = 1\}$	∞	d^2	$\text{argmin}_{\lVert\mathbf{x}\rVert=1}\mathbf{x}^T G\mathbf{x}$
Norms	$\{\mathbf{x}, \lVert\mathbf{x}\rVert \leq 1\}$	∞	d	$\text{argmin}_{\lVert\mathbf{s}\rVert\leq 1}\langle\mathbf{s}, \mathbf{g}\rangle$

8 Newton's Method

One dimensional case

The goal is to find the zero of a differentiable function $f : \mathbb{R} \to \mathbb{R}$, using an iterative method. Starting with some x_0 we compute: $x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)}$. Note that this is equivalent to solving the following linear equation: $f(x_t) + f'(x_t)(x - x_t) = 0$.
The Newton step obviously fails if $f'(x_t) = 0$ and may get out of control if $\left|f'(x_t)\right|$ is very small.

Newton's method for optimization

Suppose we want to find a global minimum x^* of a differentiable convex function $f : \mathbb{R} \to \mathbb{R}$ (assuming that a global minimum exists). We can equivalently search for a zero of the derivative f'. If f is twice differentiable the Newton Method yields: $x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)}$. For $d \geq 1$ we get:

The **Newton step** for minimizing a twice differentiable convex function:

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

Also notice that we can consider the newton method as a special case of $\mathbf{x}_{t+1} = \mathbf{x}_t - H(\mathbf{x}_t)\nabla f(\mathbf{x}_t)$, where $H(\mathbf{x}_t) \in \mathbb{R}^{d \times d}$ is some matrix. Note that gradient descent is of this form with $H(\mathbf{x}_t) = \gamma I$.
Minimization of Taylor: Let f be convex and twice differentiable at $\mathbf{x}_t \in \mathbf{dom}(f)$ with $\nabla^2 f(\mathbf{x}_t) \succ 0$ being invertible. Then the vector $\mathbf{x}_{t+1}$ resulting from the Newton step satisfies: $\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t)^T(\mathbf{x} - \mathbf{x}_t) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_t)^T \nabla^2 f(\mathbf{x}_t)(\mathbf{x} - \mathbf{x}_t)$.

Convergence result

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be twice differentiable with a critical point $\mathbf{x}^*$. Suppose that there is a ball $X \subset \mathbf{dom}(f)$ with center $\mathbf{x}^*$ such that the following holds:

(i) Bounded inverse Hessian: Thre exist a real number $\mu > 0$ such that $\left\|\nabla^2 f(\mathbf{x})^{-1}\right\| \leq \frac{1}{\mu}, \forall \mathbf{x} \in X$

(ii) Lipschitz continuous Hessians: There exists a real number $B \geq 0$ such that: $\left\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{y})\right\| \leq B\|\mathbf{x} - \mathbf{y}\| \ \forall x \in X$.

Notice that *(i)* implies that the Hessian is always invertible in X, then for $\mathbf{x}_t \in X$ and $\mathbf{x}_{t+1}$ the resulting Newton step we get $\left\|\mathbf{x}_{t+1} - \mathbf{x}^*\right\| \leq \frac{B}{2\mu}\|\mathbf{x}_t - \mathbf{x}^*\|^2$.
This yields that in this case if $\mathbf{x}_0 \in X$ satisfies $\|\mathbf{x}_0 - \mathbf{x}^*\| \leq \frac{\mu}{B}$ we get: $\|\mathbf{x}_T - \mathbf{x}^*\| \leq \frac{\mu}{B}\left(\frac{1}{2}\right)^{2^T - 1}$.
Theorem (Hessian inverses of strongly convex functions are bounded) Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be twice differentiable and strongly convex with parameter μ over an open convex subset $X \subset \mathbf{dom}(f)$, then $\nabla^2 f(\mathbf{x})$ is invertable and $\left\|\nabla^2 f(\mathbf{x})^{-1}\right\| \leq \frac{1}{\mu}$ for all $\mathbf{x} \in X$.

9 Quasi-Newton Methods

Motivation

The main computational bottleneck in Newton's method is the computation and inversion of the Hessian matrix in each step. This matrix has size $d \times d$, so it will take up to $\mathcal{O}(d^3)$ time to invert it.
In the one dimensional case we can approximate the derivative by its finite approximation and we get a **secant step**: $x_{t+1} = x_t - f(x_t)\frac{x_t - x_{t-1}}{f(x_t) - f(x_{t-1}}$. We can apply this to obtain a secant method for optimization: $x_{t+1} = x_t - f'(x_t)\frac{x_t - x_{t-1}}{f'(x_t) - f'(x_{t-1})}$.

The secant condition

Our goal is to find H_t that approximates $f''(x_t)$ and in the multidimensional case $\nabla^2 f(\mathbf{x}_t)$. The **secant condition** is:
$f'(x_t) - f'(x_{t-1})) = H_t(x_t - x_{t-1})$ and in the multidimensional case:
$\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}) = H_t(\mathbf{x}_t - x_{t-1})$.
The hope is that $H_t \approx \nabla^2 f(\mathbf{x}_t)$. We say that we have a **Quasi-Newton method** if H_t is a symmetric matrix, satisfying the secant condition.

Greenstadt's approach

For efficieny reasons (we want to avoid matrix inversions), Quasi-Newton methods typically directly deal with the inverse matrices H_t^{-1}. Suppose that we have H_{t-1}^{-1} how do we choose H_t^{-1}?
Greenstadt's approach is to update H_{t-1}^{-1} by an **error matrix** E_t to obtain $H_t^{-1} = H_{t-1}^{-1} + E_t$. Moreover the errors should be as small as possible subject to the constraint that H_t^{-1} is symetric.
We define the **Frobenius Norm** of a matrix M as: $\|M\|_F^2 = \sum_{i=1}^d \sum_{j=1}^d m_{i,j}$.
Greenstadts approach is to minimize the error term $\left\|AEA^T\right\|_F^2$ where A is some fixed invertible transformation matrix A. If $A = I$ we recover the usual Frobenius norm.
Let us fix t and simplify the notation we set $H := H_{t-1}^{-1}$, $H' := H_t^{-1}$, $E := E_t$, $\boldsymbol{\sigma} := \mathbf{x}_t - \mathbf{x}_{t-1}$, $\mathbf{y} = \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1})$, $\mathbf{r} = \boldsymbol{\sigma} - H\mathbf{y}$.
The new **update formula** is now $H' = H + E$ and the **secant condition** is $H'\mathbf{y} = \boldsymbol{\sigma}$ (or equivalently $E\mathbf{y} = \mathbf{r}$).

Greenstadt's approach can now be summarized as a convex constrained optimization problem in d^2 variables $E_{i,j}$:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\left\|AEA^T\right\|_F^2 \\ &\text{subject to} E\mathbf{y} = \mathbf{r}, \qquad E^T - E = 0 \end{aligned}$$

Such a system of equations can be solved using Lagrange multipliers. This yields the following result:

Theorem: An update matrix E^* satisfying the constraints $E\mathbf{y} = \mathbf{r}$ (secant condition in the next step) and $E^T - E = 0$ (symmetry) is a minimizer of the error function $f(E) = \frac{1}{2}\left\|AEA^T\right\|_F^2$ subject to the aformentioned constraints if and only if there exists a vector $\lambda \in \mathbb{R}^d$ and a matrix $\Gamma \in \mathbb{R}^{d \times d}$ such that $WE^*W = \lambda\mathbf{y}^T + \Gamma^T - \Gamma$, where $W = A^TA$ (a symetric and positive definite matrix).

The Greenstadt family

The new goal is to solve the following system of equations:

$$E\mathbf{y} = \mathbf{r}, \qquad E^T - E = 0, \qquad WEW = \lambda\mathbf{y}^T + \Gamma^T - \Gamma$$

which is a linear system over E, λ, Γ. This yields:

Let $M \in \mathbb{R}^{d \times d}$ be a symetric matrix and invertable matrix. Consider the quasi-Newton method: $\mathbf{x}_{t+1} = \mathbf{x}_t - H_t^{-1}\nabla f(\mathbf{x}_t)$, where $H_0 = I$ and $H_t^{-1} = H_t^{-1} + E_t$ is chosen for all $t \geq 1$ in such a way that H_t^{-1} is symmetric and satisfies the secant condition: $\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}) = H_t(\mathbf{x}_t - \mathbf{x}_{t-1})$. For any t set $H := H_{t-1}^{-1}$, $H' := H_t^{-1}$, $\boldsymbol{\sigma} := \mathbf{x}_t - \mathbf{x}_{t-1}$, $\mathbf{y} := \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1})$, and define:

$$\begin{aligned} E^* = &\frac{1}{\mathbf{y}^TM\mathbf{y}}\Big(\boldsymbol{\sigma}\mathbf{y}^TM + M\mathbf{y}\boldsymbol{\sigma}^T - H\mathbf{y}\mathbf{y}^TM - M\mathbf{y}\mathbf{y}^TH \\ &- \frac{1}{\mathbf{y}^TM\mathbf{y}}(\mathbf{y}^T\boldsymbol{\sigma} - \mathbf{y}^TH\mathbf{y})M\mathbf{y}\mathbf{y}^TM\Big) \end{aligned}$$

If the update matrix $E_t = E^*$ is used the method is call **Greenstadt method** with parameter M.

The BFGS method

The **BFGS method** is a Greenstadt family method with $M = H'$, this means that H' disappears from the formula, this yields: $E^* = \frac{1}{\mathbf{y}^T\boldsymbol{\sigma}}\left(-H\mathbf{y}\boldsymbol{\sigma}^T - \boldsymbol{\sigma}\mathbf{y}^TH + \left(1 + \frac{\mathbf{y}^TH\mathbf{y}}{\mathbf{y}^T\boldsymbol{\sigma}}\boldsymbol{\sigma}\boldsymbol{\sigma}^T\right)\right)$. Because we don't need to compute any Hessian's the cost per iteration drops to $\mathcal{O}(d^2)$
Newton and Quasi-Newton methods are often performed with scaled steps. This means that the iteration becomes: $\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t H_t^{-1}\nabla f(\mathbf{x}_t)$, for some $\alpha_t \in \mathbb{R}^+$.

10 Subgradient Methods

Definitions and first facts

Let $f : \mathbf{dom}(f) \to \mathbb{R} \cup \{+\infty\}$ be a convex function. A vector $\mathbf{g} \in \mathbb{R}^d$ is a **subgradient** of f at a point $\mathbf{x} \in \mathbf{dom}(f)$ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{g}^T(\mathbf{y} - \mathbf{x}), \forall \mathbf{y} \in \mathbf{dom}(f)$. The set of all subgradient at $\mathbf{x}$ is called the **subdifferential** of f as $\mathbf{x}$ denoted as ∂f.

If f is convex and differentiable at $\mathbf{x} \in \mathbf{dom}(f)$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$
If f is differentiable at $\mathbf{x} \in \mathbf{dom}(f)$, then $\partial f(\mathbf{x}) \subset \{\nabla f(\mathbf{x})\}$
Lemma: Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex, $\mathbf{dom}(f)$ open and $B \in \mathbb{R}_+$, then the following are equivalent:

(i) $\|\mathbf{g}\| \leq B$ for all $\mathbf{x} \in \mathbf{dom}(f)$ and $\mathbf{g} \in \partial f(\mathbf{x})$

(ii) $|f(\mathbf{x}) - f(\mathbf{y})| \leq B\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbf{dom}(f)$.

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

Properties

Lemma: Let f be a convex function and $\mathbf{x} \in \mathbf{dom}(f)$. Then $\partial f(\mathbf{x})$ is convex and closed.
The **relative interior** of set X is defined as $\text{relint}(X) = \{\mathbf{x} : \exists r > 0, \text{such that } B(\mathbf{x}, r) \cap \text{Aff}(X) \subset X\}$, which is the set of interior points relative to the affine subspaces that contain X.
Hyperplane separation theorem: Let S and T be two nonempty convex sets. Then S and T can be separated if and only if $\text{relint}(S) \cap \text{relint}(T) = \emptyset$.
Corrolary: Let S be a nonempty convex set $\mathbf{x}_0 \in \partial S$ (boundary of S). There exists a supporting hyperplane $H = \{\mathbf{x} : \mathbf{a}^T\mathbf{x} = \mathbf{a}^T\mathbf{x}_0\}$, with $\mathbf{a} \neq 0$ such that: $S \subset \{\mathbf{x} : \mathbf{a}^T\mathbf{x} \leq \mathbf{a}^T\mathbf{x}_0\}$ and $\mathbf{x}_0 \in H$.
Theorem (Existence of subgradient): Lef f be a convex function. Then $\partial f(\mathbf{x})$ is nonempty and bounded if $\mathbf{x} \in \text{relint}(\mathbf{dom}(f))$.
Lemma: Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be a function such that $\mathbf{dom}(f)$ is convex and $\partial f(\mathbf{x}) \neq \emptyset$ for all $\mathbf{x} \in \mathbf{dom}(f)$. Then f is convex.
Lemma (Monotonicity of sub-differential): The subdifferential of a convex function $f(\mathbf{x})$ at $\mathbf{x} \in \mathbf{dom}(f)$ is a monotone operator, i.e: $(\mathbf{u} - \mathbf{v})^T(\mathbf{x} - \mathbf{y}) \geq 0$, $\forall \mathbf{x}, \mathbf{y} \in \mathbf{dom}(f), \mathbf{u} \in \partial f(\mathbf{x}), \mathbf{v} \in \partial f(\mathbf{y})$.
The **directional derivative** of a function f at $\mathbf{x}$ along $\mathbf{d}$ is $f'(\mathbf{x}, \mathbf{d}) = \lim_{\delta \to 0+} \frac{f(\mathbf{x} + \delta\mathbf{d}) - f(\mathbf{x})}{\delta}$. If f is differentiable then $f'(\mathbf{x}, \mathbf{d}) = \nabla f(\mathbf{x})^T\mathbf{d}$.
Lemma: If f is convex then the ratio $\phi(\delta) = \frac{f(\mathbf{x} + \delta\mathbf{d}) - f(\mathbf{x})}{\delta}$ is non-decreasing in $\delta > 0$.
Theorem: Let f be convex and $\mathbf{y} \in \text{int}(\mathbf{dom}(f))$, then: $f'(\mathbf{x}, \text{d}) = \max_{\mathbf{g} \in \partial f(\mathbf{x})} \mathbf{g}^T\mathbf{d}$.

Determining the subdifferentiable set of a convex function at a given point is in general very difficult. The following calculus of subdifferentiable sets provides a constructive way to compute the subgradient of convex functions arising from convexity-preserving operators.

(i) **Taking conic combination**: If $h(\mathbf{x}) = \lambda f(\mathbf{x}) + \mu g(\mathbf{x})$, where $\lambda, \mu \geq 0$ and f and g are both convex then: $\partial h(\mathbf{x}) = \lambda\partial f(\mathbf{x}) + \mu\partial g(\mathbf{x}), \forall \mathbf{x} \in \text{int}(\mathbf{dom}(h))$.

(ii) **Taking affine composition**: If $h(\mathbf{x}) = f(A\mathbf{x} + \mathbf{b})$, where f is convex then $\partial h(\mathbf{x}) = A^T\partial f(A\mathbf{x} + \mathbf{b})$.

(iii) **Taking supremum**: If $h(\mathbf{x}) = \sup_{\alpha \in \mathcal{A}} f_\alpha(\mathbf{x})$ and each $f_\alpha(\mathbf{x})$ is convex then: $\partial h(\mathbf{x}) \supseteq \text{conv}\{\partial f_\alpha(\mathbf{x}) : \alpha \in \alpha(\mathbf{x})\}$ with $\alpha(\mathbf{x}) := \{\alpha : h(\mathbf{x}) = f_\alpha(\mathbf{x})\}$.

(iv) **Taking superposition**: If $h(\mathbf{x}) = F(f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$, where $F(\mathbf{y}_1, \ldots, \mathbf{y}_m)$ is non-decreasing and convex, then: $\partial h(\mathbf{x}) \supseteq \left\{\sum_{i=1}^m d_i\partial f_i(\mathbf{x}) : (d_1, \ldots, d_m) \in \partial F(y_1, \ldots, y_m)\right\}$.

Subgradient Method

Consider the generic optimization problem $\min f(\mathbf{x})$ such that $\mathbf{x} \in X$, where f is convex (possibly non differentiable) and $X \subseteq \mathbf{dom}(f)$ is closed and convex. Assume the problem is solvable with optimal solution x^*, f^*. We define two important quantities:

(i) $R^2 := \max_{\mathbf{x}, \mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|^2$, the **squared diameter** of X

(ii) $B := \sup_{\mathbf{x}, \mathbf{y} \in X} \frac{|f(\mathbf{x}) - f(\mathbf{y})|}{\|\mathbf{x} - \mathbf{y}\|_2} < \infty$ is the constant that characterizes Lipschiz continuity of f under $\|\cdot\|_2$.

The subgradient method initializes $\mathbf{x}_1 \in X$ and repeats the following step $\mathbf{x}_{t+1} = \Pi_X(\mathbf{x}_t - \gamma_t\mathbf{g}_t)$ with $\mathbf{g}_t \in \partial f(\mathbf{x}_t)$.

When f is differentiable this reduces to the projected gradient descent method. Note that unlike Gradient Descent, Subgradient Descent is not a descent method, i.e., moving along the negative direction of subgradient is not necessarily decreasing the objective function.
Convergence of subgradient descent: Assume f is convex, then Subgradient Descent satisfies $\min_{1 \leq t \leq T} f(x_t) - f^* \leq \left(\sum_{t=1}^T \gamma_t\right)^{-1}\left(\frac{1}{2}\|\mathbf{x}_1 - \mathbf{x}^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \gamma_t^2\|\mathbf{g}\|_2^2\right)$, and for $\hat{\mathbf{x_T}} = \left(\sum_{t=1}^T \gamma_t\right)^{-1}\left(\sum_{t=1}^T \gamma_t\mathbf{x}_t\right) \in X$ we have $f(\hat{\mathbf{x_T}}) - f^* \leq \left(\sum_{t=1}^T \gamma_t\right)^{-1}\left(\frac{1}{2}\|\mathbf{x}_1 - \mathbf{x}^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \gamma_t^2\|\mathbf{g}\|_2^2\right)$ **Corrolary**: Using B and R we get: $\min_{T_0 \leq t \leq T} f(\mathbf{x}_t) - f^* \leq \frac{0.5 \cdot R^2 + 0.5 \cdot \sum_{t=T_0}^T \gamma_t^2 B^2}{\sum_{t=T_0}^T \gamma_t}, \forall 1 \leq T_0 \leq T$.

Stepsizes: We define the following stepsizes:

(i) Constant stepsize $\gamma_t = \gamma > 0$

(ii) Scaled stepsize $\gamma_t = \frac{\gamma}{\|\mathbf{g}_t\|_2}$.

(iii) Non-summable but diminishing stepsize satisfying: $\sum_{t=1}^\infty \gamma_t = \infty$, $\lim_{t \to \infty} \gamma_t = 0$.

(iv) Non-summable but square-summable stepsize satisfying: $\sum_{t=1}^\infty \gamma_t = \infty$, but $\sum_{t=1}^\infty \gamma_t^2 < \infty$.

(v) Polyak stepsize: Assuming $f^* = f(\mathbf{x}^*)$ is known choose: $\gamma_t = \frac{f(\mathbf{x}_t) - f^*}{\|\mathbf{g}_t\|_2^2}$.

For convex functions, subgradient descent will always converge with the stepsizes above. In case *(i)* with $\gamma_t = \frac{B}{R\sqrt{T}}$ and *(iii)* with $\gamma_t = \frac{B}{R\sqrt{t}}$. **Convergence for strongly convex functions (1)**: Assume that f is μ-strongly convex, then subgradient descent with stepsize $\gamma_t = \frac{1}{\mu t}$ satisfies: $\min_{1 \leq t \leq T} f(\mathbf{x}_t) - f^* \leq \frac{B^2(\ln(T) + 1)}{2\mu T}$ and $f(\hat{\mathbf{x_T}}) - f^* \leq \frac{B^2(\ln(T) + 1)}{2\mu T}$, where $\hat{\mathbf{x_T}} = \frac{1}{T}\sum_{t=1}^T \mathbf{x}_t$.
Convergence for strongly convex functions (2): Assume that f is μ-strongly convex, then subgradient descent with stepsize $\gamma_t = \frac{1}{\mu(t+1)}$ satisfies: $\min_{1 \leq t \leq T} f(\mathbf{x}_t) - f^* \leq \frac{2B^2}{\mu(T+1)}$ and $f(\hat{\mathbf{x_T}}) - f^* \leq \frac{2B^2}{\mu(T+1)}$, where $\hat{\mathbf{x_T}} = \frac{1}{T}\sum_{t=1}^T \frac{2t}{T(T+1)}\mathbf{x}_t$.
While the convergence rates achieved by subgradient descent seems much worse than those achieved by gradient descent for smooth problems, one cannot improve the $\mathcal{O}(1/\sqrt{T})$ and $\mathcal{O}(1/T)$ rates for the convex and strongly convex situations, respectively, when using block-box oriented methods that only have access to the subgradient of the objective function.

11 Mirror Descent, Smoothing, Proximal Algorithms

Let $\omega : X \to \mathbb{R}$ be a function that is strictly convex, continuously differentiable on a closed convex set X. The **Bregman divergence** is defined as $V_\omega(\mathbf{x}, \mathbf{y}) := \omega(\mathbf{x}) - \omega(\mathbf{y}) - \nabla\omega(\mathbf{y})^T(\mathbf{x} - \mathbf{y}), \forall \mathbf{x}, \mathbf{y} \in X$. Note that this is not a valid distance function!

Generalized Pythagorean Theorem: If $\mathbf{x}^*$ is the Bregman projection of $\mathbf{x}_0$ onto a convex set $C \subset X$: $\mathbf{x}^* = \text{argmin}_{\mathbf{x} \in C} V_\omega(\mathbf{x}, \mathbf{x}_0)$. Then for all $\mathbf{y} \in C$ it holds that: $V_\omega(\mathbf{y}, \mathbf{x}_0) \geq V_\omega(\mathbf{y}, \mathbf{x}^*) + V_\omega(\mathbf{x}^*, \mathbf{x}_0)$.

Mirror Descent

Given an input $\mathbf{x}$ and vector $\boldsymbol{\xi}$, we will define the **prox-mapping**: $\text{prox}_{\mathbf{x}}(\boldsymbol{\xi}) = \text{argmin}_{\mathbf{u} \in X}\{V_\omega(\mathbf{u} + \mathbf{x}) + \langle\boldsymbol{\xi}, \mathbf{u}\rangle\}$, where the distance-generating function $\omega(\cdot)$ is 1-strongly convex with respect to the norm $\|\cdot\|$ on X.
The **Mirror descent algorithm** adopts the update step $\mathbf{x}_{t+1} = \text{prox}_{\mathbf{x}_t}(\gamma_t\mathbf{g}_t)$, with $\mathbf{g}_t \in \partial f(\mathbf{x}_t)$.

Note that if $\omega(\mathbf{x}) = \frac{1}{2}\|x\|_2^2$, and $\|\cdot\| = \|\cdot\|_2$, then mirror descent reduces to subgradient descent.
Three point identity: For any $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{dom}(\omega)$: $V_\omega(\mathbf{x}, \mathbf{z}) = V_\omega(\mathbf{x}, \mathbf{y}) + V_\omega(\mathbf{y}, \mathbf{z}) - \langle\nabla\omega(\mathbf{z}) - \nabla\omega(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle$.
We define the **dual norm** of $\|\cdot\|$ as $\|\mathbf{x}\|_* = \sup\{\mathbf{x}^T\mathbf{z} : \|\mathbf{z}\| \leq 1\}$. We then have **Young's inequality**: $\mathbf{x}^T\mathbf{y} \leq \frac{\|\mathbf{x}\|^2}{2} + \frac{\|\mathbf{y}\|_*^2}{2}$.
Convergence result: For Mirror descent let f be convex and $\omega(\cdot)$ be 1-strongly convex on X with respect to $\|\cdot\|$, then: $\min_{1 \leq t \leq T} f(\mathbf{x}_t) - f^* \leq \frac{V_\omega(\mathbf{x}^*, \mathbf{x}_1) + 0.5 \cdot \sum_{t=1}^T \gamma_t^2\|\mathbf{g}_t\|_*^2}{\sum_{t=1}^T \gamma_t}$, and $f\left(\frac{\sum_{t=1}^T \gamma_t\mathbf{x}_t}{\sum_{t=1}^T \gamma_t}\right) \leq \frac{V_\omega(\mathbf{x}^*, \mathbf{x}_1) + 0.5 \cdot \sum_{t=1}^T \gamma_t^2\|\mathbf{g}_t\|_*^2}{\sum_{t=1}^T \gamma_t}$. Trick: Show that $\langle\gamma_t\mathbf{g}_t, \mathbf{x}_t - \mathbf{x}^*\rangle \leq V_\omega(\mathbf{x}^*, \mathbf{x}_t) - V_\omega(\mathbf{x}^*, \mathbf{x}_{t+1}) + \frac{\gamma_t^2}{2}\|\mathbf{g}_t\|_*^2$, using **optimality condition**: $\langle\nabla\omega(\mathbf{x}_{t+1}) + \gamma_t\mathbf{g}_t - \nabla\omega(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_{t+1}\rangle \geq 0$.

Convex Conjugate Theory

For a function $f : \mathbf{dom}(f) \to \mathbb{R}$ its **convex conjugate** is given by $f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \mathbf{dom}(f)}\{\mathbf{x}^T\mathbf{y} - f(\mathbf{x})\}$. This is also known as **Legendre-Fenchel** transformation. f^* will always be convex (even if f is not).
Fenchel's inequality follows easily: $\mathbf{x}^T\mathbf{y} \leq f(\mathbf{x}) + f^*(\mathbf{y}), \forall \mathbf{x}, \mathbf{y}$. **Lemma**: If function f is convex, lower semi-continuous and proper, then $(f^*)^* = f$. Here lower semi-continuity means that $\liminf_{\mathbf{x} \to \mathbf{x}_0} f(\mathbf{x}) \geq f(\mathbf{x}_0)$.
Theorem: If f is μ-strongly convex then f^* is continuously differentiable and $\frac{1}{\mu}$-Lipschitz smooth.

Lemma: Let f and g be two proper, convex and semi-continuous functions, then *(i)* $(f+g)^*(\mathbf{x}) = \inf_{\mathbf{y}}\{f^*(\mathbf{y}) + g^*(\mathbf{x}-\mathbf{y})\}$ and *(ii)* $(\alpha f)^*(\mathbf{x}) = \alpha f^*(\frac{\mathbf{x}}{\alpha})$, for $\alpha > 0$.

Smoothing Techniques

Goal: Approximate a non-smooth function f by a smooth and convex function f_μ.
Nestorov Smoothing: We approximate $f(\mathbf{x})$, with $f_\mu(\mathbf{x}) = \max_{\mathbf{y}\in\mathbf{dom}(f^*)}\{\mathbf{x}^T\mathbf{y} - f^*(\mathbf{y}) - \mu\cdot d(\mathbf{y})\}$, where f^* is the convex conjugate of f and $d(\mathbf{y})$ is some proximity function. Notice that $f_\mu = (f^* + \mu d)^*$, hence f_μ is continuously differentiable and Lipschitz-smooth.
The **proximity function** should satisfy *(i)* $d(\mathbf{y})$ is continuous and 1-strongly convex Y; *(ii)* $d(\mathbf{y_0}) = 0$, for $\mathbf{y_0} \in \operatorname{argmin}_{\mathbf{y}\in Y} d(\mathbf{y})$; *(iii)* $d(\mathbf{y}) \geq 0$, $\forall \mathbf{y} \in Y$.
We consider the case where f can be represented as $f(\mathbf{x}) = \max_{\mathbf{y}\in Y}\{\langle A\mathbf{x}+\mathbf{b},\mathbf{y}\rangle - \phi(\mathbf{y})\}$, with $\phi(\mathbf{y})$ being a convex and continuous function and Y a convex and compact set. This generalizes the Fenchel representation. The Nesterov smoothing then reduces to $f_\mu(\mathbf{x}) = \max_{\mathbf{y}\in Y}\{\langle A\mathbf{x}+\mathbf{b},\mathbf{y}\rangle - \phi(\mathbf{y}) - \mu d(\mathbf{y})\}$. **Theoretical Guarantees**: For $f_\mu(\mathbf{x})$ we have: *(i)* $f_\mu(\mathbf{x})$ is continuously differentiable; *(ii)* $\nabla f_\mu(\mathbf{x}) = A^T y(\mathbf{x})$, where $y(\mathbf{x}) = \operatorname{argmax}_{\mathbf{y}\in Y}\{\langle A\mathbf{x}+\mathbf{b},\mathbf{y}\rangle - \phi(\mathbf{y}) - \mu d(\mathbf{y})\}$; *(iii)* $f_\mu(x)$ is $\frac{\|A\|_2^2}{\mu}$-Lipschitz smooth with $\|A\|_2 := \max_{\mathbf{x}:\|\mathbf{x}\|=1}\|A\mathbf{x}\|_2$.
Convergence: For any $\mu > 0$, let $D_Y^2 = \max_{\mathbf{y}\in Y} d(\mathbf{y})$ we have: $f(\mathbf{x}) - \mu D_Y^2 \leq f_\mu \leq f(\mathbf{x})$.
Moreau-Yosida Regularization: We consider the following approximation function: $f_\mu(\mathbf{x}) = \min_{\mathbf{y}\in\mathbf{dom}(f)}\{f(\mathbf{y}) + \frac{1}{2\mu}\|\mathbf{x}-\mathbf{y}\|_2^2\}$, where $\mu > 0$ is the smoothness parameter. f_μ is also called the Moreau envelope of f.
For a convex function f we define the **proximal operator** of f at a given point $\mathbf{x}$ as: $\operatorname{prox}_f(\mathbf{x}) := \operatorname{argmin}_{\mathbf{y}}\{f(\mathbf{y}) + \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2\}$. We immediately notice that for $\mu > 0$,we have: $\operatorname{prox}_{\mu f}(\mathbf{x}) = \operatorname{argmin}_{\mathbf{y}}\{f(\mathbf{y}) + \frac{1}{2\mu}\|\mathbf{x}-\mathbf{y}\|^2\}$. **Properties**: Let f be a convex function, then we have:

(i) Fixed Point: A point $\mathbf{x}^*$ minimizes f if and only if $\mathbf{x}^* = \operatorname{prox}_f(\mathbf{x}^*)$.
(ii) Non-Expansiveness: $\left\|\operatorname{prox}_f(\mathbf{x}) - \operatorname{prox}_f(\mathbf{y})\right\| \leq \|\mathbf{x}-\mathbf{y}\|$.
(iii) Moreau Decomposition: For any $\mathbf{x}$: $\mathbf{x} = \operatorname{prox}_f(\mathbf{x}) + \operatorname{prox}_{f^*}(\mathbf{x})$.

Danskin's theorem: The gradient of f_μ is given by $\nabla f_\mu(\mathbf{x}) = \frac{1}{\mu}(\mathbf{x} - \operatorname{prox}_{\mu f}(\mathbf{x}))$.
Since f_μ is $\frac{1}{\mu}$-smooth, gradient descent for the smoothed function works as follows $\mathbf{x}_{t+1} = \mathbf{x}_t - \mu\nabla f_\mu(\mathbf{x}_t)$, which we can rewrite as $\mathbf{x}_{t+1} = \operatorname{prox}_{\mu f}(\mathbf{x}_t)$, which is known as **proximal point algorithm**. We can also change the step size in every iteration which yields: $\mathbf{x}_{t+1} = \operatorname{prox}_{\gamma_t f}(\mathbf{x}_t)$
Convergence result: Let f be a convex function, the proximal point algorithm satisfies $f(\mathbf{x}_t) - f^* \leq \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|_2^2}{2\sum_{\tau=0}^{t-1}\gamma_\tau}$.
Randomized smoothing: The randomized smoothing paradigm uses the following function to approximate f: $f_\mu(\mathbf{x}) = \mathbb{E}_Z[f(\mathbf{x}+\mu Z)]$, where Z is an isotropic Gaussian or uniform random variable.

12 Stochastic Optimization

The **stochastic optimization problem** is $\min_{\mathbf{x}\in X} F(\mathbf{x})$ with $F(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\xi}}[f(\mathbf{x},\boldsymbol{\xi})]$, where $f(\mathbf{x},\boldsymbol{\xi})$ is a function involving the decision variable $\mathbf{x}$ and a random variable (vector) $\boldsymbol{\xi}$. The random variable $\boldsymbol{\xi}$ is some well defined variable with support $\Xi \subseteq \mathbb{R}^m$ and follows the distribution $P(\boldsymbol{\xi})$. If $\boldsymbol{\xi}$ is the uniform distribution over the index set $\{1,\ldots,n\}$, then $F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$, this is the **finite-sum problem**.

Stochastic Gradient Descent

Assume that $F(\mathbf{x},\boldsymbol{\xi})$ is a continuously differentiable for any realization $\boldsymbol{\xi} \in \Xi$. We update $\mathbf{x}_{t+1}$ as follows: $\mathbf{x}_{t+1} := \Pi_X(\mathbf{x}_t - \gamma_t\nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t))$, where $\boldsymbol{\xi}_t \sim P(\boldsymbol{\xi})$, i.i.d. Here the gradient is taken over the argument $\mathbf{x}$. In the finite-sum case we get: $\mathbf{x}_{t+1} = \Pi_X(\mathbf{x}_t - \gamma_t\nabla f_{i_t}(\mathbf{x}_t))$ where i_t is sampled uniformly at random from $\{1,\ldots,n\}$.
We also assume that the stochastic gradient is unbiased i.e. $\mathbb{E}[\nabla f(\mathbf{x},\boldsymbol{\xi})] = \nabla F(\mathbf{x})$.
Remark: We need $\gamma_t \to 0$ for $t \to \infty$ to ensure convergence.
Convergence for strongly convex functions: Assume that $F(\mathbf{x})$ is μ-strongly convex and $\exists M > 0$, such that $\mathbb{E}[\|\nabla f(\mathbf{x},\boldsymbol{\xi})\|_2^2] \leq M^2$, $\forall\mathbf{x} \in X$, then with stepsize $\gamma_t = \frac{\gamma}{t}$, with $\gamma \geq \frac{1}{2\mu}$ we get : $\mathbb{E}[\|\mathbf{x}_t - \mathbf{x}^*\|_2^2]$ where $\frac{C(\gamma)}{t}$.
Stochastic Mirror Descent, works as follows: $x_{t+1} := \operatorname{argmin}_{\mathbf{x}\in X}\{V_\omega(\mathbf{x},\mathbf{x}_t) + \langle\gamma_t G(\mathbf{x}_t,\boldsymbol{\xi}),\mathbf{x}\rangle\}$, where for a given input $\mathbf{x},\boldsymbol{\xi}$ the estimator $G(x,\boldsymbol{\xi})$ satisfies that $\mathbb{E}[G(\mathbf{x},\boldsymbol{\xi})] \in \partial F(\mathbf{x})$ and $\mathbb{E}[\|G(\mathbf{x},\boldsymbol{\xi})\|_*^2] \leq M^2$. Note that we don't require $F(\mathbf{x})$ or $f(\mathbf{x},\boldsymbol{\xi})$ to be differentiable.
Convergence for convex functions Let F be convex, then stochastic gradient descent satisfies that $\mathbb{E}[F(\hat{\mathbf{x_T}} - F(\mathbf{x}^*)] \leq \frac{R^2 + \frac{M^2}{2}\sum_{t=1}^T\gamma_t^2}{\sum_{t=1}^T\gamma_t}$, with $R^2 = \max_{\mathbf{x}\in X} V_\omega(\mathbf{x},\mathbf{x}_1)$ and $\hat{\mathbf{x_T}} = \frac{\sum_{t=1}^T\gamma_t\mathbf{x}_t}{\sum_{t=1}^T\gamma_t}$.
Convergence of SGD under constant stepsize: Assume that $F(\mathbf{x})$ is both μ-strongly convex and L-smooth. Moreover assume that stochastic gradient satisfies that $\mathbb{E}[\|\nabla f(\mathbf{x},\boldsymbol{\xi})\|_2^2] \leq \sigma^2 + c\|\nabla F(\mathbf{x})\|_2^2$, then SGD with constant stepsize γ satisfies: $\mathbb{E}[F(\mathbf{x}_t) - F(\mathbf{x}^*)] \leq \frac{\gamma L\sigma^2}{2\mu} + (1-\gamma\mu)^{t-1}[F(\mathbf{x}_1) - F(\mathbf{x}^*)]$, where $\mathbf{x}^*$ is the optimal solution.
The condition on the gradient can be viewed as a generalization of the bounded variance assumption (which we recover if $c = 1$. If $\sigma^2 = 0$, we have a **strong growth condition** with constant c. If $\sigma^2 = 0$ and $c = 1$ we recover the deterministic setting.

Adaptive Stochastic Gradient Methods

Adaptive gradient methods, are methods whose stepsizes and search directions are adjusted based on past gradients.
General Framework For $t = 1,\ldots,T$ we successively define $\mathbf{g}_t = \nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)$, $m_t = \phi(\mathbf{g}_1,\ldots,\mathbf{g}_t)$, $V_t = \psi(\mathbf{g}_1,\ldots,\mathbf{g}_t)$, for some functions ϕ,ψ to be specified $\widehat{\mathbf{x}_t} = \mathbf{x}_t - \alpha V_t^{-1/2}\mathbf{m}_t$ and $x_{t+1} = \operatorname{argmin}_{\mathbf{x}\in X}\{(\mathbf{x} - \widehat{\mathbf{x}_t})^T V_t^{1/2}(\mathbf{x} - \widehat{\mathbf{x}_t})\}$. For example:
AdaGrad: AdaGrad rescales the learning rate component-wise by the square root of the cumulative sum of the previous gradients: $\mathbf{v}_t = \mathbf{v}_{t-1} + \nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)^{\odot 2}$ and $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\gamma_0}{\epsilon+\sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)$, with $\odot$ component-wise product.
RMSProp: RMSProp uses a moving average of the squared gradients with a discount factor to slow down the decay of the learning rates: $\mathbf{v}_t = \beta\mathbf{v}_{t-1} + (1-\beta)\nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)^{\odot 2}$ and $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\gamma_0}{\epsilon+\sqrt{\mathbf{v}_t}} \odot \nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)$. $\beta \in (0,1)$ is chosen close to 1.
Adam: Adam combines RMSProp with Momentum estimation. Similar to RMSProp, Adam also keeps an exponentially decaying average of past gradients, similar to the momentum estimation. Because of the factor β_1 , β_2 , the estimates m_t and $\mathbf{v}_t$ of the first and second moments of the gradient become biased, Adam also counteract these biases by normalizing these terms. $\mathbf{v}_t = \beta_2\mathbf{v}_{t-1} + (1-\beta_2)\nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)^{\odot 2}$, $\mathbf{m}_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)$, $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\gamma_0}{\epsilon+\sqrt{\tilde{\mathbf{v}}_t}}\cdot\tilde{\mathbf{m}}_t$, here $\tilde{\mathbf{v}}_t = \frac{\mathbf{v}_t}{1-\beta^t}$ and $\tilde{m}_t = \frac{m_t}{1-\alpha^t}$ are bias corrected.

13 Finite Sum Optimization

Variance Reduction Technique

We try to reduce the variance σ^2 in order to improve the bound of SGD: **Mini-batch sampling**: use a small batch of samples instead of one to estimate the gradient at every iteration: replace $\nabla f(\mathbf{x}_t,\boldsymbol{\xi}_t)$ with $\frac{1}{b}\sum_{i=1}^b\nabla f(\mathbf{x}_t,\boldsymbol{\xi}_{t,i})$. The variance will be $\mathcal{O}(b)$ times smaller.
Importance sampling: Instead of sampling from $\boldsymbol{\xi} \sim P$, we can obtain samples from another well defined random variable $\boldsymbol{\eta}$ with nominal distribution Q, and use a different stochastic gradient, $G(\mathbf{x}_t,\boldsymbol{\xi}_t)$ becomes $G(\mathbf{x}_t,\boldsymbol{\eta}_t)\frac{P(\boldsymbol{\eta}_t)}{Q(\boldsymbol{\eta}_t)}$. The variance of the new stochastic gradient under properly chosen distribution Q could be smaller.
Momentum: add momentum to the gradient step: $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t\widehat{\mathbf{m}_t}$, where $\widehat{m_t} = c\sum_{\tau=1}^t\alpha^{t-\tau}\nabla f_{i_\tau}(\mathbf{x}_\tau)$.

Using control variate

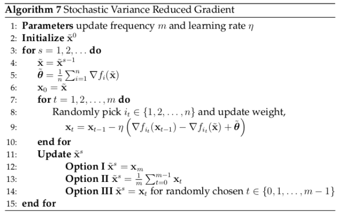
: Suppose we want to estimate $\Theta = \mathbb{E}[X]$, the expected value of a random variable X. Suppose we also have access to a random variable Y which is highly correlated with X, and we can compute $\mathbb{E}[Y]$ easily. Let's consider the following point estimator $\widehat{\Theta_\alpha}$ with $\alpha \in [0,1]$: $\widehat{\Theta_\alpha} := \alpha(X-Y) + \mathbb{E}[Y]$, then the expectation is given by $\mathbb{E}[\widehat{\Theta_\alpha}] = \alpha\mathbb{E}[X] + (1-\alpha)\mathbb{E}[Y]$ and the variance by $\mathrm{Var}[\widehat{\Theta_\alpha}] = \alpha^2(\mathrm{Var}[X] + \mathrm{Var}[Y] - 2\mathrm{Cov}[X,Y])$. As α increases from 0 to 1, the bias decreases and the variance increases.

Stochastic Variance-Reduced Algorithms

A natural question is: can we achieve best of both worlds, namely, can we design algorithms with fast convergence rate like GD but with cheap iteration cost like SGD? Here we will focus on solving the finite-sum optimization problem.

	SVRG	SAG/SAGA
memory cost	$\mathcal{O}(d)$	$\mathcal{O}(nd)$
epoch-based	yes	no
# gradients per step	at least 2	1
parameters	stepsize, epoch length	stepsize
unbiasedness	yes	yes/no
total complexity	$\mathcal{O}((n+\kappa_{\max})\log(1/\epsilon)$	$\mathcal{O}((n+\kappa_{\max})\log(1/\epsilon)$

SAG: The key idea of SAG is to keep track of the average of the past stored gradient of each component (denoted as $\mathbf{v}_i$) as an estimate of the full gradient, i.e. $\mathbf{g}_t = \frac{1}{n}\sum_{i=1}^n\mathbf{v}_i^t$. Where the past gradient $\{\mathbf{v}_i^t\}$ for each component function is updated as $\mathbf{v}_i^t = \nabla f_{i_t}(\mathbf{x}_t)$ if $i = i_t$, and $\mathbf{v}_i^{t-1}$, if $i \neq i_t$. Equivalently we can compute $\mathbf{g}_t = \mathbf{g}_{t-1} - \frac{1}{n}\mathbf{v}_{i_t}^{t-1} + \frac{1}{n}\nabla f_{i_t}(\mathbf{x}_t)$. Compared to SGD, the per-iteration cost is almost the same, but there is an additional $\mathcal{O}(nd)$ memory cost to store the past gradients of each components. The update is then $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma\mathbf{g}_t$.



Algorithm 7 Stochastic Variance Reduced Gradient

```
Parameters update frequency $m$ and learning rate $\eta$
Initialize $\tilde{\mathbf{x}}^0$
for $s = 1, 2, \ldots$ do
    $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}^{s-1}$
    $\tilde{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{\mathbf{x}})$
    $\mathbf{x}_0 = \tilde{\mathbf{x}}$
    for $t = 1, 2, \ldots, m$ do
        Randomly pick $i_t \in \{1, 2, \ldots, n\}$ and update weight,
            $\mathbf{x}_t = \mathbf{x}_{t-1} - \eta\left(\nabla f_{i_t}(\mathbf{x}_{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \tilde{\boldsymbol{\theta}}\right)$
    end for
    Update $\tilde{\mathbf{x}}^s$
        Option I $\tilde{\mathbf{x}}^s = \mathbf{x}_m$
        Option II $\tilde{\mathbf{x}}^s = \frac{1}{m}\sum_{t=0}^{m-1}\mathbf{x}_t$
        Option III $\tilde{\mathbf{x}}^s = \mathbf{x}_t$ for randomly chosen $t \in \{0, 1, \ldots, m-1\}$
end for
```

SAGA:The idea of SAGA is similar to SAG except that SAGA uses a different coefficient to keep the gradient estimator unbiased. SAGA works as follows: $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma\Big[(\nabla f_{i_t}(\mathbf{x}_t) - \mathbf{v}_{i_t}^{t-1}) + \frac{1}{n}\sum_{i=1}^n\mathbf{v}_i^{t-1}\Big]$.
SVRG: The idea of the algorithm is to use fixed reference point to build the variance-reduced gradient: $\mathbf{g}_t = \nabla f_{i_t}(\mathbf{x}_t) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \nabla F(\tilde{x})$, where the reference point $\tilde{\mathbf{x}}$ is only updated once a while.
Convergence: Assume $f_i(\mathbf{x})$ is convex and L-smooth and $F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$ is μ-strongly convex. Let $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$, assume that m is sufficiently large, and $\eta \leq \frac{1}{2L}$, so that $\rho = \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1$, then then we have geometric convergence in expectation for SVRG under Option II and III: $\mathbb{E}[F(\tilde{\mathbf{x}}^s) - F(\mathbf{x}^*)] \leq \rho^s[F(\tilde{\mathbf{x}}^0) - F(\mathbf{x}^*)]$. The trick is to prove the lemma: for any $\mathbf{x}$ we have: $\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}^*)\|_2^2 \leq 2L(F(\mathbf{x}) - F(\mathbf{x}^*))$. $\kappa_{\max} := \frac{L_{\max}}{\mu}$.

14 Min-Max Optimization

We consider the **min-max problem** of the form $\min_{\mathbf{x}\in\mathcal{X}} \max \mathbf{y} \in \mathcal{Y}\phi(\mathbf{x},\mathbf{y})$.

We say that $(\mathbf{x}^*,\mathbf{y}^*)$ is a **saddle point** if $\phi(\mathbf{x}^*,\mathbf{y}) \leq \phi(\mathbf{x}^*,\mathbf{y}^*) \leq \phi(\mathbf{x},\mathbf{y}^*)$, for any $\mathbf{x} \in \mathcal{X}$ and $\mathbf{y} \in \mathcal{Y}$. This can be seen as a **Nash equilibrium** in a simiItaneous game, where no player has an incentive to make unilateral change at the NE.

We say that $(\mathbf{x}^*,\mathbf{y}^*)$ is a **global minimax point** if $\phi(\mathbf{x}^*,\mathbf{y}) \leq \phi(\mathbf{x}^*,\mathbf{y}^*) \leq \max_{\mathbf{y}'\in\mathcal{Y}}\phi(\mathbf{x},\mathbf{y}')$ for any $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$. This can be seen as a **Stackelberg equilibrium** in a sequential game (best reponse to best reponse).

Next we define the **primal** and **dual** problems induced by the minimax optimization problem: $\mathrm{Opt}(P) = \min_{\mathbf{x}\in\mathcal{X}}\bar{\phi}(\mathbf{x})$, with $\bar{\phi}(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}}\phi(\mathbf{x},\mathbf{y})$ and $\mathrm{Opt}(D) = \min_{\mathbf{x}\in\mathcal{X}}\underline{\phi}(\mathbf{x})$, with $\underline{\phi}(\mathbf{x}) = \max_{\mathbf{y}\in\mathcal{Y}}\phi(\mathbf{x},\mathbf{y})$
Notice that weak duality holds ($\mathrm{Opt}(D) \leq \mathrm{Opt}(P)$) and hence: $\max_{\mathbf{y}\in\mathcal{Y}}\min_{\mathbf{x}\in X}\phi(\mathbf{x},\mathbf{y}) \leq \min_{\mathbf{x}\in X}\max_{\mathbf{y}\in\mathcal{Y}}\phi(\mathbf{x},\mathbf{y})$.
Existence of saddle point: The point $(\mathbf{x}^*,\mathbf{y}^*)$ is a saddle point of $\phi(\mathbf{x},\mathbf{y})$ if and only if $\max_{\mathbf{y}\in\mathcal{Y}}\min_{\mathbf{x}\in X}\phi(\mathbf{x},\mathbf{y}) = \min_{\mathbf{x}\in X}\max_{\mathbf{y}\in\mathcal{Y}}\phi(\mathbf{x},\mathbf{y})$ and $\mathbf{x}^* \in \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\bar{\phi}(\mathbf{x})$ and $\mathbf{y}^* \in \operatorname{argmax}_{\mathbf{y}\in\mathcal{Y}}\underline{\phi}(\mathbf{y})$. In other words, $(\mathbf{x}^*,\mathbf{y}^*)$ is a saddle point if and only if strong duality holds and $\mathbf{x}^*$, $\mathbf{y}^*$ are respectively the optimal solutions to the induced primal problem (P) and the dual problem (D).
Remark: A saddle point, if it exists, is also a global minimax point.And there is no advantage to the players of knowing the opponent's choice or to play second. The minimax, maximin, and the equilibrium all give the same payoff.

Minimax Theorem

von Neumann's Minimax theorem: For any payoff matrix $A \in \mathbb{R}^{m\times n}$: $\min_{\mathbf{x}\in\Delta_m}\max_{\mathbf{y}\in\Delta_n}\mathbf{x}^T A\mathbf{y} = \max_{\mathbf{y}\in\Delta_n}\min_{\mathbf{x}\in\Delta_m}\mathbf{x}^T A\mathbf{y}$, where $\Delta_m = \{\mathbf{x} \in \mathbb{R}_+^m : \sum_{i=1}^m x_i = 1\}, \Delta_n = \{\mathbf{y} \in \mathbb{R}_+^n : \sum_{i=1}^m y_i = 1\}$.

Sion-Kakutani Minimax theorem: Let sets $\mathcal{X} \subseteq \mathbb{R}^m$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ be two convex compact sets. Let $\phi(\mathbf{x},\mathbf{y}) : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ be a continuous function such that for any fixed $\mathbf{y} \in \mathcal{Y}$ it is convex in $\mathbf{x}$ and for any fixed $\mathbf{x} \in \mathcal{X}$ it is convex in $\mathbf{y}$, we call such a function **convex-concave**. Then $\phi(\mathbf{x},\mathbf{y})$ has a saddle point on $\mathcal{X}\times\mathcal{Y}$ and $\max_{\mathbf{y}\in\mathcal{Y}}\min_{\mathbf{x}\in X}\phi(\mathbf{x},\mathbf{y}) = \min_{\mathbf{x}\in X}\max_{\mathbf{y}\in\mathcal{Y}}\phi(\mathbf{x},\mathbf{y})$.

A function $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ is **strongly-convex-strongly-concave** if there exist constants $\mu_1,\mu_2 > 0$ such that: $\phi(\mathbf{x},\mathbf{y})$ is μ_1-strongly convex in $\mathbf{x} \in \mathcal{X}$ for every fixed $\mathbf{y} \in \mathcal{Y}$; $\phi(\mathbf{x},\mathbf{y})$ is μ_2-strongly concave in $\mathbf{y} \in \mathcal{Y}$ for every fixed $\mathbf{x} \in \mathcal{X}$, namely for any $\mathbf{x},\mathbf{x}_1,\mathbf{x}_2 \in \mathcal{X}$ and $\mathbf{y},\mathbf{y}_1,\mathbf{y}_2 \in \mathcal{Y}$:

(i) $\phi(\mathbf{x}_1,\mathbf{y}) \geq \phi(\mathbf{x}_2,\mathbf{y}) + \nabla_{\mathbf{x}}\phi(\mathbf{x}_2,\mathbf{y})^T(\mathbf{x}_1 - \mathbf{x}_2) + \frac{\mu_1}{2}\|\mathbf{x}_1 - \mathbf{x}_2\|^2$
(ii) $-\phi(\mathbf{x},\mathbf{y}_1) \geq -\phi(\mathbf{x},\mathbf{y}_2) - \nabla_{\mathbf{y}}\phi(\mathbf{x},\mathbf{y}_2)^T(\mathbf{y}_1 - \mathbf{y}_2) + \frac{\mu_2}{2}\|\mathbf{y}_1 - \mathbf{y}_2\|^2$

A function $\phi : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ is **L-Lipschitz smooth jointly in $\mathbf{x}$ and $\mathbf{y}$** if for any $\mathbf{x}_1,\mathbf{x}_2 \in \mathcal{X}$ and $\mathbf{y}_1,\mathbf{y}_2 \in \mathcal{Y}$:

(i) $\|\nabla_{\mathbf{x}}\phi(\mathbf{x}_1,\mathbf{y}_1) - \nabla_{\mathbf{x}}\phi(\mathbf{x}_2,\mathbf{y}_2)\| \leq L(\|\mathbf{x}_1 - \mathbf{x}_2\| + \|\mathbf{y}_1 - \mathbf{y}_2\|)$
(ii) $\|\nabla_{\mathbf{y}}\phi(\mathbf{x}_1,\mathbf{y}_1) - \nabla_{\mathbf{y}}\phi(\mathbf{x}_2,\mathbf{y}_2)\| \leq L(\|\mathbf{x}_1 - \mathbf{x}_2\| + \|\mathbf{y}_1 - \mathbf{y}_2\|)$

We measure the optimality via the **duality gap**: $\text{duality gap} = \max_{\mathbf{y}\in\mathcal{Y}}\phi(\hat{\mathbf{x}},\mathbf{y}) - \min_{\mathbf{x}\in\mathcal{X}}\phi(\mathbf{x},\hat{\mathbf{y}}) \geq 0$. When duality gap $= 0$ we have a saddle-point. If duality gap $\leq \epsilon$ we have an ϵ-saddle point.

Algorithms

Gradient Descent Ascent: The algorithm updates x and y simultaneuous at each iteration using only the gradient information: $\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta\nabla_{\mathbf{x}}\phi(\mathbf{x}_t,\mathbf{y}_t))$ and $\mathbf{y}_{t+1} = \Pi_{\mathcal{Y}}(\mathbf{y}_t - \eta\nabla_{\mathbf{y}}\phi(\mathbf{x}_t,\mathbf{y}_t))$.

Convergence of GDA: Assume that ϕ is μ-strongly-convex-strongly-concave and jointly smooth with parameter L, then GDA with stepsize $\eta < \frac{\mu}{2L^2}$ converges linearly: $\left\|\mathbf{x}_{t+1} - \mathbf{x}^*\right\|^2 + \left\|\mathbf{y}_{t+1} - \mathbf{y}^*\right\| \leq (1 + 4\eta^2L^2 - 2\eta\mu)(\|\mathbf{x}_t - \mathbf{x}^*\|^2 + \|\mathbf{y}_t - \mathbf{y}^*\|^2)$. With stepsize $\eta = \frac{\mu}{4L^2}$ this yields: $\|\mathbf{x}_T - \mathbf{x}^*\|^2 + \|\mathbf{y}_T - \mathbf{y}^*\|^2 \leq (1 - \frac{\mu^2}{4L^2})^T(\|\mathbf{x}_0 - \mathbf{x}^*\|^2 + \|\mathbf{y}_0 - \mathbf{y}^*\|^2)$.
Careful: GDA with constant stepsize may not converge for general convex-concave function. Consider the function $\phi(x,y) = xy$.

Extragradient Method (EG): The main idea of EG is to use the gradient at the current point to find a mid-point, and then use the gradient at that mid-point to find the next iterate:
$\mathbf{x}_{t+\frac{1}{2}} = \Pi_{\mathcal{X}}(\mathbf{x}_t - \eta\nabla_{\mathbf{x}}\phi(\mathbf{x}_t,\mathbf{y}_t)), \mathbf{y}_{t+1} = \Pi_{\mathcal{Y}}(\mathbf{y}_t + \eta\nabla_{\mathbf{y}}\phi(\mathbf{x}_t,\mathbf{y}_t))$,
$\mathbf{x}_{t+1} = \Pi_{\mathcal{X}}\left(\mathbf{x}_t - \eta\nabla_{\mathbf{x}}\phi(\mathbf{x}_{t+\frac{1}{2}},\mathbf{y}_{t+\frac{1}{2}})\right), \mathbf{y}_{t+1} = \Pi_{\mathcal{Y}}\left(\mathbf{y}_t - \eta\nabla_{\mathbf{y}}\phi(\mathbf{x}_{t+\frac{1}{2}},\mathbf{y}_{t+\frac{1}{2}})\right)$.

Convergence of EG: Assume that $\mathcal{D}_{\mathcal{X}} := \max_{\mathbf{x},\mathbf{x}'}\left\|\mathbf{x} - \mathbf{x}'\right\| < \infty$ and $\mathcal{D}_{\mathcal{Y}} := \max_{\mathbf{y},\mathbf{y}'}\left\|\mathbf{y} - \mathbf{y}'\right\| < \infty$. Assume that ϕ is convex-concave and jointly L-smooth then EG with stepsize $\eta \leq \frac{1}{2L}$ satisfies by denoting $\hat{\mathbf{x}}_T = \frac{1}{T}\sum_{t=1}^T\mathbf{x}_{t+\frac{1}{2}}$ and $\hat{\mathbf{y}}_T = \frac{1}{T}\sum_{t=1}^T\mathbf{y}_{t+\frac{1}{2}}$, setting $\eta = \frac{1}{2L}$, this implies that the duality gap: $\epsilon_{\mathrm{sad}}(\hat{\mathbf{x}}_T,\hat{\mathbf{y}}_T) \leq \frac{D_{\mathcal{X}}^2 + D_{\mathcal{Y}}^2)}{2\eta T} = \frac{L(D_{\mathcal{X}}^2 + D_{\mathcal{Y}}^2)}{T}$.
Connections to Proximal Point Algorithm (PPA): At each iteration, PPA performs the update: $(\mathbf{x}_{t+1},\mathbf{y}_{t+1}) = \operatorname{argmin}_{\mathbf{x}\in\mathcal{X}}\operatorname{argmax}_{\mathbf{y}\in\mathcal{Y}}\{\phi(\mathbf{x},\mathbf{y}) + \frac{1}{2\eta}\|\mathbf{x} - \mathbf{x}_t\|^2 - \frac{1}{2\eta}\|\mathbf{y} - \mathbf{y}_t\|^2\}$.

Variational Inequalities

Variational Inequality (VI) Problem: Let $\mathcal{Z} \subset \mathbb{R}^d$ be a nonempty subset and consider mapping $F : \mathcal{Z} \to \mathbb{R}^d$. The goal of a VI is to find a (strong) solution $\mathbf{z}^* \in \mathcal{Z}$ such that $\langle F(\mathbf{z}^*),\mathbf{z} - \mathbf{z}^*\rangle \geq 0$ for all $\mathbf{z} \in \mathcal{Z}$. This is known as the **Stampacchia Variational Inequality (SVI)**. A closely relevant problem is the **Minty Variational Inequality (MVI)**, which aims to find a (weak) solution $\mathbf{z}^*$ such that $\langle F(\mathbf{z}),\mathbf{z} - \mathbf{z}^*\rangle \geq 0$ for all $\mathbf{z} \in \mathcal{Z}$.

An operator $F : \mathcal{Z} \to \mathbb{R}^d$ is said to be **monotone** if $\langle F(\mathbf{u}) - f(\mathbf{v}),\mathbf{u} - \mathbf{v}\rangle \geq 0$, $\forall\mathbf{u},\mathbf{v} \in \mathcal{Z}$, it is said to be **$\mu$-strongly monotone** with modulus $\mu > 0$ if $\langle F(\mathbf{u}) - f(\mathbf{v}),\mathbf{u} - \mathbf{v}\rangle \geq \mu\|\mathbf{u} - \mathbf{v}\|^2$, $\forall\mathbf{u},\mathbf{v} \in \mathcal{Z}$. **Equivalence of SVI and MVI** *(i)* If F is monotone, then a solution to SVI is also a solution to MVI. *(ii)* If F is continuous and $\mathcal{Z}$ is convex, then a solution to MVI is also a solution to SVI.
Accuracy Measure: A natural inaccuracy measure if a candidate solution $\hat{\mathbf{z}}$ to MVI is the dual gap function: $\epsilon_{\mathrm{VI}}(\hat{\mathbf{z}}) := \max_{\mathbf{z}\in\mathcal{Z}}\langle F(\mathbf{z}),\hat{\mathbf{z}} - \mathbf{z}\rangle$.
Extragradient Method (EG) and Optimistic Gradient Descent Ascent (OGDA) can be directly extended to solving VIs.