Convex optimization: homework

Version: January 10, 2025 (typos/modifications since 1st version will be corrected in red)

1 Generalities

Each student should submit a short report with his own answers to the exercises below. The report can take (entirely or partially) the form of a (commented) jupyter notebook if it is convenient for you.

The deadline for the assignment is on **20th of February 2025**. To report typos or ask questions if something is unclear, please reach me at `adrien.taylor@inria.fr`.

2 Exercises

Convex sets. Which of the following sets are convex? Provide either a proof or a counter-example.

- an hyperbolic set: $\{x \in \mathbb{R}^2_+ \;:\; x_1 x_2 \geqslant 1\}$.
- the set of points closer to one set than another $\{x \;:\; \mathrm{dist}(x, S) \leqslant \mathrm{dist}(x, T)\}$, with $S, T \subseteq \mathbb{R}^n$ and $\mathrm{dist}(x, S) = \inf_{z \in S} \|x - z\|_2$.
- the set $\{x \;:\; x + S_2 \subseteq S_1\}$ with $S_1, S_2 \subseteq \mathbb{R}^n$ and S_1 convex.
- the set $\{x \;:\; \exists y \in S_2,\, x + y \in S_1\}$ with $S_1, S_2 \subseteq \mathbb{R}^n$, S_1, S_2 convex.

Convex functions. Determine which of those functions are convex, concave, quasi-convex or quasi-concave.

- $f(x_1, x_2) = x_1 x_2$ on $\mathbb{R}^2$,
- $f(x_1, x_2) = x_1 x_2$ on $\mathbb{R}^2_{++}$,
- $f(x_1, x_2) = x_1 / x_2$ on $\mathbb{R}^2_{++}$,
- $f(X) = \mathrm{Tr}(X^{-1})$ on $\mathbb{S}^n_{++}$ (hint: prove convexity along lines: let $X \in \mathbb{S}^n_{++}$ and $Y \in \mathbb{S}^n$ define $S(t) = X + tY$ and proceed).

Fenchel conjugation.

- (Fenchel conjugate) Compute the Fenchel conjugate of the squared ℓ_2-norm $\|x\|_2^2$ and that of the ℓ_1-norm $\|x\|_1$.
- (Infimal convolution and smoothing) An extremely useful operation related to the Fenchel conjugation is the *infimal convolution*. Relate the Fenchel conjugate of $f(x) = \min_{u+v=x}\{g(u) + h(v)\}$ to the conjugates of g and h.

  The infimal convolution is often used for creating *smoothed approximations* to nonsmooth functions by convolution with the squared ℓ_2-norm. Let $g(x) = \|x\|_1$ and $h(x) = \frac{1}{2\alpha}\|x\|_2^2$. Compute the infimal convolution $f(x) = \min_{u+v=x}\{g(u) + h(v)\}$ and relate it to the biconjugate f^{**}. (Hint: work componentwise)
- (Fenchel conjugate) Compute the Fenchel conjugate of the logistic loss: $\log(1 + \exp(x))$.

Duality.

- Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Show that $\max_{\lambda\in\mathbb{R}^m} -\frac{1}{2}\|\lambda - b\|_2^2 + \frac{1}{2}\|b\|_2^2 - i_{\{\|\cdot\|_\infty \leqslant 1\}}\left(\frac{A^T\lambda}{\alpha}\right)$ is a dual for the LASSO: $\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \alpha\|x\|_1$.
- Let $h_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \dots, m)$ and the problem $\min_{w\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^m h_i(w) + \frac{\lambda}{2}\|w\|_2^2$. Compute the Lagrange dual of the problem by introducing the following intermediate constrained problem:
$$\min_{\substack{w\in\mathbb{R}^n\\ w_1,\dots,w_m\in\mathbb{R}^n}} \left\{ \frac{1}{m}\sum_{i=1}^m h_i(w_i) + \frac{\lambda}{2}\|w\|_2^2 \text{ s.t. } w_i = w \quad (i = 1, \dots, m)\right\}.$$
Apply this result to the regularized logistic regression problem
$$\min_{w\in\mathbb{R}^n} \frac{1}{m}\sum_{i=1}^m \log\left(1 + \exp\left(-y_i x_i^T w\right)\right) + \frac{\lambda}{2}\|w\|_2^2,$$
where $(x_1, y_1), \dots, (x_m, y_m) \in \mathbb{R}^n \times \{-1, 1\}$ is a dataset (consisting of m points within two categories depending on whether $y_i = 1$ or $y_i = -1$).
- Let $A_0, A_1, \dots, A_m \in \mathbb{S}^n$, $b_1, \dots, b_m \in \mathbb{R}$ and consider the problem
$$\min_{X\in\mathbb{S}^n} \{\mathrm{Tr}(A_0X) \quad \text{s.t. } X \succcurlyeq 0,\ \mathrm{Tr}(A_1X) = b_1, \dots,\ \mathrm{Tr}(A_mX) = b_m\}.$$
What is the Lagrange dual to this problem? (denote by λ_i the dual variable associated to the constraint $\mathrm{Tr}(A_iX) = b_i$, and by S the dual variable associated to $X \succcurlyeq 0$). Assume that both primal and dual optimal values are attained (by respectively $X_\star$ and $S_\star$): show that $\mathrm{Tr}(S_\star X_\star) = 0$ if and only if strong duality holds.

3 Classification via support vector machines

Problem statement. Let $\{(x_i, y_i)\}_{i=1,\dots,m} \in \mathbb{R}^n \times \{-1, 1\}$ be a set of measurements classified in two categories (respectively denoted by $y_i = 1$ or $y_i = -1$). For convenience, we assume that the last entry of each x_i is 1 (to handle the constant part of the affine functions below). Our goal is to find a (linear) predicting function $f : \mathbb{R}^n \to \mathbb{R}$ for being able to predict the classification of a new entry. For doing that, we use the linear *support vector machine* (SVM) framework, which aims at finding an hyperplane $w^Tx = 0$ separating the two sets (recall that the last component of x is 1, so that w^Tx is in fact affine). More precisely, we consider the (linear) soft-margin SVM problem (parametrized by some $\alpha > 0$) given by
$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\|w\|_2^2 + \alpha\sum_{i=1}^m \max\left\{0, 1 - w^Tx_iy_i\right\} \tag{SVM}$$
Remark: it is not expected that you spent too much time running methods that are too slow. If any method takes more than a few minutes to run, you can consider that it timeouts for this homework.

1. Dual problem. For convenience, denote by $X = [y_1x_1 \,|\, y_2x_2 \,|\, \dots \,|\, y_mx_m] \in \mathbb{R}^{n\times m}$. Show that (SVM) can be reformulated as
$$\begin{aligned} \min_{\substack{w\in\mathbb{R}^n\\ s\in\mathbb{R}^m}}\ & \frac{1}{2}\|w\|_2^2 + \alpha\sum_{i=1}^m s_i \\ \text{s.t.}\ & w^Tx_iy_i \geqslant 1 - s_i \\ & s_i \geqslant 0 \end{aligned}$$
and that this formulation yields the following Lagrange dual:
$$\max_{0\leqslant\lambda\leqslant\alpha} \left\{ D(\lambda) \triangleq -\frac{1}{2}\lambda^TX^TX\lambda + \sum_{i=1}^m \lambda_i \right\} \tag{Dual-SVM}$$
and a natural estimate of the primal variable $w = \sum_{i=1}^m \lambda_ix_iy_i = X\lambda$.

2. Algorithms. We propose two natural ways (there are many others, including incorporating momentum into the following methods), namely a **dual projected gradient ascent** and a **dual randomized coordinate ascent**. The iterates of the different algorithms are written with an exponent as $\lambda^{(k)}$ (kth iterate for the dual variable).

- Projected gradient ascent for (Dual-SVM) is given by

$$\lambda^{(k+1)} = \mathrm{Proj}_{[0,\alpha]}\left(\lambda^{(k)} + h^{(k)}\nabla D(\lambda^{(k)})\right),$$

  for some initial $\lambda^{(0)} \in \mathbb{R}^m$ (typically zero) and for some step size $h^{(k)} > 0$. A typical choice for $h^{(k)}$ is $h^{(k)} = \left(\lambda_{\max}(X^TX)\right)^{-1}$. A common alternative consists in performing a backtracking line-search ensuring sufficient increase as follows: set up an initial guess $h^{(0)} > 0$ and run:

$$\begin{aligned}
&\text{For } k = 0, 1, 2, \dots\\
&\quad \text{(a) } \lambda^{(k+1)} = \mathrm{Proj}_{[0,\alpha]}\left(\lambda^{(k)} + h^{(k)}\nabla D(\lambda^{(k)})\right)\\
&\quad \text{(b) If } D(\lambda^{(k+1)}) \leqslant D(\lambda^{(k)}) + \frac{1}{2h^{(k)}}\|\lambda^{(k+1)} - \lambda^{(k)}\|_2^2\\
&\qquad \text{\# the increment is not large enough, so we decrease the step-size:}\\
&\qquad h^{(k)} \leftarrow h^{(k)}/2 \text{ and return to (a) without incrementing } k\\
&\quad (c) \text{ Else:}\\
&\qquad \text{\# The increment is large enough: proceed with next iteration}\\
&\qquad h^{(k+1)} \leftarrow h^{(k)}.
\end{aligned} \tag{ProjGrad}$$

  Note that the operation $\mathrm{Proj}_{[0,\alpha]}(\cdot)$ corresponds to a (componentwise) projection on $[0, \alpha]$.

- Randomized coordinate ascent for (Dual-SVM) consists in optimizing (exactly) one dimension (corresponding to the i_kth dual coordinate) at a time (for the record: $\lambda_i^{(k)}$ denotes the ith component of the kth iterate for the dual variable). Show that the initial $w^{(0)} = 0$ and $\lambda^{(0)} = 0$ with the update rule

$$\begin{aligned}
&\text{For } k = 0, 1, 2, \dots\\
&\quad \text{Pick } i_k \in \{1, 2, \dots, m\} \text{ (uniformly at random)}\\
&\quad \bar{\lambda} = \lambda_{i_k}^{(k)}\\
&\quad \lambda_{i_k}^{(k+1)} = \mathrm{Proj}_{[0,\alpha]}\left(\lambda_{i_k}^{(k)} + \frac{1 - y_{i_k}x_{i_k}^T w^{(k)}}{\|x_{i_k}\|^2}\right)\\
&\quad w^{(k+1)} = w^{(k)} + y_{i_k}x_{i_k}(\lambda_{i_k}^{(k+1)} - \bar{\lambda})
\end{aligned} \tag{DCA}$$

  optimizes one (random) dimension at a time, exactly.

Experiment on synthetic data that you can visualize.

3. Experiments. Experiment with one (or a few) datasets, for instance from the LIBSVM library (or see Git) that contains many test datasets for SVMs.

Your implementation can be tested on a few datasets that are well-suited for linear SVMs, such as the (tiny) sonar dataset and the (small) mushroom dataset. Larger datasets can be tested as well, e.g., the news20 or the real-sim ones.

Report on your experiments depending on what you deem relevant here (e.g., timeouts, overflows, plots, tables, timings, numbers of misclassified samples, primal-dual gaps, etc.).

4. Formulation as a linear program. Show that the ℓ_1 version of the SVM problem

$$\min_{w \in \mathbb{R}^n} \|w\|_1 + \alpha \sum_{i=1}^{m} \max\left\{0, 1 - w^T x_i y_i\right\} \qquad (\ell_1\text{-SVM})$$

can be formulated as a linear optimization problem. Use an off-the-shelf solver (e.g., download and use CVXPY to interface with different solvers—see linear programming and this example for help) to solve the ℓ_1 problem. What value do you obtain? How does the classifier behave as compared to your home-made solution above, on the same datasets?

5. An interior-point strategy for linear SVMs. Using the formulation as a linear program, propose an interior-point strategy for solving the problem. What subproblems do you have to solve, and what is the computational cost of an iteration?