QP

A PRE–PRINT

Dr. Luke A. Wendt
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Luke.A.Wendt@gmail.com
Luke-A-Wendt.info

December 12, 2019

ABSTRACT

This repo explores general formulation of **LP** (linear programs) and **QP** (quadratic programs) with applications in modeling, prediction, and control. These formulations have well-known solution methods that scale and are fully developed technologies integrated into most programming languages (making them readily available for embedded applications). State-of-the-art implamentation uses hardware accelerated distributed optimization specialized for sparse representations and parallelization.

Keywords linprog · **LP** · quadprog · **QP**

Contents

1 Notation

1.1 Sets

$\mathbf{Z}$	: integers (zahlen)
$\mathbf{R}$	: reals
$\mathbf{R}_{+}$	: positive
$\mathbf{R}_{0+}$	: non–negative
$\mathbf{R}^{n}$	: column vector
$\mathbf{R}^{m\times n}$	: matrix
$\mathbf{R}[a,b]$	: $\{x \in \mathbf{R} \mid a \leq x \leq b\}$
$\mathbf{R}(a,b)$	: $\{x \in \mathbf{R} \mid a < x < b\}$
$\mathbf{R}[a,b]^{n}$	: $\{x_i \in \mathbf{R}[a,b] \mid i \in \mathbf{Z}[0,n-1]\}$
$\{a,b\}^{n}$	: $\{x_i \in \{a,b\} \mid i \in \mathbf{Z}[0,n-1]\}$
$\mathbf{S}^{n}$	: symmetric matrix
$\mathbf{S}^{n}_{+}$	: positive definite matrix
$\mathbf{S}^{n}_{0+}$	: non–negative definite matrix
$\mathbf{P}$	: polyhedron
$\mathbf{D}$	: domain

1.2 Matrices

$\mathbf{1}$	: ones
$\mathbf{0}$	: zeros
$\boldsymbol{I}$	: identity matrix
$\mathbf{e}$	: element vector
$(\cdot)^{\intercal}$	: transpose
$(\cdot)^{+}$	: pseudo inverse
$(\cdot)_{\bullet}$	: index enumeration
diag	: block diagonal
row	: row–wise concatenation
col	: column–wise concatenation

1.3 Probability

$\widetilde{(\cdot)}$	: random variable
$\langle\cdot\rangle$	: expectation

1.4 Optimization

$(\cdot)^{\circ}$	: target solution
$(\cdot)^{*}$	: optimal solution

1.5 Functions

$$\text{Boolean}(x \in \mathbf{D}) = \begin{cases} 1 & \text{if } x \in \mathbf{D} \\ 0 & \text{else} \end{cases}$$

$$\begin{aligned} \text{Indicator}(x \in \mathbf{D}) &= -\log(\text{Boolean}(x \in \mathbf{D})) \\ &= \begin{cases} 0 & \text{if } x \in \mathbf{D} \\ \infty & \text{else} \end{cases} \end{aligned}$$

$$(x)_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{else} \end{cases} \qquad \text{hinge}$$

$$\text{Deadzone}(x, a, b) = (x - b)_+ - (-x - a)_+, \quad \text{for} \quad a < 0 < b$$

$$\text{Saturation}(x, a, b) = \begin{cases} b & \text{if } x > b \\ a & \text{if } x < a \\ x & \text{else} \end{cases}, \quad \text{for} \quad a < b$$

$$\text{Window}(x, a, b) = \text{Boolean}(x \in \mathbf{R}[a, b]), \quad \text{for} \quad a < b$$

Note: For the Deadzone(), Saturation(), and Window() functions,

$$\begin{aligned} f(x, c) &= f(x, -c, c), \\ f(x) &= f(x, 1). \end{aligned}$$

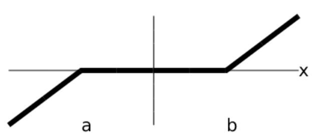


Figure 1: Deadzone

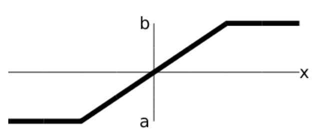


Figure 2: Saturation

Figure 3: Window

2 Introduction

2.1 Polyherons

Polyheron sets are defined by

$$\mathbf{P} := \left\{ x \in \mathbf{R}^n \;\middle|\; \begin{array}{l} A_{\text{ub}} x \le b_{\text{ub}}, \\ A_{\text{eq}} x = b_{\text{eq}}, \\ x_{\text{lb}} \le x \le x_{\text{ub}} \end{array} \right\}.$$

2.2 LP : Linear Programming

Linear optimization is formulated as [1, p. 146]

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = c^\mathsf{T} x \\ \text{s.t.} \quad & x \in \mathbf{P}, \end{aligned}$$

Note: For $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^n$, $x \le y$ implies element-wise inequality.
Note: x_{lb} and x_{ub} can be absorbed into A_{ub} and b_{ub}.
The solver is called with

$$x^* = \textbf{LP}\ (c, \mathbf{P}).\text{solve}().$$

2.3 QP : Quadratic Programming

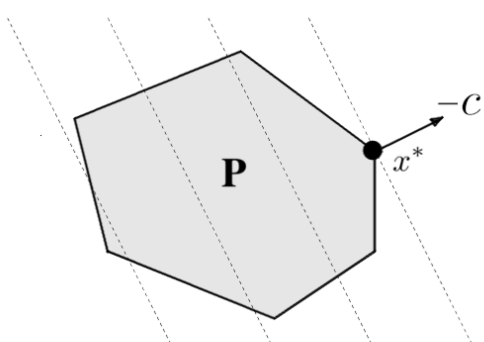
Quadratic optimization is formulated as [1, p. 152]

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \frac{1}{2} x^\mathsf{T} Q x + c^\mathsf{T} x \\ \text{s.t.} \quad & x \in \mathbf{P}, \end{aligned}$$

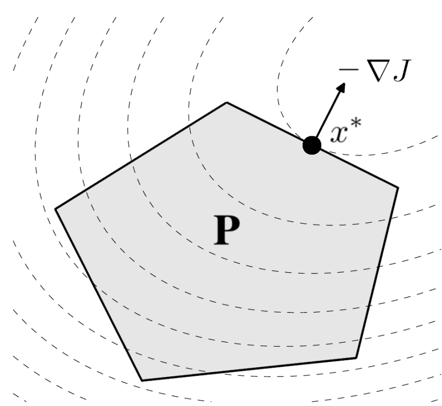
The solver is called with

$$x^* = \textbf{QP}\ (Q, c, \mathbf{P}).\text{solve}().$$

Note: **LP** is a subproblem of **QP** .



(a) **LP**



(b) **QP**

Figure 4: Comparison of **LP** and **QP**

3 Linear p–Norms

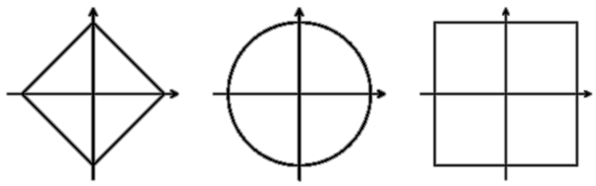


Figure 5: $\|\cdot\|_p =$ constant.

For $p \in \{1, 2, \infty\}$, consider

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_p,$$

$$\text{s.t.} \quad x \in \mathbf{P}.$$

The ℓ_p contours are depicted in Figure 5. The weighted linear p–norm is given by

$$\|Ax - b\|_{p,W} = \|W(Ax - b)\|_p.$$

For $w_i > 0$, a typical choice of weighting is

$$W = \text{diag}\{w_i\}_{i=0}^{m-1}.$$

3.1 Linear 2–Norm

For $x \in \mathbf{R}^n$,

$$\|x\|_2^2 := \sum_{i=0}^{n-1} x_i^2.$$

For $A \in \mathbf{R}^{m \times n}$,

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_2^2$$

$$\text{s.t.} \quad x \in \mathbf{P},$$

is the **QP**

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J &= (Ax - b)^\intercal (Ax - b) \\ &= x^\intercal A^\intercal A x - 2b^\intercal A x + b^\intercal b \\ &= \frac{1}{2} x^\intercal Q x + c^\intercal x + r \end{aligned}$$

$$\text{s.t.} \quad x \in \mathbf{P},$$

where

$$\begin{aligned} Q &= \quad 2A^\intercal A, \\ c &= -2A^\intercal b, \\ r &= \quad b^\intercal b. \end{aligned}$$

3.1.1 Weight

For non–negative definite $M = W^\intercal W$,

$$\begin{aligned}\|Ax-b\|_{2,W}^2 &= \|W(Ax-b)\|_2^2\\ &= (Ax-b)^\intercal W^\intercal W(Ax-b)\\ &= (Ax-b)^\intercal M(Ax-b)\\ &= x^\intercal A^\intercal MAx - 2b^\intercal MAx + b^\intercal Mb,\end{aligned}$$

which gives

$$\begin{aligned}Q &= 2A^\intercal MA,\\ c &= -2A^\intercal Mb.\end{aligned}$$

3.1.2 Expectation

Consider the addition of random variables. The expected linear 2–norm is given by

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & J = \langle \|(A+\widetilde{A})(x+\widetilde{x}) - (b+\widetilde{b})\|_{2,W}^2\rangle\\ \text{s.t.} \quad & x \in \mathbf{P},\end{aligned}$$

which, for non–negative definite $M = W^\intercal W$, can be solved as the **QP**

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad J &= x^\intercal\langle (A+\widetilde{A})^\intercal M(A+\widetilde{A})\rangle x\\ &\quad + 2\langle (A+\widetilde{A})^\intercal M((A+\widetilde{A})\widetilde{x} - (b+\widetilde{b}))\rangle^\intercal x\\ &\quad + \langle ((A+\widetilde{A})\widetilde{x} - (b+\widetilde{b}))^\intercal M((A+\widetilde{A})\widetilde{x} - (b+\widetilde{b}))\rangle\\ &= \frac{1}{2}x^\intercal Qx + c^\intercal x + r\\ \text{s.t.} \quad & x \in \mathbf{P},\end{aligned}$$

where

$$Q = 2A^\intercal MA + 2A^\intercal M\langle\widetilde{A}\rangle + 2\langle\widetilde{A}^\intercal\rangle MA + 2\langle\widetilde{A}^\intercal M\widetilde{A}\rangle,$$

$$\begin{aligned}c &= 2\left(A^\intercal MA\langle\widetilde{x}\rangle + A^\intercal M\langle\widetilde{A}\widetilde{x}\rangle + \langle\widetilde{A}^\intercal MA\widetilde{x}\rangle + \langle\widetilde{A}^\intercal M\widetilde{A}\widetilde{x}\rangle\right)\\ &\quad - 2\left(A^\intercal Mb + \langle\widetilde{A}^\intercal\rangle Mb + A^\intercal M\langle\widetilde{b}\rangle + \langle\widetilde{A}^\intercal M\widetilde{b}\rangle\right),\end{aligned}$$

and

$$\begin{aligned}\operatorname{vec}\left(\langle\widetilde{A}^\intercal M\widetilde{A}\rangle\right)^\intercal &= \vec{M}^\intercal\langle\widetilde{A}\otimes\widetilde{A}\rangle,\\ \operatorname{vec}\left(\langle\widetilde{A}^\intercal MA\widetilde{x}\rangle\right)^\intercal &= \operatorname{vec}(MA)^\intercal\langle\widetilde{A}\otimes\widetilde{x}\rangle,\\ \operatorname{vec}\left(\langle\widetilde{A}^\intercal M\widetilde{A}\widetilde{x}\rangle\right)^\intercal &= \vec{M}^\intercal\langle\widetilde{A}\otimes\widetilde{A}\widetilde{x}\rangle,\\ \operatorname{vec}\left(\langle\widetilde{A}^\intercal M\widetilde{b}\rangle\right)^\intercal &= \vec{M}^\intercal\langle\widetilde{A}\otimes\widetilde{b}\rangle.\end{aligned}$$

If the random variables are independent,

$$\begin{aligned}\langle\widetilde{A}\otimes\widetilde{x}\rangle &= \langle\,\widetilde{A}\,\rangle\otimes\langle\,\widetilde{x}\,\rangle,\\ \langle\widetilde{A}\otimes\widetilde{A}\widetilde{x}\rangle &= \langle\widetilde{A}\otimes\widetilde{A}\rangle\,(\,\boldsymbol{I}_n\otimes\langle\widetilde{x}\rangle\,)\,,\\ \langle\widetilde{A}\otimes\widetilde{b}\rangle &= \langle\,\widetilde{A}\,\rangle\otimes\langle\,\widetilde{b}\,\rangle.\end{aligned}$$

3.1.3 QP to Linear 2-Norm

For the constraint $x \in \mathbf{P}$, consider

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J &= \frac{1}{2} x^\intercal Q x + c^\intercal x \\ &= \|Ax - b\|_2^2 + r, \end{aligned}$$

with

$$\begin{aligned} Q &= \quad 2A^\intercal A, \\ c &= -2A^\intercal b. \end{aligned}$$

If $Q \in \mathbf{S}_+^n$, using SVD, the square root of Q and its inverse can be computed with

$$\begin{aligned} Q^{1/2} &= US^{1/2} \; U^\intercal, \\ Q^{-1/2} &= US^{-1/2} U^\intercal. \end{aligned}$$

The linear 2-norm can be expressed as

$$\begin{aligned} A &= \quad \frac{1}{\sqrt{2}} Q^{1/2}, \\ b &= -\frac{1}{\sqrt{2}} Q^{-1/2} c. \end{aligned}$$

Note: This choice of A gives $A = A^\intercal$.
Note: The constant offset does not affect the solution.

3.2 Linear 1–Norm

For $x \in \mathbf{R}^n$,

$$\|x\|_1 := \sum_{i=0}^{n-1} |x_i|.$$

For $A \in \mathbf{R}^{m \times n}$,

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_1$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to [1, p. 294]

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \mathbf{1}^\intercal y$$

$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ -y \le\ & Ax - b \le y, \end{aligned}$$

which can be expressed as the **LP**

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ \mathbf{1}^m \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ \begin{bmatrix} A & -\boldsymbol{I}_m \\ -A & -\boldsymbol{I}_m \end{bmatrix} & \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} b \\ -b \end{bmatrix}. \end{aligned}$$

Note: For $r_i = a_i^\intercal x - b_i$,

$$\|Ax - b\|_1 = \textstyle\sum_{i=0}^{m-1} \text{sign}(r_i) r_i.$$

3.3 Linear Inf–Norm

For $x \in \mathbf{R}^n$,

$$\|x\|_\infty := \max_{i \in \mathbf{Z}[0,n-1]} |x_i|.$$

For $A \in \mathbf{R}^{m \times n}$

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_\infty$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to [1, p. 294]

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad J = y$$

$$\begin{gathered} \text{s.t.} \quad x \in \mathbf{P}, \\ -y\mathbf{1} \leq Ax - b \leq y\mathbf{1}, \end{gathered}$$

which can be expressed as the **LP**

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\begin{gathered} \text{s.t.} \quad x \in \mathbf{P} \\ \begin{bmatrix} -A & -\mathbf{1}^m \\ A & -\mathbf{1}^m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} -b \\ b \end{bmatrix}. \end{gathered}$$

3.4 Combined Linear p–Norms

Combinations of $p \in \{1, 2, \infty\}$ result in **LP** or **QP** .

3.4.1 Linear 1–1–Norm

For $A \in \mathbf{R}^{m \times n}$ and $C \in \mathbf{R}^{r \times n}$,

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_1 + \|Cx - d\|_1$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to

$$\min_{\{x,y,z\} \in \{\mathbf{R}^n, \mathbf{R}^m, \mathbf{R}^r\}} \quad J = \mathbf{1}^\mathsf{T} y + \mathbf{1}^\mathsf{T} z$$

$$\begin{gathered} \text{s.t.} \quad x \in \mathbf{P}, \\ -y \leq Ax - b \leq y, \\ -z \leq Cx - d \leq z, \end{gathered}$$

which can be expressed as the **LP**

$$\min_{\{x,y,z\} \in \{\mathbf{R}^n, \mathbf{R}^m, \mathbf{R}^r\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ \mathbf{1}^m \\ \mathbf{1}^r \end{bmatrix}^\mathsf{T} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{gathered} \text{s.t.} \quad x \in \mathbf{P}, \\ \begin{bmatrix} -A & -\boldsymbol{I}_m & \mathbf{0}^{m \times r} \\ A & -\boldsymbol{I}_m & \mathbf{0}^{m \times r} \\ -C & \mathbf{0}^{r \times m} & -\boldsymbol{I}_r \\ C & \mathbf{0}^{r \times m} & -\boldsymbol{I}_r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} -b \\ b \\ -d \\ d \end{bmatrix}. \end{gathered}$$

3.4.2 Linear 2–2–Norm

For $A \in \mathbf{R}^{m\times n}$ and $C \in \mathbf{R}^{r\times n}$,

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & J = \|Ax-b\|_2^2 + \|Cx-d\|_2^2 \\ \text{s.t.} \quad & x \in \mathbf{P}\end{aligned}$$

is equivalent to

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & J = x^\intercal(A^\intercal A + C^\intercal C)x - 2(b^\intercal A + d^\intercal C)x + b^\intercal b + d^\intercal d \\ \text{s.t.} \quad & x \in \mathbf{P},\end{aligned}$$

which is a **QP** with

$$\begin{aligned} Q &= \quad 2(A^\intercal A + C^\intercal C), \\ c &= -2(A^\intercal b + C^\intercal d).\end{aligned}$$

3.4.3 Linear inf–inf–Norm

For $A \in \mathbf{R}^{m\times n}$ and $C \in \mathbf{R}^{r\times n}$,

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & J = \|Ax-b\|_\infty + \|Cx-d\|_\infty \\ \text{s.t.} \quad & x \in \mathbf{P}\end{aligned}$$

is equivalent to

$$\begin{aligned}\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}\}} \quad & J = y + z \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & -y\mathbf{1}^m \le Ax - b \le y\mathbf{1}^m, \\ & -z\mathbf{1}^r \le Cx - d \le z\mathbf{1}^r,\end{aligned}$$

which can be expressed as the **LP**

$$\begin{aligned}\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}\}} \quad & J = \begin{bmatrix} \mathbf{0}^n \\ 1 \\ 1 \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \begin{bmatrix} -A & -\mathbf{1}^m & \mathbf{0}^m \\ A & -\mathbf{1}^m & \mathbf{0}^m \\ -C & \mathbf{0}^r & -\mathbf{1}^r \\ C & \mathbf{0}^r & -\mathbf{1}^r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \le \begin{bmatrix} -b \\ b \\ -d \\ d \end{bmatrix}.\end{aligned}$$

3.4.4 Linear 1–2–Norm

For $C \in \mathbf{R}^{m\times n}$,

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & J = \|Ax-b\|_2^2 + \|Cx-d\|_1 \\ \text{s.t.} \quad & x \in \mathbf{P}\end{aligned}$$

is equivalent to

$$\begin{aligned}\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad & J = x^\intercal A^\intercal Ax - b^\intercal Ax + b^\intercal b + \mathbf{1}^\intercal y \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & -y \le Cx - d \le y,\end{aligned}$$

which can be expressed as the **QP**

$$\begin{aligned}\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad & J = \frac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^\intercal \begin{bmatrix} 2A^\intercal A & \mathbf{0}^{n\times m} \\ \mathbf{0}^{m\times n} & \mathbf{0}^{m\times m} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -2A^\intercal b \\ \mathbf{1}^m \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \end{bmatrix} + b^\intercal b \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \begin{bmatrix} -C & -\boldsymbol{I}_m \\ C & -\boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} -d \\ d \end{bmatrix}.\end{aligned}$$

3.4.5 Linear 1–Inf–Norm

For $A \in \mathbf{R}^{m\times n}$ and $C \in \mathbf{R}^{r\times n}$,

$$\min_{x\in\mathbf{R}^n} \quad J = \|Ax-b\|_1 + \|Cx-d\|_\infty$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to

$$\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R}^m,\mathbf{R}\}} \quad J = \mathbf{1}^\mathsf{T} y + z$$

$$\begin{gathered}\text{s.t.} \quad x \in \mathbf{P},\\ -y \le Ax-b \le y,\\ -z\mathbf{1} \le Cx-d \le z\mathbf{1},\end{gathered}$$

which can be expressed as the **LP**

$$\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R}^m,\mathbf{R}\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ \mathbf{1}^m \\ 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{gathered}\text{s.t.} \quad x \in \mathbf{P},\\ \begin{bmatrix} -A & -\boldsymbol{I}_m & \mathbf{0}^m \\ A & -\boldsymbol{I}_m & \mathbf{0}^m \\ -C & \mathbf{0}^{r\times m} & -\mathbf{1}^r \\ C & \mathbf{0}^{r\times m} & -\mathbf{1}^r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \le \begin{bmatrix} -b \\ b \\ -d \\ d \end{bmatrix}.\end{gathered}$$

3.4.6 Linear 2–Inf–Norm

For $C \in \mathbf{R}^{m\times n}$,

$$\min_{x\in\mathbf{R}^n} \quad J = \|Ax-b\|_2^2 + \|Cx-d\|_\infty$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to

$$\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad J = x^\mathsf{T} A^\mathsf{T} Ax - b^\mathsf{T} Ax + b^\mathsf{T} b + y$$

$$\begin{gathered}\text{s.t.} \quad x \in \mathbf{P},\\ -y\mathbf{1} \le Cx-d \le y\mathbf{1},\end{gathered}$$

which can be expressed as the **QP**

$$\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad J = \frac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^\mathsf{T} \begin{bmatrix} 2A^\mathsf{T} A & \mathbf{0}^n \\ \mathbf{0}^{1\times n} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -2A^\mathsf{T} b \\ 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} x \\ y \end{bmatrix} + b^\mathsf{T} b$$

$$\begin{gathered}\text{s.t.} \quad x \in \mathbf{P},\\ \begin{bmatrix} -C & -\mathbf{1}^m \\ C & -\mathbf{1}^m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} -d \\ d \end{bmatrix}.\end{gathered}$$

3.4.7 Linear 1–2–Inf–Norm

For $B \in \mathbf{R}^{m\times n}$ and $D \in \mathbf{R}^{r\times n}$,

$$\min_{x\in\mathbf{R}^n} \quad J = \|Ax-b\|_2^2 + \|Bx-c\|_1 + \|Dx-e\|_\infty$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to

$$\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R}^m,\mathbf{R}\}} \quad J = x^\intercal A^\intercal Ax - b^\intercal Ax + b^\intercal b + \mathbf{1}^\intercal y + z$$

$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ & -y \le Bx - c \le y, \\ & -z\mathbf{1} \le Dx - e \le z\mathbf{1}, \end{aligned}$$

which can be expressed as the **QP**

$$\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R}^m,\mathbf{R}\}} \quad J = \frac{1}{2}\begin{bmatrix} x \\ y \\ z \end{bmatrix}^\intercal \begin{bmatrix} 2A^\intercal A & \mathbf{0}^{n\times m} & \mathbf{0}^n \\ \mathbf{0}^{m\times n} & \mathbf{0}^{m\times m} & \mathbf{0}^m \\ \mathbf{0}^{1\times n} & \mathbf{0}^{1\times m} & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} -2A^\intercal b \\ \mathbf{1}^m \\ 1 \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \\ z \end{bmatrix} + b^\intercal b$$

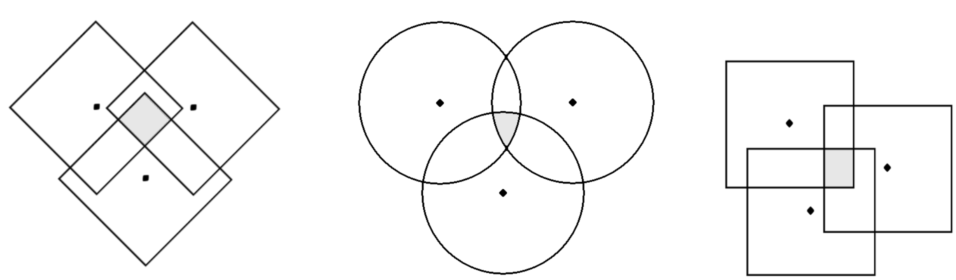
$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ & \begin{bmatrix} -B & -\boldsymbol{I}_m & \mathbf{0}^m \\ B & -\boldsymbol{I}_m & \mathbf{0}^m \\ -D & \mathbf{0}^{r\times m} & -\mathbf{1}^r \\ D & \mathbf{0}^{r\times m} & -\mathbf{1}^r \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \le \begin{bmatrix} -c \\ c \\ -e \\ e \end{bmatrix}. \end{aligned}$$

3.5 Linear p–Norm Constraint

For $A_i \in \mathbf{R}^{m_i\times n}$ and $t_i > 0$, consider the problem

$$\min_{x\in\mathbf{R}^n} \quad J = f(x)$$

$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ & \|A_i x - b_i\|_p \le t_i. \end{aligned} \tag{1}$$



(a) 1–norm (b) 2–norm (c) inf–norm

Figure 6: Comparison of p–norm inequality constraints

3.5.1 Linear 1–Norm Constraint

For $p = 1$, Equation [1] is equivalent to

$$\min_{\{x,y_i\}\in\{\mathbf{R}^n,\mathbf{R}^{m_i}\}} \quad J = f(x)$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ \mathbf{1}^\intercal y_i &\leq t_i, \\ -y_i \leq A_i x - b_i &\leq y_i. \end{aligned}$$

Figure 6a depicts the intersection of 1–norm inequality constraints.

If $f(x)$ is linear, this problem is an **LP** .

If $f(x)$ is quadratic, this problem is a **QP** .

3.5.2 Linear 2–Norm Constraint

For $p = 2$, Equation [1] can be squared to get

$$\|A_i x - b_i\|_2^2 \leq t_i^2,$$

which can be expressed as

$$\frac{1}{2} x^\intercal Q_i x + c_i^\intercal x + r_i \leq 0,$$

where

$$\begin{aligned} Q_i &= 2A_i^\intercal A_i, \\ c_i &= -2A_i^\intercal b_i, \\ r_i &= b_i^\intercal b_i - t_i^2. \end{aligned}$$

Figure 6b depicts the intersection of 2–norm inequality constraints.

If $f(x)$ is quadratic, this problem is a **QCQP** .

3.5.3 Linear Inf–Norm Constraint

For $p = \infty$, Equation [1] is equivalent to

$$\min_{\{x,y_i\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad J = f(x)$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ y_i &\leq t_i, \\ -y_i\mathbf{1} \leq A_i x - b_i &\leq y_i\mathbf{1}. \end{aligned}$$

Figure 6c depicts the intersection of inf–norm inequality constraints.

If $f(x)$ linear, this problem is an **LP** .

If $f(x)$ quadratic, this problem is a **QP** .

3.6 Piecewise–Linear Minimization

Consider the problem

$$\begin{aligned} \min_{x\in\mathbf{R}^n} \quad \max_{i\in\mathbf{Z}[0,m-1]} \quad & J_i = a_i^\intercal x - b_i, \\ \text{s.t.} \quad & x \in \mathbf{P}. \end{aligned}$$

This problem is equivalent to the **LP** [1, p. 150]

$$\begin{aligned} \min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad & y \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & a_i^\intercal x - b_i \le y, \end{aligned}$$

which can be expressed as

$$\begin{aligned} \min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad & \begin{bmatrix} \mathbf{0}^n \\ 1 \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \end{bmatrix} \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \begin{bmatrix} A & -\mathbf{1}^m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le b, \end{aligned}$$

where

$$A = \text{row}\{a_i^\intercal\}_{i=0}^{m-1},$$

$$b = \text{row}\{b_i\}_{i=0}^{m-1}.$$

3.7 Robust p–Norm from a Finite Set

For $A_i \in \mathbf{R}^{q_i\times n}$, consider the problem

$$\begin{aligned} \min_{x\in\mathbf{R}^n} \quad \max_{i\in\mathbf{Z}[0,m-1]} \quad & J_i = \|A_i x - b_i\|_p, \\ \text{s.t.} \quad & x \in \mathbf{P}. \end{aligned}$$

This problem is equivalent to [1, p. 321]

$$\begin{aligned} \min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad & y \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \|A_i x - b_i\|_p \le y. \end{aligned} \tag{2}$$

If $p \in \{1, \infty\}$, this is an **LP** .

3.7.1 Robust 1–Norm from a Finite Set

For $p = 1$ and $i \in \mathbf{Z}[0, m-1]$, Equation [2] is equivalent to the **LP**

$$\begin{aligned} \min_{\{x,y,z_i\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}^{q_i}\}} \quad & J = y \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \mathbf{1}^\intercal z_i \le y, \\ & -z_i \le A_i x - b_i \le z_i. \end{aligned}$$

3.7.2 Robust Inf–Norm from a Finite Set

For $p = \infty$ and $i \in \mathbf{Z}[0, m-1]$, Equation [2] is equivalent to the **LP**

$$\begin{aligned} \min_{\{x,y,z_i\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}\}} \quad & J = y \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & z_i \le y, \\ & -z_i\mathbf{1} \le A_i x - b_i \le z_i\mathbf{1}. \end{aligned}$$

4 Penalty Functions

4.1 Deadzone Penalty

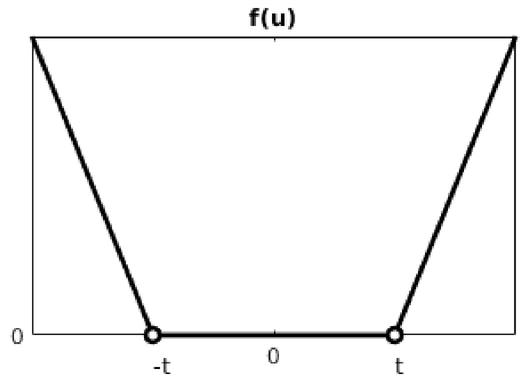


Figure 7: Deadzone Penalty

Consider the problem

$$\min_{x \in \mathbf{R}^n} \quad J = \sum_{i=0}^{m-1} f(a_i^\intercal x - b_i)$$
$$\text{s.t.} \quad x \in \mathbf{P},$$

where $f(u)$ is the deadzone

$$\begin{aligned} f(u) &= \begin{cases} 0 & \text{if } |u| \leq t \\ |u| - t & \text{else} \end{cases} \\ &= \max(\, -u - t,\ 0,\ u - t\,), \end{aligned} \tag{3}$$

$$f'(u) = \begin{cases} 0 & \text{if } |u| \leq t \\ \text{sign}(u) & \text{else} \end{cases} .$$

Note: The deadzone penalty is not a norm because it fails the condition $f(x) = 0$ if and only if $x = 0$.
This problem is equivalent to [1, p. 344]

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \mathbf{1}^\intercal y$$
$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ -y - t\mathbf{1} \leq\ & Ax - b \leq y + t\mathbf{1}, \\ & y \geq \mathbf{0}, \end{aligned}$$

which can be expressed as the **LP**

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ \mathbf{1}^m \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \end{bmatrix}$$
$$\text{s.t.} \quad x \in \mathbf{P},$$
$$\begin{bmatrix} A & -\boldsymbol{I}_m \\ -A & \boldsymbol{I}_m \\ \mathbf{0}^{n \times m} & -\boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} t\mathbf{1}^m + b \\ t\mathbf{1}^m - b \\ \mathbf{0}^m \end{bmatrix}.$$

4.2 Huber Penalty

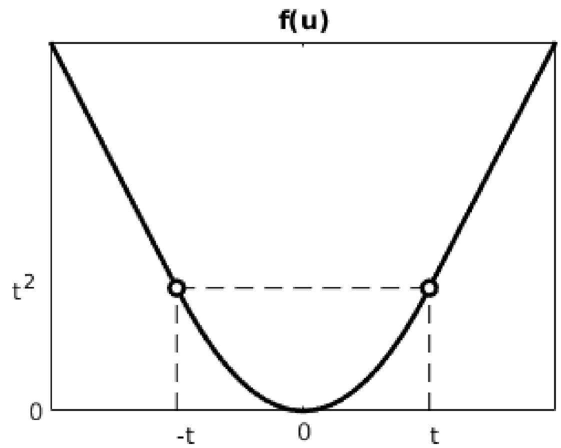


Figure 8: Huber Penalty

Consider the problem

$$\min_{x \in \mathbf{R}^n} \quad J = \sum_{i=0}^{m-1} f(a_i^\intercal x - b_i)$$

$$\text{s.t.} \quad x \in \mathbf{P},$$

where $f(u)$ is the Huber function

$$f(u) = \begin{cases} u^2 & \text{if } |u| \leq t \\ t(2|u| - t) & \text{else} \end{cases}, \tag{4}$$

$$f'(u) = \begin{cases} 2u & \text{if } |u| \leq t \\ 2t \operatorname{sign}(u) & \text{else} \end{cases}$$
$$= 2 \operatorname{Saturation}(u, t).$$

Note: The Huber penalty is a norm.
This problem is equivalent to [1, p. 190]

$$\min_{\{x,y,z\} \in \{\mathbf{R}^n, \mathbf{R}^m, \mathbf{R}^m\}} \quad J = y^\intercal y + 2t\mathbf{1}^\intercal z$$

$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ -y - z \leq & Ax - b \leq y + z, \\ \mathbf{0} \leq & y \leq t\mathbf{1}, \\ & z \geq \mathbf{0}, \end{aligned}$$

which can be expressed as the **QP**

$$\min_{\{x,y,z\} \in \{\mathbf{R}^n, \mathbf{R}^m, \mathbf{R}^m\}} \quad J = \frac{1}{2} \begin{bmatrix} x \\ y \\ z \end{bmatrix}^\intercal \begin{bmatrix} \mathbf{0}^{n\times n} & \mathbf{0}^{n\times m} & \mathbf{0}^{n\times m} \\ \mathbf{0}^{m\times n} & 2\boldsymbol{I}_m & \mathbf{0}^{m\times m} \\ \mathbf{0}^{m\times n} & \mathbf{0}^{m\times m} & \mathbf{0}^{m\times m} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{0}^m \\ 2t\mathbf{1}^m \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\text{s.t.} \quad x \in \mathbf{P},$$

$$\begin{bmatrix} A & -\boldsymbol{I}_m & -\boldsymbol{I}_m \\ -A & -\boldsymbol{I}_m & -\boldsymbol{I}_m \\ \mathbf{0}^{m\times n} & \boldsymbol{I}_m & \mathbf{0}^{m\times m} \\ \mathbf{0}^{m\times n} & -\boldsymbol{I}_m & \mathbf{0}^{m\times m} \\ \mathbf{0}^{m\times n} & \mathbf{0}^{m\times m} & -\boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} b \\ -b \\ t\mathbf{1}^m \\ \mathbf{0}^m \\ \mathbf{0}^m \end{bmatrix}.$$

4.3 Rectified Linear Penalty

Consider the problem

$$\min_{x \in \mathbf{R}^n} \quad J = \sum_{i=0}^{m-1} f(a_i^\intercal x - b_i)$$

$$\text{s.t.} \quad x \in \mathbf{P},$$

where $f(u)$ is the rectified linear penalty function

$$\begin{aligned} f(u) &= (u)_+ \\ &= \max(0, u). \end{aligned}$$

This problem can be expressed as the **LP**

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \mathbf{1}^\intercal y$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ Ax - b &\leq y, \\ 0^n &\leq y. \end{aligned}$$

Note: Rectified linear penalty is convex and can be used as a constraint function.
Note: Rectified linear penalty is the preffered nonlinearity for deep learning [41].

4.4 Leaky Linear Penalty

$$\min_{x \in \mathbf{R}^n} \quad J = \sum_{i=0}^{m-1} f(a_i^\intercal x - b_i)$$

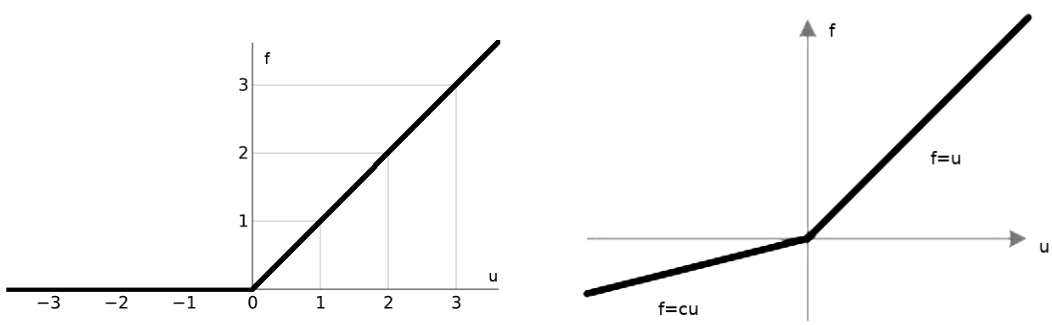
where $f(u)$ is the leaky–linear penalty function

$$\begin{aligned} f(u) &= (u)_+ - t(-u)_+ \\ &= \max(tu, u), \end{aligned}$$

with $t \in \mathbf{R}(0,1)$. This problem can be expressed as the **LP**

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \mathbf{1}^\intercal y$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ Ax - b &\leq y, \\ Ax - b &\leq y/t. \end{aligned}$$



(a) linear (b) leaky–linear

Figure 9: Rectified Linear Penalty

5 Duality

Consider the primal problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f(x) \\ \text{s.t.} \quad & A_{\text{ub}} x \le b_{\text{ub}}, \\ & A_{\text{eq}} x = b_{\text{eq}}. \end{aligned}$$

Let

$$A = \begin{bmatrix} A_{\text{ub}} \\ A_{\text{eq}} \end{bmatrix}, \quad b = \begin{bmatrix} b_{\text{ub}} \\ b_{\text{eq}} \end{bmatrix}.$$

For $A_{\text{ub}} \in \mathbf{R}^{r \times n}$ and $A \in \mathbf{R}^{m \times n}$, the dual problem formulation is

$$\begin{aligned} \max_{y \in \mathbf{R}^m} \quad \inf_{x \in \mathbf{R}^n} \quad & L(x, y) \\ \text{s.t.} \quad & \begin{bmatrix} \boldsymbol{I}_r & \mathbf{0}^{r \times (m-r)} \end{bmatrix} y \ge \mathbf{0}^r, \end{aligned}$$

where the Lagrangian is given by

$$L(x, y) = f(x) + y^\mathsf{T}(Ax - b).$$

If strong duality holds, the following steps compute x^*:

- Compute $\inf_{x \in \mathbf{R}^n} L(x, y)$ to get $L(x^*, y)$.
  Note: This step may not give an explicit representation of x^*.
  Implicit constraints for $L(x^*, y) > -\infty$ can be made explicit.
- Solve the dual problem to get y^*.
- The primal problem is solved with $\min_{x \in \mathbf{R}^n} L(x, y^*)$.

Note: The dual problem may be concave even when the primal problem is not convex.

5.1 Norm Duality

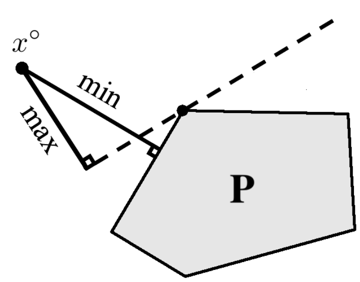


Figure 10: Norm Duality

Consider the problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|x - x^\circ\| \\ \text{s.t.} \quad & x \in \mathbf{P}. \end{aligned}$$

The dual problem is illustrated in Figure 10 [36, p. 9].

Conisder the problem

$$\min_{x \in \mathbf{R}^n} \quad J = \|x\|_p$$
$$\text{s.t.} \quad Ax = b.$$

The dual problem is given by

$$\max_{y \in \mathbf{R}^m} \inf_{x \in \mathbf{R}^n} \quad L(x, y) = \|x\|_p + y^\intercal (Ax - b),$$

which becomes [1, p. 221][36, p. 107][36, p. 123]

$$\max_{y \in \mathbf{R}^m} \quad L(x^*, y) = \begin{cases} -b^\intercal y & \text{if} \quad \|A^\intercal y\|_q \leq 1 \\ -\infty & \text{else} \end{cases} ,$$

where $q = p/(p-1)$ for $p \in \mathbf{R}[1, \infty)$.

Note: This does not hold for $p = \infty$.

The dual problem can be stated as

$$\min_{y \in \mathbf{R}^m} \quad -L = b^\intercal y$$
$$\text{s.t.} \quad \|A^\intercal y\|_q \leq 1.$$

5.1.1 1–Norm

For $p = 1$, $q = \infty$ and this becomes the **LP**

$$\min_{\{y,z\} \in \{\mathbf{R}^m, \mathbf{R}\}} \quad -L = b^\intercal y + z$$
$$\text{s.t.} \quad z \leq 1,$$
$$-z\mathbf{1}^n \leq A^\intercal y \leq z\mathbf{1}^n,$$

or

$$\min_{\{y,z\} \in \{\mathbf{R}^m, \mathbf{R}\}} \quad -L = \begin{bmatrix} b \\ 1 \end{bmatrix}^\intercal \begin{bmatrix} y \\ z \end{bmatrix}$$
$$\text{s.t.} \quad \begin{bmatrix} \mathbf{0}^{1\times m} & 1 \\ A^\intercal & -\mathbf{1}^n \\ -A^\intercal & -\mathbf{1}^n \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \leq \begin{bmatrix} 1 \\ \mathbf{0}^n \\ \mathbf{0}^n \end{bmatrix}.$$

5.1.2 2–Norm

For $p = 2$, $q = 2$ and this becomes the **QCQP**

$$\min_{y \in \mathbf{R}^m} \quad -L = b^\intercal y$$
$$\text{s.t.} \quad y^\intercal Q y \leq 1,$$

where $Q = AA^\intercal$.

5.2 LP Duality

Consider the primal **LP**

$$\min_{x \in \mathbf{R}^n} \quad J = c^\intercal x$$

$$\begin{aligned} \text{s.t.} \quad & A_{\text{ub}} x \leq b_{\text{ub}}, \\ & A_{\text{eq}} x = b_{\text{eq}}. \end{aligned}$$

Let

$$A = \begin{bmatrix} A_{\text{ub}} \\ A_{\text{eq}} \end{bmatrix}, \quad b = \begin{bmatrix} b_{\text{ub}} \\ b_{\text{eq}} \end{bmatrix}.$$

For $A_{\text{ub}} \in \mathbf{R}^{r \times n}$ and $A \in \mathbf{R}^{m \times n}$, the dual problem formulation is

$$\max_{y \in \mathbf{R}^m} \quad \inf_{x \in \mathbf{R}^n} \quad L(x, y)$$

$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{I}_r & \mathbf{0}^{r \times (m-r)} \end{bmatrix} y \geq \mathbf{0}^r,$$

where the Lagrangian is given by

$$\begin{aligned} L(x, y) &= c^\intercal x + y^\intercal (Ax - b) \\ &= (c^\intercal + y^\intercal A)x - y^\intercal b, \end{aligned}$$

which gives

$$L(x^*, y) = \begin{cases} -b^\intercal y & \text{if } c^\intercal + y^\intercal A = 0 \\ -\infty & \text{else} \end{cases}.$$

Making dual constraints explicit gives the **LP**

$$\min_{y \in \mathbf{R}^m} \quad -L(x^*, y) = b^\intercal y$$

$$\begin{aligned} \text{s.t.} \quad \begin{bmatrix} -\boldsymbol{I}_r & \mathbf{0}^{r \times (m-r)} \end{bmatrix} y &\leq \mathbf{0}^r, \\ A^\intercal y &= -c. \end{aligned}$$

5.2.1 Standard Form

If the primal problem is in standard form, i.e., $A_{\text{ub}} = -\boldsymbol{I}_n$ and $b_{\text{ub}} = \mathbf{0}^n$, the dual can be simplified to [1, p. 224]

$$\min_{y_{\text{eq}} \in \mathbf{R}^{m-r}} \quad -L\left(x^*, y = \begin{bmatrix} y_{\text{ub}} \\ y_{\text{eq}} \end{bmatrix}\right) = b_{\text{eq}}^\intercal y_{\text{eq}}$$

$$\text{s.t.} \quad -A_{\text{eq}}^\intercal y_{\text{eq}} \leq c,$$

where

$$y_{\text{ub}}^* = A_{\text{eq}}^\intercal y_{\text{eq}}^* + c.$$

5.3 QP Duality

Consider the primal **QP**

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & J = \frac{1}{2}x^\intercal Q x + c^\intercal x \\ \text{s.t.} \quad & A_{\text{ub}}x \le b_{\text{ub}}, \\ & A_{\text{eq}}x = b_{\text{eq}}.\end{aligned}$$

Let

$$A = \begin{bmatrix} A_{\text{ub}} \\ A_{\text{eq}} \end{bmatrix}, \quad b = \begin{bmatrix} b_{\text{ub}} \\ b_{\text{eq}} \end{bmatrix}.$$

For $A_{\text{ub}} \in \mathbf{R}^{r\times n}$ and $A \in \mathbf{R}^{m\times n}$, the dual problem formulation is

$$\begin{aligned}\max_{y\in\mathbf{R}^m} \quad \inf_{x\in\mathbf{R}^n} \quad & L(x,y) \\ \text{s.t.} \quad & \begin{bmatrix} \boldsymbol{I}_r & \mathbf{0}^{r\times(m-r)} \end{bmatrix} y \ge \mathbf{0}^r,\end{aligned}$$

where the Lagrangian is given by

$$\begin{aligned}L(x,y) &= \frac{1}{2}x^\intercal Q x + c^\intercal x + y^\intercal(Ax - b) \\ &= \frac{1}{2}x^\intercal Q x + (c^\intercal + y^\intercal A)x - y^\intercal b.\end{aligned}$$

5.3.1 Positive Definite Duality

For $Q \in \mathbf{S}^n_+$,

$$\frac{\partial L}{\partial x} = x^\intercal Q + c^\intercal + y^\intercal A.$$

Setting this to zero and solving gives

$$x^* = -Q^{-1}(A^\intercal y + c)$$

and

$$\begin{aligned}L(x^*,y) &= -\frac{1}{2}(c^\intercal + y^\intercal A)Q^{-1}(A^\intercal y + c) - y^\intercal b \\ &= -\frac{1}{2}y^\intercal A Q^{-1}A^\intercal y - c^\intercal Q^{-1}A^\intercal y - \frac{1}{2}c^\intercal Q^{-1}c - b^\intercal y.\end{aligned}$$

The dual problem is the **QP**

$$\begin{aligned}\min_{y\in\mathbf{R}^m} \quad & -L(x^*,y) = \frac{1}{2}y^\intercal Q_{\text{dual}}\, y + c_{\text{dual}}^\intercal\, y \\ \text{s.t.} \quad & \begin{bmatrix} -\boldsymbol{I}_r & \mathbf{0}^{r\times(m-r)} \end{bmatrix} y \le \mathbf{0}^r,\end{aligned}$$

where

$$\begin{aligned}Q_{\text{dual}} &= AQ^{-1}A^\intercal, \\ c_{\text{dual}} &= b + AQ^{-1}c.\end{aligned}$$

Note: If $n > m$, the dual problem provides a computationally effecient alternative with m free variables to solve.

5.3.2 Non–Negative Definite Duality

For $Q \in \mathbf{S}^n_{0+}$, SVD gives $Q = U_+ S_+ V_+^\intercal$ with $U_+ = V_+$. Choose

$$x = \begin{bmatrix} V_+ & V_0 \end{bmatrix} \begin{bmatrix} x_+ \\ x_0 \end{bmatrix}.$$

Note: The primal problem can be solved with an ADMM of **LP** and positive definite **QP** .
The Lagrangian becomes

$$\begin{aligned} L(x, y) = &\frac{1}{2} x_+^\intercal S_+ x_+ + \begin{bmatrix} c_+^\intercal & c_0^\intercal \end{bmatrix} \begin{bmatrix} x_+ \\ x_0 \end{bmatrix} \\ &+ y^\intercal \left(\begin{bmatrix} A_+ & A_0 \end{bmatrix} \begin{bmatrix} x_+ \\ x_0 \end{bmatrix} - b_{\text{eq}} \right) \end{aligned}$$

where

$$\begin{aligned} A_+ &:= A V_+, \quad A_0 := A V_0, \\ c_+^\intercal &:= c^\intercal V_+, \quad c_0^\intercal := c^\intercal V_0. \end{aligned}$$

Minimize x_+ with

$$\begin{aligned} 0 &= \frac{\partial L}{\partial x_+} \\ &= x_+^\intercal S_+ + c_+^\intercal + y^\intercal A_+, \end{aligned}$$

which gives

$$x_+^* = -S_+^{-1} (A_+^\intercal y + c_+).$$

Minimizing x_0,

$$\begin{aligned} L(x^*, y) &= \frac{1}{2} x_+^{*\intercal} S_+ x_+^* + \begin{bmatrix} c_+^\intercal & c_0^\intercal \end{bmatrix} \begin{bmatrix} x_+^* \\ x_0 \end{bmatrix} + y^\intercal \left(\begin{bmatrix} A_+ & A_0 \end{bmatrix} \begin{bmatrix} x_+^* \\ x_0 \end{bmatrix} - b \right) \\ &= \begin{cases} \frac{1}{2} x_+^{*\intercal} S_+ x_+^* + \left(c_+^\intercal + y^\intercal A_+ \right) x_+^* - b^\intercal y & \text{if} \quad c_0 + A_0^\intercal y = \mathbf{0} \\ -\infty & \text{else} \end{cases} \\ &= \begin{cases} -\frac{1}{2} x_+^{*\intercal} S_+ x_+^* - b^\intercal y & \text{if} \quad c_0 + A_0^\intercal y = \mathbf{0} \\ -\infty & \text{else} \end{cases} \end{aligned}$$

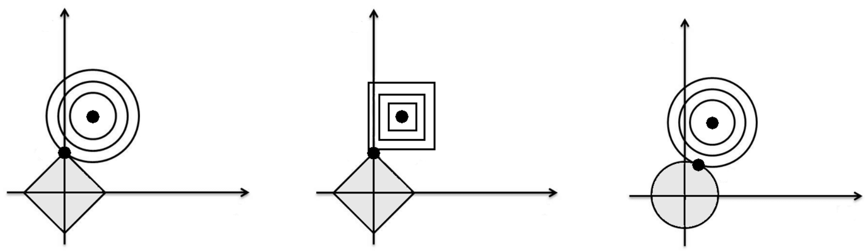
Making dual constraints explicit gives the **QP**

$$\begin{aligned} \min_{y \in \mathbf{R}^m} \quad & -L(x^*, y) = \frac{1}{2} y^\intercal Q_{\text{dual}}\, y + c_{\text{dual}}^\intercal\, y \\ \text{s.t.} \quad & \begin{bmatrix} -\boldsymbol{I}_r & \mathbf{0}^{r \times (m-r)} \end{bmatrix} y \leq \mathbf{0}^r, \\ & A_0^\intercal y = -c_0, \end{aligned}$$

where

$$\begin{aligned} Q_{\text{dual}} &= A_+ S_+^{-1} A_+^\intercal, \\ c_{\text{dual}} &= b + A_+ S_+^{-1} c_+. \end{aligned}$$

6 Regularization



(a) $p = 2,\ q = 1$ (b) $p = \infty,\ q = 1$ (c) $p = 2,\ q = 2$

Figure 11: ℓ_p residual with ℓ_q constraint

Consider

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_p$$
$$\text{s.t.} \quad \|x\|_q \leq t,$$

where $q \in \{1, 2, \infty\}$. The dual problem is [32]

$$\max_{\lambda \in \mathbf{R}} \min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_p + \lambda(\|x\|_q - t)$$
$$\text{s.t.} \quad \lambda \geq 0.$$

The optimal pair $\{x^*, \lambda^*\}$ that meet the threshold t will depend on $\{A, b\}$. Conversely, there exists a t for every fixed selection of $\lambda > 0$. Selecting $\lambda > 0$, the regularized problem is defined as

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|_p + \lambda \|x\|_q.$$

Figure 11 shows ℓ_1 constraints with ℓ_2 residuals will preferentially seek solutions with ordinates at zero. This also holds true for ℓ_1 constraints with ℓ_∞ residuals, but it does not hold true for ℓ_2 or ℓ_∞ constraints.

6.1 2–Norm Residual with 1–Norm Regularization

This is known as LASSO (Least Absolute Shrinkage and Selection Operator).
Note: This form of regularization seeks solutions with zeros on the ordinates.

The regularized problem is given by

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J &= \|Ax - b\|_2^2 + \lambda \|x\|_1 \\ &= \frac{1}{2} x^\intercal Q x + c^\intercal x + r + \lambda \|x\|_1, \end{aligned}$$

where

$$\begin{aligned} Q &= 2A^\intercal A, \\ c &= -2A^\intercal b, \\ r &= b^\intercal b, \end{aligned}$$

which can be expressed as the **QP**

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^n\}} \quad J = \frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^\intercal \begin{bmatrix} Q & \mathbf{0}^{n \times n} \\ \mathbf{0}^{n \times n} & \mathbf{0}^{n \times n} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c \\ \mathbf{1}^n \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \end{bmatrix}$$
$$\text{s.t.} \quad \begin{bmatrix} -\lambda \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \lambda \boldsymbol{I}_n & -\boldsymbol{I}_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} \mathbf{0}^n \\ \mathbf{0}^n \end{bmatrix}.$$

6.2 Inf–Norm Residual with 1–Norm Regularization

Note: This form of regularization seeks solutions with zeros on the ordinates.

For $A \in \mathbf{R}^{m\times n}$, the regularized problem is given by

$$\min_{x\in\mathbf{R}^n} \quad J = \|Ax-b\|_\infty + \lambda\|x\|_1,$$

which can be expressed as the **LP**

$$\begin{aligned}
\min_{\{x,y,z\}\in\{\mathbf{R}^n,\mathbf{R}^n,\mathbf{R}\}} \quad & J = \begin{bmatrix} \mathbf{0}^n \\ \mathbf{1}^n \\ 1 \end{bmatrix}^\intercal \begin{bmatrix} x \\ y \\ z \end{bmatrix} \\
\text{s.t.} \quad & \begin{bmatrix} -\lambda\boldsymbol{I}_n & -\boldsymbol{I}_n & \mathbf{0}^n \\ \lambda\boldsymbol{I}_n & -\boldsymbol{I}_n & \mathbf{0}^n \\ -A & \mathbf{0}^{m\times n} & -\mathbf{1}^m \\ A & \mathbf{0}^{m\times n} & -\mathbf{1}^m \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \le \begin{bmatrix} \mathbf{0}^n \\ \mathbf{0}^n \\ -b \\ b \end{bmatrix}.
\end{aligned}$$

6.3 2–Norm Residual with 2–Norm Regularization

The regularized problem is given by the **QP**

$$\begin{aligned}
\min_{x\in\mathbf{R}^n} \quad J &= \|Ax-b\|_2^2 + \lambda\|x\|_2^2 \\
&= \frac{1}{2}x^\intercal Qx + c^\intercal x + b^\intercal b,
\end{aligned}$$

where

$$\begin{aligned}
Q &= \quad 2A^\intercal A + \lambda\boldsymbol{I}, \\
c &= -2A^\intercal b.
\end{aligned}$$

Computing $A = U_+S_+V_+^\intercal$ with SVD, the problem can be regularized with uniform circles (as depicted in Figure 11). Using

$$x = V_+S_+^{-1}y + V_0z,$$

the reformulation is give by

$$\begin{aligned}
\min_{\{y,z\}\in\{\mathbf{R}^r,\mathbf{R}^{n-r}\}} \quad J &= \|U_+y-b\|_2^2 + \lambda\|V_+S_+^{-1}y + V_0z\|_2^2 \\
&= \|y - U_+^\intercal b\|_2^2 + \lambda\|y\|_{2,S_+^{-1}}^2 + \lambda\|z\|_2^2 + b^\intercal U_0U_0^\intercal b \\
&= \frac{1}{2}\begin{bmatrix} y \\ z \end{bmatrix}^\intercal Q \begin{bmatrix} y \\ z \end{bmatrix} + c^\intercal \begin{bmatrix} y \\ z \end{bmatrix} + b^\intercal b,
\end{aligned}$$

where

$$\begin{aligned}
Q &= \begin{bmatrix} \boldsymbol{I}_r + \lambda S_+^{-2} & \mathbf{0}^{r\times n-r} \\ \mathbf{0}^{n-r\times r} & \lambda\boldsymbol{I}_{n-r} \end{bmatrix}, \\
c &= \begin{bmatrix} -2U_+^\intercal b \\ \mathbf{0}^{n-r} \end{bmatrix}.
\end{aligned}$$

Note: The residual is now circular, and the the regularizaton constraint is elliptical and aligned with the axis.

6.4 Generalized Regularization

A more general form of regularization can be given by

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\| + \sum_{i=0}^{n-1} f_i(x_i),$$

where $f_i(\cdot)$ is any convex penalty function that can be formulated as an **LP** or **QP** , e.g., deadzone or Huber.

6.5 Non–Convex Regularization

For $p \in \mathbf{R}(0,1)$, consider

$$\begin{aligned}\min_{x \in \mathbf{R}^n} \quad J &= \|Ax - b\|_{1/p} + \|x\|_p \\ &= \left(\sum_{i=0}^{m-1} |a_i^\intercal x - b_i|^{1/p}\right)^p + \left(\sum_{i=0}^{m-1} |x_i|^p\right)^{1/p}.\end{aligned}$$

This problem formulation gives exceptional sparsity for large p. It is equivalent to $1/p$–regularized p–constrained problem illustrated in Figure 12a. This problem is non–convex, which means multiple extremum may occur. Figure 12b illustrates non–unique solution.

The gradient is given by

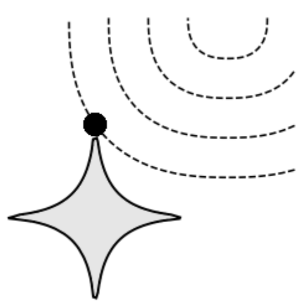
$$\frac{\partial J}{\partial x_j} = \|r\|_{1/p}^{-1} \left(\sum_{i=0}^{m-1} |r_i|^{1/p-1} \text{sign}(r_i) A_{ij}\right) + \|x\|_p^{-1} |x_j|^{p-1} \text{sign}(x_j),$$

where

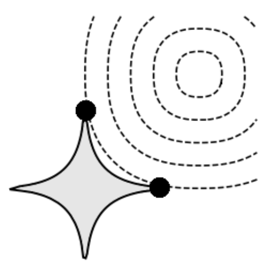
$$r_i = a_i^\intercal x - b_i,$$

and the Hessian is given by

$$\begin{aligned}\frac{\partial^2 J}{\partial x_j \partial x_k} = &\frac{p-1}{p} \|r\|_{1/p}^{-2} \left(\sum_{i=0}^{m-1} A_{ij}\, \text{sign}(r_i) |r_i|^{1/p-1}\right) \left(\sum_{i=0}^{m-1} A_{ik}\, \text{sign}(r_i) |r_i|^{1/p-1}\right) \\ &+ (1/p - 1)\|r\|_{1/p}^{-1} \left(\sum_{i=0}^{m-1} A_{ij} A_{ik} |r_i|^{1/p-2}\right) \\ &+ p(1/p-1)\|x\|_p^{-2} \text{sign}(x_j x_k) |x_j x_k|^{p-1} \\ &+ (p-1)\|x\|_p^{-1} |x_j|^{p-2} \text{Boolean}(j = k)\end{aligned}$$



(a) preference for a single axis



(b) possible non–unique solutions

Figure 12: non–convex regularization

7 Nonlinear Programming

Specialized solutions exist for specific forms of nonlinearity, but nonlinear programs do not have a general solution. Convex nonlinear programs expressed in standard form can be solved with interior point methods with gaurantees on determining feasibility and finding a global minimum. Non–convex nonlinear programs may have infinately many local extremum, with no gaurantees on determining feasibility or finding a global minimum. Nonlinear programs are typically solved by computing local first and second derivatives and taking small constrained or regulated steps starting at a random initial guess. For the following section, let $d = \|x - x_0\|_2$.

7.1 Linearizing Nonlinearity Inside Norm

Consider the nonlinear function $f(x) : \mathbf{R}^n \to \mathbf{R}^m$, and consider

$$\begin{aligned} J &= \|f(x)\| \\ &= \|f_0 + g_0^\intercal(x - x_0) + O(d^2)\| \\ &= \|A_0 x + b_0 + O(d^2)\|, \end{aligned}$$

where $f_0 = f(x_0)$,

$$\begin{aligned} A &= g^\intercal, \\ b &= f - g^\intercal x, \end{aligned}$$

and

$$g = \nabla f.$$

If $d^2 \le nr^2$ for a sufficiently small $r > 0$, the problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|f(x)\| \\ \text{s.t.} \quad & x \in \mathbf{P} \end{aligned}$$

can be solved with Algorithm 1 or 2. **Note**: Any of the penalty functions that result in **LP** or **QP** can be used.

$x_0 \in \mathbf{P}$, $r > 0$
while x_0 not converged **do**

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|A_0 x + b_0\| \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ -r\mathbf{1}^n \le & x - x_0 \le r\mathbf{1}^n. \end{aligned}$$

$x_0 \leftarrow x^*$
end

Algorithm 1: Nonlinear Solver with Step Bounding Box. Alternative: p–norm constraint with Equation [1].

$x_0 \in \mathbf{P}$, $\lambda > 0$
while x_0 not converged **do**

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|A_0 x + b_0\| + \lambda\|x - x_0\| \\ \text{s.t.} \quad & x \in \mathbf{P}. \end{aligned}$$

$x_0 \leftarrow x^*$
end

Algorithm 2: Nonlinear Solver with Step Regularization

7.2 2–Norm of Nonlinear Function

Consider the nonlinear function $f(x) : \mathbf{R}^n \to \mathbf{R}^m$. For positive definite $W \in \mathbf{S}_+^m$, consider

$$J = \|f(x)\|_{2,W}^2.$$

The linear expansion is

$$J = J_0 + g_0^\intercal (x - x_0) + O(d^2).$$

The quadratic expansion is

$$\begin{aligned} J &= J_0 + g_0^\intercal (x - x_0) + \frac{1}{2}(x - x_0)^\intercal H_0 (x - x_0) + O(d^3) \\ &= \frac{1}{2} x^\intercal Q_0 x + c_0^\intercal x + \text{constant} + O(d^3), \end{aligned}$$

where

$$\begin{aligned} Q_0 &= H_0, \\ c_0 &= g_0 - H_0 x_0. \end{aligned}$$

If $d^3 \le nr^3$ for a sufficiently small $r > 0$, the problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|f(x)\|_{2,W}^2 \\ \text{s.t.} \quad & x \in \mathbf{P} \end{aligned}$$

can be solved with Algorithm 3 or 4. If $\mathbf{P} = \mathbf{R}^n$, Algorithm 5 or 6 can be used.
Note: These algorithms can be extended to mixtures of ℓ_p.

7.2.1 Non–Diagonal Weight

For $M = W^\intercal W$, the gradient is calculated with

$$\begin{aligned} g^\intercal &= \nabla J^\intercal \\ &= 2 f^\intercal M h, \end{aligned}$$

where

$$h = \frac{\partial f}{\partial x},$$

and the Hessian is calculated with

$$\begin{aligned} H &= \nabla^2 J \\ &= \frac{\partial}{\partial x} 2h^\intercal M f \\ &= 2 \operatorname{col} \left\{ \frac{\partial h^\intercal}{\partial x_i} M f \right\}_{i=0}^{n-1} + 2h^\intercal M h \\ &= 2 \operatorname{col} \left\{ \frac{\partial h^\intercal}{\partial x_i} \right\}_{i=0}^{n-1} (\boldsymbol{I}_n \otimes M f) + 2h^\intercal M h. \end{aligned}$$

Alternatively,

$$\begin{aligned} H &= 2(\boldsymbol{I}_n \otimes f^\intercal M) \frac{\partial \operatorname{vec}(h^\intercal)}{\partial x} + 2h^\intercal M h \\ &= 2 \operatorname{row} \left\{ f^\intercal M \frac{\partial h}{\partial x_i} \right\}_{i=0}^{n-1} + 2h^\intercal M h. \end{aligned}$$

Note: $H = H^\intercal$.

7.2.2 Diagonal Weight

If $W = \text{diag}\{w_i\}_{i=0}^{m-1}$

$$J = \sum_{i=0}^{m-1} w_i^2 f_i^2.$$

The gradient is given by

$$\begin{aligned} g &= \nabla J \\ &= \sum_{i=0}^{m-1} 2w_i^2 f_i \nabla f_i \end{aligned}$$

The Hessian is given by

$$\begin{aligned} H &= \frac{\partial}{\partial x} \nabla J \\ &= \sum_{i=0}^{m-1} 2w_i^2 (\nabla f_i \nabla f_i^\intercal + f_i \nabla^2 f_i). \end{aligned}$$

7.2.3 Gradient Search

Consider

$$\min_{x \in \mathbf{R}^n} \quad J = J_0 + g_0^\intercal (x - x_0) + O(d^2).$$

If $\mathbf{P} = \mathbf{R}^n$, ignoring $O(d^2)$, stepping in the direction of maximum decline with $s \in \mathbf{R}$ and $M \in \mathbf{S}_+^n$,

$$\begin{aligned} x^*(s) &= x_0 - 2^s M g_0, \\ J^*(s) &= J_0 - 2^s g_0^\intercal M g_0 + O(2^{2s}). \end{aligned}$$

Algorithm 5 searches for an optimal s and then takes a scaled step.
Note: M may be normazlied by $\|g_0\|$ or $\|g_0\|^2$ at each iteration.

7.2.4 Hessian Search

Consider

$$\min_{x \in \mathbf{R}^n} \quad J = J_0 + g_0^\intercal (x - x_0) + \frac{1}{2}(x - x_0)^\intercal H_0 (x - x_0) + O(d^3).$$

If $\mathbf{P} = \mathbf{R}^n$, ignoring the $O(d^3)$ term, the optimal solution is found by solving

$$0 = g_0^\intercal + (x^* - x_0)^\intercal H_0,$$

which gives the pure–Newton iteration

$$x^* = x_0 - H_0^+ g_0.$$

Note: Selecting the matrix condition of the pseudo inverse with SVD can provide a well-conditioned step.
If the pure–Newton iteration is scaled by $s > 0$ with

$$x(s) = x_0 - s H_0^+ g_0,$$

then,

$$J(s) = J_0 + (s^2/2 - s) g_0^\intercal H_0^+ g_0 + O(s^3),$$

where $(s^2/2 - s)$ is negative for $s \in (0, 2)$, equal to zero at $s \in \{0, 2\}$, and minimum at $s = 1$.
Algorithm 6 searches for an optimal s and then takes a scaled step.

$x_0 \in \mathbf{P}$, $r > 0$
while x_0 not converged **do**

$$\min_{x \in \mathbf{R}^n} \quad J = \frac{1}{2} x^\intercal Q_0 x + c_0^\intercal x$$

$$\text{s.t.} \quad x \in \mathbf{P},$$
$$-r\mathbf{1}^n \leq x - x_0 \leq r\mathbf{1}^n.$$

$x_0 \leftarrow x^*$
end

Algorithm 3: Nonlinear ℓ_2 Solver with Step Bounding Box. Alternative: p–norm constraint with Equation [1].

$x_0 \in \mathbf{P}$, $\lambda > 0$
while x_0 not converged **do**

$$\min_{x \in \mathbf{R}^n} \quad J = \frac{1}{2} x^\intercal Q_0 x + c_0^\intercal x + \lambda \|x - x_0\|$$

$$\text{s.t.} \quad x \in \mathbf{P}.$$

$x_0 \leftarrow x^*$
end

Algorithm 4: Nonlinear ℓ_2 Solver with Step Regularization

$x_0 \in \mathbf{P}$, $M \in \mathbf{S}_+^n$
while x_0 not converged **do**

$$x(s) = x_0 - 2^s M g_0,$$

$$\min_{s \in \mathbf{R}} J(x(s)).$$

$x_0 \leftarrow x^*$
end

Algorithm 5: Unconstrained Nonlinear ℓ_2 Solver with Gradient Line Search

$x_0 \in \mathbf{P}$
while x_0 not converged **do**

$$x(s) = x_0 - sH_0^+ g_0,$$

$$\min_{s \in \mathbf{R}(0,2)} J(x(s))$$

$x_0 \leftarrow x^*$
end

Algorithm 6: Unconstrained Nonlinear ℓ_2 Solver with Hessian Line Search

7.3 Nonlinear Equality Constraints

Consider the nonlinear function $f_{\text{eq}}(x) : \mathbf{R}^n \to \mathbf{R}^m$, and consider the linear p-norm with nonlinear constraint

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|Ax - b\| \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & f_{\text{eq}}(x) = 0. \end{aligned} \tag{5}$$

7.3.1 Penalty Approach

The nonlinear constraint in Equation 5 can be brought into the objective function with

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\| + \|f_{\text{eq}}(x)\|_W$$

$$\text{s.t.} \quad x \in \mathbf{P}.$$

Note: The minimum singular value of W must be much larger than the maximum singular value of A.
If the residual and regularization are ℓ_2, then

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J &= \|Ax - b\|_2^2 + \|f_{\text{eq}}(x)\|_{2,W}^2 \\ &= x^\intercal A^\intercal Ax - 2b^\intercal Ax + b^\intercal b + \|f_{\text{eq}}(x_0)\|_{2,W}^2 + g_0^\intercal (x - x_0) + \frac{1}{2}(x - x_0)^\intercal H_0 (x - x_0) + O(d^3) \\ &= \frac{1}{2} x^\intercal Q_0 x + c_0^\intercal x + \text{constant} + O(d^3) \end{aligned}$$

$$\text{s.t.} \quad x \in \mathbf{P},$$

where

$$\begin{aligned} Q_0 &= \quad 2A^\intercal A + H_0 \\ c_0 &= -2A^\intercal b + g_0 - H_0 x_0, \end{aligned}$$

which can be solved with Algorithm 3 or 4.

7.3.2 Linearization

The nonlinear constraint in Equation 5 can be linearized with

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|$$

$$\begin{aligned} \text{s.t.} \quad & x \in \mathbf{P}, \\ & B_0 x = c_0 + O(d^2), \end{aligned}$$

where

$$\begin{aligned} B &= g^\intercal, \\ c &= g^\intercal x - f_{\text{eq}}, \end{aligned}$$

and

$$g = \nabla f_{\text{eq}}.$$

The constraint $d^2 \leq nr^2$ can be added as a box (Algrothim 7), p–norm (Algorithm 8), or regularization (Algorithm 9).

$x_0 \in \mathbf{P}$, $r > 0$
while x_0 not converged **do**

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ B_0\, x &= c_0 \\ -r\mathbf{1}^n \leq & x - x_0 \leq r\mathbf{1}^n. \end{aligned}$$

$x_0 \leftarrow x^*$
end

Algorithm 7: Nonlinear Constrained Solver with Step Bounding Box.

$x_0 \in \mathbf{P}$, $r > 0$, $p \in \{1, \infty\}$
while x_0 not converged **do**

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\|$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ B_0\, x &= c_0, \\ \|x - x_0\|_p &\leq n^{1/p} r. \end{aligned}$$

$x_0 \leftarrow x^*$
end

Algorithm 8: Nonlinear Constrained Solver with p–norm constraint solved with Equation [1]

$x_0 \in \mathbf{P}$, $\lambda > 0$, $a > 1$
while $\lambda < \lambda^\circ$ **do**
  while x_0 not converged **do**

$$\min_{x \in \mathbf{R}^n} \quad J = \|Ax - b\| + \lambda \|B_0\, x - c_0\|$$

$$\text{s.t.} \quad x \in \mathbf{P}.$$

  $x_0 \leftarrow x^*$
  end
  $\lambda \leftarrow a\lambda$
end

Algorithm 9: Nonlinear Constrained Solver with Penalty

7.3.3 Nonlinear Inequality

Consider the nonlinear inequality $f_{\text{ub}}(x) \leq 0$. A slack variable $y \in \mathbf{R}^m$ can be used to recover nonlinear equality with

$$\min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad J = \|Ax - b\|$$

$$\begin{aligned} \text{s.t.} \quad x &\in \mathbf{P}, \\ f_{\text{ub}}(x) + y &= 0, \\ y &\geq 0. \end{aligned}$$

7.3.4 Logarithmic Barrier

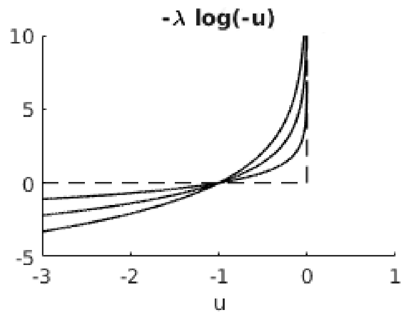


Figure 13: Log–Barrier Penalty

Adding a slack variable, the nonlinar constraint in Equation 5 can be re–expressed as the nonlinear inequality problem

$$\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad J = \|Ax-b\| + \|y\|_W$$

$$\text{s.t.} \quad -f_{\text{eq}}(x) - y \le 0, \\ f_{\text{eq}}(x) - y \le 0.$$

Note: These constraints force $y \ge 0$.

For $\lambda > 0$, the logrithmic barrier problem is given by [1, p. 562]

$$\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad J = \|Ax-b\| + \|y\|_W + \lambda h(x,y),$$

where

$$h = -\sum_{i=0}^{m-1} \left(\log(y_i+f_i) + \log(y_i-f_i)\right).$$

The gradient of h is

$$\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \nabla h,$$

where

$$\begin{aligned} g_1^{\intercal} &= \frac{\partial h}{\partial x} \\ &= -\sum_{i=0}^{m-1}\left(\frac{1}{y_i+f_i} - \frac{1}{y_i-f_i}\right)\frac{\partial f_i}{\partial x}, \end{aligned}$$

$$\begin{aligned} g_2^{\intercal} &= \frac{\partial h}{\partial y} \\ &= -\sum_{i=0}^{m-1}\left(\frac{1}{y_i+f_i} + \frac{1}{y_i-f_i}\right)\mathbf{e}_i^{\intercal}. \end{aligned}$$

Note: This problem formulation is equivalent to the regularized problem

$$\min_{x\in\mathbf{R}^n} \quad J = \|Ax-b\| + \|f_{\text{eq}}(x)\|_W.$$

7.4 Generalized Reformulations

There are several generalized problem reformulations.

7.4.1 Projection

Consider nonlinear f, f_{ub}, and f_{eq}. Any problem can be stated as

$$\begin{aligned}\min_{x\in\mathbf{R}^n} \quad & \|f(x)\| \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & f_{\text{ub}}(x) \le b_{\text{ub}}, \\ & f_{\text{eq}}(x) = b_{\text{eq}}.\end{aligned}$$

Consider a one-to-one projective basis $h(x) : \mathbf{R}^n \to \mathbf{R}^m$, and project over the domain $x \in \mathbf{P}$ to get

$$\begin{aligned}f\ \ (x) &\approx A\ \ h(x), \\ f_{\text{ub}}(x) &\approx A_{\text{ub}}h(x), \\ f_{\text{eq}}(x) &\approx A_{\text{eq}}h(x),\end{aligned}$$

where

$$\begin{aligned}A\ \ &= \left(\int_{x\in\mathbf{P}} f\ \ h^\intercal dx\right)\left(\int_{x\in\mathbf{P}} hh^\intercal dx\right)^+, \\ A_{\text{ub}} &= \left(\int_{x\in\mathbf{P}} f_{\text{ub}}h^\intercal dx\right)\left(\int_{x\in\mathbf{P}} hh^\intercal dx\right)^+, \\ A_{\text{eq}} &= \left(\int_{x\in\mathbf{P}} f_{\text{eq}}h^\intercal dx\right)\left(\int_{x\in\mathbf{P}} hh^\intercal dx\right)^+.\end{aligned}$$

The problem can be approximated by

$$\begin{aligned}\min_{y\in\mathbf{R}^m} \quad & \|Ay\| \\ \text{s.t.} \quad & y \in \mathbf{D} \\ & A_{\text{ub}}y \le b_{\text{ub}}, \\ & A_{\text{eq}}y = b_{\text{eq}},\end{aligned}$$

where

$$\mathbf{D} = \{h(x)|x \in \mathbf{P}\}.$$

Alternatively, the problem can be approximated by the nonlinear constrained problem

$$\begin{aligned}\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad & \|Ay\| \\ \text{s.t.} \quad & x \in \mathbf{P} \\ & A_{\text{ub}}y \le b_{\text{ub}}, \\ & A_{\text{eq}}y = b_{\text{eq}}, \\ & y = h(x).\end{aligned}$$

The penalty approach can be used to solve this problem.

7.4.2 Epigraph

Any problem is equivalent to

$$\begin{aligned} \min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad & J = y \\ \text{s.t.} \quad & f(x) - y \leq 0 \\ & x \in \mathbf{D}. \end{aligned}$$

7.4.3 Indicator Function

Any problem is equivalent to

$$\begin{aligned} \min_{x\in\mathbf{R}^n} \quad & f(x) \\ \text{s.t.} \quad & \text{Indicator}(x \in \mathbf{D}) \leq 0, \end{aligned}$$

where

$$\text{Indicator}(x \in \mathbf{D}) = \begin{cases} 0 & \text{if} \quad x \in \mathbf{D} \\ \infty & \text{else} \end{cases}.$$

7.5 Convex Programming

For $i \in \mathbf{Z}[0, m]$ and convex $f_i(x)$, a general convex problem can be expressed as

$$\begin{aligned} \min_{x\in\mathbf{R}^n} \quad & J = f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \\ & Ax = b. \end{aligned}$$

Note: The equality constraint must be linear [1, p. 137].
For $i \in \mathbf{Z}[1, m]$, the constraints define the set

$$\mathbf{D} = \{x \in \mathbf{R}^n | f_i(x) \leq 0, Ax = b\}.$$

For all $i \in \mathbf{Z}[0, m]$, $x \in \mathbf{D}$, $y \in \mathbf{D}$, and $p \in \mathbf{R}[0, 1]$, the problem is convex if

$$f_i(px + (1 - p)y) \leq pf_i(x) + (1 - p)f_i(y).$$

Any convex problem can be solved with interior point methods.
Several convex problems are given special consideration with specialized simplifications and solutions, e.g.,

- **LP** : Linear Programs,
- **QP** : Quadratic Programs,
- **SOCP** : Second Order Cone Programs [1, p. 158],
- **QCQP** : Quadratic Constrained Quadratic Programs [1, p. 152][42],
- **GP** : Geometric Programs [1, p. 161],
- **LMI** : Linear Matrix Inequalities [1, p. 169],
- **SDP** : Semi Definite Programs [1, p. 168][42].

Solvers for each of these specialized convex problems can be found in many programming languages.
It is often worth the effort to formulate a problem as one of these standard problems.

7.5.1 QCQP : Quadratic Constrained Quadratic Programming

For $i \in \mathbf{Z}[0, m-1]$, any convex **QCQP** can be expressed as

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \frac{1}{2} x^\intercal Q x + c^\intercal x \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \|A_i x - b_i\|_2^2 \leq 1. \end{aligned}$$

Note: If $A_i \in \mathbf{R}^{n \times n}$ with rank$(A_i) = n$,
then $\|A_i x - b_i\|_2^2 = \|A_i(x - A_i^{-1} b_i)\|_2^2$ is an ellipse centered at $y_i = A_i^{-1} b_i$ with covariance $(A_i^\intercal A_i)^{-1}$.

A **QCQP** can also be expressed as

$$\begin{aligned} \min_{\{x, z_i\} \in \{\mathbf{R}^n, \mathbf{R}^{m_i}\}} \quad & J = \frac{1}{2} x^\intercal Q x + c^\intercal x \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & \|z_i\|_2^2 \leq 1, \\ & A_i x - z_i = b_i. \end{aligned}$$

7.5.2 SDP : Semi–Definite Programming

For $A_i \in \mathbf{S}^n$ and $B \in \mathbf{S}^n$, the primal **SDP** is

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = c^\intercal x \\ \text{s.t.} \quad & \sum_{i=0}^{n-1} x_i A_i \preceq B. \end{aligned}$$

This form is also known as an **LMI** (linear matrix inequality).

The Lagrangian is

$$L(x, Y) = c^\intercal x + \text{tr}\left(Y \sum_{i=0}^{n-1} x_i A_i\right) - \text{tr}\,(YB)\,,$$

with

$$\inf_{x \in \mathbf{R}^n} L(x, Y) = \begin{cases} -\text{tr}\,(BY) & \text{if} \quad \text{tr}\,(A_i Y) + c_i = 0 \\ -\infty & \text{else} \end{cases}.$$

The dual equals the primal if the primal inequality is strictly feasible. The dual problem is given by

$$\begin{aligned} \min_{Y \in \mathbf{R}^{n \times n}} \quad & J = \text{tr}\,(BY) \\ \text{s.t.} \quad & \text{tr}\,(A_i Y) = -c_i, \\ & Y \succeq 0, \end{aligned}$$

For $i \in \mathbf{Z}[0, n-1]$, the positive definite constraint gives inequality constraints of the form

$$-\det(M_i) \leq 0,$$

where M_i are the principle minors of Y, i.e., $M_i = Y[0:i, 0:i]$.

If the **SDP** is optimized with the interior point method,

$$\frac{\partial}{\partial Y} \text{tr}(YB) = B^\intercal = B,$$
$$\frac{\partial}{\partial Y} \log\det(M_i) = \text{tr}\left(M_i^{-1} \frac{\partial M_i}{\partial Y}\right).$$

Note: If $B \in \mathbf{S}^n$ and $Y \in \mathbf{S}^n$, $\text{tr}\,(BY) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} B_{ij} Y_{ij} = \vec{B}^\intercal \vec{Y}$.

Note: $x^\intercal Q x = \text{tr}\,(Q x x^\intercal) = \text{tr}\,(QX)$ with $X \succeq 0$ and rank$(X) = 1$.

7.6 Applications

Figure 14 depicts many of the common applications in machine learning. All of these applications can be framed as optimization problems. Ideally, optimization should be convex for reliable performance. Some of these optimization problems can be cast as **LP** or **QP** , or they can be solved with iterative **LP** or **QP** methods.

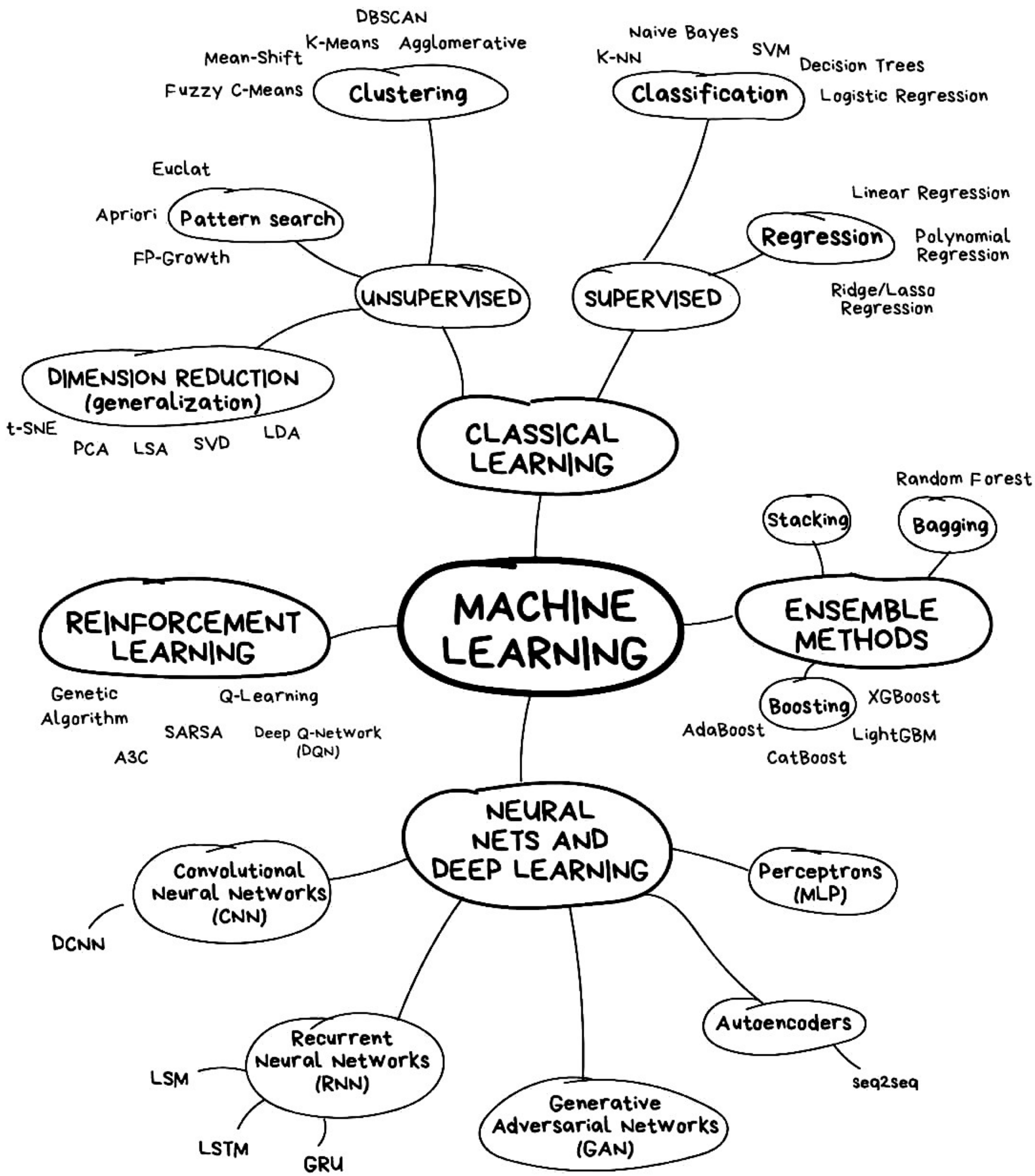


Figure 14: Machine Learning

8 Sigmoids

Sigmoids are a common form of nonlinearity in generalized machine learning applications.
Many of them can be expressed as derivatives of the convex penalty functions discussed above.

8.1 Boolean Sigmoids

Boolean sigmoids map $s : \mathbf{R} \to \mathbf{R}[0, 1]$. Figure 15 shows a sigmoid computed from the gradient of ReLu, and Figure 16 shows a sigmoid computed from the gradient of a right-sidded Huber penalty.

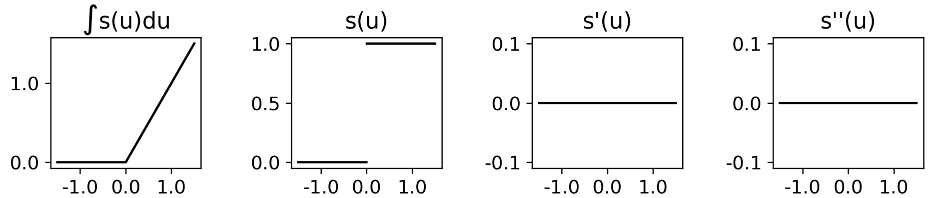


Figure 15: $s(u) = \text{Boolean}(u > 0)$

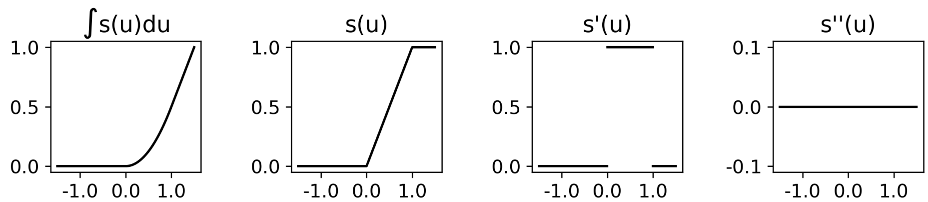


Figure 16: $s(u) = \text{Saturation}(u, 0, 1)$

The logistic sigmoid is defined as

$$s(u) = (1 + e^{-u})^{-1},$$

with

$$\begin{aligned} s'(u) &= s(1 - s), \\ s''(u) &= s(1 - s)^2 - s^2(1 - s), \\ \int s(u)du &= \log(1 + e^u). \end{aligned}$$

Note: This function curves primarily for $u \in \mathbf{R}[-2\pi, 2\pi]$.

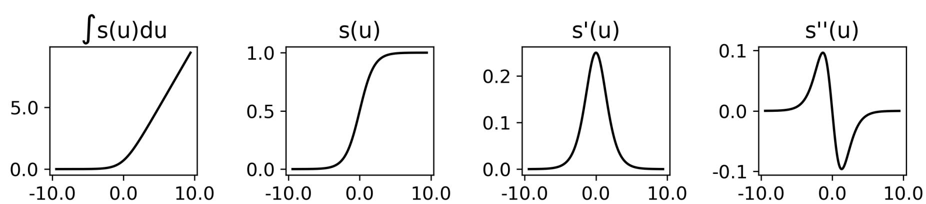


Figure 17: $s(u) = (1 + e^{-u})^{-1}$

In many applications sigmoid functions that map $s : \mathbf{R} \to \mathbf{R}[-1, 1]$ are used instead of $s : \mathbf{R} \to \mathbf{R}[0, 1]$.
It is trivial to re–map in either case.

8.2 Piecewise–Polynomial Sigmoids

There are many ways to obtain an s–shaped curve. Consider a piecewise polynomial given by

$$s_n(u) = \begin{cases} \sum_{i=0}^{2n+1} (a_i/i!)u^i & \text{if} \quad u \in \mathbf{R}[-1,1] \\ \text{sign}(u) & \text{else} \end{cases} ,$$

where $s(-1) = -1$, $s(1) = 1$, and all derivatives at $u \in \{-1, 1\}$ are zero. For $n \in \mathbf{Z}[0, 3]$ and $|u| < 1$, this gives

$$\begin{aligned} s_0(u) &= u, \\ s_1(u) &= (3u - u^3)/2, \\ s_2(u) &= (15u - 10u^3 + 3u^5)/8, \\ s_3(u) &= (35u - 35u^3 + 21u^5 - 5u^7)/16, \end{aligned}$$

Figures 18–21 show $n \in \mathbf{Z}[0, 3]$.
Note: The Huber penalty function at $t = 1$ is given by $f_0 = \int s_0(u)du$.

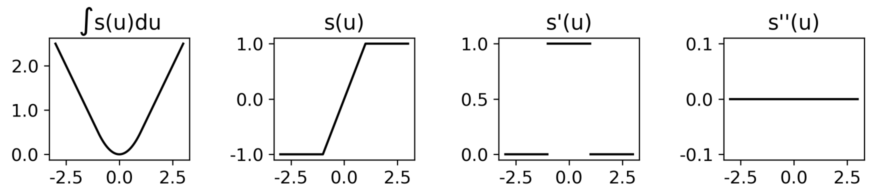


Figure 18: $s_0(u) = \text{Saturation}(u)$

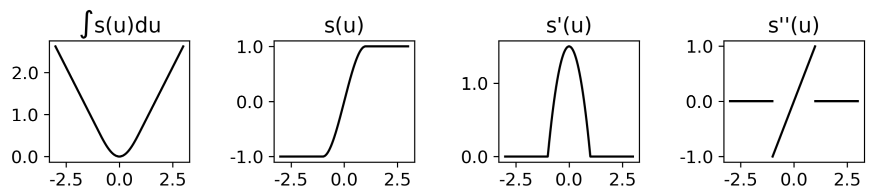


Figure 19: $s_1(u)$

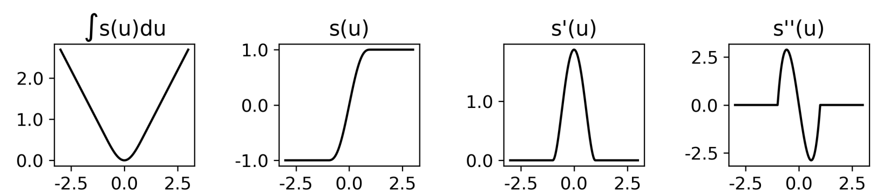


Figure 20: $s_2(u)$

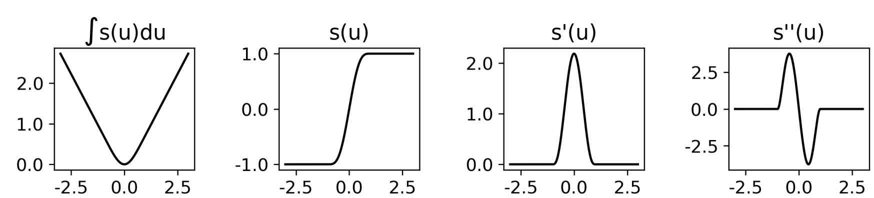


Figure 21: $s_3(u)$

Dividing $f_i = \int s_i(u)du$ by u gives alternative sigmoid functions. For $|u| < 1$,

$$
\begin{aligned}
f_0/u &= u/2,\\
f_1/u &= \left(6u - u^3\right)/8,\\
f_2/u &= \left(15u - 5u^3 + u^5\right)/16,\\
f_3/u &= \left(140u - 70u^3 + 28u^5 - 5u^7\right)/128.
\end{aligned}
$$

For $|u| > 1$,

$$\frac{f_i}{u} = \text{sign}(u) + \frac{f_i(1) - 1}{u}.$$

The gradient approaches zero asymptotically. Figures 22–25 show $n \in \mathbf{Z}[0, 3]$.

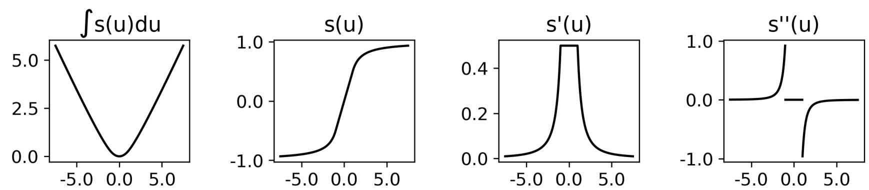


Figure 22: $s_0(u) \leftarrow f_0(u)/u$

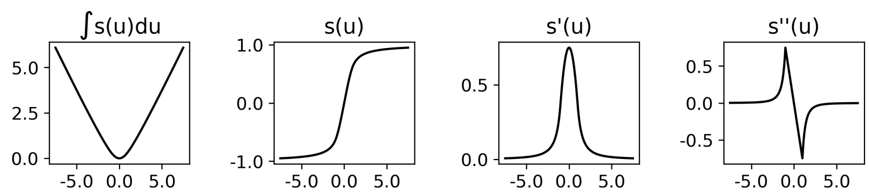


Figure 23: $s_1(u) \leftarrow f_1(u)/u$

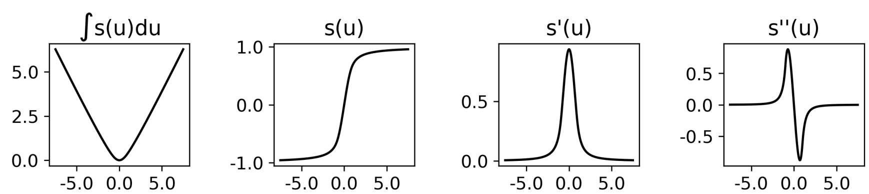


Figure 24: $s_2(u) \leftarrow f_2(u)/u$

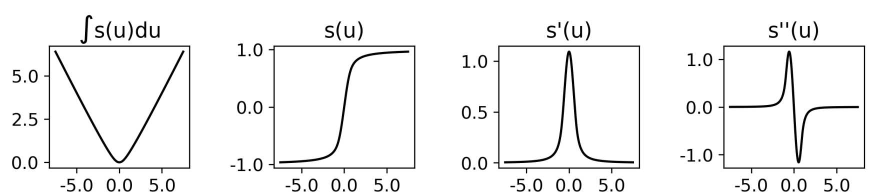


Figure 25: $s_3(u) \leftarrow f_3(u)/u$

Note: For $r_i = a_i^\intercal x - b_i$,

$$
\begin{aligned}
\|Ax - b\|_1 &\approx \textstyle\sum_{i=0}^{m-1} r_i s(r_i),\\
\frac{\partial}{\partial x}\|Ax - b\|_1 &\approx \textstyle\sum_{i=0}^{m-1} (s(r_i) + r_i s'(r_i)) a_i^\intercal.
\end{aligned}
$$

8.3 Deadzone Sigmoid

For the deadzone penalty $f(u)$, consider the sigmoid function given by

$$s(u) = \frac{f(u)}{u}$$
$$= \begin{cases} 0 & \text{if } |u| \leq t \\ \text{sign}(u) - t/u & \text{else} \end{cases}.$$

Figure 26 plots the sigmoid curve. **Note**: In the limit of $|u| \to \infty$, the sigmoid approaches sign(u) asymptotically.
Note: The asymptotic approach is much slower than a logistic sigmoid.
A constraint of the form

$$y = s(u)$$

can be expressed as

$$yu = \begin{cases} 0 & \text{if } |u| \leq t \\ |u| - t & \text{else} \end{cases} \quad \text{or} \quad (yu + t)^2 = \begin{cases} t^2 & \text{if } |u| \leq t \\ u^2 & \text{else} \end{cases}.$$

The first derivative is

$$s'(u) = \begin{cases} 0 & \text{if } |u| \leq t \\ t/u^2 & \text{else} \end{cases},$$

and the second derivative is

$$s''(u) = \begin{cases} 0 & \text{if } |u| \leq t \\ -2t/u^3 & \text{else} \end{cases}.$$

Note: When $t = 0$, $s(u) = \text{sign}(u)$ and the constraint can be expressed as $yu = |u| = \max(-u, u)$.

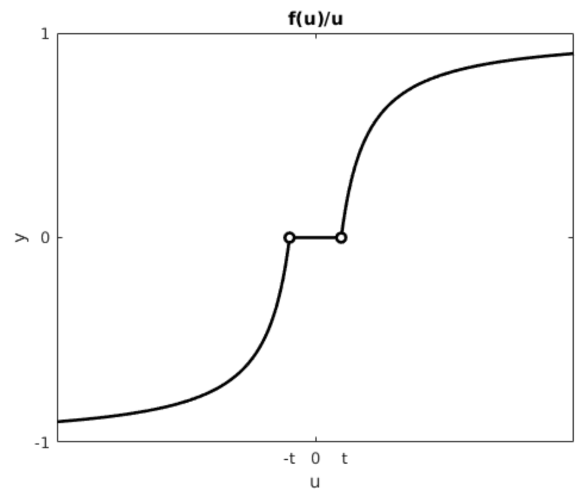


Figure 26: Deadzone Sigmoid

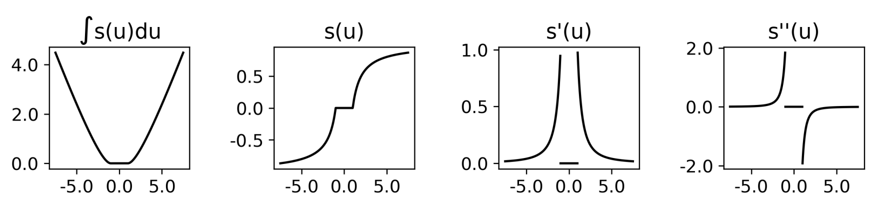


Figure 27: Deadzone Sigmoid

8.4 Huber Sigmoid

For the Huber penalty $f(u)$, consider the sigmoid function given by

$$\begin{aligned} s(u) &= \frac{f(u)}{2u} \\ &= \frac{1}{2}\begin{cases} u & \text{if } |u| \leq t \\ 2t\ \mathrm{sign}(u) - t^2/u & \text{else} \end{cases} . \end{aligned}$$

Figure 28 plots the sigmoid curve.
Note: In the limit of $|u| \to \infty$, the sigmoid approaches $\mathrm{sign}(u)$ asymptotically. **Note**: The asymptotic approach is much slower than a logistic sigmoid.
A constraint of the form

$$y = s(u)$$

can be expressed as

$$2uy = \begin{cases} u^2 & \text{if } |u| \leq t \\ 2t|u| - t^2 & \text{else} \end{cases} \quad \text{or} \quad (2yu + t^2)^2 = \begin{cases} (u^2 + t^2)^2 & \text{if } |u| \leq t \\ 4t^2u^2 & \text{else} \end{cases} ,$$

which has a unique solution for any choice of $y \in \mathbf{R}[-t, t]$. The Huber sigmoid has a continuous first derivative

$$s'(u) = \frac{1}{2}\begin{cases} 1 & \text{if } |u| \leq t \\ t^2/u^2 & \text{else} \end{cases} ,$$

and a discontinuous second derivative

$$s''(u) = \begin{cases} 0 & \text{if } |u| \leq t \\ -t^2/u^3 & \text{else} \end{cases} .$$

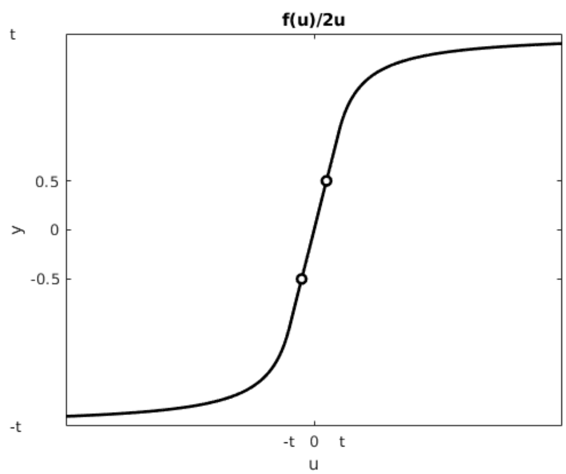


Figure 28: Huber Sigmoid

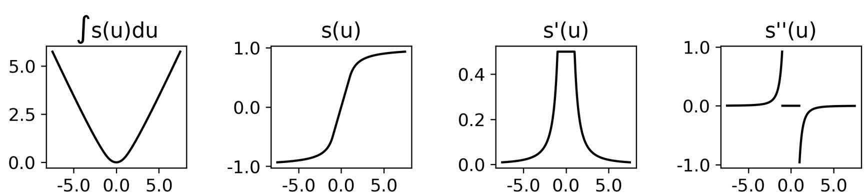


Figure 29: Huber Sigmoid

8.5 Misc. Sigmoids

8.5.1 Trig Sigmoids

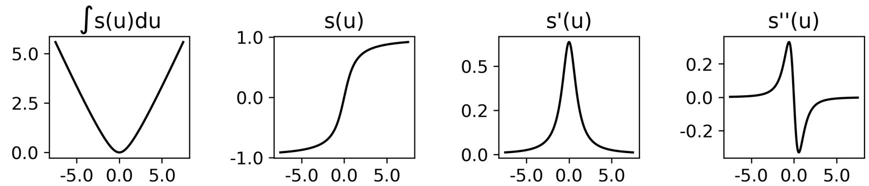


Figure 30: $s(u) = 2\,\text{atan}(u)/\pi$

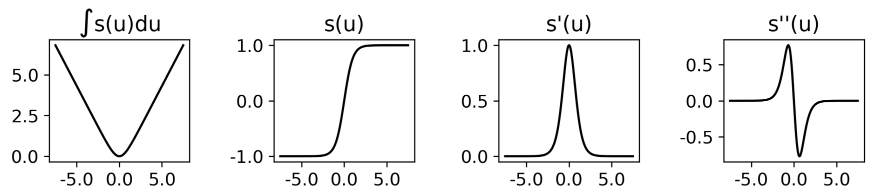


Figure 31: $s(u) = \tanh(u)$

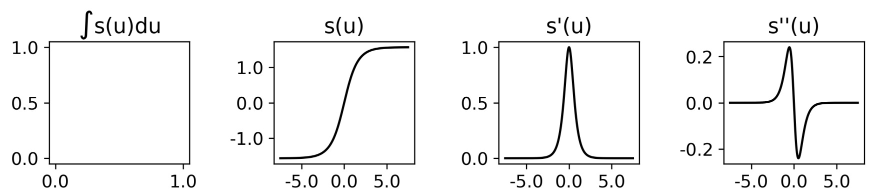


Figure 32: $s(u) = 2\,\text{atan}\left(\tanh(u/2)\right)$

8.5.2 Exponential Sigmoids

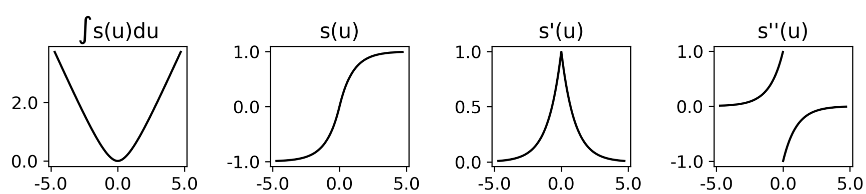


Figure 33: $s'(u) = \exp(-|u|)$

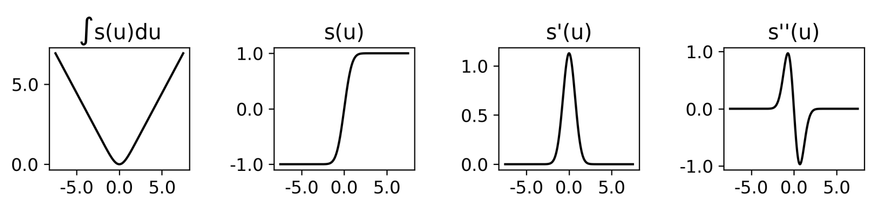


Figure 34: $s(u) = \text{erf}(u), \quad s'(u) = 2\exp(-u^2)/\sqrt{\pi}$

8.5.3 Ramp Sigmoid

Ramp velocity profiles are common in robotics applications, e.g. Figure 35.

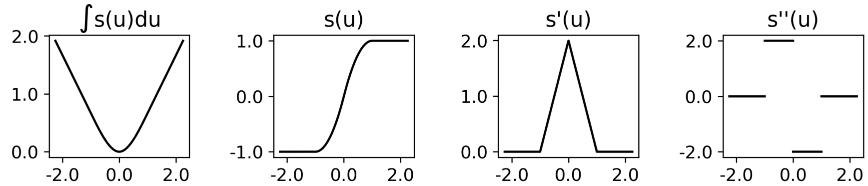


Figure 35: Ramp

8.5.4 Parabolic Sigmoids

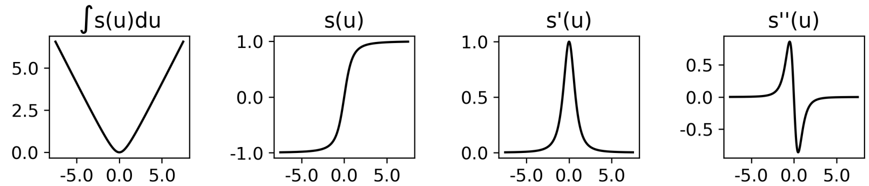


Figure 36: $s(u) = u/\sqrt{1+u^2}$, $\quad \int s(u)du = \sqrt{1+u^2} - 1$

8.5.5 Reciprocal Sigmoids

To obtain sigmoids with slow asymptotic convergence, consider reciprocal functions, e.g., linear–reciprocal (Figure 37)

$$s(u) = \begin{cases} (1/u - 1)^{-1} & \text{if} \quad u < 0 \\ (1/u + 1)^{-1} & \text{else} \end{cases},$$

or log–reciprocal (Figure 38)

$$s(u) = \begin{cases} \log(2)\log(2-u)^{-1} - 1 & \text{if} \quad u < 0 \\ -\log(2)\log(2+u)^{-1} + 1 & \text{else} \end{cases}.$$

These sigmoids both have an $s'(u)$ that converges to 0 very slowly.

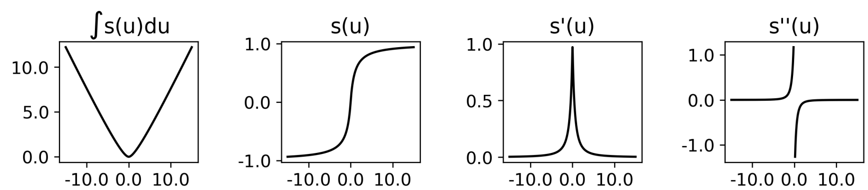


Figure 37: Reciprocal Linear

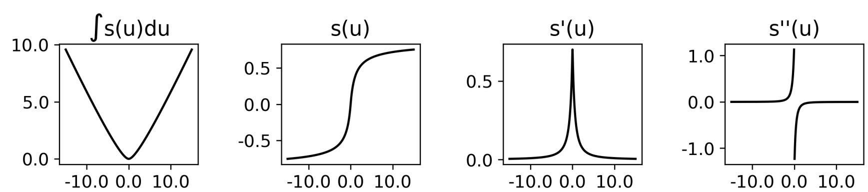


Figure 38: Reciprocal Log

9 NN: Neural Networks

Classical multi–layer perceptrons have been around for several decades. They are well–suited to learning mappings between Boolean tables but suffered early set backs (not living up to the early hype). In recent years, there has a been a resurgence in neural network research with fantastic results in classification [52] and generation of complex data sets from easily interperted low level features [49], [50]. The discoveries that have made this possible are pre– and post–filters that provide appropriately conditioned output for classical multi–layer perceptrons to excel, e.g., VAE (variational auto–encoders) and CNN (convolutional neural networks). These filters convert complex data into easily seperable or predictable features. In the case of CNN, these features are much larger than the input. In the case of VAE, these features are much smaller than the input.

Using only multi–layer perceptrons injected with noise, generative methods (mapping low feature spaces to complex data spaces) have recently proven more powerful than state–of–the–art Markov–based methods. One of my rearch goals is to combine cascading RBST with the machinery of RNN, GAN, and LSTM to find scalable real–world application, similar to the results of [48] but on a much larger scale.

The forward calculation of an n-layer network mapping x to y is shown in Algorithm 10. Neural networks are nonlinear and non–convex, but their optimization is greatly aided with a deeper understanding of **QP** . Using the nonlinear **QP** solvers in this repo, many modifications to vanilla back–propogated ℓ_2 error can be explored, e.g., p–norm regularization, p–norm constraint, upper and lower parameter bounds, relaxations with slack variables, and robust penalties like Huber and dedzone. Enhanced solvers that take advantage of sparsity and parallelization are also of interest.

$z_0 \leftarrow s\left(A_0 x - b_0\right)$
for $i \in \mathbf{Z}[1, n-1]$ **do**
| $z_i \leftarrow s\left(A_i z_{i-1} - b_i\right)$
end
return $y \leftarrow z_{n-1}$

Algorithm 10: Forward NN

9.1 1–Layer Perceptron

For $h : \mathbf{R}^n \to \mathbf{R}^m$, let

$$s(h) = \text{row}\{s(h_i)\}_{i=0}^{m-1},$$
$$s'(h) = \text{diag}\{s'(h_i)\}_{i=0}^{m-1},$$

where s could be any of the proposed functions in the sigmoid section.
For $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^m$, consider

$$\min_{\{A,b\} \in \{\mathbf{R}^{m\times n}, \mathbf{R}^m\}} \quad \sum_{\{x,y\}} J(x, y),$$

where

$$\begin{aligned} J(x, y) &= \frac{1}{2}\|s(Ax - b) - y\|_{2,W(x,y)}^2 \\ &= \frac{1}{2}\|z - y\|_{2,W(x,y)}^2, \end{aligned}$$

and

$$\begin{aligned} z &= s(Ax - b) \\ &= s((\boldsymbol{I}_m \otimes x^{\mathsf{T}})\vec{A} - b). \end{aligned}$$

Let

$$w = W^{\mathsf{T}} W(z - y).$$

The partials are

$$\frac{\partial J}{\partial x} = w^\intercal \frac{\partial z}{\partial x},$$

$$\frac{\partial J}{\partial \vec{A}} = w^\intercal \frac{\partial z}{\partial \vec{A}},$$

$$\frac{\partial J}{\partial b} = w^\intercal \frac{\partial z}{\partial b},$$

where

$$\frac{\partial z}{\partial x} = s'(Ax - b)A,$$

$$\frac{\partial z}{\partial \vec{A}} = s'(Ax - b)(\boldsymbol{I}_m \otimes x^\intercal),$$

$$\frac{\partial z}{\partial b} = -s'(Ax - b).$$

9.2 2–Layer Perceptron

For $z_i \in \mathbf{R}^{m_i}$ with $i \in \mathbf{Z}[0, n-1]$ and $n = 2$, consider

$$\min_{\{A_i, b_i\} \in \{\mathbf{R}^{m_i \times m_{i-1}}, \mathbf{R}^{m_i}\}} \quad \sum_{\{x,y\}} J(x, y),$$

where

$$\begin{aligned} J(x, y) &= \frac{1}{2}\|s(A_1 s(A_0 x - b_0) - b_1) - y\|^2_{2,W(x,y)} \\ &= \frac{1}{2}\|z_1 - y\|^2_{2,W(x,y)}, \end{aligned}$$

and

$$\begin{aligned} z_0 &= s(A_0 x - b_0), \\ z_1 &= s(A_1 z_0 - b_1). \end{aligned}$$

Let

$$w = W^\intercal W(z_1 - y).$$

The partials for A_i and b_i are

$$\begin{aligned} \frac{\partial J}{\partial \vec{A}_1} &= w^\intercal \frac{\partial z_1}{\partial \vec{A}_1}, \\ \frac{\partial J}{\partial \vec{A}_0} &= w^\intercal \frac{\partial z_1}{\partial z_0} \frac{\partial z_0}{\partial \vec{A}_0}, \end{aligned}$$

$$\begin{aligned} \frac{\partial J}{\partial b_1} &= w^\intercal \frac{\partial z_1}{\partial b_1}, \\ \frac{\partial J}{\partial b_0} &= w^\intercal \frac{\partial z_1}{\partial z_0} \frac{\partial z_0}{\partial b_0}, \end{aligned}$$

where

$$\frac{\partial z_1}{\partial z_0} = s'(A_1 z_0 - b_1)A_1,$$

$$\begin{aligned}\frac{\partial z_1}{\partial \vec{A}_1} &= s'(A_1 z_0 - b_1)(\boldsymbol{I}_{m_1} \otimes z_0^\intercal),\\ \frac{\partial z_1}{\partial b_1} &= -s'(A_1 z_0 - b_1),\end{aligned}$$

$$\begin{aligned}\frac{\partial z_0}{\partial \vec{A}_0} &= s'(A_0 x - b_0)(\boldsymbol{I}_{m_0} \otimes x^\intercal)\\ \frac{\partial z_0}{\partial b_0} &= -s'(A_0 x - b_0).\end{aligned}$$

9.3 3–Layer Perceptron

For $z_i \in \mathbf{R}^{m_i}$ with $i \in \mathbf{Z}[0, n-1]$ and $n = 3$, consider

$$\min_{\{A_i, b_i\} \in \{\mathbf{R}^{m_i \times n_i}, \mathbf{R}^{m_i}\}} \sum_{\{x,y\}} J(x, y)$$

where

$$\begin{aligned}J(x, y) &= \frac{1}{2}\|s(A_2 s(A_1 s(A_0 x - b_0) - b_1) - b_2) - y\|_{2,W(x,y)}^2\\ &= \frac{1}{2}\|z_2 - y\|_{2,W(x,y)}^2,\end{aligned}$$

and

$$\begin{aligned}z_0 &= s(A_0 x - b_0),\\ z_1 &= s(A_1 z_0 - b_1),\\ z_2 &= s(A_2 z_1 - b_2).\end{aligned}$$

Let

$$w = W^\intercal W(z_2 - y).$$

The partials for A_i and b_i are

$$\begin{aligned}\frac{\partial J}{\partial \vec{A}_2} &= w^\intercal s'(A_2 z_1 - b_2)(\boldsymbol{I}_{m_2} \otimes z_1^\intercal),\\ \frac{\partial J}{\partial \vec{A}_1} &= w^\intercal s'(A_2 z_1 - b_2)A_2 s'(A_1 z_0 - b_1)(\boldsymbol{I}_{m_1} \otimes z_0^\intercal),\\ \frac{\partial J}{\partial \vec{A}_0} &= w^\intercal s'(A_2 z_1 - b_2)A_2 s'(A_1 z_0 - b_1)A_1 s'(A_0 x - b_0)(\boldsymbol{I}_{m_0} \otimes x^\intercal),\end{aligned}$$

$$\begin{aligned}\frac{\partial J}{\partial b_2} &= -w^\intercal s'(A_2 z_1 - b_2),\\ \frac{\partial J}{\partial b_1} &= -w^\intercal s'(A_2 z_1 - b_2)A_2 s'(A_1 z_0 - b_1),\\ \frac{\partial J}{\partial b_0} &= -w^\intercal s'(A_2 z_1 - b_2)A_2 s'(A_1 z_0 - b_1)A_1 s'(A_0 x - b_0).\end{aligned}$$

9.4 n–Layer Perceptron

In general, for $z_i \in \mathbf{R}^{m_i}$ with $i \in \mathbf{Z}[0, n-1]$, consider

$$\min_{\{A_i, b_i\} \in \{\mathbf{R}^{m_i \times m_{i-1}}, \mathbf{R}^{m_i}\}} \quad \sum_{\{x,y\}} = J(x, y),$$

where

$$J(x, y) = \frac{1}{2} \|z_{n-1} - y\|^2_{2, W(x,y)}.$$

For $i \in \mathbf{Z}[1, n-1]$,

$$z_i = s(A_i z_{i-1} - b_i),$$

with

$$z_0 = s(A_0 x - b_0).$$

Let

$$w = W^\intercal W (z_{n-1} - y).$$

The partials are given by

$$\begin{aligned}
\frac{\partial J}{\partial \vec{A}_i} &= w^\intercal \frac{\partial z_{n-1}}{\partial z_{n-2}} \frac{\partial z_{n-2}}{\partial z_{n-3}} \cdots \frac{\partial z_{i+1}}{\partial z_i} \frac{\partial z_i}{\partial \vec{A}_i}, \\
\frac{\partial J}{\partial b_i} &= w^\intercal \frac{\partial z_{n-1}}{\partial z_{n-2}} \frac{\partial z_{n-2}}{\partial z_{n-3}} \cdots \frac{\partial z_{i+1}}{\partial z_i} \frac{\partial z_i}{\partial b_i},
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial z_j}{\partial z_{j-1}} &= s'(A_j z_{j-1} - b_j) A_j, \\
\frac{\partial z_j}{\partial \vec{A}_j} &= s'(A_j z_{j-1} - b_j)(\boldsymbol{I}_{m_j} \otimes z_{j-1}^\intercal), \\
\frac{\partial z_j}{\partial b_j} &= -s'(A_j z_{j-1} - b_j).
\end{aligned}$$

9.5 n–Layer Perceptron with States

For $z_i \in \mathbf{R}^{m_i}$ with $i \in \mathbf{Z}[0, n-1]$, consider

$$\min_{\{z_i(x), A_i, b_i\} \in \{\mathbf{R}^{m_i}, \mathbf{R}^{m_i \times m_{i-1}}, \mathbf{R}^{m_i}\}} \quad \sum_{\{x,y\}} J(x, y) = \frac{1}{2} \|z_{n-1} - y\|^2_{2, W(x,y)},$$

with

$$f_{\text{eq}}(z, A, b | x) = \begin{bmatrix} z_0 - s(A_0 x - b_0) \\ z_1 - s(A_1 z_0 - b_1) \\ \vdots \\ z_{n-1} - s(A_{n-1} z_{n-2} - b_{n-1}) \end{bmatrix} = \mathbf{0}^M,$$

where

$$r = \sum_{i=0}^{n-1} m_i.$$

This formulation resembles MPC. The nonlinear equality constraint can be implamented with any of the methods presented in seciton 7.3. Adding states adds a degree of freedom in the optimization that might help in cases where the gradient would traditionally vanish. The disadvantage to this approach is, for q sample pair of $\{x, y\}$, there are rq new free variables to optimize.

9.6 ML: Maximum Likelihood

Consider the probability model

$$p = \text{nn}(x),$$

with

$$\begin{aligned} p &= \text{Probability}(\widetilde{y} = 1), \\ 1 - p &= \text{Probability}(\widetilde{y} = 0). \end{aligned}$$

Collect $x \in \{0,1\}^n$ and sort them by $y = 1$ and $y = 0$.
ML formulates the problem

$$\begin{aligned} \min_{\{A_i, b_i\} \in \{\mathbf{R}^{m_i \times m_{i-1}}, \mathbf{R}^{m_i}\}} \quad J &= -l \\ &= -\log\left(\prod_{x \text{ with } y=1} p(x) \prod_{x \text{ with } y=0} (1 - p(x))\right) \\ &= -\sum_{x \text{ with } y=1} \log\left(\text{nn}(x)\right) - \sum_{x \text{ with } y=0} \log\left(1 - \text{nn}(x)\right). \end{aligned}$$

With $\text{nn}(x) = z_{n-1}(x)$ and $z_{n-1} \in \mathbf{R}[0,1]$, the partials are given by

$$\frac{\partial J}{\partial \vec{A}_i} = \begin{cases} -\dfrac{1}{z_{n-1}} \dfrac{\partial z_{n-1}}{\partial z_{n-2}} \dfrac{\partial z_{n-2}}{\partial z_{n-3}} \cdots \dfrac{\partial z_{i+1}}{\partial z_i} \dfrac{\partial z_i}{\partial \vec{A}_i} & \text{if} \quad y = 1 \\ \dfrac{1}{1 - z_{n-1}} \dfrac{\partial z_{n-1}}{\partial z_{n-2}} \dfrac{\partial z_{n-2}}{\partial z_{n-3}} \cdots \dfrac{\partial z_{i+1}}{\partial z_i} \dfrac{\partial z_i}{\partial \vec{A}_i} & \text{if} \quad y = 0 \end{cases},$$

$$\frac{\partial J}{\partial b_i} = \begin{cases} -\dfrac{1}{z_{n-1}} \dfrac{\partial z_{n-1}}{\partial z_{n-2}} \dfrac{\partial z_{n-2}}{\partial z_{n-3}} \cdots \dfrac{\partial z_{i+1}}{\partial z_i} \dfrac{\partial z_i}{\partial b_i} & \text{if} \quad y = 1 \\ \dfrac{1}{1 - z_{n-1}} \dfrac{\partial z_{n-1}}{\partial z_{n-2}} \dfrac{\partial z_{n-2}}{\partial z_{n-3}} \cdots \dfrac{\partial z_{i+1}}{\partial z_i} \dfrac{\partial z_i}{\partial b_i} & \text{if} \quad y = 0 \end{cases},$$

where

$$\frac{\partial z_j}{\partial z_{j-1}} = s'(A_j z_{j-1} - b_j) A_j,$$

$$\frac{\partial z_j}{\partial \vec{A}_j} = s'(A_j z_{j-1} - b_j)(\boldsymbol{I}_{m_j} \otimes z_{j-1}^{\intercal}),$$

$$\frac{\partial z_j}{\partial b_j} = -s'(A_j z_{j-1} - b_j),$$

and

$$s(u) = (1 + e^{-u})^{-1}.$$

10 MPC: Model Predictive Control

Let

$$\begin{aligned} u_t &= \text{input}, \\ y_t &= \text{output}, \\ z_t &= \text{hidden internal dynamic state}, \\ \theta &= \text{hidden internal constant state}. \end{aligned}$$

MPC uses models of the form

$$\begin{aligned} z_{t+1} &= f(z_t, u_t|\theta) \\ y_t &= g(z_t, u_t|\theta) \end{aligned}$$

The models may be nonlinear or generalized deep neural networks, but if they can be expressed as linear, then **QP** can be used to solve them in one iteration. Time will be analyzed on a sliding window where $t = 0$ depicts the present (Figure 40). The world takes input and produces output over this moving time window (Figure 41). It is assumed that the world has some hidden internel dynamic states that need to be determined. The quantitative expression of these states depends on the model chosen to approximate the world. An estimator uses the model and the observed input and output history to produce a best current guess of the dynamic and constant hidden states (Figure 43). Once the current best guess is given for the states, the controller uses the model along with target input and output to find the optimal future input (Figure 44). These calculations are done each time step on hardware,

- estimating z_0^* and θ^* from $u[-T, 0)$ and $y[-T, 0]$,
- and computing and applying u_0^* from z_0^* and θ^*.

Lectures on MPC can be found here [15] and here [3]. There are many interesting applications in walking and flying robotics, including [18], [20], and [21]. Additional examples can be found at [24] and [26]. Python packages can be found here [44]. Pytorch extensions can be found here [45].

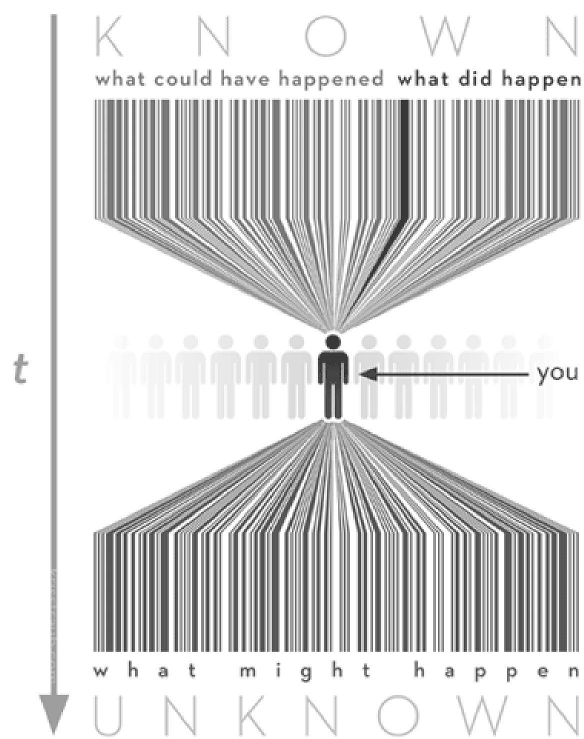


Figure 39: MPC

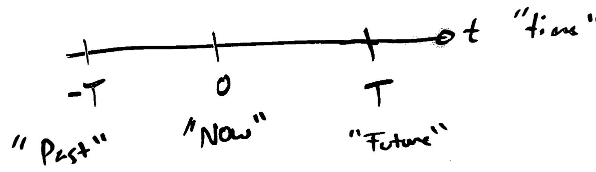


Figure 40: Time

Figure 41: World

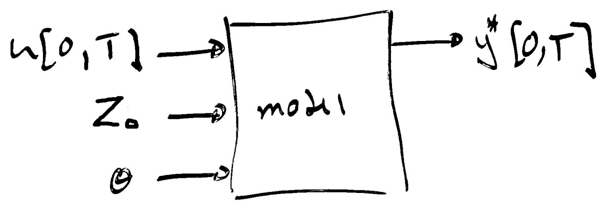


Figure 42: World Model

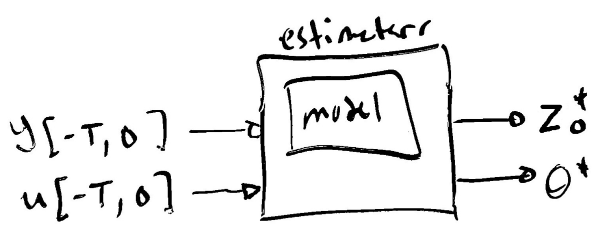


Figure 43: World Model Estimator

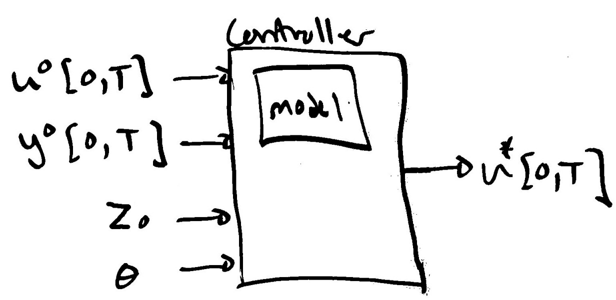


Figure 44: World Model Controller

10.1 LTV: Linear Time Variant

For $i \in \mathbf{Z}[0, q-1]$, consider the discrete-time system

$$\begin{aligned} z_{i+1} &= A_i z_i + B_i u_i, \\ z_0 &= \text{fixed}, \end{aligned}$$

where $u_i \in \mathbf{R}^m$ and $z_i \in \mathbf{R}^n$. For any selected time horizon q, this system can be expressed as

$$A_{\text{eq}} x = b_{\text{eq}},$$

where

$$x_i = \begin{bmatrix} u_i \\ z_{i+1} \end{bmatrix}.$$

For $q = 3$,

$$\begin{aligned} \mathbf{0}^n &= A_0 z_0 + B_0 u_0 - z_1, \\ \mathbf{0}^n &= A_1 z_1 + B_1 u_1 - z_2, \\ \mathbf{0}^n &= A_2 z_2 + B_2 u_2 - z_3, \end{aligned}$$

which gives

$$\left[\begin{array}{cc|cc|cc} B_0 & -\boldsymbol{I}_n & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_1 & B_1 & -\boldsymbol{I}_n & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A_2 & B_2 & -\boldsymbol{I}_n \end{array}\right] \left[\begin{array}{c} u_0 \\ z_1 \\ \hline u_1 \\ z_2 \\ \hline u_2 \\ z_3 \end{array}\right] = \begin{bmatrix} -A_0 z_0 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

For $q > 0$,

$$\begin{aligned} A_{\text{eq}} &= \begin{bmatrix} \mathbf{0}^{n \times (m+n)(q-1)} & \mathbf{0}^{n \times (m+n)} \\ \text{diag}\{\begin{bmatrix} \mathbf{0}^{n \times m} & A_i \end{bmatrix}\}_{i=1}^{q-1} & \mathbf{0}^{n(q-1) \times (m+n)} \end{bmatrix} + \text{diag}\{\begin{bmatrix} B_i & -\boldsymbol{I}_n \end{bmatrix}\}_{i=0}^{q-1}, \\ b_{\text{eq}} &= \begin{bmatrix} -A_0 z_0 \\ \mathbf{0}^{n(q-1)} \end{bmatrix}. \end{aligned}$$

10.1.1 Block Components

The u and z components can be accessed with

$$\begin{aligned} u_{0:q-1} &= Gx, \\ z_{1:q} &= Fx, \end{aligned}$$

where

$$\begin{aligned} G &= \text{diag}\left\{\begin{bmatrix} \boldsymbol{I}_m & \mathbf{0}^{m \times n} \end{bmatrix}\right\}_{i=0}^{q-1}, \\ F &= \text{diag}\left\{\begin{bmatrix} \mathbf{0}^{n \times m} & \boldsymbol{I}_n \end{bmatrix}\right\}_{i=0}^{q-1}. \end{aligned}$$

10.1.2 Rate Regularization

For weight $W_i \in \mathbf{R}^{m \times m}$, the input rate can be regularized by

$$\begin{aligned} \sum_{i=0}^{q-2} \|u_{i+1} - u_i\|_{W_i} &= \|u_{1:q-1} - u_{0:q-2}\|_W \\ &= \|u_{0:q-1}\|_{WD}, \end{aligned}$$

where [1, p. 312]

$$D = \text{diag}(\, \mathbf{0}^{m\times m},\, \boldsymbol{I}_{m(q-1)}\,) - \text{diag}(\, \boldsymbol{I}_{m(q-1)},\, \mathbf{0}^{m\times m}\,).$$

For weight $W_i \in \mathbf{R}^{n\times n}$, the state rate can be regularized by

$$\begin{aligned}\sum_{i=0}^{q-1} \|z_{i+1} - z_i\|_{W_i} &= \|z_{1:q} - z_{0:q-1}\|_W \\ &= \|Az_{1:q} - b\|_{WD},\end{aligned}$$

where

$$\begin{aligned}A &= \begin{bmatrix} \mathbf{0}^{n\times nq} \\ \boldsymbol{I}_{nq} \end{bmatrix}, \\ b &= \begin{bmatrix} -z_0 \\ \mathbf{0}^{nq} \end{bmatrix},\end{aligned}$$

and

$$D = \text{diag}(\, \mathbf{0}^{n\times n},\, \boldsymbol{I}_{nq}\,) - \text{diag}(\, \boldsymbol{I}_{nq},\, \mathbf{0}^{n\times n}\,).$$

Note: Rate can also be directly constrained with $d_{\text{lb}} \le Dx \le d_{\text{ub}}$.

10.1.3 2–Norm Objective

Consider the **QP**

$$\begin{aligned}\min_{x\in\mathbf{R}^{(m+n)q}} \quad J &= \sum_{i=0}^{q-1} \|u_i - u_i^\circ\|^2_{2,R_i^{1/2}} + \|z_{i+1} - z_{i+1}^\circ\|^2_{2,P_{i+1}^{1/2}} \\ &= \|Ax - b\|_2^2 \\ &\text{s.t.} \quad A_{\text{eq}}x = b_{\text{eq}},\end{aligned}$$

where

$$\begin{aligned}A &= \text{diag}\left\{ \begin{bmatrix} R_i^{1/2} & \mathbf{0}^{m\times n} \\ \mathbf{0}^{n\times m} & P_{i+1}^{1/2} \end{bmatrix} \right\}_{i=0}^{q-1}, \\ b &= \text{row}\left\{ \begin{bmatrix} R_i^{1/2} u_i^\circ \\ P_{i+1}^{1/2} z_{i+1}^\circ \end{bmatrix} \right\}_{i=0}^{q-1}.\end{aligned}$$

Expanding the objective gives

$$J = \frac{1}{2} x^\intercal Q x + c^\intercal x + r,$$

where

$$\begin{aligned}Q &= \quad 2\,\text{diag}\left\{ \begin{bmatrix} R_i & \mathbf{0}^{m\times n} \\ \mathbf{0}^{n\times m} & P_{i+1} \end{bmatrix} \right\}_{i=0}^{q-1}, \\ c &= -2\,\text{row}\left\{ \begin{bmatrix} R_i u_i^\circ \\ P_{i+1} z_{i+1}^\circ \end{bmatrix} \right\}_{i=0}^{q-1}.\end{aligned}$$

The diagonal structure has an effecient solution method using the Schur complement [1, p. 552].

10.1.4 Generalized Norm

The objective can be split into state and input with

$$\begin{aligned}\min_{x\in\mathbf{R}^{(m+n)q}} \quad J &= \|Gx - u^\circ\|_R + \|Fx - z^\circ\|_Q \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & A_{\text{eq}}x = b_{\text{eq}},\end{aligned}$$

where **P** adds additional constraints and $\|\cdot\|$ can be any norm that results in a **LP** or **QP** .

10.2 NTV: Nonlinear Time Variant

For $u_i \in \mathbf{R}^m$ and $z_i \in \mathbf{R}^n$, consider the problem

$$\begin{aligned} \min_{x \in \mathbf{R}^{(n+m)q}} \quad & J = \sum_{i=0}^{q-1} \left\| z_{i+1} - z_{i+1}^{\circ} \right\|_{P_{i+1}^{1/2}}^2 + \left\| u_i - u_i^{\circ} \right\|_{R_i^{1/2}}^2 \\ \text{s.t.} \quad & z_{i+1} = f(t_i, z_i, u_i), \\ & z_0 = \text{fixed}. \end{aligned}$$

For $i \in \mathbf{Z}[0, q-1]$, let $f_i := f(t_i, z_i, u_i)$ with

$$A_i = \frac{\partial f_i}{\partial z_i}, \quad B_i = \frac{\partial f_i}{\partial u_i},$$

$$x_i = \begin{bmatrix} u_i \\ z_{i+1} \end{bmatrix}.$$

The penalty approach gives

$$\min_{x \in \mathbf{R}^{(m+n)q}} \quad J = \|x - x^{\circ}\|_{2,W}^2 + \|f - Fx\|_{2,M}^2,$$

where

$$z_{1:q} = Fx.$$

The gradient is

$$\begin{aligned} g^{\mathsf{T}} &= \nabla J^{\mathsf{T}} \\ &= 2(x - x^{\circ})^{\mathsf{T}} W^{\mathsf{T}} W + 2(f - Fx)^{\mathsf{T}} M^{\mathsf{T}} M \left(h - F\right), \end{aligned}$$

where

$$\begin{aligned} h &= \frac{\partial f}{\partial x} \\ &= \left[\begin{array}{cc|cc|cc} B_0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_1 & B_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A_2 & B_2 & \mathbf{0} \end{array}\right] \\ &= \begin{bmatrix} \mathbf{0}^{n \times (m+n)(q-1)} & \mathbf{0}^{n \times (m+n)} \\ \operatorname{diag}\{\begin{bmatrix} \mathbf{0}^{n \times m} & A_i \end{bmatrix}\}_{i=1}^{q-1} & \mathbf{0}^{n(q-1) \times (m+n)} \end{bmatrix} + \operatorname{diag}\{\begin{bmatrix} B_i & \mathbf{0}^{n \times n} \end{bmatrix}\}_{i=0}^{q-1}. \end{aligned}$$

The Hessian is given by

$$\begin{aligned} H &= \frac{\partial g}{\partial x} \\ &= 2W^{\mathsf{T}} W + 2\left(h - F\right)^{\mathsf{T}} M^{\mathsf{T}} M \left(h - F\right) \\ &\quad + 2 \operatorname{col}\left\{ \left[\operatorname{col}\left\{ \frac{\partial h^{\mathsf{T}}}{\partial u_{ij}} \right\}_{j=0}^{m-1} \quad \operatorname{col}\left\{ \frac{\partial h^{\mathsf{T}}}{\partial z_{ij}} \right\}_{j=0}^{n-1} \right] \right\}_{i=0}^{q-1} M^{\mathsf{T}} M \left(f - Fx\right). \end{aligned}$$

10.3 LTI: Linear Time Invariant

If A and B are constant,

$$A_{\text{eq}} = \begin{bmatrix} \mathbf{0}^{n\times(m+n)(q-1)} & \mathbf{0}^{n\times(m+n)} \\ \boldsymbol{I}_{q-1} \otimes \begin{bmatrix} \mathbf{0}^{n\times m} & A \end{bmatrix} & \mathbf{0}^{n(q-1)\times(m+n)} \end{bmatrix} + \boldsymbol{I}_q \otimes \begin{bmatrix} B & -\boldsymbol{I}_n \end{bmatrix},$$

$$b_{\text{eq}} = \begin{bmatrix} -Az_0 \\ \mathbf{0}^{n(q-1)} \end{bmatrix}.$$

If P and R are constant (as well as the reference points), then

$$\begin{aligned} J &= \sum_{i=0}^{q-1} \|u_i - u^\circ\|_{R^{1/2}}^2 + \|z_{i+1} - z^\circ\|_{P^{1/2}}^2 \\ &= \frac{1}{2} x^\intercal Q x + c^\intercal x + r, \end{aligned}$$

where

$$\begin{aligned} Q &= 2\boldsymbol{I}_q \otimes \begin{bmatrix} R & \mathbf{0}^{m\times n} \\ \mathbf{0}^{n\times m} & P \end{bmatrix}, \\ c &= -2\mathbf{1}^q \otimes \begin{bmatrix} Ru^\circ \\ Pz^\circ \end{bmatrix}. \end{aligned}$$

If the bounds are constant with $z_{\text{lb}} \le z_i \le z_{\text{ub}}$ and $u_{\text{lb}} \le u_i \le u_{\text{ub}}$, then

$$\begin{aligned} \text{lb} &= \mathbf{1}^q \otimes \begin{bmatrix} u_{\text{lb}} \\ z_{\text{lb}} \end{bmatrix}, \\ \text{ub} &= \mathbf{1}^q \otimes \begin{bmatrix} u_{\text{ub}} \\ z_{\text{ub}} \end{bmatrix}. \end{aligned}$$

Figure 45 shows an example application.

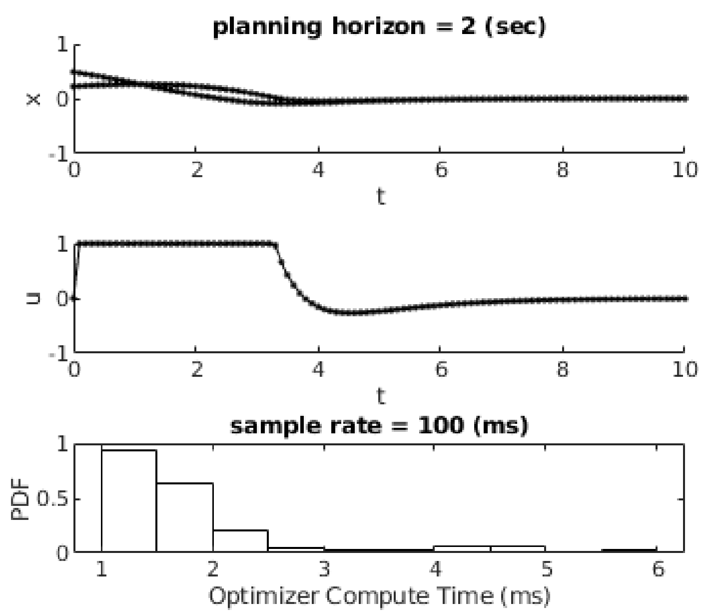


Figure 45: LTI MPC with unstable A and $|u| < 1$

10.4 System Identification

Linear prediction of output from past input and output gives

$$\begin{aligned} z_{i+1} &= Az_i + Bu_i \\ &= (\boldsymbol{I}_n \otimes z_i^{\mathsf{T}})\vec{A} + (\boldsymbol{I}_n \otimes u_i^{\mathsf{T}})\vec{B} \\ &= \begin{bmatrix} \boldsymbol{I}_n \otimes z_i^{\mathsf{T}} & \boldsymbol{I}_n \otimes u_i^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \vec{A} \\ \vec{B} \end{bmatrix}. \end{aligned}$$

For $i \in \mathbf{Z}[0, q-1]$, optimal paramaterization can be computed with

$$\min_{x \in \mathbf{R}^{n(n+m)}} \quad J = \|A_{\text{eq}}x - b_{\text{eq}}\|_2^2 + \lambda\|x\|_1,$$

where

$$\begin{aligned} A_{\text{eq}} &= \begin{bmatrix} \boldsymbol{I}_n \otimes z_0^{\mathsf{T}} & \boldsymbol{I}_n \otimes u_0^{\mathsf{T}} \\ \vdots & \vdots \\ \boldsymbol{I}_n \otimes z_{q-1}^{\mathsf{T}} & \boldsymbol{I}_n \otimes u_{q-1}^{\mathsf{T}} \end{bmatrix}, \\ b_{\text{eq}} &= \begin{bmatrix} z_1 \\ \vdots \\ z_q \end{bmatrix}, \\ x &= \begin{bmatrix} \vec{A} \\ \vec{B} \end{bmatrix}. \end{aligned}$$

Note: Previous estimates of x can be added to the objective function with

$$\min_{x \in \mathbf{R}^{n(n+m)}} \quad J = \|A_{\text{eq}}x - b_{\text{eq}}\|_2^2 + \lambda\|x\|_1 + \lambda_{\text{update}}\|x - x_{\text{previous}}\|_p.$$

If the model paramaters have known ranges, e.g., $\vec{A}_{\text{lb}} \leq \vec{A} \leq \vec{A}_{\text{ub}}$, these can be added with a $x \in \mathbf{P}$ constraint.

10.5 Non-Uniform Time Samples

The accuracy of any model prediction will degrade the farther it gets from the present. Including the prediction uncertainty into the cost weight typically aleviates this. The indexed time sample does not need to be uniform. Figure 46 shows a 3-layer sample. Alternatively, a continuous exponential time sample can be used (depicted in Figure 47). Only layerd samples are implamentable on hardware.

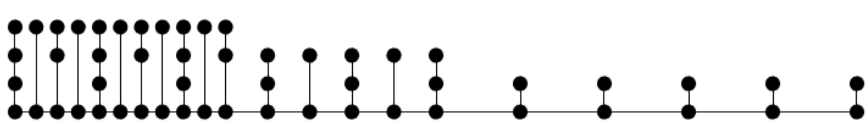


Figure 46: Layerd Time Sample

Figure 47: Exponential Time Sample

10.6 Continuous–Time Problems

Consider the continuous time problem

$$\min_{\{z,u\}:\mathbf{R}\to\{\mathbf{R}^n,\mathbf{R}^m\}} J = \int_{t_0}^{t_0+T} J(t,z,u)dt$$

$$\text{s.t.} \quad \begin{bmatrix} z(t) \\ u(t) \end{bmatrix} \in \mathbf{P}(t),$$
$$\dot{z} = f(t,z,u).$$

10.6.1 Fixed–Time Step

For a fixed $d_i > 0$ with $T = \sum_{i=0}^{q-1} d_i$, the problem can be re–expressed as

$$\min_{x\in\mathbf{R}^{(n+m)q}} \quad J(x) = \sum_{i=0}^{q-1} J(t_i, z_i, u_i)d_i$$

$$\begin{aligned} \text{s.t.} \quad x_i &\in \mathbf{P}_i, \\ x_i &= \begin{bmatrix} u_i \\ z_{i+1} \end{bmatrix}, \\ z_{i+1} &= z_i + f(t_i, z_i, u_i)d_i, \\ t_{i+1} &= t_i + d_i, \\ t_0 &= \text{fixed}, \\ z_0 &= \text{fixed}. \end{aligned}$$

10.6.2 Variable–Time Step

For a variable $d_i \geq 0$, the problem can be re–expressed as

$$\min_{x\in\mathbf{R}^{(n+m+1)q}} \quad J(x) = \sum_{i=0}^{q-1} J(t_i, z_i, u_i)d_i$$

$$\begin{aligned} \text{s.t.} \quad x_i &\in \mathbf{P}_i, \\ x_i &= \begin{bmatrix} d_i \\ u_i \\ z_{i+1} \end{bmatrix}, \\ z_{i+1} &= z_i + f(t_i, z_i, u_i)d_i, \\ t_{i+1} &= t_i + d_i, \\ t_0 &= \text{fixed}, \\ z_0 &= \text{fixed}, \\ t_{q-1} &= t_0 + T, \\ d_i &\geq 0. \end{aligned}$$

Note: For $L_i > 0$, an additional Lipshitz constraint can be added with

$$|z_{i+1} - z_i| \leq L_i d_i.$$

10.6.3 Chance-Constrained MPC

Random variables can be added to both the objective function and the constraints, e.g.,

$$z_{i+1} = z_i + (d_i + \widetilde{d}_i) f(z_i, u_i + \widetilde{u}_i, \widetilde{w}_i) + \widetilde{v}_i.$$

A probablistic formulation of MPC can include constraints that minimize the probability of failure, as defined by the probability of going outside of $x \in \mathbf{P}$. Interesting examples include: [16] and [17]. For $p \in \mathbf{R}(0, 1)$, the generalized problem formulation is

$$\begin{aligned}
\min_{\{z,u\} \in \{\mathbf{R}^{nq}, \mathbf{R}^{mq}\}} \quad & J = \langle J_q(z_q) + \sum_{i=0}^{q-1} J_i(z_i, u_i) \rangle \\
\text{s.t.} \quad & \text{Probability} \left(\bigwedge_{i=1}^{q} z_i \in \mathbf{D}_i \middle| z_0 \right) \geq 1 - p, \\
& z_{i+1} = f(z_i, u_i, \widetilde{w}_i),
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
\min_{\{z,u\} \in \{\mathbf{R}^{nq}, \mathbf{R}^{mq}\}} \quad & J = \langle J_q(z_q) + \sum_{i=0}^{q-1} J_i(z_i, u_i) \rangle \\
\text{s.t.} \quad & \left\langle \sum_{i=1}^{q} \text{Boolean} \left(z_i \notin \mathbf{D}_i \right) \middle| z_0 \right\rangle \leq p, \\
& z_{i+1} = f(z_i, u_i, \widetilde{w}_i),
\end{aligned}$$

10.7 PDE: Partial Differential Equation

PDE problems optimize free variables which are differentially constrained over time and space. This differes from MPC which has only time, resulting in only 1 index. Adding space results in $m > 1$ indicies. There are many potential applications including:

- eletro–magnetics,
- thermo–dynamics,
- surfaces optimization,
- classical mechanics,
- quantum mechanics.

Consider optimizing the f in $f(x)$, not the x. When discretized, x becomes the index. The first step is to remap the domain of x to the unit cube. Then, for $x_i \in \mathbf{R}[0, 1]^m$, $\Delta x_i = 1/n$. Integration of a function of $f(x)$ involves m summations over $\mathbf{Z}[0, n-1]$ divided by n^m. The partial across any of the x_i is simply n times the difference across that index. For irregular shapes within the unit cube, only formulate the problem for indices that occur within the shape.

10.7.1 Optimal Heat Diffusion

Consider $f : \mathbf{R}[0, 1]^2 \to \mathbf{R}$ with the Laplace diffusion problem

$$\begin{aligned}
\min \quad & J = \int_{x \in \mathbf{R}[0,1]^2} g(f(x)) dx \\
\text{s.t.} \quad & \frac{\partial f}{\partial x_1} + \frac{\partial f}{\partial x_2} = 0, \\
& \text{Boundary}(f) = \text{fixed}.
\end{aligned}$$

Let f_{ij} denote the value of f at $x = \begin{bmatrix} i/n \\ j/n \end{bmatrix}$ for $\begin{bmatrix} i \\ j \end{bmatrix} \in \mathbf{Z}[0, n-1]^2$.

$$\min_{f \in \mathbf{R}^{n \times n}} \quad J = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g(f_{ij})$$

$$\text{s.t.} \quad n(f_{i+1,j} - f_{ij}) + n(f_{i,j+1} - f_{ij}) = 0$$
$$f \in \mathbf{D}.$$

If $\mathbf{D} = \mathbf{P}$ and $g(\cdot)$ is linear or quadratic, this can be solved as an **LP** or **QP** .

10.7.2 Minimum Surface

Consider $f : \mathbf{R}[0,1]^2 \to \mathbf{R}$ with surface area [1, p. 159]

$$\begin{aligned} A &= \int_{\mathbf{R}[0,1]^2} \sqrt{1 + \|\nabla f(x)\|_2^2} dx \\ &= \int_{\mathbf{R}[0,1]^2} \left\| \begin{bmatrix} \nabla f(x) \\ 1 \end{bmatrix} \right\|_2 dx. \end{aligned}$$

The minimum surface problem is to find f that minimizes A subject to boundary constraints. Let f_{ij} denote the value of f at $x = \begin{bmatrix} i/n \\ j/n \end{bmatrix}$ for $\begin{bmatrix} i \\ j \end{bmatrix} \in \mathbf{Z}[0, n-1]^2$. An approximate expression for the gradient of f is

$$\nabla f(x) \approx n \begin{bmatrix} f_{i+1,j} - f_{i,j} \\ f_{i,j+1} - f_{i,j} \end{bmatrix},$$

which gives

$$A \approx \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \left\| \begin{bmatrix} n f_{i+1,j} - f_{i,j} \\ n f_{i,j+1} - f_{i,j} \\ 1 \end{bmatrix} \right\|_2.$$

The problem can then be stated as

$$\min_{f \in \mathbf{R}^{n \times n}} \quad J = A$$

$$\text{s.t.} \quad f \in \mathbf{D},$$

which, if $\mathbf{D} = \mathbf{P}$, can be recast as the **SOCP** if

$$\min_{\{f,t\} \in \{\mathbf{R}^{n \times n}, \mathbf{R}^{n \times n}\}} \quad J = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} t_{ij}$$

$$\text{s.t.} \quad f \in \mathbf{P},$$
$$\left\| \begin{bmatrix} n(f_{i+1,j} - f_{i,j}) \\ n(f_{i,j+1} - f_{i,j}) \\ 1 \end{bmatrix} \right\|_2 \leq t_{ij},$$

which can be converted to a **QCQP** by squaring the conic constraints.
Note: At this point, additional quadratic constraints can be added to **D**.

11 SVM: Support Vector Machines

For $x \in \mathbf{R}^n$, an SVM classifies with

$$\text{svm}(x) = \begin{cases} 1 & \text{if } a^\intercal x - b > 0 \\ -1 & \text{if } a^\intercal x - b < 0 \\ -1 + 2\widetilde{r} & \text{else} \end{cases} ,$$

where $\widetilde{r} \sim \text{Bernoulli}(1/2)$. Figure 48 illustrates this classification.

There are many ways to optimizate an SVM. The primary goals are to find **LP** and **QP** formulations. Secondary goals are to take advantage of dimension reduction through duality or sparsity. SVM references can be found at [22] and [23].

11.0.1 SVM as Single–Layer NN

An SVM can be approximated by a single–layer NN. Consider,

$$\text{svm}(x) \approx s\left(a^\intercal x - b\right),$$

where $s(\cdot)$ is a sigmoid function.

This approximation allows gradient calculation with respect to a and b.

If there are m class labels,

$$s\left(Ax - b\right) = \begin{bmatrix} s(a_0^\intercal x - b_0) \\ \vdots \\ s(a_{m-1}^\intercal x - b_{m-1}) \end{bmatrix}.$$

For $i \in \mathbf{Z}[0, q-1]$ samples $x_i \in \mathbf{R}^n$ and $y_i \in \{-1, 1\}$, the SVM can be optimized with

$$\begin{aligned} \min_{\{a,b\} \in \{\mathbf{R}^n, \mathbf{R}\}} J &= \textstyle\sum_{i=0}^{q-1} \|\text{svm}\left(a^\intercal x_i - b\right) - y_i\| \\ &\approx \textstyle\sum_{i=0}^{q-1} \|s\left(a^\intercal x_i - b\right) - y_i\|. \end{aligned}$$

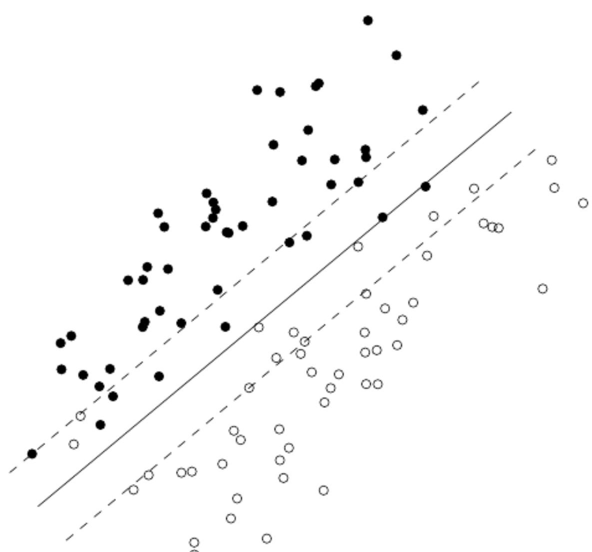


Figure 48: SVM

11.1 Primal Methods

Sort x_i with label $+1$ and y_j with label -1. For the following calculations,

$$x = \text{col}\{x_i\}_{i=0}^{m-1},$$
$$y = \text{col}\{y_j\}_{j=0}^{r-1}.$$

The support vector between $x_i \in \mathbf{R}^n$ and $y_j \in \mathbf{R}^n$ can be comptued with [1, p. 427]

$$\begin{aligned} \min_{\{a,b,u,v\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}^m,\mathbf{R}^r\}} \quad & J = \|a\| + \lambda(\mathbf{1}^\intercal u + \mathbf{1}^\intercal v) \\ \text{s.t.} \quad & a^\intercal x_i - b \geq 1 - u_i, \quad u_i \geq 0, \\ & a^\intercal y_j - b \leq -1 + v_j, \quad v_j \geq 0. \end{aligned}$$

11.1.1 1–Norm

If $p = 1$, this is the **LP**

$$\min_{\{a,b,u,v,w\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}^m,\mathbf{R}^r,\mathbf{R}^n\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ 0 \\ \lambda\mathbf{1}^m \\ \lambda\mathbf{1}^r \\ \mathbf{1}^n \end{bmatrix}^\intercal \begin{bmatrix} a \\ b \\ u \\ v \\ w \end{bmatrix}$$

$$\text{s.t.} \quad \begin{bmatrix} -x^\intercal & \mathbf{1}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} & \mathbf{0}^{m\times n} \\ y^\intercal & -\mathbf{1}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r & \mathbf{0}^{r\times n} \\ -\boldsymbol{I}_n & \mathbf{0}^n & \mathbf{0}^{n\times m} & \mathbf{0}^{n\times r} & -\boldsymbol{I}_n \\ \boldsymbol{I}_n & \mathbf{0}^n & \mathbf{0}^{n\times m} & \mathbf{0}^{n\times r} & -\boldsymbol{I}_n \\ \mathbf{0}^{m\times n} & \mathbf{0}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} & \mathbf{0}^{m\times n} \\ \mathbf{0}^{r\times n} & \mathbf{0}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r & \mathbf{0}^{r\times n} \end{bmatrix} \begin{bmatrix} a \\ b \\ u \\ v \\ w \end{bmatrix} \leq \begin{bmatrix} -\mathbf{1}^m \\ -\mathbf{1}^r \\ \mathbf{0}^n \\ \mathbf{0}^n \\ \mathbf{0}^m \\ \mathbf{0}^r \end{bmatrix}.$$

11.1.2 2–Norm

If $p = 2$ and the norm is squared, this is the **QP**

$$\min_{\{a,b,u,v\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}^m,\mathbf{R}^r\}} \quad J = \frac{1}{2}\begin{bmatrix} a \\ b \\ u \\ v \end{bmatrix}^\intercal \begin{bmatrix} 2\boldsymbol{I}_n & \mathbf{0}^n & \mathbf{0}^{n\times m} & \mathbf{0}^{n\times r} \\ \mathbf{0}^{1\times n} & 0 & \mathbf{0}^{1\times m} & \mathbf{0}^{1\times r} \\ \mathbf{0}^{m\times n} & \mathbf{0}^m & \mathbf{0}^{m\times m} & \mathbf{0}^{m\times r} \\ \mathbf{0}^{r\times n} & \mathbf{0}^r & \mathbf{0}^{r\times m} & \mathbf{0}^{r\times r} \end{bmatrix} \begin{bmatrix} a \\ b \\ u \\ v \end{bmatrix} + \begin{bmatrix} \mathbf{0}^n \\ 0 \\ \lambda\mathbf{1}^m \\ \lambda\mathbf{1}^r \end{bmatrix}^\intercal \begin{bmatrix} a \\ b \\ u \\ v \end{bmatrix}$$

$$\text{s.t.} \quad \begin{bmatrix} -x^\intercal & \mathbf{1}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} \\ y^\intercal & -\mathbf{1}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r \\ \mathbf{0}^{m\times n} & \mathbf{0}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} \\ \mathbf{0}^{r\times n} & \mathbf{0}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r \end{bmatrix} \begin{bmatrix} a \\ b \\ u \\ v \end{bmatrix} \leq \begin{bmatrix} -\mathbf{1}^m \\ -\mathbf{1}^r \\ \mathbf{0}^m \\ \mathbf{0}^r \end{bmatrix}.$$

11.1.3 Inf–Norm

If $p = \infty$, this is the **LP**

$$\min_{\{a,b,u,v,w\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}^m,\mathbf{R}^r,\mathbf{R}\}} \quad J = \begin{bmatrix} \mathbf{0}^n \\ 0 \\ \lambda\mathbf{1}^m \\ \lambda\mathbf{1}^r \\ 1 \end{bmatrix}^\intercal \begin{bmatrix} a \\ b \\ u \\ v \\ w \end{bmatrix}$$

$$\text{s.t.} \quad \begin{bmatrix} -x^\intercal & \mathbf{1}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} & \mathbf{0}^m \\ y^\intercal & -\mathbf{1}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r & \mathbf{0}^r \\ -\boldsymbol{I}_n & \mathbf{0}^n & \mathbf{0}^{n\times m} & \mathbf{0}^{n\times r} & -\mathbf{1}^n \\ \boldsymbol{I}_n & \mathbf{0}^n & \mathbf{0}^{n\times m} & \mathbf{0}^{n\times r} & -\mathbf{1}^n \\ \mathbf{0}^{m\times n} & \mathbf{0}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} & \mathbf{0}^m \\ \mathbf{0}^{r\times n} & \mathbf{0}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r & \mathbf{0}^r \end{bmatrix} \begin{bmatrix} a \\ b \\ u \\ v \\ w \end{bmatrix} \leq \begin{bmatrix} -\mathbf{1}^m \\ -\mathbf{1}^r \\ \mathbf{0}^n \\ \mathbf{0}^n \\ \mathbf{0}^m \\ \mathbf{0}^r \end{bmatrix}.$$

11.2 Dual Methods

There is a computational advantage to using the dual if the m samples are smaller than the n variables of x.

11.2.1 Dual

For $x_i \in \mathbf{R}^n$ and $y_i \in \{-1, +1\}$, if the data is linearly seperable, then

$$\begin{cases} a^\intercal x_i - b \leq y_i & \text{if} \quad y_i = -1 \\ a^\intercal x_i - b \geq y_i & \text{if} \quad y_i = +1 \end{cases}.$$

These constraint can be re–expressed as

$$y_i(a^\intercal x_i - b) \geq 1,$$

which can be written in matrix form as

$$\left[\begin{array}{cc} -\text{row}\{y_i x_i^\intercal\}_{i=0}^{m-1} & y \end{array}\right] \left[\begin{array}{c} a \\ b \end{array}\right] \leq -\mathbf{1}^m.$$

Consider the primal problem

$$\begin{aligned} \min_{\{a,b\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad & J = \frac{1}{2}\|a\|_2^2 \\ \text{s.t.} \quad & A_{\text{ub}} \left[\begin{array}{c} a \\ b \end{array}\right] \leq b_{\text{ub}}. \end{aligned}$$

The dual problem is

$$\begin{aligned} \max_{z \in \mathbf{R}^m} \quad \inf_{\{a,b\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad & L(a, b, z) \\ \text{s.t.} \quad & z \geq 0, \end{aligned}$$

where

$$L(a, b, z) = \frac{1}{2}\|a\|_2^2 + z^\intercal \left(A_{\text{ub}} \left[\begin{array}{c} a \\ b \end{array}\right] - b_{\text{ub}} \right).$$

Solving

$$\mathbf{0} = \left[\begin{array}{cc} \frac{\partial L}{\partial a} & \frac{\partial L}{\partial b} \end{array}\right] = \left[\begin{array}{cc} a^\intercal - z^\intercal \text{row}\{y_i x_i^\intercal\}_{i=0}^{m-1} & z^\intercal y \end{array}\right]$$

gives

$$\begin{aligned} a &= \text{col}\{x_i y_i\}_{i=0}^{m-1} z, \\ 0 &= y^\intercal z, \end{aligned}$$

with equality constraint on z.

The dual problem is [34, p. 313]

$$\begin{aligned} \min_{z \in \mathbf{R}^m} \quad & \frac{1}{2} z^\intercal Q_{\text{dual}}\, z + c_{\text{dual}}^\intercal\, z \\ \text{s.t.} \quad & z \geq \mathbf{0}^m, \\ & y^\intercal z = 0, \end{aligned}$$

where

$$\begin{aligned} [Q_{\text{dual}}]_{ij} &= y_i y_j x_i^\intercal x_j \\ c_{\text{dual}} &= -\mathbf{1}^m. \end{aligned}$$

To recover the primal solution,

$$a^\intercal x = \sum_{i=0}^{m-1} z_i^* y_i x_i^\intercal x.$$

For each $z_i > 0$, x_i is on the margin. Solve b_i with $b_i = y_i - x_i^\intercal a$. Then compute $b = \text{mean}(b_i)$.

11.2.2 Soft Dual

For $c_i \geq 0$, the constraints can be relaxed with

$$y_i(a^\intercal x_i - b) \geq 1 - c_i,$$

where

$$x_i = \begin{cases} \text{correctly classified outside of the margin} & \text{if} \quad c_i = 0 \\ \text{correctly classified inside the margin} & \text{if} \quad c_i \in \mathbf{R}(0,1) \\ \text{missclassified} & \text{if} \quad c_i \geq 1 \end{cases}.$$

Consider the primal problem

$$\begin{aligned} \min_{\{a,b,c\} \in \{\mathbf{R}^n, \mathbf{R}, \mathbf{R}^m\}} \quad & J = \frac{1}{2}\|a\|_2^2 + \lambda \mathbf{1}^\intercal c \\ \text{s.t.} \quad & A_{\text{ub}} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \leq b_{\text{ub}}, \end{aligned}$$

where

$$\begin{aligned} A_{\text{ub}} &= \begin{bmatrix} -\text{row}\{y_i x_i^\intercal\}_{i=0}^{m-1} & y & -\boldsymbol{I}_m \\ \mathbf{0}^{m \times n} & \mathbf{0}^{m \times 1} & -\boldsymbol{I}_m \end{bmatrix}, \\ b_{\text{ub}} &= \begin{bmatrix} -\mathbf{1}^m \\ \mathbf{0}^m \end{bmatrix}. \end{aligned}$$

The dual problem is

$$\begin{aligned} \max_{\{z,w\} \in \{\mathbf{R}^m, \mathbf{R}^m\}} \quad & \inf_{\{a,b\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad L(a,b,z,w) \\ \text{s.t.} \quad & \begin{bmatrix} z \\ w \end{bmatrix} \geq \mathbf{0}^{2m}, \end{aligned}$$

where

$$L(a,b,z) = \frac{1}{2}\|a\|_2^2 + \lambda \mathbf{1}^\intercal c + \begin{bmatrix} z \\ w \end{bmatrix}^\intercal \left(A_{\text{ub}} \begin{bmatrix} a \\ b \\ c \end{bmatrix} - b_{\text{ub}} \right).$$

Solving

$$\mathbf{0} = \begin{bmatrix} \frac{\partial L}{\partial a} & \frac{\partial L}{\partial b} & \frac{\partial L}{\partial c} \end{bmatrix} = \begin{bmatrix} a^\intercal - z^\intercal \text{row}\{y_i x_i^\intercal\}_{i=0}^{m-1} & z^\intercal y & \lambda \mathbf{1}^m - z - w \end{bmatrix}$$

gives

$$\begin{aligned} a &= \text{col}\{x_i y_i\}_{i=0}^{m-1} z, \\ 0 &= y^\intercal z, \\ w &= \lambda \mathbf{1}^m - z. \end{aligned}$$

The dual problem is [34, p. 316]

$$\begin{aligned} \min_{z \in \mathbf{R}^m} \quad & \frac{1}{2} z^\intercal Q_{\text{dual}}\, z + c_{\text{dual}}^\intercal\, z \\ \text{s.t.} \quad & \mathbf{0}^m \leq z \leq \lambda \mathbf{1}^m \\ & y^\intercal z = 0, \end{aligned}$$

where

$$\begin{aligned} [Q_{\text{dual}}]_{ij} &= y_i y_j x_i^\intercal x_j, \\ c_{\text{dual}} &= -\mathbf{1}^m. \end{aligned}$$

To recover the primal solution,

$$a^\intercal x = \sum_{i=0}^{m-1} z_i^* y_i x_i^\intercal x.$$

For each $z_i \in \mathbf{R}(0,\lambda)$, x_i is on the margin. Solve b_i with $b_i = y_i - x_i^\intercal a^*$. Then compute $b^* = \text{mean}(b_i)$.

11.2.3 Soft Dual (Alternative)

For $\lambda \in \mathbf{R}[0,1]$, consider the primal problem

$$\min_{\{a,b,c,d\}\in\{\mathbf{R}^n,\mathbf{R},\mathbf{R}^m,\mathbf{R}\}} \quad J = \frac{1}{2}\|a\|_2^2 + \frac{1}{m}\mathbf{1}^\intercal c - \lambda d$$
$$\text{s.t.} \quad y_i(a^\intercal x_i - b) \geq d - c_i, \quad c \geq \mathbf{0}^m, \quad d \geq 0.$$

The margin is now given by $2d/\|a\|_2$.

The dual problem is [34, p. 319]

$$\min_{z\in\mathbf{R}^m} \quad \frac{1}{2} z^\intercal Q_{\text{dual}}\, z$$
$$\text{s.t.} \quad \mathbf{0}^m \leq z \leq \frac{1}{m}\mathbf{1}^m, \quad \mathbf{1}^\intercal z \leq \lambda, \quad y^\intercal z = 0,$$

where

$$[Q_{\text{dual}}]_{ij} = y_i y_j x_i^\intercal x_j.$$

To recover the primal solution,

$$a^\intercal x = \sum_{i=0}^{m-1} z_i^* y_i x_i^\intercal x.$$

11.2.4 Kernels

The kernel trick comes from noting that both

$$[Q_{\text{dual}}]_{ij} = y_i y_j K(x_i, x_j)$$

and

$$\begin{aligned} a^\intercal x &= \textstyle\sum_{i=0}^{m-1} z_i^* y_i x_i^\intercal x \\ &= \textstyle\sum_{i=0}^{m-1} z_i^* y_i K(x_i, x). \end{aligned}$$

contain inner products.

If $x = f(u)$ is a mapping from a low dimensional space to a very large feature space, then replacing $x_i^\intercal x_j = f(u_i)^\intercal f(u_j)$ with a kernel $K(u_i, u_j)$ gives [34, p. 321]

$$[Q_{\text{dual}}]_{ij} = y_i y_j K(u_i, u_j)$$

and

$$\begin{aligned} a^\intercal x &= \textstyle\sum_{i=0}^{m-1} z_i^* y_i f(u_i)^\intercal f(u) \\ &= \textstyle\sum_{i=0}^{m-1} z_i^* y_i K(u_i, u). \end{aligned}$$

This can significantly reduce computation without imposing restrictive limits on n. In some cases, a feature vector may be near infinite or not even known, but the inner product between $f(u_i)$ and $f(u_j)$ may have a known form or at least a good heuristic.

Some popular kernels include:

- polynomials of degree q, computed with $(u_i^\intercal u_j + 1)^q$,
- sigmoid functions computed from $u_i^\intercal u_j$,
- radial basis functions computed from dist(u_i, u_j).

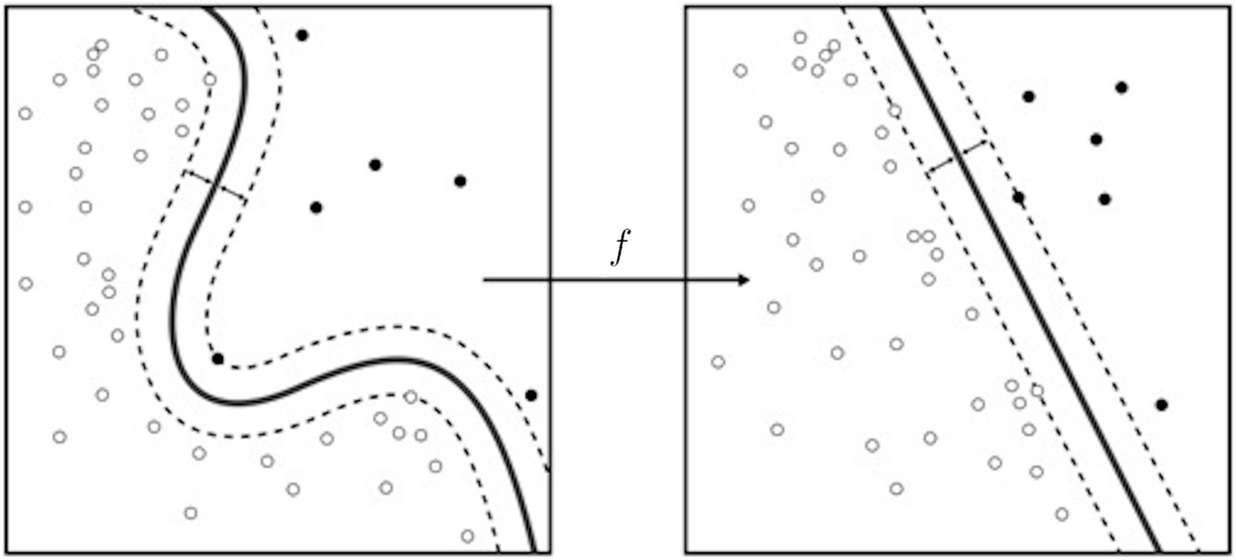


Figure 49: SVM with the kernel trick

11.2.5 Recovering Bias

Sort x_i with label $+1$ and y_j with label -1. For the following calculations,

$$\begin{aligned} x &= \text{col}\{x_i\}_{i=0}^{m-1}, \\ y &= \text{col}\{y_j\}_{j=0}^{r-1}. \end{aligned}$$

Given a^* from the dual problem, the support vector between $x_i \in \mathbf{R}^n$ and $y_j \in \mathbf{R}^n$ can be comptued with

$$\min_{\{b,u,v\}\in\{\mathbf{R},\mathbf{R}^m,\mathbf{R}^r\}} \quad J = \mathbf{1}^\intercal u + \mathbf{1}^\intercal v$$

$$\begin{aligned} \text{s.t.} \quad x_i^\intercal a^* - b &\geq 1 - u_i, \quad u_i \geq 0, \\ y_j^\intercal a^* - b &\leq -1 + v_j, \quad v_j \geq 0. \end{aligned}$$

These constraints can be re–expressed as

$$\begin{aligned} b - u_i &\leq -1 + x_i^\intercal a^*, \quad -u_i \leq 0, \\ -b - v_j &\leq -1 - y_j^\intercal a^*, \quad -v_j \leq 0. \end{aligned}$$

The bias can be computed with the primary **LP**

$$\min_{\{b,u,v\}\in\{\mathbf{R},\mathbf{R}^m,\mathbf{R}^r\}} \quad J = \begin{bmatrix} 0 \\ \mathbf{1}^m \\ \mathbf{1}^r \end{bmatrix}^\intercal \begin{bmatrix} b \\ u \\ v \end{bmatrix}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{1}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} \\ -\mathbf{1}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r \\ \mathbf{0}^m & -\boldsymbol{I}_m & \mathbf{0}^{m\times r} \\ \mathbf{0}^r & \mathbf{0}^{r\times m} & -\boldsymbol{I}_r \end{bmatrix} \begin{bmatrix} b \\ u \\ v \end{bmatrix} \leq \begin{bmatrix} -\mathbf{1}^m \\ -\mathbf{1}^r \\ \mathbf{0}^m \\ \mathbf{0}^r \end{bmatrix} + \begin{bmatrix} \text{row}\{x_i^\intercal a^*\}_{i=0}^{m-1} \\ -\text{row}\{y_j^\intercal a^*\}_{j=0}^{r-1} \\ \mathbf{0}^m \\ \mathbf{0}^r \end{bmatrix}$$

12 Miscellaneous Applications

12.1 Minimum Complexity Modeling

For samples $i \in \mathbf{Z}[0, q-1]$, find a linear map from $x_i \in \mathbf{R}^n$ to $y_i \in \mathbf{R}^m$ with minimum non–zero paramaters. This objective can be expressed as

$$\min_{A \in \mathbf{R}^{m \times n}} \quad J = \lambda \|\vec{A}\|_1 + \textstyle\sum_{i=0}^{q-1} \|Ax_i - y_i\|_{2,W_i}^2.$$

Vectorizing from left to right and top to bottom and using the Kronecker product, the objective function becomes

$$\begin{aligned} J &= \lambda \|\vec{A}\|_1 + \textstyle\sum_{i=0}^{q-1} \|(\boldsymbol{I}_m \otimes x_i^\intercal)\vec{A} - y_i\|_{2,W_i}^2 \\ &= \frac{1}{2}\vec{A}^\intercal Q \vec{A} + c^\intercal \vec{A} + r + \lambda \|\vec{A}\|_1 + \textstyle\sum_{i=0}^{q-1} y_i^\intercal y_i, \end{aligned}$$

where

$$\begin{aligned} Q &= 2\textstyle\sum_{i=0}^{q-1} (\boldsymbol{I}_m \otimes x_i) W_i^\intercal W_i (\boldsymbol{I}_m \otimes x_i^\intercal) \\ &= 2\textstyle\sum_{i=0}^{q-1} (\boldsymbol{I}_m \otimes x_i x_i^\intercal) && \text{if} \quad W_i = \boldsymbol{I}, \\ c &= -2\textstyle\sum_{i=0}^{q-1} (\boldsymbol{I}_m \otimes x_i) W_i^\intercal W_i y_i \\ &= -2\textstyle\sum_{i=0}^{q-1} (\boldsymbol{I}_m \otimes x_i) y_i && \text{if} \quad W_i = \boldsymbol{I}. \end{aligned}$$

Note: Additional convex paramater constraints such as non–negativity can easily be added with $\vec{A} \in \mathbf{P}$.

12.2 Linear Fractional

Consider

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \frac{a^\intercal x - b}{c^\intercal x - d} \\ \text{s.t.} \quad & x \in \mathbf{P}, \\ & c^\intercal x - d > 0. \end{aligned}$$

The problem can be stated as the **LP** [1, p. 151]

$$\begin{aligned} \min_{\{y,z\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad & J = a^\intercal y - bz \\ \text{s.t.} \quad & A_{\text{ub}} y - b_{\text{ub}} z \leq \mathbf{0}, \\ & A_{\text{eq}} y - b_{\text{eq}} z = \mathbf{0}, \\ & c^\intercal y - dz = 1, \\ & z \geq 0, \end{aligned}$$

which can be expressed as

$$\begin{aligned} \min_{\{y,z\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad & J = \begin{bmatrix} a \\ b \end{bmatrix}^\intercal \begin{bmatrix} y \\ z \end{bmatrix} \\ \text{s.t.} \quad & \begin{bmatrix} A_{\text{ub}} & -b_{\text{ub}} \\ \mathbf{0}^{1 \times n} & -1 \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} \leq \mathbf{0}^{m+1}, \\ & \begin{bmatrix} A_{\text{eq}} & -b_{\text{eq}} \\ c^\intercal & -d \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{0}^r \\ 1 \end{bmatrix}, \end{aligned}$$

where $x = y/z$. The solution exists if $c^\intercal x > d$.
Note: These problem fomulations are common in projective geometry.

12.3 Sampled Convex Sets

For a convex set $\mathbf{D} \subset \mathbf{R}^n$, consider

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \|Ax - b\|_p \\ \text{s.t.} \quad & x \in \mathbf{D}. \end{aligned}$$

The constraint $x \in \mathbf{D}$ can be approximated with m sample points $s_i \in \mathbf{D}$ using the convex combination [1, p. 24]

$$x = \textstyle\sum_{i=0}^{m-1} s_i y_i, \quad 0 \leq y_i \leq 1, \quad \textstyle\sum_{i=0}^{m-1} y_i = 1, \quad y_i \in \mathbf{R}.$$

The problem can then be expressed as

$$\begin{aligned} \min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad & J = \|Ax - b\|_p \\ \text{s.t.} \quad & A_{\text{eq}} \begin{bmatrix} x \\ y \end{bmatrix} = b_{\text{eq}}, \\ & A_{\text{ub}} \begin{bmatrix} x \\ y \end{bmatrix} \leq b_{\text{ub}}, \end{aligned}$$

where

$$\begin{aligned} y &= \text{row}\,\{y_i\}_{i=0}^{m-1} \\ A_{\text{eq}} &= \begin{bmatrix} -\boldsymbol{I}_n & s \\ \mathbf{0}^{1\times n} & \mathbf{1}^{1\times m} \end{bmatrix}, \\ b_{\text{eq}} &= \begin{bmatrix} \mathbf{0}^n \\ 1 \end{bmatrix}, \\ A_{\text{ub}} &= \begin{bmatrix} \mathbf{0}^{m\times n} & -\boldsymbol{I}_m \\ \mathbf{0}^{m\times n} & \boldsymbol{I}_m \end{bmatrix}, \\ b_{\text{ub}} &= \begin{bmatrix} \mathbf{0}^m \\ \mathbf{1}^m \end{bmatrix}. \end{aligned}$$

Note: Eliminate interior sample points before optimizing to reduce computation.

12.4 Interset Distance

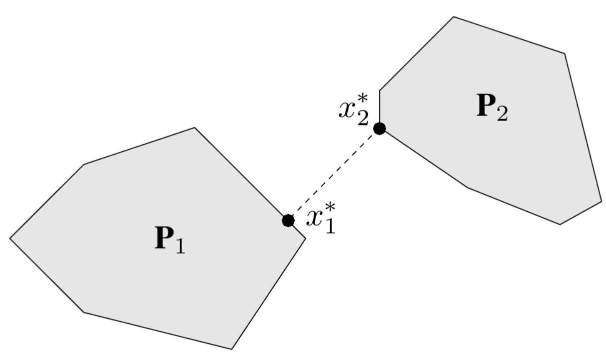


Figure 50: minimum distance between disjoint polyhedron sets

The problem

$$\begin{aligned} \min_{\{x_1,x_2\} \in \{\mathbf{R}^n, \mathbf{R}^n\}} \quad & J = \|x_1 - x_2\|_p \\ \text{s.t.} \quad & x_1 \in \mathbf{P}_1, \\ & x_2 \in \mathbf{P}_2, \end{aligned}$$

is an **LP** for $p \in \{1, \infty\}$ and a **QP** for $p = 2$.

12.5 SPINV: Sparse Pseudo Inverse

For $A \in \mathbf{R}^{m\times n}$ with $m > n$ and $\operatorname{rank}(A) = n$, a sparse representation of $B = A^+$ can be computed using [7]

$$\begin{aligned}\min_{B\in\mathbf{R}^{n\times m}} \quad J &= \|BA - \boldsymbol{I}_n\|_2^2 + \lambda\|\vec{B}\|_1 \\ &= \|(\boldsymbol{I}_n \otimes A^\intercal)\vec{B} - \vec{\boldsymbol{I}}_n\|_2^2 + \lambda\|\vec{B}\|_1.\end{aligned}$$

Note: If $A \in \mathbf{S}^n_+$, then $B = A^{-1} \in \mathbf{S}^n_+$, where the symmetry can be enforced with the constraint $B = B^\intercal$.

12.6 k–means

Given m sampls of $s_j \in \mathbf{D} \subset \mathbf{R}^n$, k cluster centers can be found using Algorithm 11 [34, p. 149]. See [54, p. 195] for its derivation from the EM algorithm. This is a non-convex optimization, and many potential cluster may exist. Convergence to a particular cluster depends on the initialization of the algorithm. Many different norms may be considered. Different norms create different cell geometries. A 2–norm creates convex polyhedron cells (see Figure 51b), but a 1–norm creates non-convex cells (see Figure 51a).

For $j \in \mathbf{Z}[0, m-1]$, sample $s_i \in \mathbf{D}$
For $i \in \mathbf{Z}[0, k-1]$, initialize means $x_i \in \mathbf{D}$
while x not converged **do**
  for $j \in \mathbf{Z}[0, m-1]$ **do**

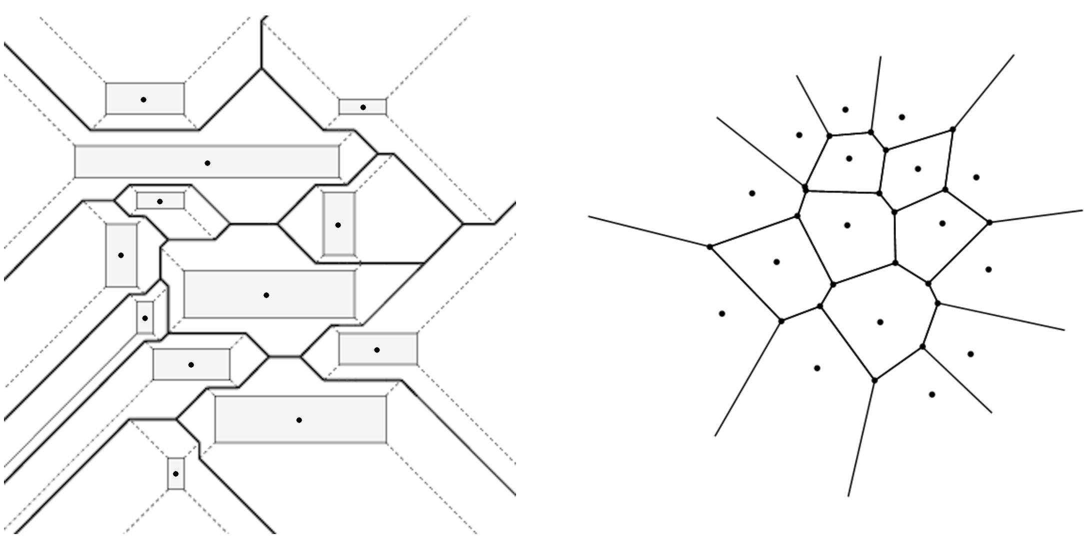
$$b_{ij} \leftarrow \begin{cases} 1 & \text{if} \quad \|s_j - x_i\| = \min_q \|s_j - x_q\| \\ 0 & \text{else} \end{cases}$$

  end
  for $i \in \mathbf{Z}[0, k-1]$ **do**

$$x_i \leftarrow \frac{\sum_{j=0}^{m-1} b_{ij} s_j}{\sum_{j=0}^{m-1} b_{ij}}$$

  end
end

Algorithm 11: k–means



(a) 1–Norm Voronoi (b) 2–Norm Voronoi

Figure 51: Voronoi Cells

12.7 2–Norm Voronoi

A polyhedron can be expressed as [1, p. 60]

$$\begin{aligned}\mathbf{P}_0 &= \{x \in \mathbf{R}^n | \|x - x_0\|_2 \leq \|x - x_i\|_2, i \in \mathbf{Z}[1, m-1]\} \\ &= \{x \in \mathbf{R}^n | Ax \leq b\}.\end{aligned}$$

Squaring the constraints gives

$$\|x - x_0\|_2^2 \leq \|x - x_i\|_2^2.$$

Expanding the constraints gives

$$x^\intercal x - 2x_0^\intercal x + x_0^\intercal x_0 \leq x^\intercal x - 2x_i^\intercal x + x_i^\intercal x_i.$$

Collecting terms gives

$$2(x_i - x_0)^\intercal x \leq x_i^\intercal x_i - x_0^\intercal x_0,$$

which gives the relationship

$$\begin{aligned}a_i &= 2(x_i - x_0), \\ b_i &= x_i^\intercal x_i - x_0^\intercal x_0.\end{aligned}$$

Therefore,

$$\begin{aligned}x_i &= \frac{1}{2}a_i + x_0, \\ b_i &= \left(\frac{1}{2}a_i + x_0\right)^\intercal \left(\frac{1}{2}a_i + x_0\right) - x_0^\intercal x_0 \\ &= \frac{1}{4}a_i^\intercal a_i + a_i^\intercal x_0.\end{aligned}$$

Let

$$\begin{aligned}A &= \text{row}\{a_i^\intercal\}_{i=1}^{m-1}, \\ b &= \text{row}\{b_i\}_{i=1}^{m-1}, \\ c &= \frac{1}{4}\text{row}\{a_i^\intercal a_i\}_{i=1}^{m-1}.\end{aligned}$$

Then,

$$b = c + Ax_0.$$

For the SVD $A = U_+ S_+ V_+^\intercal$,

$$x_0 = A^+(b - c) + U_0 d,$$

where $d \in \mathbf{R}^{n-r}$ can be freely choosen.

Mutually exclusive polyhedral sets can be computed with

$$\begin{aligned}\mathbf{P}_j &= \{x \in \mathbf{R}^n | \|x - x_j\|_2 \leq \|x - x_i\|_2, i \neq j\} \\ &= \{x \in \mathbf{R}^n | A_j x \leq b_j\}.\end{aligned}$$

If $\text{rank}(A_j) = n$ for all j,

$$\mathbf{R}^n = \bigcup_{j=0}^{m-1} \mathbf{P}_j,$$

and, for $i \neq j$,

$$\text{Interior}(\mathbf{P}_i) \cap \text{Interior}(\mathbf{P}_j) = \emptyset.$$

Note: When only two points are present,

$$\begin{aligned}\|x-x_0\|_2^2 \leq \|x-x_1\|_2^2 \quad &\Leftrightarrow \quad (x-x_0)^\intercal(x-x_0) \leq (x-x_1)^\intercal(x-x_1)\\ &\Leftrightarrow \quad x^\intercal x - 2x_0^\intercal x + x_0^\intercal x_0 \leq x^\intercal x - 2x_1^\intercal x + x_1^\intercal x_1\\ &\Leftrightarrow \quad 2(x_1-x_0)^\intercal x \leq x_1^\intercal x_1 - x_0^\intercal x_0,\end{aligned}$$

defines a seperating hyperplane $\{x \in \mathbf{R}^n | a^\intercal x \leq b\}$, with $a = 2(x_1 - x_0)$ and $b = x_1^\intercal x_1 - x_0^\intercal x_0$.

12.8 Noisy Unkown Nonlinear Maps

For $i \in \mathbf{Z}[0, m-1]$, consider

$$y_i = f(a_i^\intercal x - b_i + \widetilde{w}_i),$$

where $x \in \mathbf{R}^n$ is a vector to be estimated, $\{y_i, a_i, b_i\} \in \{\mathbf{R}, \mathbf{R}^n, \mathbf{R}\}$ are measured, and $\widetilde{w}_i$ is IID. The function $f : \mathbf{R} \to \mathbf{R}$ is unkown, but it is known that $f' \in \mathbf{R}[c, d]$ with $0 < c < d$. Solving for $\widetilde{w}_i$,

$$\widetilde{w}_i = f^{-1}(y_i) - a_i^\intercal x + b_i^\intercal.$$

The probability of observing a sequence of y_i is then

$$\prod_{i=0}^{m-1} p(f^{-1}(y_i) - a_i^\intercal x + b_i^\intercal).$$

The objective is then to minimize the negative log–likelihood

$$\min_{\{x,z_i\}\in\{\mathbf{R}^n,\mathbf{R}\}} \quad J = -\sum_{i=0}^{m-1} \log p(z_i - a_i^\intercal x + b_i^\intercal).$$

$$\text{s.t.} \quad z_i = f^{-1}(y_i).$$

The constraints can be expressed in terms of the inverse with

$$\frac{\partial}{\partial y_i} f^{-1}(y_i) \in \mathbf{R}[d^{-1}, c^{-1}],$$

which, for $i \in \mathbf{Z}[0, m-1]$ and $j \in \mathbf{Z}[0, m-1]$, gives the constraint

$$\frac{|y_i - y_j|}{d} \leq |z_i - z_j| \leq \frac{|y_i - y_j|}{c}.$$

This result is easily extended to nested nonlinear models of the form

$$\begin{aligned}x_{j+1} &= f(a_{ij}^\intercal x_j - b_{ij} + \widetilde{w}_{ij}),\\ x_{m-1} &= y_i.\end{aligned}$$

12.9 Reconstruction

Consider the difference vector

$$\begin{bmatrix} x_1 - x_0 \\ x_2 - x_1 \\ \vdots \\ x_{n-2} - x_{n-3} \\ x_{n-1} - x_{n-2} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \end{bmatrix}$$

$$= \underbrace{\left(\begin{bmatrix} \mathbf{0}^{n-1} & \boldsymbol{I}_{n-1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{I}_{n-1} & \mathbf{0}^{n-1} \end{bmatrix} \right)}_{D} x,$$

where $D \in \mathbf{R}^{n-1 \times n}$. Reconstruction using derivative penalty is computed with

$$\min_{x \in \mathbf{R}^n} \quad J = \|x - x_{\text{data}}\| + \lambda \|Dx\|.$$

12.9.1 Quadratic Smoothing

Minimum quadratic smoothing is computed with

$$\min_{x \in \mathbf{R}^n} \quad J = \|x - x_{\text{data}}\|_2^2 + \lambda \|Dx\|_2^2,$$

which has the closed form solution

$$x = (\boldsymbol{I}_n + \lambda D^{\mathsf{T}} D)^{-1} x_{\text{data}}.$$

12.9.2 Total Varation

Minimum total variation is computed with

$$\min_{x \in \mathbf{R}^n} \quad J = \|x - x_{\text{data}}\|_2^2 + \lambda \|Dx\|_1.$$

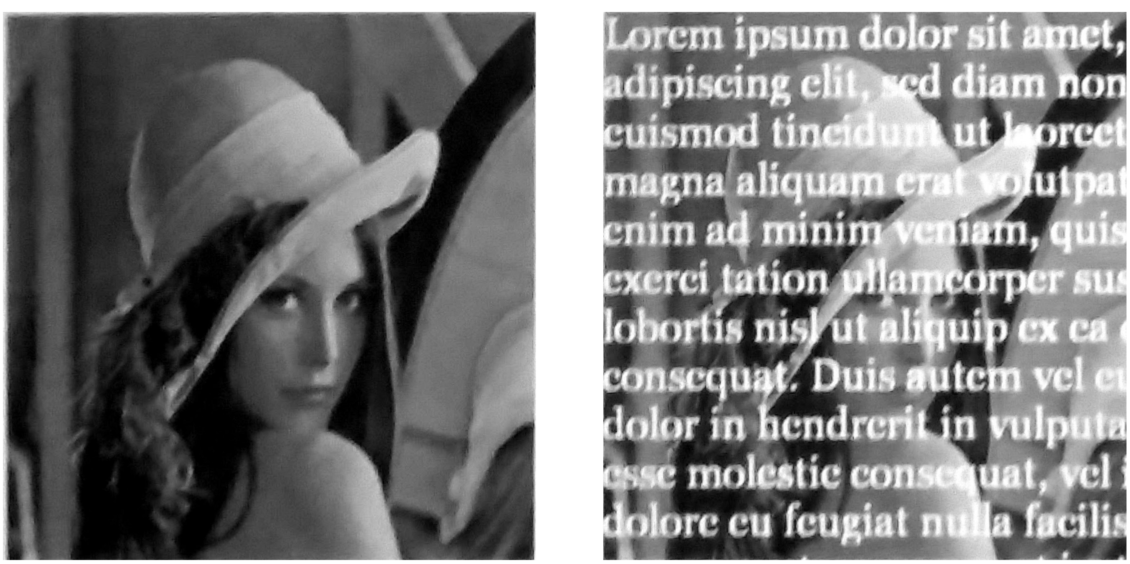


Figure 52: Total varation reconstruction on left from corrupted image on right.

13 Simplifications

13.1 Affine Transforms

Consider $x = Ay - b$. The constraint $x \in \mathbf{P}$ implies

$$Ay - b \in \mathbf{P},$$

which gives

$$\begin{aligned} A_{\text{ub}}(Ay - b) &\leq b_{\text{ub}}, \\ A_{\text{eq}}(Ay - b) &= b_{\text{eq}}, \\ x_{\text{lb}} \leq (Ay - b) &\leq x_{\text{ub}}. \end{aligned}$$

This can be equivalently stated as

$$\begin{bmatrix} A_{\text{ub}}A \\ A \\ -A \end{bmatrix} y \leq \begin{bmatrix} b_{\text{ub}} + A_{\text{ub}}b \\ x_{\text{ub}} + b \\ -x_{\text{lb}} - b \end{bmatrix},$$
$$(A_{\text{eq}}A)y = (b_{\text{eq}} + A_{\text{eq}}b).$$

13.2 Full-Rank Inverse

For $A \in \mathbf{R}^{n \times n}$, if $\text{rank}(A) = n$,

$$\min_{x \in \mathbf{R}^n} \; J = f(Ax - b)$$
$$x \in \mathbf{P},$$

is equivalent to

$$\min_{y \in \mathbf{R}^n} \; J = f(y)$$
$$A^{-1}(y - b) \in \mathbf{P},$$

with $x^* = A^{-1}(y^* - b)$.

13.3 Full-Rank Right Pseudo Inverse

For $A \in \mathbf{R}^{m \times n}$, if $\text{rank}(A) = m$, then $A^+ = A^\intercal(AA^\intercal)^{-1}$, $AA^+ = \boldsymbol{I}_m$, and

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \; J &= f(Ax - b) \\ &= f(A(x - A^+b)) \\ \text{s.t.} \quad & x \in \mathbf{P} \end{aligned}$$

is equivalent to

$$\min_{y \in \mathbf{R}^n} \; J = f(Ay)$$
$$\text{s.t.} \quad y + A^+b \in \mathbf{P},$$

with $x^* = y^* + A^+b$. **Note**: This problem is ill-conditioned without constraint.

13.4 Full-Rank Left Pseudo Inverse

For $A \in \mathbf{R}^{m \times n}$, if $\text{rank}(A) = n$, then $A^+ = (A^\intercal A)^{-1}A^\intercal$, $A^+A = \boldsymbol{I}_n$, and

$$\min_{x \in \mathbf{R}^n} \; J = f(Ax - b).$$
$$x \in \mathbf{P}$$

is equivalent to

$$\min_{y \in \mathbf{R}^m} \; J = f(y)$$
$$\text{s.t.} \quad A^+(y + b) \in \mathbf{P},$$

with $x^* = A^+(y^* + b)$.

13.5 Simplifying Inequality Constriant

Consider the general problem

$$\min_{x\in\mathbf{R}^n} \quad J = f(Ax-b)$$

$$\begin{aligned}\text{s.t.} \quad A_{\text{ub}}x &\le b_{\text{ub}},\\ A_{\text{eq}}x &= b_{\text{eq}}.\end{aligned}$$

13.5.1 Slack Variables

For $A_{\text{ub}} \in \mathbf{R}^{m\times n}$, introduce a slack vector $y \in \mathbf{R}^m$ to get

$$\min_{x\in\mathbf{R}^n} \quad J = f(Ax-b)$$

$$\begin{aligned}\text{s.t.} \quad A_{\text{ub}}x + y &= b_{\text{ub}},\\ A_{\text{eq}}x &= b_{\text{eq}},\\ y &\ge 0,\end{aligned}$$

which can be written as

$$\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad J = f\left(\left[\begin{array}{cc} A & \mathbf{0} \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right] - b\right)$$

$$\begin{aligned}\text{s.t.} \quad \left[\begin{array}{cc} A_{\text{ub}} & \boldsymbol{I}_m \\ A_{\text{eq}} & \mathbf{0} \end{array}\right]\left[\begin{array}{c} x \\ y \end{array}\right] &= \left[\begin{array}{c} b_{\text{ub}} \\ b_{\text{eq}} \end{array}\right],\\ y &\ge 0.\end{aligned}$$

13.5.2 Non–Negative Variables

For the problem

$$\min_{x\in\mathbf{R}^n} \quad J = f(Ax-b)$$

$$\begin{aligned}\text{s.t.} \quad A_{\text{ub}}x + y &= b_{\text{ub}},\\ A_{\text{eq}}x &= b_{\text{eq}},\\ y &\ge 0,\end{aligned}$$

let $x = x_{\text{pos}} - x_{\text{neg}}$ with $x_{\text{pos}} \ge 0$ and $x_{\text{neg}} \ge 0$, to get

$$\min_{\{x_{\text{pos}},x_{\text{neg}},y\}\in\{\mathbf{R}^n,\mathbf{R}^n,\mathbf{R}^m\}} \quad J = f\left(\left[\begin{array}{ccc} A & -A & \mathbf{0} \end{array}\right]\left[\begin{array}{c} x_{\text{pos}} \\ x_{\text{neg}} \\ y \end{array}\right] - b\right)$$

$$\begin{aligned}\text{s.t.} \quad \left[\begin{array}{ccc} A_{\text{ub}} & -A_{\text{ub}} & \boldsymbol{I}_m \\ A_{\text{eq}} & -A_{\text{eq}} & \mathbf{0} \end{array}\right]\left[\begin{array}{c} x_{\text{pos}} \\ x_{\text{neg}} \\ y \end{array}\right] &= \left[\begin{array}{c} b_{\text{ub}} \\ b_{\text{eq}} \end{array}\right],\\ \left[\begin{array}{c} x_{\text{pos}} \\ x_{\text{neg}} \\ y \end{array}\right] &\ge \mathbf{0}^{2n+m}.\end{aligned}$$

The problem is now in standard form [1, p. 147].

13.6 Removing Equality Constraint

Consider $A_{\text{eq}} = U_+ S_+ V_+^\intercal$. A valid problem has $U_0^\intercal b_{\text{eq}} = \mathbf{0}$. Multiply $U_+^\intercal$ across the equality constraint to get

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J =& f(Ax - b) \\ \text{s.t.} \quad & f_{\text{ub}}(x) \le 0 \\ & S_+ V_+^\intercal x = U_+^\intercal b_{\text{eq}}. \end{aligned}$$

If $x = V_+ S_+^{-1} y + V_0 z$, then $y = U_+^\intercal b_{\text{eq}}$, and the problem reduces to

$$\begin{aligned} \min_{z \in \mathbf{R}^{n-r}} \quad J =& f(AV_0 z + AA_{\text{eq}}^+ b_{\text{eq}} - b) \\ \text{s.t.} \quad & f_{\text{ub}}(V_0 z + A_{\text{eq}}^+ b_{\text{eq}}) \le 0 \end{aligned}$$

13.7 Relaxing Equality Constrains

Consider

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f(x) \\ \text{s.t.} \quad & f_{\text{ub}}(x) \le 0, \\ & Ax = b, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f(x) \\ \text{s.t.} \quad & f_{\text{ub}}(x) \le 0, \\ & \begin{bmatrix} A \\ -A \end{bmatrix} x \le \begin{bmatrix} b \\ -b \end{bmatrix}. \end{aligned}$$

However, there is no interior to this set. For a fixed $y > 0$, consider the relaxed problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f(x) \\ \text{s.t.} \quad & f_{\text{ub}}(x) \le 0, \\ & Ax - y \le b, \\ & Ax + y \ge b. \end{aligned}$$

For $W \in \mathbf{S}_+^m$, consider the relaxed problem

$$\begin{aligned} \min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad & J = f(x) + \|y\|_W \\ \text{s.t.} \quad & f_{\text{ub}}(x) \le 0, \\ & \begin{bmatrix} A & -\boldsymbol{I}_m \\ -A & -\boldsymbol{I}_m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \le \begin{bmatrix} b \\ -b \end{bmatrix}. \end{aligned}$$

Note: These constraints force $y \ge 0$.

This is equivalent to the problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J =& f(x) + \|Ax - b\|_W \\ \text{s.t.} \quad & f_{\text{ub}}(x) \le 0. \end{aligned}$$

13.8 2–Norm Simplification

Consider

$$A = U_+ S_+ V_+^\intercal.$$

Let

$$\begin{aligned} x &= \underbrace{\begin{bmatrix} V_+ & V_0 \end{bmatrix}}_{V} \begin{bmatrix} S_+^{-1} y \\ z \end{bmatrix} \\ &= V_+ S_+^{-1} y + V_0 z. \end{aligned}$$

The problem

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \ J &= \|Ax - b\|_2^2 \\ &= x^\intercal V_+ S_+ U_+^\intercal U_+ S_+ V_+^\intercal x - 2b^\intercal U_+ S_+ V_+^\intercal x + b^\intercal b \\ &= y^\intercal y - 2b^\intercal U_+ y + b^\intercal b + \underbrace{b^\intercal U_+ U_+^\intercal b - b^\intercal U_+ U_+^\intercal b)}_{0} \end{aligned}$$

$$\text{s.t.} \quad x \in \mathbf{P}$$

is equivalent to

$$\min_{\{y,z\} \in \{\mathbf{R}^r, \mathbf{R}^{n-r}\}} \quad J = \|y - U_+^\intercal b\|_2^2 + b^\intercal U_0 U_0^\intercal b,$$

$$\text{s.t.} \quad V_+ S_+^{-1} y + V_0 z \in \mathbf{P},$$

with

$$x^* = V_+ S_+^{-1} y^* + V_0 z^*.$$

14 Solvers

There are well–known algorithms for solving **LP** and **QP** in polynomial–time. Many specialized solutions exists for **LP** and **QP** with specific properties, but generalized interior point methods [1, p. 569] typically have similar run–time. Matlab has an **LP** [10] and **QP** [11] solver built into its Optimization Toolbox. Several **QP** solvers are available for Python, e.g., the cvxopt library [35]. There are also **QP** solvers available for C++, e.g., [13], and Java, e.g., [14]. For larger problems or problems that need to be solved quickly, simplifying assumptions and relaxations can be employed as shown in [5]. Further, ADMM [6] can be used for massively parallel optimization. If the problem has sparsity, [25] and [29] can be used. Effecient data structures should be employed wherever there are repeated copies of a sub–matrix or structures with diagonals or zeros. It is also adventageous to decouple problems wherever it is possible to do so. For nonlinear problems, solvers like Matlab's fmincon [12] can be used, which iteratively solve with **QP** . A nonlinear solver library for C++ can be found at [31]. A robot-centeric solver library can be found at [30]

14.1 Interior Point

For $i \in \mathbf{Z}[0, m]$, let $f_i(x)$ be convex functions. Consider

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, \\ & Ax = b, \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f_0(x) + \sum_{i=1}^{m} \text{Indicator}(f_i(x) \leq 0) \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

For $t > 0$, this objective function can be approximated by the log–barrier with

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f_0(x) + \sum_{i=1}^{m} -\frac{1}{t} \log(-f_i(x)) \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

Let

$$h(x) = -\sum_{i=1}^{m} \log(-f_i(x)). \tag{6}$$

Then,

$$\begin{aligned} \nabla h &= -\sum_{i=1}^{m} \frac{1}{f_i} \nabla f_i \\ \nabla^2 h &= -\sum_{i=1}^{m} \frac{1}{f_i} \nabla^2 f_i + \sum_{i=1}^{m} \frac{1}{f_i^2} (\nabla f_i)(\nabla f_i)^\intercal \end{aligned}$$

The approximate problem can be restated as

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = t f_0(x) + h(x) \\ \text{s.t.} \quad & Ax = b. \end{aligned}$$

The optimal solution x^* is a function of t and can be computed with Algorithms 12.

$x \in \mathbf{P}$, $t > 0$ $s > 1$, small $c > 0$
while $m/t > c$ **do**
- $x \leftarrow \underset{x \in \mathbf{R}^n}{\text{argmin}} \quad t f_0(x) + h(x) \quad \text{s.t.} \quad Ax = b,$ where h comes from 6,
- $t \leftarrow st$.

end

Algorithm 12: Barrier Method

14.2 Feasible Initial Conditions

A feasible x in the domain

$$\mathbf{D} = \{x \in \mathbf{R}^n | f_i(x) \leq 0, Ax = b\}$$

can be found by introduce a slack variable and minimize it until the the constraints are satisfied, e.g.,

$$\begin{aligned} \min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad & J = \|y\|_1 \\ \text{s.t.} \quad & f_i(x) \leq y_i, \\ & Ax = b, \end{aligned}$$

or

$$\begin{aligned} \min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}} \quad & J = \|y\|_2^2 \\ \text{s.t.} \quad & f_i(x) \leq y_i, \\ & Ax = b, \end{aligned}$$

or

$$\begin{aligned} \min_{\{x,y\} \in \{\mathbf{R}^n, \mathbf{R}\}} \quad & J = |y| \\ \text{s.t.} \quad & f_i(x) \leq y, \\ & Ax = b. \end{aligned}$$

Note: This provides a diagnostic of which constraints are not satisfiable.

14.3 Polyhedron Constraints

For $x \in \mathbf{P}$ with $\mathbf{P} = \{x \in \mathbf{R}^n | Ax \leq b\}$,

$$f_i(x) = a_i^\intercal x - b_i.$$

The log-barrier is then computed with

$$h(x) = -\sum_{i=0}^{m-1} \log(b_i - a_i^\intercal x).$$

The gradient and Hessian are computed with

$$\begin{aligned} \nabla h(x) &= \sum_{i=0}^{m-1} \frac{a_i}{b_i - a_i^\intercal x} \\ &= A_{\text{ub}}^\intercal \text{col}\{d_i\}_{i=0}^{m-1}, \end{aligned}$$

$$\begin{aligned} \nabla^2 h(x) &= \sum_{i=0}^{m-1} \frac{a_i a_i^\intercal}{(b_i - a_i^\intercal x)^2} \\ &= A_{\text{ub}}^\intercal \text{diag}\{d_i^2\}_{i=0}^{m-1} A_{\text{ub}}, \end{aligned}$$

where

$$d_i = \frac{1}{b_i - a_i^\intercal x}.$$

The equality constraint can be removed using SVD as described in Section 13.6

14.3.1 Positive Quadrant Constraints

The constraint $x \geq 0$, i.e., $-x \leq 0$, simplifies calculations with a log-barrier of

$$h(x) = -\sum_{i=0}^{m-1} \log(x_i).$$

The gradient and Hessian are given by

$$\begin{aligned}\nabla h(x) &= -\sum_{i=0}^{m-1} \frac{\mathbf{e}_i}{x_i} = -\mathrm{col}\{x_i^{-1}\}_{i=0}^{m-1},\\ \nabla^2 h(x) &= \sum_{i=0}^{m-1} \frac{\mathbf{e}_i \mathbf{e}_i^{\mathsf{T}}}{x_i^2} = \quad \mathrm{diag}\{x_i^{-2}\}_{i=0}^{m-1}.\end{aligned}$$

14.4 ADMM: Alternating Direction Method of Multipliers

Consider the convex optimization problem

$$\begin{aligned}\min_{\{x,y\}\in\{\mathbf{R}^n,\mathbf{R}^m\}} \quad & J = f(x) + g(y)\\ \text{s.t.} \quad & Ax + By = c.\end{aligned}$$

The regularized Lagrangian is given by

$$L(x, y, z) = f(x) + g(y) + z^{\mathsf{T}}(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|_2^2.$$

The ADMM is computed with Algorithm 13.

$\{x, y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}$
while $\{x, y\}$ not converged **do**

- $x \leftarrow \underset{x\in\mathbf{R}^n}{\operatorname{argmin}} \quad L(x, y, z)$,
- $y \leftarrow \underset{y\in\mathbf{R}^m}{\operatorname{argmin}} \quad L(x, y, z)$,
- $z \leftarrow z + \rho(Ax + By - c)$.

end

Algorithm 13: ADMM

14.5 Scaled ADMM

The regularized Lagrangian can be expanded with

$$\begin{aligned}L(x, y, z) &= f(x) + g(y) + z^{\mathsf{T}}(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|_2^2\\ &= f(x) + g(y) + \frac{\rho}{2}\|Ax + By - c + u\|_2^2 + r,\end{aligned}$$

where

$$u = z/\rho.$$

The scaled ADMM is computed with Algorithm 14.

$\{x, y\} \in \{\mathbf{R}^n, \mathbf{R}^m\}$
while $\{x, y\}$ not converged **do**

- $x \leftarrow \underset{x\in\mathbf{R}^n}{\operatorname{argmin}} \quad f(x) + \frac{\rho}{2}\|Ax + by - c + u\|_2^2$,
- $y \leftarrow \underset{y\in\mathbf{R}^m}{\operatorname{argmin}} \quad g(y) + \frac{\rho}{2}\|Ax + by - c + u\|_2^2$,
- $u \leftarrow u + \rho(Ax + By - c)$.

end

Algorithm 14: Scaled ADMM

14.6 Examples

If $A = \boldsymbol{I}_n$,

$$x^* = \underset{x \in \mathbf{R}^n}{\text{argmin}} \quad f(x) + \frac{\rho}{2}\|x - v\|_2^2.$$

- If

  $$\begin{aligned} f(x) &= \text{Indicator}(x \in \mathbf{D}) \\ &= \begin{cases} 0 & \text{if} \quad x \in \mathbf{D} \\ \infty & \text{else} \end{cases}, \end{aligned}$$

  then

  $$\begin{aligned} x^* = \underset{x \in \mathbf{R}^n}{\text{argmin}} \quad & \|x - v\|_2^2 \\ \text{s.t.} \quad & x \in \mathbf{D}. \end{aligned}$$

- If

  $$f(x) = \lambda\|x\|_1,$$

  then

  $$\begin{aligned} x_i^* &= \text{Deadzone}(v_i, \lambda/\rho) \\ &= (x - \lambda/\rho)_+ - (-x - \lambda/\rho)_+ \end{aligned}$$

  This result can be derived from independently optimizing

  $$\underset{x_i \in \mathbf{R}}{\text{argmin}} \quad \lambda|x_i| + \frac{\rho}{2}(x_i - v_i)^2,$$

  which gives

  $$\begin{aligned} 0 &= \lambda \text{sign}(x_i) + \rho(x_i - v_i), \\ v_i &= \frac{\lambda}{\rho}\text{sign}(x_i) + x_i, \\ x_i &= \text{Deadzone}(v_i, \lambda/\rho). \end{aligned}$$

- If

  $$f(x) = \frac{1}{2}x^\intercal Q x + c^\intercal x + r,$$

  then

  $$x^* = (Q + \rho A^\intercal A)^{-1}(\rho A^\intercal v - c).$$

- If

  $$f(x) = \frac{1}{2}x^\intercal Q x + c^\intercal x + \text{Indicator}(x \in \mathbf{P}),$$

  then

  $$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = \frac{1}{2}x^\intercal Q x + c^\intercal x + \frac{\rho}{2}\|x - v\|_2^2 \\ \text{s.t.} \quad & x \in \mathbf{P} \end{aligned}$$

14.7 Consensus ADMM

The problem

$$\min_{x \in \mathbf{R}^n} \quad J = \sum_{i=0}^{m} f_i(x)$$

is equivalent to

$$\begin{aligned} \min_{x_i \in \mathbf{R}} \quad & J = \sum_{i=0}^{m} f_i(x_i) \\ \text{s.t.} \quad & x_i - z = 0. \end{aligned}$$

This problem can be solved with Algorithm 15.

$x_i \in \mathbf{R}^n$
while x_i not converged **do**
- **for** *parallel* $i \in \mathbf{Z}[0, m]$ **do**
  - $x_i \leftarrow \underset{x_i \in \mathbf{R}^n}{\text{argmin}} \quad f_i(x_i) + y_i^\intercal(x_i - z) + \frac{\rho}{2}\|x_i - z\|_2^2$,
  - $y_i \leftarrow y_i + \rho(x_i - z)$,

  end
- $z \leftarrow \frac{1}{1+m}\sum_{i=0}^{m}(x_i + y_i/\rho)$.

end

Algorithm 15: Consensus ADMM

14.7.1 Constraint Decomposition

Any convex problem can be expressed as

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad & J = f_0(x) \\ \text{s.t.} \quad & x \in \mathbf{P}, \end{aligned}$$

where

$$\mathbf{P} = \bigcap_{i=1}^{m} \mathbf{P}_i,$$

which is equivalent to

$$\min_{x \in \mathbf{R}^n} \quad J = f_0(x) + \sum_{i=1}^{m} \text{Indicator}(x \in \mathbf{P}_i).$$

14.8 Logistic Regression

Logistic regression can be expressed as

$$\begin{aligned} \min_{\{x_i, x_j\} \in \{\mathbf{R}^n, \mathbf{R}^n\}} \quad & J = -\sum_i \log(p(x_i)) - \sum_j \log(1 - p(x_j)) \\ \text{s.t.} \quad & x_i - z = 0, \\ & x_j - z = 0. \end{aligned}$$

14.9 Partial QP

14.9.1 Decompostion with SVD

Consider the problem

$$\min_{z \in \mathbf{R}^n} \quad J = \frac{1}{2} z^\intercal Q z + c^\intercal z$$
$$\text{s.t.} \quad z \in \mathbf{P},$$

with

$$\left[\begin{array}{c} A_{\text{ub}} \\ A_{\text{eq}} \end{array} \right] = U_+ S_+ V_+^\intercal .$$

Let

$$\begin{aligned} z &= V_0 x + V_+ y \\ &= \left[\begin{array}{cc} V_0 & V_+ \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right]. \end{aligned}$$

The problem becomes

$$\min_{y \in \mathbf{R}^r} \min_{x \in \mathbf{R}^{n-r}} \quad J = \frac{1}{2} \left[\begin{array}{c} x \\ y \end{array} \right]^\intercal \left[\begin{array}{cc} V_0^\intercal Q V_0 & V_0^\intercal Q V_+ \\ V_+^\intercal Q V_0 & V_+^\intercal Q V_+ \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right] + \left[\begin{array}{c} V_0^\intercal c \\ V_+^\intercal c \end{array} \right]^\intercal \left[\begin{array}{c} x \\ y \end{array} \right]$$
$$\text{s.t.} \quad V_+ y \in \mathbf{P}$$

14.9.2 Sum of Squares Decomposition

If $A \in \mathbf{S}^n$, $D \in \mathbf{S}^m$, and $B \in \mathbf{R}^{n \times m}$, then

$$\left[\begin{array}{c} x \\ y \end{array} \right]^\intercal \left[\begin{array}{cc} A & B \\ B^\intercal & D \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right] = (x + A^{-1} B y)^\intercal A (x + A^{-1} B y) + y^\intercal (D - B^\intercal A^{-1} B) y.$$

This gives a test for global positivity with

$$\left[\begin{array}{cc} A & B \\ B^\intercal & D \end{array} \right] \in \mathbf{S}_+^{n+m} \quad \Leftrightarrow \quad A \in \mathbf{S}_+^n \quad \text{and} \quad D - B^\intercal A^{-1} B \in \mathbf{S}_+^m .$$

This can be applied recursively to certify non–negativity.

14.9.3 Partial Min

If $A \in \mathbf{S}_+^n$, then partial quadratic optimization can be computed with

$$\begin{aligned} \min_{x \in \mathbf{R}^n} \quad J &= \left[\begin{array}{c} x \\ y \end{array} \right]^\intercal \left[\begin{array}{cc} A & B \\ B^\intercal & D \end{array} \right] \left[\begin{array}{c} x \\ y \end{array} \right] \\ &= y^\intercal (D - B^\intercal A^{-1} B) y, \end{aligned}$$

where

$$x^* = -A^{-1} B y.$$

The problem

$$\min_{y \in \mathbf{R}^m} \quad J(x^*, y)$$
$$\text{s.t.} \quad y \in \mathbf{P},$$

becomes

$$\min_{y \in \mathbf{R}^m} \quad J = y^\intercal (D - B^\intercal A^{-1} B) y$$
$$\text{s.t.} \quad y \in \mathbf{P}.$$

15 Appendix

15.1 SVD: Singular Value Decomposition

For $A \in \mathbf{R}^{m \times n}$,

$$\begin{aligned} A &= USV^\intercal \\ &= \begin{bmatrix} U_+ & U_0 \end{bmatrix} \begin{bmatrix} S_+ & \mathbf{0}^{r \times n-r} \\ \mathbf{0}^{m-r \times r} & \mathbf{0}^{m-r \times n-r} \end{bmatrix} \begin{bmatrix} V_+^\intercal \\ V_0^\intercal \end{bmatrix} \\ &= U_+ S_+ V_+^\intercal, \end{aligned}$$

where $r = \text{rank}(A)$, $S_+ = \text{diag}\{\sigma_i\}_{i=0}^{r-1}$ with $\sigma_i > 0$ for $i \in \mathbf{Z}[0, r-1]$, and

$$\begin{aligned} U &\in \mathbf{R}^{m \times m}, \quad \text{s.t.} \quad U^\intercal U = UU^\intercal = \boldsymbol{I}_m, \quad U^{-1} = U^\intercal, \\ V &\in \mathbf{R}^{n \times n}, \quad \text{s.t.} \quad V^\intercal V = VV^\intercal = \boldsymbol{I}_n, \quad V^{-1} = V^\intercal. \end{aligned}$$

Orthonormality gives

$$\begin{aligned} U_+^\intercal U_+ &= \boldsymbol{I}_r, \\ V_+^\intercal V_+ &= \boldsymbol{I}_r, \\ \\ U_+^\intercal U_0 &= \mathbf{0}^{r \times m-r}, \\ V_+^\intercal V_0 &= \mathbf{0}^{r \times n-r}, \\ \\ U_+ U_+^\intercal &= \boldsymbol{I}_m - U_0 U_0^\intercal, \\ V_+ V_+^\intercal &= \boldsymbol{I}_n - V_0 V_0^\intercal. \end{aligned}$$

Note: $U_0^\intercal A = \mathbf{0}^{m-r \times n}$ and $AV_0 = \mathbf{0}^{m \times n-r}$.

15.2 Generalized Pseudo Inverse

For $A \in \mathbf{R}^{m \times n}$, if $\text{rank}(A) < \min(m, n)$, the pseudo inverse is given by

$$A^+ = \begin{cases} V_+ S_+^{-1} U_+^\intercal + BU_0^\intercal & \text{left} \\ V_+ S_+^{-1} U_+^\intercal + V_0 B & \text{right} \end{cases},$$

where B can be freely choosen. Unless otherwise specified, assume $B = \mathbf{0}$.

15.3 Matrix Factorization

Adding $A \in \mathbf{R}^{m \times n}$ and $B \in \mathbf{R}^{m \times n}$, for SVD $A = U_+ S_+ V_+^\intercal$,

$$\begin{aligned} A + B &= \begin{bmatrix} U_+ & U_0 \end{bmatrix} \begin{bmatrix} S_+ & \mathbf{0}^{r \times n-r} \\ \mathbf{0}^{m-r \times r} & \mathbf{0}^{m-r \times n-r} \end{bmatrix} \begin{bmatrix} V_+^\intercal \\ V_0^\intercal \end{bmatrix} + UU^\intercal BV^\intercal V \\ &= U \begin{bmatrix} U_+^\intercal BV_+ + S_+ & U_+^\intercal BV_0 \\ U_0^\intercal BV_+ & U_0^\intercal BV_0 \end{bmatrix} V^\intercal. \end{aligned}$$

If $A \in \mathbf{R}^{n \times n}$,

$$\begin{aligned} U^\intercal (A+B) U &= \begin{bmatrix} S_+ V_+^\intercal U_+ & S_+ V_+^\intercal U_0 \\ \mathbf{0}^{n-r \times r} & \mathbf{0}^{n-r \times n-r} \end{bmatrix} + \begin{bmatrix} U_+^\intercal BU_+ & U_+^\intercal BU_0 \\ U_0^\intercal BU_+ & U_0^\intercal BU_0 \end{bmatrix}, \\ V^\intercal (A+B) V &= \begin{bmatrix} V_+^\intercal U_+ S_+ & \mathbf{0}^{r \times n-r} \\ V_0^\intercal U_+ S_+ & \mathbf{0}^{n-r \times n-r} \end{bmatrix} + \begin{bmatrix} V_+^\intercal BV_+ & V_+^\intercal BV_0 \\ V_0^\intercal BV_+ & V_0^\intercal BV_0 \end{bmatrix}. \end{aligned}$$

15.4 Kronecker Product

For $A \in \mathbf{R}^{m \times n}$,

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix}.$$

In general, $A \otimes B \neq B \otimes A$. Some useful properties include:

$$\boldsymbol{I}_q \otimes A = \text{diag}\{A\}_{i=0}^{q-1}, \qquad \mathbf{1}^q \otimes A = \text{row}\{A\}_{i=0}^{q-1},$$

$$\begin{aligned} A \otimes (B + C) &= A \otimes B + A \otimes C, & (A + B) \otimes C &= A \otimes C + B \otimes C, \\ (cA) \otimes B &= A \otimes (cB) = c(A \otimes B), & A \otimes (B \otimes C) &= (A \otimes B) \otimes C, \\ A \otimes B &= C(B \otimes A)D, & AC \otimes BD &= (A \otimes B)(C \otimes D), \end{aligned}$$

$$\begin{aligned} (A \otimes B)^\intercal &= A^\intercal \otimes B^\intercal, & \left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right)^\intercal &= \textstyle\bigotimes_{i=0}^{q-1} A_i^\intercal, \\ (A \otimes B)^p &= A^p \otimes B^p, & \left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right)^p &= \textstyle\bigotimes_{i=0}^{q-1} A_i^p, \\ \|A \otimes B\|_2^2 &= \|A\|_2^2 \|B\|_2^2, & \left\|\textstyle\bigotimes_{i=0}^{q-1} A_i\right\|_2^2 &= \textstyle\prod_{i=0}^{q-1} \|A_i\|_2^2, \\ (A \otimes B)^{-1} &= A^{-1} \otimes B^{-1}, & \left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right)^{-1} &= \textstyle\bigotimes_{i=0}^{q-1} A_i^{-1}, \\ (A \otimes B)^+ &= A^+ \otimes B^+, & \left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right)^+ &= \textstyle\bigotimes_{i=0}^{q-1} A_i^+, \\ \text{eig}(A \otimes B) &= \text{eig}(A) \otimes \text{eig}(B), & \text{eig}\left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right) &= \textstyle\bigotimes_{i=0}^{q-1} \text{eig}(A_i), \\ \text{rank}(A \otimes B) &= \text{rank}(A)\text{rank}(B), & \text{rank}\left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right) &= \textstyle\prod_{i=0}^{q-1} \text{rank}(A_i), \\ \text{svd}(A \otimes B) &= \text{svd}(A) \otimes \text{svd}(B), & \text{svd}\left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right) &= \textstyle\bigotimes_{i=0}^{q-1} \text{svd}(A_i), \\ \text{tr}(A \otimes B) &= \text{sum}(\text{eig}(A) \otimes \text{eig}(B)), & \text{tr}\left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right) &= \text{sum}\left(\textstyle\bigotimes_{i=0}^{q-1} \text{eig}(A_i)\right), \\ \det(A \otimes B) &= \det(A)^{\text{rank}(B)} \det(B)^{\text{rank}(A)}, & \det\left(\textstyle\bigotimes_{i=0}^{q-1} A_i\right) &= \textstyle\prod_{i=0}^{q-1} \det(A_i)^{\prod_{j \neq i} \text{rank}(A_j)}, \\ \frac{\partial (A \otimes B)}{\partial s} &= \frac{\partial A}{\partial s} \otimes B + A \otimes \frac{\partial B}{\partial s}, & \frac{\partial}{\partial x_j} \bigotimes_{i=0}^{q-1} f_i(x_i) &= \bigotimes_{i<j} f_i \otimes \frac{\partial f_j}{\partial x_j} \otimes \bigotimes_{i>j} f_i \end{aligned}$$

For independent $\widetilde{A}$ and $\widetilde{B}$,

$$\langle \widetilde{A} \otimes \widetilde{B} \rangle = \langle \widetilde{A} \rangle \otimes \langle \widetilde{B} \rangle.$$

For $x_i \in \mathbf{R}^{n_i}$ and $y_i \in \mathbf{R}^{n_i}$,

$$\begin{aligned} \textstyle\prod_{i=0}^{q-1} y_i^\intercal x_i &= \left(\textstyle\bigotimes_{i=0}^{q-1} y_i^\intercal\right)\left(\textstyle\bigotimes_{i=0}^{q-1} x_i\right) \\ &= \left(\textstyle\bigotimes_{i=0}^{q-1} y_i\right)^\intercal \left(\textstyle\bigotimes_{i=0}^{q-1} x_i\right). \end{aligned}$$

15.5 Vectorization

For $A \in \mathbf{R}^{m \times n}$, row-wise vectorization gives

$$\text{vec}\begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix} = [\ A_{11} \quad \cdots \quad A_{1n} \quad \cdots \quad A_{m1} \quad \cdots \quad A_{mn}\ ]^\intercal,$$

with

$$\begin{aligned}\mathrm{vec}\,(A\ \ ) &= (\boldsymbol{I}_m \otimes A^\intercal)\vec{\boldsymbol{I}}_m = (A\ \ \otimes \boldsymbol{I}_n\,)\vec{\boldsymbol{I}}_n\,,\\ \mathrm{vec}\,(A^\intercal) &= (\boldsymbol{I}_n\ \otimes A\ \ )\vec{\boldsymbol{I}}_n\ = (A^\intercal \otimes \boldsymbol{I}_m)\vec{\boldsymbol{I}}_m.\end{aligned}$$

The inverse operation is defined by

$$A = \mathrm{mat}(\,\vec{A},\ m,\ n\,).$$

For $AB \in \mathbf{R}^{m\times n}$,

$$\mathrm{vec}\,(AB) = (\boldsymbol{I}_m \otimes B^\intercal)\vec{A} = (A \otimes \boldsymbol{I}_n)\vec{B},$$

$$\mathrm{vec}\,(ABC) = (A \otimes C^\intercal)\vec{B}.$$

For $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^m$,

$$\begin{aligned}\mathrm{vec}\,(x \otimes y^\intercal) &= \mathrm{vec}\,(xy^\intercal) &&= (\boldsymbol{I}_n \otimes y)x &&= x \otimes y,\\ \mathrm{vec}\,(x^\intercal \otimes y) &= \mathrm{vec}\,(yx^\intercal) &&= (\boldsymbol{I}_m \otimes x)y &&= y \otimes x.\end{aligned}$$

For row-wise concatenation,

$$\mathrm{vec}\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} \vec{A} \\ \vec{B} \end{bmatrix}.$$

15.6 Trace

For $A \in \mathbf{R}^{n\times n}$,

$$\begin{aligned}\mathrm{tr}(A) &= \textstyle\sum_{i=0}^{n-1} A_{ii}\\ &= \mathrm{tr}(A^\intercal)\\ &= \vec{\boldsymbol{I}}^\intercal \vec{A},\\ \mathrm{tr}(AB) &= \textstyle\sum_{i=0}^{n-1}\sum_{j=0}^{m-1} A_{ij}B_{ji}\\ &= \mathrm{tr}(BA)\\ &= \mathrm{tr}\left(A\textstyle\sum_{i=0}^{n-1}\lambda_i v_i v_i^{\mathrm{c}\intercal}\right) = \textstyle\sum_{i=0}^{n-1}\lambda_i v_i^{\mathrm{c}\intercal} A v_i,\\ \mathrm{tr}(A+B) &= \mathrm{tr}(A) + \mathrm{tr}(B),\\ \mathrm{tr}(A^\intercal A) &= \mathrm{svd}(A)^\intercal \mathrm{svd}(A),\\ \mathrm{tr}(A\otimes B) &= \mathrm{tr}(A)\mathrm{tr}(B),\\ \mathrm{tr}(A) &= \log(\det(\exp(A))),\\ \mathrm{tr}(ABA^{-1}) &= \mathrm{tr}(B),\\ \mathrm{tr}(A^p) &= \textstyle\sum_{i=0}^{n-1}\lambda_i^p,\\ \frac{\partial}{\partial B}\mathrm{tr}(ABC) &= A^\intercal C^\intercal,\\ \frac{\partial}{\partial A}\mathrm{tr}(ABA^\intercal) &= AB^\intercal + AB.\end{aligned}$$

For $x \in \mathbf{R}^n$,

$$\mathrm{tr}\,(xx^\intercal) = x^\intercal x.$$

For $a \in \mathbf{R}^n$ and $b \in \mathbf{R}^m$, $[ab^\intercal]_{ij} = a_i b_j$.

For $a_i \in \mathbf{R}^m$, $b_i \in \mathbf{R}^n$, $A = \mathrm{col}\{a_i\}_i$, $B = \mathrm{col}\{b_i\}_i$,

$$[AB^\intercal]_{ij} = \left[\sum_k a_k b_k^\intercal\right]_{ij} = \sum_k [a_k b_k^\intercal]_{ij} = \sum_k [a_k]_i\,[a_k]_j = \sum_k A_{ik}B_{jk}.$$

15.7 2–Norm

For $A \in \mathbf{R}^{m\times n}$ and $x \in \mathbf{R}^n$,

$$
\begin{aligned}
\|Ax\|_2^2 &= x^\intercal A^\intercal Ax, \\
&= \vec{A}^\intercal(\boldsymbol{I}_m \otimes x)(\boldsymbol{I}_m \otimes x^\intercal)\vec{A}, \\
&= \vec{A}^\intercal(\boldsymbol{I}_m \otimes xx^\intercal)\vec{A}, \\
&= (\vec{A}^\intercal \otimes \vec{A}^\intercal)\mathrm{vec}\,(\boldsymbol{I}_m \otimes xx^\intercal)\,, \\
&= \mathrm{vec}\,(A^\intercal A)^\intercal\,\mathrm{vec}\,(xx^\intercal)\,, \\
&= \mathrm{tr}(Axx^\intercal A^\intercal), \\
&= \mathrm{tr}(A^\intercal Axx^\intercal), \\
&\le \max(\mathrm{svd}(A))^2\|x\|_2^2, \\
&\le \|\vec{A}\|_2^2\|x\|_2^2.
\end{aligned}
$$

15.8 Determinant

Some useful properties include:

$$
\begin{aligned}
\det(A)\boldsymbol{I}_n &= \mathrm{adj}(A)A \\
\det(\boldsymbol{I}_n) &= 1, \\
\det(U) &= \pm 1, && \text{where} \quad U^\intercal U = UU^\intercal = \boldsymbol{I}_n, \\
\det(A^\intercal) &= \det(A), \\
\det(A^{-1}) &= \det(A)^{-1}, \\
\det(aA) &= a^n\det(A), \\
\det(AB) &= \det(A)\det(B), && \det\left(\textstyle\prod_{i=0}^{q-1} A_i\right) = \textstyle\prod_{i=0}^{q-1}\det(A_i), \\
\det(A^p) &= \det(A)^p, \\
\det(A) &= \exp(\mathrm{tr}(\log(A))), \\
\det(A) &= \det(BAB^{-1}) \\
\det(A) &= \textstyle\prod_{i=0}^{n-1}\lambda_i, && \text{where} \quad \det(A - \lambda_i\boldsymbol{I}_n) = 0, \\
\frac{\partial}{\partial A}\det(A) &= \mathrm{adj}(A)^\intercal, && \frac{\partial}{\partial s}\det(A) = \det(A)\mathrm{tr}\left(A^{-1}\frac{\partial A}{\partial s}\right), \\
\frac{\partial}{\partial A}\log(\det(A)) &= A^{-\intercal} && \frac{\partial}{\partial s}\log(\det(A)) = \mathrm{tr}\left(A^{-1}\frac{\partial A}{\partial s}\right) \\
\det\begin{bmatrix} A & B \\ C & D \end{bmatrix} &= \det(A)\det(D - CA^{-1}B) \\
&= \det(A - BC) \quad \text{if} \quad D = \boldsymbol{I}.
\end{aligned}
$$

For $x \in \mathbf{R}^n$ and $y \in \mathbf{R}^n$,

$$
\det(\boldsymbol{I}_n + xy^\intercal) = 1 + x^\intercal y.
$$

For $A \in \mathbf{S}^n$,$M_i = A[0 : i, 0 : i]$, and $i \in \mathbf{Z}[1, n-1]$,

$$
\begin{aligned}
A \succeq 0 \quad &\Leftrightarrow \quad \det(M_i) \ge 0, \quad A \preceq 0 \quad \Leftrightarrow \quad -A \succeq 0, \\
A \succ 0 \quad &\Leftrightarrow \quad \det(M_i) > 0, \quad A \prec 0 \quad \Leftrightarrow \quad -A \succ 0.
\end{aligned}
$$

For $A \in \mathbf{S}^n$ and $B \in \mathbf{S}^n$,

$$
\begin{aligned}
A \succeq B \quad &\Leftrightarrow \quad A - B \succeq 0, \\
A \succ B \quad &\Leftrightarrow \quad A - B \succ 0.
\end{aligned}
$$

15.9 Pseudo Inverse

For $x \in \mathbf{R}^n$,

$$x^+ = \begin{cases} (x^\intercal x)^{-1} x^\intercal & \text{if } \|x\| > 0 \\ 0 & \text{else} \end{cases}.$$

For $A \in \mathbf{R}^{m \times n}$ and $p \in \mathbf{R}_+$,

$$\begin{aligned}
A &= (A^+)^+ \\
&= A(A^+A)^p && \text{and} \quad A = (AA^+)^p A, \\
A^+ &= (A^\intercal A)^+ A^\intercal && \text{and} \quad A^+ = A^\intercal (AA^\intercal)^+, \\
(A^\intercal)^+ &= (A^+)^\intercal, \\
(aA)^+ &= a^{-1} A^+, \\
(A^\intercal A)^+ &= A^+ A^{+\intercal}, && \text{and} \quad (AA^\intercal)^+ = A^{+\intercal} A^+, \\
(A^+ A)^p &= A^+ A, && \text{and} \quad (AA^+)^p = AA^+, \\
\text{tr}(AA^+) &= \text{rank}(A), \\
(\boldsymbol{I}_n - A^+A)^p &= \boldsymbol{I}_n - A^+ A, && \text{and} \quad (\boldsymbol{I}_m - AA^+)^p = \boldsymbol{I}_m - AA^+ \\
(AB)^+ &= (A^+AB)^+(ABB^+)^+, \\
\begin{bmatrix} A & B \end{bmatrix}^+ &= \begin{bmatrix} ((\boldsymbol{I} - BB^+)A)^+ \\ ((\boldsymbol{I} - AA^+)B)^+ \end{bmatrix}, \\
\begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{bmatrix}^+ &= \begin{bmatrix} A^+ & \mathbf{0} \\ \mathbf{0} & B^+ \end{bmatrix}.
\end{aligned}$$

15.10 Inverse Matrix

For $A \in \mathbf{R}^{n \times n}$ with rank$(A) = n$,

$$\begin{aligned}
A^+ &= A^{-1}, \\
(A^\intercal)^{-1} &= (A^{-1})^\intercal, \\
(AB)^{-1} &= B^{-1} A^{-1}, \\
\frac{\partial}{\partial s} A^{-1} &= -A^{-1} \frac{\partial A}{\partial s} A^{-1}, \\
\left(\prod_{i=0}^{q-1} A_i \right)^{-1} &= \prod_{j=q-1}^{0} A_j^{-1}.
\end{aligned}$$

The block inverse is given by

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} E & -EBD^{-1} \\ -D^{-1}CE & F \end{bmatrix},$$

where

$$\begin{aligned}
E &= (A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}BFCA^{-1}, \\
F &= (D - CA^{-1}B)^{-1} = D^{-1} + D^{-1}CEBD^{-1}.
\end{aligned}$$

For $A \in \mathbf{S}_+^n$,

$$\begin{bmatrix} A & b \\ b^\intercal & d \end{bmatrix}^{-1} = \begin{bmatrix} E & -Ebd^{-1} \\ -d^{-1}b^\intercal E & f \end{bmatrix},$$

where

$$\begin{aligned}
f^{-1} &= d - b^\intercal A^{-1} b \\
E &= A^{-1} + A^{-1} b b^\intercal A^{-1} f.
\end{aligned}$$

15.11 Derivatives

15.11.1 Scalar to Scalar

For $f : \mathbf{R} \to \mathbf{R}$, the scalar first and second derivative are denoted by f' and f''.
Note: The same notation will be used for scalar to vector derivatives.

15.11.2 Vector to Scalar

For the scalar function $f : \mathbf{R}^n \to \mathbf{R}$,

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_0} & \cdots & \frac{\partial f}{\partial x_{n-1}} \end{bmatrix} \in \mathbf{R}^{1\times n}.$$

The gradient is

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_0} \\ \vdots \\ \frac{\partial f}{\partial x_{n-1}} \end{bmatrix} \in \mathbf{R}^n,$$

where

$$\frac{\partial f}{\partial x} = \nabla f^\intercal.$$

The Hessian is

$$\nabla^2 f = \begin{bmatrix} \frac{\partial}{\partial x_0}\frac{\partial f}{\partial x_0} & \cdots & \frac{\partial}{\partial x_{n-1}}\frac{\partial f_0}{\partial x_0} \\ \vdots & & \vdots \\ \frac{\partial}{\partial x_0}\frac{\partial f}{\partial x_{n-1}} & \cdots & \frac{\partial}{\partial x_{n-1}}\frac{\partial f}{\partial x_{n-1}} \end{bmatrix} \in \mathbf{R}^{n\times n},$$

where

$$\nabla^2 f = \frac{\partial}{\partial x}\nabla f.$$

15.11.3 Vector to Vector

For $f : \mathbf{R}^n \to \mathbf{R}^m$, the partial derivative is

$$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_0}{\partial x_0} & \cdots & \frac{\partial f_0}{\partial x_{n-1}} \\ \vdots & & \vdots \\ \frac{\partial f_{m-1}}{\partial x_0} & \cdots & \frac{\partial f_{m-1}}{\partial x_{n-1}} \end{bmatrix} \in \mathbf{R}^{m\times n},$$

where

$$\frac{\partial f}{\partial x} = \text{row}\left\{\nabla f_i^\intercal\right\}_{i=0}^{m-1}.$$

Note:

$$\text{vec}\left(\frac{\partial f}{\partial x}\right) = \text{row}\left\{\nabla f_i\right\}_{i=0}^{m-1} \in \mathbf{R}^{mn}.$$

The second partial is given by

$$\left\{\frac{\partial}{\partial x_j}\frac{\partial f}{\partial x}\right\}_{j=0}^{n-1} = \left\{\text{row}\left\{\frac{\partial}{\partial x_j}\nabla f_i^\intercal\right\}_{i=0}^{m-1}\right\}_{j=0}^{n-1} \in \mathbf{R}^{m\times n\times n}.$$

Note:

$$\frac{\partial}{\partial x}\text{vec}\left(\frac{\partial f}{\partial x}\right) = \text{row}\left\{\nabla^2 f_i\right\}_{i=0}^{m-1} \in \mathbf{R}^{mn\times n}.$$

15.11.4 Vector to Scalar to Scalar

For $h = g(f)$, with $f : \mathbf{R}^n \to \mathbf{R}$ and $g : \mathbf{R} \to \mathbf{R}$,

$$\frac{\partial h}{\partial x} = g'(f)\frac{\partial f}{\partial x}.$$

The gradient is

$$\nabla h = g'(f)\nabla f.$$

The Hessian is

$$\nabla^2 h = g'(f)\nabla^2 f + g''(f)\nabla f \nabla f^\intercal.$$

15.11.5 Vector to Vector to Scalar

For $h = g(f)$, with $f : \mathbf{R}^n \to \mathbf{R}^m$ and $g : \mathbf{R}^m \to \mathbf{R}$,

$$\frac{\partial h}{\partial x} = \frac{\partial g}{\partial f}\frac{\partial f}{\partial x},$$

where

$$\frac{\partial g}{\partial f} \in \mathbf{R}^{1\times m}, \quad \frac{\partial f}{\partial x} \in \mathbf{R}^{m\times n}.$$

The gradient is

$$\begin{aligned}\nabla h &= \mathrm{col}\{\nabla f_i\}_{i=0}^{m-1}\left(\frac{\partial g}{\partial f}\right)^\intercal \\ &= \left(\frac{\partial g}{\partial f}\otimes \boldsymbol{I}_n\right)\mathrm{row}\{\nabla f_i\}_{i=0}^{m-1}.\end{aligned}$$

The Hessian is

$$\nabla^2 h = \mathrm{col}\{\nabla f_i\}_{i=0}^{m-1}\frac{\partial}{\partial x}\left(\frac{\partial g}{\partial f}\right)^\intercal + \left(\frac{\partial g}{\partial f}\otimes \boldsymbol{I}_n\right)\mathrm{row}\{\nabla^2 f_i\}_{i=0}^{m-1}.$$

15.11.6 Vector to Vector to Repeated Scalar

For $f : \mathbf{R}^n \to \mathbf{R}^m$ and $g : \mathbf{R} \to \mathbf{R}$, let

$$h = \mathrm{row}\{g(f_i)\}_{i=0}^{m-1}.$$

The partial is

$$\frac{\partial h}{\partial x} = \mathrm{diag}\{g'(f_i)\}_{i=0}^{m-1}\frac{\partial f}{\partial x}$$

with

$$\begin{aligned}\mathrm{vec}\left(\frac{\partial h}{\partial x}\right) &= \mathrm{row}\{\nabla h_i\}_{i=0}^{m-1} \\ &= \left(\mathrm{diag}\{g'(f_i)\}_{i=0}^{m-1}\otimes \boldsymbol{I}_n\right)\mathrm{row}\{\nabla f_i\}_{i=0}^{m-1} \\ &= \left(\boldsymbol{I}_m \otimes \frac{\partial f}{\partial x}\right)\mathrm{vec}\left(\mathrm{diag}\{g'(f_i)\}_{i=0}^{m-1}\right).\end{aligned}$$

The second partial is

$$\left\{\frac{\partial}{\partial x_j}\frac{\partial h}{\partial x}\right\}_{j=0}^{n-1} = \left\{\mathrm{diag}\left\{g''(f_i)\frac{\partial f_i}{\partial x_j}\right\}_{i=0}^{m-1}\frac{\partial f}{\partial x} + \mathrm{diag}\left\{g'(f_i)\right\}_{i=0}^{m-1}\frac{\partial}{\partial x_j}\frac{\partial f}{\partial x}\right\}_{j=0}^{n-1} \in \mathbf{R}^{m\times n\times n},$$

with

$$\begin{aligned}\frac{\partial}{\partial x}\mathrm{vec}\left(\frac{\partial h}{\partial x}\right) &= \mathrm{row}\{\nabla^2 h_i\}_{i=0}^{m-1} \\ &= \mathrm{row}\{g'(f_i)\nabla^2 f_i + g''(f_i)\nabla f_i \nabla f_i^\intercal\}_{i=0}^{m-1}.\end{aligned}$$

References

[1] Convex Optimization
S. Boyd, L. Vandenberghe
http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

[2] ADMM
S. Boyd
https://stanford.edu/~boyd/admm.html

[3] Model Predictive Control
S. Boyd
https://stanford.edu/class/ee364b/lectures/mpc_slides.pdf

[4] ℓ_1-MAGIC
https://statweb.stanford.edu/~candes/l1magic/

[5] L1-L2 Optimization in Signal and Image Processing
M. Zibulevsky, M. Elad
https://ieeexplore.ieee.org/document/5447114#full-text-section

[6] ADMM: Alternating Direction Method of Multipliers
S. Boyd
http://stanford.edu/~boyd//papers/pdf/admm_slides.pdf

[7] Beyond Moore–Penrose: Sparse Pseudoinverse
I. Dokmanic, M. Kolundzija, M. Vetterli
https://infoscience.epfl.ch/record/182698/files/06638923.pdf

[8] Quadratic Programming in Python
S. Caron
https://scaron.info/blog/quadratic-programming-in-python.html

[9] Python: quadprog 0.1.6
https://pypi.org/project/quadprog/

[10] Matlab: linprog
https://www.mathworks.com/help/optim/ug/linprog.html

[11] Matlab: quadprog
https://www.mathworks.com/help/optim/ug/quadprog.html

[12] Matlab: fmincon
https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html

[13] C++: CGAL 4.13 - Linear and Quadratic Programming Solver
K. Fischer, B. Gärtner, S. Schönherr, F. Wessendorp
https://doc.cgal.org/latest/QP_solver/index.html

[14] JAVA: QuadraticProgramming
https://docs.roguewave.com/imsl/java/5.0.1/api/com/imsl/math/QuadraticProgramming.html

[15] A Lecture on Model Predictive Control
J. H. Lee
http://cepac.cheme.cmu.edu/pasilectures/lee/LecturenoteonMPC-JHL.pdf

[16] Chance-Constrained Optimal Path Planning With Obstacles
L. Blackmore, M. Ono, B. C. Williams
https://ieeexplore.ieee.org/abstract/document/5970128

[17] Chance-constrained dynamic programming with application to risk-aware robotic space exploration
M. Ono, M. Pavone, Y. Kuwata, J. Balaram
https://link.springer.com/article/10.1007%2Fs10514-015-9467-7

[18] Online Planning for Autonomous Running Jumps Over Obstacles in High-Speed Quadrupeds
H. W. Parl, P. M. Wensing, S. Kim
https://dspace.mit.edu/handle/1721.1/97236
https://youtu.be/_luhn7TLfWU

[19] A numerically stable dual method for solving strictly convex quadratic programs
D. Goldfarb, A. U. Idnani
https://www.semanticscholar.org/paper/A-numerically-stable-dual-method-for-solving-convex-Goldfarb-Idnani/d1984defd4ccd17ff944219ebd34420e3fb78239

[20] Quadrocopter pole acrobatics
D. Brescianini, M. Hehn, R. D'Andrea
https://ieeexplore.ieee.org/document/6696851
https://youtu.be/XxFZ-VStApo

[21] Iterative learning of feed-forward corrections for high-performance tracking
F. L. Mueller, A. P. Schoellig, R. D'Andrea
https://ieeexplore.ieee.org/document/6385647

[22] Linear programming support vector machines
W. Zhou, L. Zhang, L. Jiao
https://www.sciencedirect.com/science/article/abs/pii/S0031320301002102

[23] Efficient Large Scale Linear Programming Support Vector Machines
S. Sra
https://link.springer.com/content/pdf/10.1007/11871842_78.pdf

[24] Underactuated Robotics
R. Tedrake
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-832-underactuated-robotics-spring-2009/index.htm

[25] Sparse Optimization Methods
S. Wright
http://pages.cs.wisc.edu/~swright/talks/sjw-toulouse.pdf

[26] Nonlinear Optimization for Optimal Control
P. Abbeel
https://people.eecs.berkeley.edu/~pabbeel/cs287-fa12/slides/NonlinearOptimizationForOptimalControl.pdf

[27] Model-predictive control with stochastic collision avoidance using Bayesian policy optimization
O. Andersson, M. Wzorek, P. Rudol, P. Doherty
https://ieeexplore.ieee.org/abstract/document/7487661

[28] Constrained Differential Optimization
J. C. Platt, A. H. Barr
https://papers.nips.cc/paper/4-constrained-differential-optimization.pdf

[29] Sparse Linear Programming via Primal and Dual Augmented Coordinate Descent
I. E.-H. Yen, K. Zhong, C.-J. Hsieh, P. K. Ravikumar, I. S. Dhillon
http://papers.nips.cc/paper/5917-sparse-linear-programming-via-primal-and-dual-augmented-coordinate-descent

[30] Numerical Optimization for Robotics
http://roboptim.net/index.html

[31] Ceres Solver
http://ceres-solver.org/

[32] Karush-Kuhn-Tucker conditions
G. Gordon, R. Tibshirani
http://www.cs.cmu.edu/~ggordon/10725-F12/slides/16-kkt.pdf

[33] Derivation of the soft thresholding operator
A. Ang
https://angms.science/doc/CVX/ISTA0.pdf

[34] Introduction to Machine Learning (2nd ed.)
E. Alpaydin
https://www.amazon.com/Introduction-Machine-Learning-Adaptive-Computation/dp/026201243X

[35] CVXOPT: Python Software for Convex Optimization
https://cvxopt.org/
https://courses.csail.mit.edu/6.867/wiki/images/a/a7/Qp-cvxopt.pdf

[36] Optimization by Vector Space Methods
D. G. Luenberger
https://www.amazon.com/Optimization-Vector-Space-Methods-Luenberger/dp/047118117X

[37] Solvers
https://en.wikipedia.org/wiki/Quadratic_programming#Solvers_and_scripting_(programming)_languages

[38] The Matrix cookbook
K. B. Petersen, M. S. Pedersen
https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

[39] A geometric interpretation of v-SVM classifiers
D. J. Crisp, C. J. C. Burges
http://papers.nips.cc/paper/1687-a-geometric-interpretation-of-v-svm-classifiers.pdf

[40] Support Vector Machines as Probabilistic Models
V. Franc, A. Zien, B. Scholkopf
http://is.tuebingen.mpg.de/fileadmin/user_upload/files/publications/2011/ICML-2011-Franc.pdf

[41] A Gentle Introduction to the Rectified Linear Unit (ReLU) for Deep Learning Neural Networks
J. Brownlee
https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[42] Semidefinite Programming Relaxations and Algebraic Optimization in Control
http://www.mit.edu/~parrilo/cdc03_workshop/

[43] ADMM for Efficient Deep Learning with Global Convergence
https://arxiv.org/abs/1905.13611

[44] OSQP: Model predictive control (MPC)
https://osqp.org/docs/examples/mpc.html

[45] MPC Pytorch
https://locuslab.github.io/mpc.pytorch/

[46] git: qpsolves
https://github.com/stephane-caron/qpsolvers

[47] Generative Adversarial Nets
I. J. Goodfellow, H. Bouget-Abadie, B. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio
http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[48] World Models
D. Ha, J. Schmidhuber
https://arxiv.org/abs/1803.10122
https://worldmodels.github.io/

[49] Image–to–Image Translation with Conditional Adversarial Networks
P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros
https://arxiv.org/abs/1611.07004

[50] A Style-Based Generator Architecture for Generative Adversarial Networks
T. Karras, S. Laine, T. Aila
https://arxiv.org/abs/1812.04948
https://github.com/NVlabs/stylegan

[51] MorphNet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks
A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-H. Yang, E. Choi
https://arxiv.org/pdf/1711.06798.pdf
https://github.com/google-research/morph-net

[52] You Only Look Once
https://arxiv.org/pdf/1506.02640.pdf
https://pjreddie.com/darknet/yolo/

[53] CycleGAN
H.-Y. Zhu, T. Park, P. Isola, A. A. Efros
https://arxiv.org/pdf/1703.10593.pdf
https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

[54] Machine Learning
T. M. Mitchell