ECE236C. Optimization Methods for Large-Scale Systems
Exercises

Lieven Vandenberghe

June 7, 2022

1 *Barzilai–Borwein step sizes.* Consider the gradient method

$$x_{k+1} = x_k - t_k \nabla f(x_k).$$

We assume f is convex and differentiable, with $\mathbf{dom}\, f = \mathbf{R}^n$, and that ∇f is Lipschitz continuous with respect to a norm $\|\cdot\|$:

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\| \quad \text{for all } x,\, y,$$

where L is a positive constant. Define

$$s_k = x_k - x_{k-1}, \qquad y_k = \nabla f(x_k) - \nabla f(x_{k-1})$$

and assume $y_k \neq 0$. Use the properties in lecture 1 (pages 1.10–1.15) to show that the following two choices for t_k satisfy $t_k \geq 1/L$:

$$t_k = \frac{\|s_k\|^2}{s_k^T y_k}, \qquad t_k = \frac{s_k^T y_k}{\|y_k\|_*^2}.$$

2 *Heavy-ball method* [Polyak]. We consider a "two-step" variant of the gradient method:

$$x_{k+1} = x_k - t\nabla f(x_k) + s(x_k - x_{k-1}), \quad k = 1, 2, \ldots,$$

with $x_1 = x_0$. The step sizes t and s are fixed. The term $s(x_k - x_{k-1})$ is a *momentum* term added to suppress the typical zigzagging in the gradient method.

We examine the convergence of the method applied to a strictly convex quadratic function $f(x) = (1/2)x^T Ax + b^T x + c$. The notation m and L will be used for the smallest and largest eigenvalues of the symmetric positive definite matrix A:

$$m = \lambda_{\min}(A) > 0, \qquad L = \lambda_{\max}(A) \geq m.$$

(a) Verify that the iteration can be written as a linear recursion

$$z_{k+1} = Mz_k + q, \quad k = 1, 2, \ldots,$$

where

$$z_k = \begin{bmatrix} x_k \\ x_{k-1} \end{bmatrix}, \qquad M = \begin{bmatrix} (1+s)I - tA & -sI \\ I & 0 \end{bmatrix}, \qquad q = \begin{bmatrix} -tb \\ 0 \end{bmatrix}.$$

If the sequence converges, the limit $z^\star = Mz^\star + q$ is $z^\star = (-A^{-1}b, -A^{-1}b)$.

(b) The speed of convergence depends on the spectral radius $\rho(M)$ of the matrix M. The spectral radius of a matrix is the largest absolute value of its eigenvalues. If $\rho(M) < 1$, then the iterates z_k converge to $z^\star$. For large k the distance $\|z_k - z^\star\|$ decreases as $\rho(M)^k$.

Express the eigenvalues of M in terms of the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A. Show that $\rho(M) = \sqrt{s}$ if

$$s < 1, \qquad \frac{(1-\sqrt{s})^2}{m} \leq t \leq \frac{(1+\sqrt{s})^2}{L}. \tag{1}$$

(c) Find s, t that minimize the spectral radius subject to the constraints (1). Show that for the optimal step sizes,

$$\rho(M) = \frac{\sqrt{L} - \sqrt{m}}{\sqrt{L} + \sqrt{m}} = \frac{\sqrt{\gamma} - 1}{\sqrt{\gamma} + 1},$$

where $\gamma = L/m$. Compare this with the linear convergence rate

$$\|x_k - x^\star\|_2 \leq \left(\frac{\gamma - 1}{\gamma + 1}\right)^k \|x_0 - x^\star\|_2$$

of the gradient method (page 1.31 of the lecture notes).

3 Let $F(x) = Ax + b$ be an affine function, with A an $n \times n$-matrix. What properties of the matrix A correspond to the following conditions (a)–(e) on F? Distinguish three cases for each subproblem: (1) A is symmetric, so $F(x)$ is the gradient of a quadratic function, (2) A is skew-symmetric ($A + A^T = 0$), and (3) A is a general non-symmetric matrix.

(a) *Monotonicity:*

$$(F(x) - F(y))^T(x - y) \geq 0 \quad \text{for all } x, y.$$

(b) *Strict monotonicity:*

$$(F(x) - F(y))^T(x - y) > 0 \quad \text{for all } x \text{ and } y \neq x.$$

(c) *Strong monotonicity (for the Euclidean norm):*

$$(F(x) - F(y))^T(x - y) \geq m\|x - y\|_2^2 \quad \text{for all } x, y,$$

where m is a positive constant.

(d) *Lipschitz continuity (for the Euclidean norm):*

$$\|F(x) - F(y)\|_2 \leq L\|x - y\|_2 \quad \text{for all } x,\, y,$$

where L is a positive constant.

(e) *Co-coercivity (for the Euclidean norm):*

$$(F(x) - F(y))^T(x - y) \geq \frac{1}{L}\|F(x) - F(y)\|_2^2 \quad \text{for all } x,\, y,$$

where L is a positive constant.

4 For each of the following convex functions on $\mathbf{R}^n$, explain how to calculate a subgradient at a given x.

(a) $f(x) = \sup_{0 \leq t \leq 1} p(t)$ where $p(t) = x_1 + x_2 t + \cdots + x_n t^{n-1}$.

(b) $f(x) = x_{[1]} + x_{[2]} + \cdots + x_{[k]}$ where $x_{[i]}$ denotes the ith largest element of x.

(c) $f(x) = \|Ax - b\|_2 + \|x\|_2$ where $A \in \mathbf{R}^{m \times n}$.

(d) $f(x) = \lambda_{\max}(W + \mathbf{diag}(x))$ where $W \in \mathbf{S}^n$.

(e) $f(x) = \inf_{y \in \mathbf{R}} \|x - y\mathbf{1}\|_1 = \|x - \mathrm{med}(x)\mathbf{1}\|_1$, where $\mathrm{med}(x)$ is the median of the elements of x.

(f) $f(x) = \inf_y \|Ay - x\|_\infty$ where $A \in \mathbf{R}^{n \times m}$.

(g) $f(x) = \sup_{Ay \preceq b} x^T y$, where $A \in \mathbf{R}^{m \times n}$ and the polyhedron defined by $Ay \preceq b$ is nonempty and bounded.

5 *Relaxation method for linear inequalities* [Agmon, Motzkin, and Schoenberg]. We consider the problem of solving a set of linear inequalities $a_i^T x \leq b_i$, $i = 1, \ldots, m$. We assume that the inequalities are strictly feasible, and that $a_i \neq 0$ for all i. The problem is a special case of the problem on page 3.12 of the lecture notes, where C_i is the halfspace

$$C_i = \{x \mid a_i^T x \leq b_i\}, \quad i = 1, \ldots, m.$$

As in the lecture notes, we denote by $f_i(x)$ the Euclidean distance of x to C_i, and by $f(x)$ the maximum of $f_1(x), \ldots, f_m(x)$:

$$f_i(x) = \max\{0, \frac{a_i^T x - b_i}{\|a_i\|_2}\}, \qquad f(x) = \max\{f_1(x), \ldots, f_m(x)\}.$$

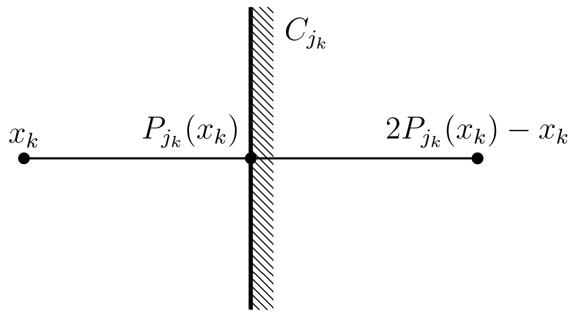
The Euclidean projection of x on the halfspace C_i is denoted by $P_i(x)$:

$$P_i(x) = x - f_i(x)\frac{a_i}{\|a_i\|_2}.$$

The subgradient method with step size $t_k = \lambda f(x_k)$ uses the iteration

$$x_{k+1} = x_k + \lambda(P_{j_k}(x_k) - x_k) \quad \text{where } j_k = \operatorname*{argmax}_{i=1,\ldots,m} \frac{a_i^T x_k - b_i}{\|a_i\|_2}, \tag{2}$$

until x_k is feasible. The constant $\lambda \in (0, 2]$ is an algorithm parameter. If $\lambda = 1$, the new point x_{k+1} is the projection of x_k on the halfspace C_{j_k} farthest from x_k. If $\lambda = 2$, the new point x_{k+1} is the reflection of x_k through the boundary hyperplane of C_{j_k}.



Algorithm (3) was proposed by Agmon, Motzkin, and Schoenberg in 1954. Other variants, with different rules to select j_k (for example, cyclic or random), have also been studied. In the neural network literature, the recursion is known as the perceptron learning algorithm for training linear classifiers.

Motzkin and Schoenberg showed that for $\lambda \in (0, 2)$ the algorithm either finds a solution in a finite number of iterations or converges to a point in the boundary of $C = \cap_{i=1,\ldots,m} C_i$. For $\lambda = 2$ they showed that the algorithm finds a solution in a finite number of iterations. The following is an outline of the proof with some questions to complete.

(a) Show that the projection $P_i(x)$ on the halfspace C_i satisfies the property

$$\|z - P_i(x)\|_2 \leq \|z - x\|_2 \quad \text{for all } z \in C_i.$$

Use this to show that the iterates (3) satisfy

$$\|z - x_{k+1}\|_2 \leq \|z - x_k\|_2 \quad \text{for all } z \in C.$$

(b) We use the result in part (a) to show that the sequence x_k converges.

A first consequence of (a) is that the iterates x_k are bounded. A standard result from analysis says that every bounded sequence has at least one limit point (a limit of a converging subsequence). To show that the entire sequence converges we show that there is at most one limit point. Consider any $z \in C$. From part (a) the distances $\|x_k - z\|_2$ form a nonincreasing sequence of nonnegative numbers. Therefore this sequence converges to a limit, which we denote by $r(z) = \lim_{k\to\infty} \|x_k - z\|_2$. Every limit point of the sequence x_k must lie on the sphere $\{x \mid \|x - z\|_2 = r(z)\}$. Now suppose $\hat{x}$ and $\tilde{x}$ are two distinct limit points of the sequence x_k. Since $\|\hat{x} - z\|_2 = \|\tilde{x} - z\|_2 = r(z)$, the point z is at the same distance from $\hat{x}$ and $\tilde{x}$. This is true for any $z \in C$. Explain why this contradicts the assumption that the inequalities are strictly feasible, *i.e.*, the polyhedron C has nonempty interior.

(c) Let $\bar{x}$ be the limit of x_k. We show that $\bar{x} \in C$. The iteration (3) satisfies

$$f(x_k) = \frac{\|x_{k+1} - x_k\|_2}{\lambda}.$$

Since x_k converges, $\lim_{k\to\infty} f(x_k) = 0$. Since the function f is continuous, $f(\bar{x}) = \lim_{k\to\infty} f(x_k) = 0$. Hence $\bar{x} \in C$.

(d) In the last part of the problem we show that if $\lambda = 2$, then $x_k \in C$ after a finite number of iterations. We prove this by contradiction. Suppose $x_k \notin C$ for all k, and let j_k be the index of the halfspace selected in iteration k of (3). Verify that

$$\frac{|a_{j_k}^T \bar{x} - b_{j_k}|}{\|a_{j_k}\|_2} \leq \frac{\|x_{k+1} - x_k\|_2}{2} + \frac{|a_{j_k}^T (\bar{x} - x_k)|}{\|a_{j_k}\|_2}.$$

The left-hand side is the distance of $\bar{x}$ to $H_{j_k} = \{x \mid a_{j_k}^T x = b_{j_k}\}$. The right-hand side converges to zero as $k \to \infty$. Since j_k is chosen from a finite set $\{1, \ldots, m\}$, we must have $\bar{x} \in H_{j_k}$ for all k after some finite number of iterations K. Show that this implies that $\|\bar{x} - x_{k+1}\|_2 = \|\bar{x} - x_k\|_2$ for all $k \geq K$, and therefore $\|\bar{x} - x_k\|_2$ remains constant for $k \geq K$. This contradicts the assumption that x_k is an infinite non-constant sequence with limit $\bar{x}$.

6 For each f, find the subdifferential ∂f, the conjugate f^*, the subdifferential of the conjugate ∂f^*, and verify graphically that ∂f and ∂f^* are inverses.

(a) $f(x) = \exp(|x|)$.

(b) $f(x) = -\sqrt{1 - x^2}$ with domain $[-1, 1]$.

(c) The Huber penalty

$$f(x) = \begin{cases} x^2/2 & |x| \leq 1 \\ |x| - 1/2 & |x| > 1. \end{cases}$$

(d) $f(x) = \max\{0, |x| - 1\}$.

(e) $f(x) = \log(1 + \exp(x))$.

7 Give a formula or simple algorithm for evaluating the proximal mapping

$$\mathrm{prox}_f(x) = \operatorname*{argmin}_u \left(f(u) + \frac{1}{2}\|u - x\|_2^2 \right)$$

of each of the following functions on $\mathbf{R}^n$.

(a) $f(x) = \|x\|_1$ with domain $\mathbf{dom}\, f = \{x \in \mathbf{R}^n \mid \|x\|_\infty \leq 1\}$.

(b) $f(x) = \|Ax - b\|_1$ where $AA^T = D$ with D positive diagonal.

(c) $f(x) = \max_{k=1,\ldots,n} x_k$.

(d) $f(x) = \|x\|_2$ with domain $\mathbf{R}^n_+$.

(e) The function

$$f(x) = \inf_{t \geq 0} (rt + \sum_{i=1}^{n} \max\{x_i - t, 0\})$$

where r is an integer between 1 and n. Taking the dual of the optimization problem in the definition, we can derive the equivalent expression

$$f(x) = \sum_{i=1}^{r} \max\{x_{[i]}, 0\},$$

where $x_{[1]} \geq x_{[2]} \geq \cdots \geq x_{[n]}$ are the components of x sorted in descending order.

(f) $f(x) = \|Ax\|_2$ with A nonsingular.

Hints. For the function (a) the minimization in the definition of prox_f is separable. In problem (b), combine the property on page 6.8 with the scaling rule on page 6.4. The functions in (c), (d), (e) can be expressed as support functions, and the proximal operators follow from the property on page 6.18. If we omit the $t \geq 0$ constraint in the definition of f in part (e), the answer is given in the example on page 6.18, since it can be shown that the function

$$\tilde{f}(x) = \inf_{t} (rt + \sum_{i=1}^{n} \max\{x_i - t, 0\})$$

is equal to $\tilde{f}(x) = \sum_{i=1}^{r} x_{[i]}$. The functions $\tilde{f}$ and f in part (e) are used in finance (the conditional value at risk for discrete distributions) and machine learning (ν-support vector regression and classification). The function in (f) is a norm and the proximal mapping can be computed via projection on the unit ball for the dual norm (page 6.19).

8 Give the proximal mapping of the following two functions.

(a) $f(X) = -\log\det X$ where $X \in \mathbf{S}^n$ and $\mathbf{dom}\, f = \mathbf{S}^n_{++}$.

(b) $f(X) = \|X\|_*$ where $X \in \mathbf{R}^{m\times n}$ and $\|\cdot\|_*$ is the trace norm (sum of singular values). This is the dual norm of the spectral norm (maximum singular value).

We use the Frobenius norm $\|\cdot\|_F$ to define the proximal mappings of functions of matrices:

$$\mathrm{prox}_f(X) = \operatorname*{argmin}_{U} \left(f(U) + \frac{1}{2}\|U - X\|_F^2 \right).$$

9 The Moreau envelope of a closed convex function f is defined as

$$f_{(\lambda)}(x) = \inf_u \left(f(u) + \frac{1}{2\lambda}\|u - x\|_2^2 \right)$$

(lecture 8, page 11). Prove the following formula for the proximal mapping of $f_{(\lambda)}$:

$$\mathrm{prox}_{\mu f_{(\lambda)}}(x) = \frac{\lambda}{\lambda+\mu} x + \frac{\mu}{\lambda+\mu} \mathrm{prox}_{(\lambda+\mu)f}(x).$$

As an example, applying this to $f(x) = \|x\|_1$ gives a formula for the proximal mapping of the Huber penalty. Another example (for $f(x) = \delta_C(x)$) is the formula for the prox-operator of the squared Euclidean distance on page 6.21.

10 We have discussed the following technique for smoothing a nondifferentiable convex function $f(x)$: find the conjugate $f^*(y)$, add a small strongly convex term $d(y)$ to it, and take the conjugate $(f^* + d)^*$ of the modified conjugate. The Moreau–Yosida smoothing in lecture 8 is an example with $d(y) = (t/2)\|y\|_2^2$.

In this problem, we work out two other examples. Find $(f^* + d)^*$ for the following combinations of f and d. In both problems, the variable x is an n-vector and μ is a positive constant.

(a) $f(x) = \|x\|_1$ and $d(y) = \mu \sum_{i=1}^n (1 - \sqrt{1 - y_i^2})$.

(b) $f(x) = \max_{i=1,\ldots,n} x_i$ and $d(y) = \mu(\sum_{i=1}^n y_i \log y_i + \log n)$.

11 *Projection on order cone.*

Ordering constraints $x_1 \leq x_2 \leq \cdots \leq x_n$ arise in many applications. In this problem we discuss the Euclidean projection on the cone defined by these inequalities, *i.e.*, the problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x - a\|_2^2 \\ \text{subject to} & x_1 \leq x_2 \leq \cdots \leq x_n. \end{array} \tag{3}$$

This is known in statistics as the *isotonic regression problem.* It can be written as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x - a\|_2^2 \\ \text{subject to} & Ax \preceq 0, \end{array} \tag{4}$$

where A is the $(n-1) \times n$ matrix

$$A = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{bmatrix}.$$

The following algorithm is called the *Pool Adjacent Violators Algorithm*. We use the following notation. If β is a subset of $\{1, 2, \ldots, n\}$, then a_β is the subvector of a with elements indexed by β, and $\operatorname{avg}(a_\beta)$ denotes the average of the elements of the vector a_β. Thus, if $\beta = \{2, 3, 4\}$, then

$$a_\beta = (a_2, a_3, a_4), \qquad \operatorname{avg}(a_\beta) = \frac{a_2 + a_3 + a_4}{3}.$$

Pool Adjacent Violators Algorithm. Initially, $l = 1$ and $\beta_1 = \{1\}$. For $i = 2, \ldots, n$, execute the following steps.

(a) Set $l := l + 1$ and define $\beta_l = \{i\}$.

(b) While $\operatorname{avg}(a_{\beta_{l-1}}) \geq \operatorname{avg}(a_{\beta_l})$, merge the sets β_{l-1} and β_l:

$$\beta_{l-1} := \beta_{l-1} \cup \beta_l, \qquad l := l - 1.$$

An example is shown in Table 1.

When the algorithm terminates, the sets $\beta_1, \ldots, \beta_l$ partition $\{1, 2, \ldots, n\}$. We show that the optimal solution of (5) is given by

$$x_{\beta_i} = \operatorname{avg}(a_{\beta_i})\mathbf{1}, \quad i = 1, \ldots, l. \tag{5}$$

(a) Show that x is optimal for (6) if and only if there exists an $(n-1)$-vector z with

$$Ax \preceq 0, \qquad z \succeq 0, \qquad z^T Ax = 0, \qquad x + A^T z = a.$$

(b) Verify that after cycle $i = 1, \ldots, n$ in the algorithm, the following properties hold.

(i) The sets β_i are nonempty sets of consecutive indices in $\{1, 2, \ldots, n\}$ and they follow each other, *i.e.*, $\max \beta_k + 1 = \min \beta_{k+1}$ for $k = 1, \ldots, l-1$. Together, they partition $\{1, 2, \ldots, \max \beta_l\}$.

(ii) The averages of the subvectors a_{β_k} are strictly increasing:

$$\operatorname{avg}(a_{\beta_k}) < \operatorname{avg}(a_{\beta_{k+1}}), \quad k = 1, \ldots, l-1.$$

(iii) The cumulative sums of the vectors $a_{\beta_k} - \operatorname{avg}(a_{\beta_k})\mathbf{1}$ are nonnegative:

$$\operatorname{cs}(a_{\beta_k}) \succeq 0, \quad k = 1, \ldots, l,$$

where $\operatorname{cs}(u)$ is defined as

$$\operatorname{cs}(u) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} (u - \operatorname{avg}(u)\mathbf{1}).$$

(Note that the last element of $\operatorname{cs}(u)$ is necessarily zero.)

i	Subvectors $a_{\beta_1}, \ldots, a_{\beta_l}$	Averages $\mathrm{avg}(a_{\beta_1}), \ldots, \mathrm{avg}(a_{\beta_l})$
1	$\boxed{7}$	7
2	$\boxed{7}\ \boxed{-8}$	$7, -8$
	$\boxed{7, -8}$	$-1/2$
3	$\boxed{7, -8}\ \boxed{-6}$	$-1/2, -6$
	$\boxed{7, -8, -6}$	$-7/3$
4	$\boxed{7, -8, -6}\ \boxed{18}$	$-7/3, 18$
5	$\boxed{7, -8, -6}\ \boxed{18}\ \boxed{-9}$	$-7/3, 18, -9$
	$\boxed{7, -8, -6}\ \boxed{18, -9}$	$-7/3, 9/2$
6	$\boxed{7, -8, -6}\ \boxed{18, -9}\ \boxed{4}$	$-7/3, 9/2, 4$
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}$	$-7/3, 13/3$
7	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}$	$-7/3, 13/3, 16$
8	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}\ \boxed{17}$	$-7/3, 13/3, 16, 17$
9	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}\ \boxed{17}\ \boxed{-10}$	$-7/3, 13/3, 16, 17, -10$
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}\ \boxed{17, -10}$	$-7/3, 13/3, 16, 7/2$
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16, 17, -10}$	$-7/3, 13/3, 23/3$
10	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16, 17, -10}\ \boxed{-8}$	$-7/3, 13/3, 23/3, -8$
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16, 17, -10, -8}$	$-7/3, 13/3, 15/4$
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4, 16, 17, -10, -8}$	$-7/3, 4$

Table 1: The projection of the vector $a = (7, -8, -6, 18, -9, 4, 16, 17, -10, -8)$ on the order cone is $x = (-7/3, -7/3, -7/3, 4, 4, 4, 4, 4, 4, 4)$.

(c) Show that the optimality conditions in part (a) are satisfied by the vector x defined in (7) and the $(n-1)$-vector z defined by

$$z_{\beta_i} = \mathrm{cs}(a_{\beta_i}), \quad i = 1, \ldots, l-1, \qquad (z_{\bar{\beta}_l}, 0) = \mathrm{cs}(a_{\beta_l}),$$

where $\bar{\beta}_l = \beta_l \setminus \{n\}$.

(d) Explain why the complexity of the algorithm is linear in n.

12 In the lecture we derived ADMM from the Douglas–Rachford splitting method applied to a dual problem. One can also derive the Douglas–Rachford splitting from ADMM. Show that ADMM applied to the problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(u) \\ \text{subject to} & x - u = 0 \end{array}$$

(with variables x and u) gives the Douglas–Rachford splitting method in its equivalent form on page 11.5.

13 Describe an efficient implementation of ADMM for each of the following four optimization problems with variable $x \in \mathbf{R}^n$.

We use the notation $H(x)$ for the linear function that maps an n-vector x to the $p \times q$ Hankel matrix

$$H(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_q \\ x_2 & x_3 & x_4 & \cdots & x_{q+1} \\ x_3 & x_4 & x_5 & \cdots & x_{q+2} \\ \vdots & \vdots & \vdots & & \vdots \\ x_p & x_{p+1} & x_{p+2} & \cdots & x_n \end{bmatrix},$$

for some fixed p, q with $p + q - 1 = n$. The matrix D in parts (b) and (d) is the $(n-1) \times n$ finite-difference matrix

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}.$$

The matrix norm $\|\cdot\|_*$ is the trace norm or nuclear norm (sum of the singular values). See problem 2(b) of homework 4 for the proximal mapping of the trace norm.

By "efficient implementation" we mean that the cost per iteration should be dominated by the cost of a singular value decomposition of a $p \times q$ matrix (assuming p and q are not small).

(a) Given $a \in \mathbf{R}^n$, solve

$$\text{minimize} \quad \|H(x)\|_* + \frac{1}{2}\|x - a\|_2^2.$$

(b) Given $a \in \mathbf{R}^n$ and $\gamma > 0$, solve

$$\begin{array}{ll} \text{minimize} & \|H(x)\|_* + \frac{1}{2}\|x-a\|_2^2 \\ \text{subject to} & \|Dx\|_2 \leq \gamma. \end{array}$$

(c) Given $a \in \mathbf{R}^n$, solve

$$\text{minimize} \quad \|H(x)\|_* + \|x-a\|_1.$$

(d) Given $a \in \mathbf{R}^n$ and $\gamma > 0$, solve

$$\begin{array}{ll} \text{minimize} & \|H(x)\|_* + \|x-a\|_1 \\ \text{subject to} & \|Dx\|_2 \leq \gamma. \end{array}$$

14 *Linearized ADMM.* Consider the standard problem

$$\text{minimize} \quad f(x) + g(Ax) \tag{6}$$

where f and g are closed convex functions. In lecture 12 (page 12.31) we derived the *proximal method of multipliers* from the proximal point method applied to the primal–dual optimality conditions. Here we write the proximal method of multipliers as

$$\begin{aligned} (x_{k+1}, y_{k+1}) &= \operatorname*{argmin}_{x,y} \left(f(x) + g(y) + \frac{\tau}{2}\|Ax - y + u_k\|_2^2 + \frac{1}{2\sigma}\|x - x_k\|_2^2\right) \\ u_{k+1} &= u_k + Ax_{k+1} - y_{k+1}. \end{aligned}$$

The two parameters τ and σ correspond to $\tau = \sigma = t$ on page 12.31. Using different values can be interpreted as a simple "preconditioning" of the proximal point method (see page 12.29). The variable u_k corresponds to $u_k = z_k/t$ on page 12.31.

We note that the iteration is similar to the augmented Lagrangian method, with an extra term $\|x - x_k\|_2^2$ added to the augmented Lagrangian. Motivated by the interpretation of ADMM as a simplified augmented Lagrangian method, we can replace the joint minimization over x, y by an alternating minimization:

$$\begin{aligned} x_{k+1} &= \operatorname*{argmin}_{x} \left(f(x) + \frac{\tau}{2}\|Ax - y_k + u_k\|_2^2 + \frac{1}{2\sigma}\|x - x_k\|_2^2\right) \\ y_{k+1} &= \operatorname{prox}_{(1/\tau)g}(Ax_{k+1} + u_k) \\ u_{k+1} &= u_k + Ax_{k+1} - y_{k+1}. \end{aligned}$$

For general f and A, the optimization problem in the x-update may be expensive, because the second term in the cost function contributes a quadratic term x^TA^TAx. To avoid this, one can make a further simplification and linearize the second term around x_k:

$$\frac{1}{2}\|Ax - y_k + u_k\|_2^2 \approx \frac{1}{2}\|Ax_k - y_k + u_k\|_2^2 + (Ax_k - y_k + u_k)^T A(x - x_k).$$

If we omit the constant terms (in x), the simplified x-update is

$$\begin{aligned}
x_{k+1} &= \operatorname*{argmin}_x \left(f(x) + \tau(Ax_k - y_k + u_k)^T Ax + \frac{1}{2\sigma}\|x - x_k\|_2^2\right) \\
&= \operatorname*{argmin}_x \left(f(x) + \frac{1}{2\sigma}\|x - x_k + \tau\sigma A^T(Ax_k - y_k + u_k)\|_2^2\right) \\
&= \mathrm{prox}_{\sigma f}(x_k - \tau\sigma A^T(Ax_k - y_k + u_k)).
\end{aligned}$$

The resulting method is known as *linearized ADMM:*

$$\begin{aligned}
x_{k+1} &= \mathrm{prox}_{\sigma f}(x_k - \tau\sigma A^T(Ax_k - y_k + u_k)) \\
y_{k+1} &= \mathrm{prox}_{(1/\tau)g}(Ax_{k+1} + u_k) \\
u_{k+1} &= u_k + Ax_{k+1} - y_{k+1}.
\end{aligned}$$

Show that linearized ADMM is equivalent to PDHG applied to the dual of (9),

$$\text{maximize} \quad -g^*(z) - f^*(-A^T z).$$

PDHG for this problem is

$$\begin{aligned}
z_{k+1} &= \mathrm{prox}_{\tau g^*}(z_k + \tau A\tilde{x}_k) \\
\tilde{x}_{k+1} &= \mathrm{prox}_{\sigma f}(\tilde{x}_k - \sigma A^T(2z_{k+1} - z_k)).
\end{aligned}$$

15 *Proximal gradient method as Bregman proximal point algorithm* [O'Connor]. The following iteration is an extension of the proximal point algorithm (page 8.2, with $t_k = 1$) to a Bregman distance d:

$$x_{k+1} = \operatorname*{argmin}_x \left(f(x) + d(x, x_k)\right). \tag{7}$$

We apply this to a cost function $f(x) = g(x) + h(x)$, where g and h are convex, and g is differentiable with a Lipschitz continuous gradient. As we have seen in lecture 1 (page 1.17), this means that the function

$$\phi(x) = \frac{1}{2t}x^T x - g(x)$$

is convex for $0 < t \leq 1/L$, if L is the Lipschitz constant for the Euclidean norm.

Find the Bregman distance d generated by this kernel ϕ. Show that the proximal point iteration (10) with this distance reduces to the proximal gradient iteration

$$x_{k+1} = \mathrm{prox}_{th}(x_k - t\nabla g(x_k)).$$

16 *Exponential method of multipliers.* We consider a convex problem with m linear inequality constraints, and the dual problem:

$$\text{Primal:} \quad \begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax \preceq b \end{array} \qquad \text{Dual:} \quad \begin{array}{ll} \text{maximize} & -b^T z - f^*(-A^T z) \\ \text{subject to} & z \succeq 0. \end{array}$$

The dual variable z is an m-vector. In lecture 8 we interpreted the augmented Lagrangian method as the proximal point method applied to the dual problem. Here we work out what happens if we replace the squared Euclidean distance in the proximal point method with the relative entropy

$$d(u,v) = \sum_{i=1}^m (u_i \log(u_i/v_i) - u_i + v_i).$$

The Bregman proximal point iteration for the dual problem is

$$z_{k+1} = \operatorname*{argmin}_u \left(b^T u + f^*(-A^T u) + \frac{1}{t_k} d(u, z_k) \right),$$

where t_k is a positive step size and the starting point z_0 is a positive vector. Show that this is equivalent to the following iteration:

$$\begin{aligned} \hat{x} &= \operatorname*{argmin}_x \, (f(x) + \frac{1}{t_k} \sum_{i=1}^m z_{k,i} e^{t_k(a_i^T x - b_i)}) \\ z_{k+1,i} &= z_{k,i} e^{t_k(a_i^T \hat{x} - b_i)}, \quad i = 1, \ldots, m. \end{aligned}$$

Here a_i^T is the ith row of A, and $z_{k,i}$ is the ith component of the m-vector z_k.

17 [Polyak] In this problem we compare the convergence results for the conjugate gradient method (lecture 13) with the gradient method (lecture 1). We consider the minimization of a quadratic function

$$f(x) = \frac{1}{2} x^T A x - b^T x$$

with A positive definite and $\lambda_{\max}(A) = L$. From the last expression on page 13.15 we have the following bound on the error after k iterations:

$$\begin{aligned} 2(f(x_k) - f^\star) &\leq \left(\sum_{i=1}^n \frac{d_i^2}{\lambda_i^2} \right) \inf_{\deg(q) \leq k,\, q(0)=1} \left(\max_{i=1,\ldots,n} \lambda_i q(\lambda_i)^2 \right) \\ &= \|x^\star\|_2^2 \inf_{\deg(q) \leq k,\, q(0)=1} \left(\max_{i=1,\ldots,n} \lambda_i q(\lambda_i)^2 \right). \end{aligned} \tag{8}$$

The second line follows from $\|\Lambda^{-1} d\|_2 = \|Q \Lambda^{-1} d\|_2 = \|Q \Lambda^{-1} Q^T b\|_2 = \|A^{-1} b\|_2$.

The infimum in (11) is over all polynomials q that satisfy $q(0) = 1$ and have degree k or less. We will use Chebyshev polynomials to construct a polynomial q that satisfies these conditions, and therefore gives an upper bound on the right-hand side of (11).

The Chebyshev polynomial of degree m, denoted by T_m, is defined by the recursion

$$T_0(t) = 1, \qquad T_1(t) = t, \qquad T_{m+1}(t) = 2t T_m(t) - T_{m-1}(t) \quad \text{for } m \geq 1.$$

The following properties will be needed.

- The Chebyshev polynomials of odd degree only contain odd powers of t. The coefficient of t in $T_{2k+1}(t)$ is $(-1)^k(2k+1)$. For example,

$$T_1(t) = t, \qquad T_3(t) = 4t^3 - 3t, \qquad T_5(t) = 16t^5 - 20t^3 + 5t, \qquad \ldots.$$

- $|T_m(t)| \leq 1$ for $t \in [-1, 1]$.

Verify that the polynomial

$$q(t) = \frac{(-1)^k}{2k+1} \frac{T_{2k+1}(\sqrt{t/L})}{\sqrt{t/L}}$$

is a polynomial of degree k and satisfies $q(0) = 1$. Use this polynomial in (11) to show that after k iterations of the conjugate gradient method (started at $x_0 = 0$),

$$f(x_k) - f^\star \leq \frac{L}{2(2k+1)^2} \|x_0 - x^\star\|_2^2. \tag{9}$$

The corresponding result for the gradient method (page 1.26) with fixed step size $t = 1/L$ is

$$f(x_k) - f^\star \leq \frac{L}{2k} \|x_0 - x^\star\|_2^2.$$

While the bound (12) only holds for quadratic functions, the faster $1/k^2$ convergence has motivated research on accelerated gradient methods.

18 *Perturbation lemma.* With the notation of page 14.3, show that if A is invertible and $\|A^{-1}B\| < 1$, then

$$\|(A+B)^{-1} - A^{-1}\| \leq \frac{\|A^{-1}B\|}{1 - \|A^{-1}B\|} \|A^{-1}\|.$$

19 [Deuflhard] Consider a nonlinear equation $f(x) = 0$ where $f : \mathbf{R}^n \to \mathbf{R}^n$ is differentiable. Suppose $f(\hat{x}) \neq 0$ and the Jacobian matrix $f'(\hat{x})$ of f at $\hat{x}$ is nonsingular. Show that the Newton direction $v = -f'(\hat{x})^{-1} f(\hat{x})$ is a descent direction of the function $g_A(x) = \|Af(x)\|_2^2$, for *any* nonsingular matrix A. In other words, show that

$$\nabla g_A(\hat{x})^T v < 0 \quad \text{for all nonsingular } A. \tag{10}$$

Are there other directions v (other than the Newton direction) with this property?

20 [Myklebust and Tunçel] *Quasi-Newton update with two secant equations.* Let y_1, y_2, s_1, s_2 be n-vectors that satisfy

$$\begin{bmatrix} s_1^T y_1 & s_1^T y_2 \\ s_2^T y_1 & s_2^T y_2 \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix} \succ 0.$$

Suppose H is a given symmetric positive definite $n \times n$ matrix. We construct an update H_+ of H as follows.

(a) Define $\hat{y} = y_2 - (\beta/\alpha) y_1$ and $\hat{s} = s_2 - (\beta/\alpha) s_1$.

(b) Make two consecutive BFGS updates:

$$\begin{aligned}\hat{H} &= H + \frac{1}{y_1^T s_1} y_1 y_1^T - \frac{1}{s_1^T H s_1} H s_1 s_1^T H \\ H_+ &= \hat{H} + \frac{1}{\hat{y}^T \hat{s}} \hat{y}\hat{y}^T - \frac{1}{\hat{s}^T \hat{H} \hat{s}} \hat{H}\hat{s}\hat{s}^T \hat{H}.\end{aligned}$$

Show that H_+ is positive definite and satisfies the two secant equations

$$H_+ s_1 = y_1, \qquad H_+ s_2 = y_2.$$