Sequential Convex Programming

• sequential convex programming
• alternating convex optimization
• convex-concave procedure
Methods for nonconvex optimization problems

- **convex optimization methods** are (roughly) always global, always fast

- for general nonconvex problems, we have to give up one
  - **local optimization methods** are fast, but need not find global solution (and even when they do, cannot certify it)
  - **global optimization methods** find global solution (and certify it), but are not always fast (indeed, are often slow)

- **this lecture**: local optimization methods that are based on solving a sequence of convex problems
Sequential convex programming (SCP)

- a local optimization method for nonconvex problems that leverages convex optimization
  - convex portions of a problem are handled ‘exactly’ and efficiently

- SCP is a heuristic
  - it can fail to find optimal (or even feasible) point
  - results can (and often do) depend on starting point
    (can run algorithm from many initial points and take best result)

- SCP often works well, i.e., finds a feasible point with good, if not optimal, objective value
Problem

we consider nonconvex problem

minimize \( f_0(x) \)
subject to \( f_i(x) \leq 0, \quad i = 1, \ldots, m \)
\( h_j(x) = 0, \quad j = 1, \ldots, p \)

with variable \( x \in \mathbb{R}^n \)

\begin{itemize}
\item \( f_0 \) and \( f_i \) (possibly) nonconvex
\item \( h_j \) (possibly) non-affine
\end{itemize}
Basic idea of SCP

- maintain estimate of solution $x^{(k)}$, and convex trust region $\mathcal{T}^{(k)} \subset \mathbb{R}^n$
- form convex approximation $\hat{f}_i$ of $f_i$ over trust region $\mathcal{T}^{(k)}$
- form affine approximation $\hat{h}_i$ of $h_i$ over trust region $\mathcal{T}^{(k)}$
- $x^{(k+1)}$ is optimal point for approximate convex problem

$$\begin{aligned}
\text{minimize} & \quad \hat{f}_0(x) \\
\text{subject to} & \quad \hat{f}_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad \hat{h}_i(x) = 0, \quad i = 1, \ldots, p \\
& \quad x \in \mathcal{T}^{(k)}
\end{aligned}$$
Trust region

• typical trust region is box around current point:

\[ T^{(k)} = \{ x \mid |x_i - x_i^{(k)}| \leq \rho_i, \ i = 1, \ldots, n \} \]

• if \( x_i \) appears only in convex inequalities and affine equalities, can take \( \rho_i = \infty \)
Affine and convex approximations via Taylor expansions

• (affine) first order Taylor expansion:

\[ \hat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) \]

• (convex part of) second order Taylor expansion:

\[ \hat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + \left( \frac{1}{2} \right)(x - x^{(k)})^T P(x - x^{(k)}) \]

\[ P = \left( \nabla^2 f(x^{(k)}) \right)_+, \text{ PSD part of Hessian} \]

• give local approximations, which don’t depend on trust region radii \( \rho_i \)
Particle method

- particle method:
  - choose points \( z_1, \ldots, z_K \in \mathcal{T}^{(k)} \)
    (e.g., all vertices, some vertices, grid, random, \ldots)
  - evaluate \( y_i = f(z_i) \)
  - fit data \((z_i, y_i)\) with convex (affine) function
    (using convex optimization)

- advantages:
  - handles nondifferentiable functions, or functions for which evaluating
    derivatives is difficult
  - gives regional models, which depend on current point and trust
    region radii \( \rho_i \)
Fitting affine or quadratic functions to data

fit convex quadratic function to data \((z_i, y_i)\)

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{K} \left( (z_i - x^{(k)})^T P(z_i - x^{(k)}) + q^T (z_i - x^{(k)}) + r - y_i \right)^2 \\
\text{subject to} & \quad P \succeq 0
\end{align*}
\]

with variables \(P \in S^n, q \in \mathbb{R}^n, r \in \mathbb{R}\)

- can use other objectives, add other convex constraints
- no need to solve exactly
- this problem is solved for each nonconvex constraint, each SCP step
Quasi-linearization

• a cheap and simple method for affine approximation

• write \( h(x) \) as \( A(x)x + b(x) \) (many ways to do this)

• use \( \hat{h}(x) = A(x^{(k)})x + b(x^{(k)}) \)

• example:

\[
  h(x) = (1/2)x^T P x + q^T x + r = ((1/2)P x + q)^T x + r
\]

• \( \hat{h}_{q1}(x) = ((1/2)Px^{(k)} + q)^T x + r \)

• \( \hat{h}_{tay}(x) = (Px^{(k)} + q)^T (x - x^{(k)}) + r \)
Example

• nonconvex QP

\[
\begin{align*}
\text{minimize} & \quad f(x) = \frac{1}{2} x^T P x + q^T x \\
\text{subject to} & \quad \|x\|_\infty \leq 1
\end{align*}
\]

with \( P \) symmetric but not PSD

• use approximation

\[
f(x^{(k)}) + (Px^{(k)} + q)^T (x - x^{(k)}) + \frac{1}{2}(x - x^{(k)})^T P_+ (x - x^{(k)})
\]
• example with $x \in \mathbb{R}^{20}$

• SCP with $\rho = 0.2$, started from 10 different points

• runs typically converge to points between $-60$ and $-50$

• dashed line shows lower bound on optimal value $\approx -66.5$
Lower bound via Lagrange dual

- write constraints as $x_i^2 \leq 1$ and form Lagrangian

\[
L(x, \lambda) = \frac{1}{2}x^T P x + q^T x + \sum_{i=1}^{n} \lambda_i (x_i^2 - 1)
\]

\[
= \frac{1}{2}x^T (P + \text{diag}(\lambda)) x + q^T x
\]

- $g(\lambda) = -(1/2)q^T (P + \text{diag}(\lambda))^{-1} q$; need $P + \text{diag}(\lambda) \succ 0$

- solve dual problem to get best lower bound:

\[
\begin{align*}
\text{maximize} & \quad -(1/2)q^T (P + \text{diag}(\lambda))^{-1} q \\
\text{subject to} & \quad \lambda \succeq 0
\end{align*}
\]
Some (related) issues

• approximate convex problem can be infeasible

• how do we evaluate progress when \(x^{(k)}\) isn’t feasible?
  need to take into account
  – objective \(f_0(x^{(k)})\)
  – inequality constraint violations \(f_i(x^{(k)})_+\)
  – equality constraint violations \(|h_i(x^{(k)})|\)

• controlling the trust region size
  – \(\rho\) too large: approximations are poor, leading to bad choice of \(x^{(k+1)}\)
  – \(\rho\) too small: approximations are good, but progress is slow
Exact penalty formulation

• instead of original problem, we solve unconstrained problem

\[
\text{minimize} \quad \phi(x) = f_0(x) + \lambda \left( \sum_{i=1}^{m} f_i(x) + \sum_{i=1}^{p} |h_i(x)| \right)
\]

where \( \lambda > 0 \)

• for \( \lambda \) large enough, minimizer of \( \phi \) is solution of original problem

• for SCP, use convex approximation

\[
\hat{\phi}(x) = \hat{f}_0(x) + \lambda \left( \sum_{i=1}^{m} \hat{f}_i(x) + \sum_{i=1}^{p} |\hat{h}_i(x)| \right)
\]

• approximate problem always feasible
Trust region update

- judge algorithm progress by decrease in $\phi$, using solution $\tilde{x}$ of approximate problem

- decrease with approximate objective: $\hat{\delta} = \phi(x^{(k)}) - \hat{\phi}(\tilde{x})$ (called predicted decrease)

- decrease with exact objective: $\delta = \phi(x^{(k)}) - \phi(\tilde{x})$

- if $\delta \geq \alpha \hat{\delta}$, $\rho^{(k+1)} = \beta^{\text{succ}} \rho^{(k)}$, $x^{(k+1)} = \tilde{x}$
  \((\alpha \in (0, 1), \beta^{\text{succ}} \geq 1; \text{typical values } \alpha = 0.1, \beta^{\text{succ}} = 1.1)\)

- if $\delta < \alpha \hat{\delta}$, $\rho^{(k+1)} = \beta^{\text{fail}} \rho^{(k)}$, $x^{(k+1)} = x^{(k)}$
  \((\beta^{\text{fail}} \in (0, 1); \text{typical value } \beta^{\text{fail}} = 0.5)\)

- interpretation: if actual decrease is more (less) than fraction $\alpha$ of predicted decrease then increase (decrease) trust region size
Nonlinear optimal control

- 2-link system, controlled by torques $\tau_1$ and $\tau_2$ (no gravity)
• dynamics given by \( M(\theta)\ddot{\theta} + W(\theta, \dot{\theta})\dot{\theta} = \tau \), with

\[
M(\theta) = \begin{bmatrix}
(m_1 + m_2)l_1^2 & m_2l_1l_2(s_1s_2 + c_1c_2) \\
m_2l_1l_2(s_1s_2 + c_1c_2) & m_2l_2^2
\end{bmatrix}
\]

\[
W(\theta, \dot{\theta}) = \begin{bmatrix}
0 & m_2l_1l_2(s_1c_2 - c_1s_2)\dot{\theta}_2 \\
m_2l_1l_2(s_1c_2 - c_1s_2)\dot{\theta}_1 & 0
\end{bmatrix}
\]

\( s_i = \sin \theta_i, \ c_i = \cos \theta_i \)

• nonlinear optimal control problem:

\[
\text{minimize} \quad J = \int_0^T \|\tau(t)\|_2^2 dt
\]

\[
\text{subject to} \quad \theta(0) = \theta_{\text{init}}, \quad \dot{\theta}(0) = 0, \quad \theta(T) = \theta_{\text{final}}, \quad \dot{\theta}(T) = 0
\]

\[
\|\tau(t)\|_\infty \leq \tau_{\text{max}}, \quad 0 \leq t \leq T
\]
Discretization

- discretize with time interval $h = T/N$
- $J \approx h \sum_{i=1}^{N} \| \tau_i \|_2^2$, with $\tau_i = \tau(ih)$
- approximate derivatives as
  \[
  \dot{\theta}(ih) \approx \frac{\theta_{i+1} - \theta_{i-1}}{2h}, \quad \ddot{\theta}(ih) \approx \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2}
  \]
- approximate dynamics as set of nonlinear equality constraints:
  \[
  M(\theta_i) \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2} + W \left( \theta_i, \frac{\theta_{i+1} - \theta_{i-1}}{2h} \right) \frac{\theta_{i+1} - \theta_{i-1}}{2h} = \tau_i
  \]
- $\theta_0 = \theta_1 = \theta_{\text{init}}$; $\theta_N = \theta_{N+1} = \theta_{\text{final}}$
• discretized nonlinear optimal control problem:

minimize \[ h \sum_{i=1}^{N} \| \tau_i \|_2^2 \]
subject to \[ \theta_0 = \theta_1 = \theta_{\text{init}}, \quad \theta_N = \theta_{N+1} = \theta_{\text{final}} \]
\[ \| \tau_i \|_\infty \leq \tau_{\text{max}}, \quad i = 1, \ldots, N \]
\[ M(\theta_i) \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2} + W \left( \theta_i, \frac{\theta_{i+1} - \theta_{i-1}}{2h} \right) \frac{\theta_{i+1} - \theta_{i-1}}{2h} = \tau_i \]

• replace equality constraints with quasilinearized versions

\[ M(\theta_i^{(k)}) \frac{\theta_{i+1}^{(k)} - 2\theta_i^{(k)} + \theta_{i-1}^{(k)}}{h^2} + W \left( \theta_i^{(k)}, \frac{\theta_{i+1}^{(k)} - \theta_{i-1}^{(k)}}{2h} \right) \frac{\theta_{i+1}^{(k)} - \theta_{i-1}^{(k)}}{2h} = \tau_i \]

• trust region: only on \( \theta_i \)

• initialize with \( \theta_i = \left( \frac{(i - 1)}{(N - 1)} \right) (\theta_{\text{final}} - \theta_{\text{init}}), \quad i = 1, \ldots, N \)
Numerical example

• $m_1 = 1, m_2 = 5, l_1 = 1, l_2 = 1$

• $N = 40, T = 10$

• $\theta_{\text{init}} = (0, -2.9), \theta_{\text{final}} = (3, 2.9)$

• $\tau_{\text{max}} = 1.1$

• $\alpha = 0.1, \beta^{\text{succ}} = 1.1, \beta^{\text{fail}} = 0.5, \rho^{(1)} = 90^\circ$

• $\lambda = 2$
SCP progress

\[ \phi(x(k)) \]

Prof. S. Boyd, EE364b, Stanford University
Convergence of $J$ and torque residuals
Predicted and actual decreases in $\phi$
Trajectory plan

\begin{align*}
&\tau_1(t) \\
&\tau_2(t) \\
&\theta_1(t) \\
&\theta_2(t)
\end{align*}
‘Difference of convex’ programming

• express problem as

\[
\begin{align*}
\text{minimize} & \quad f_0(x) - g_0(x) \\
\text{subject to} & \quad f_i(x) - g_i(x) \leq 0, \quad i = 1, \ldots, m
\end{align*}
\]

where \( f_i \) and \( g_i \) are convex

• \( f_i - g_i \) are called ‘difference of convex’ functions

• problem is sometimes called ‘difference of convex programming’
Convex-concave procedure

- obvious convexification at $x^{(k)}$: replace $f(x) - g(x)$ with
  \[
  \hat{f}(x) = f(x) - g(x^{(k)}) - \nabla g(x^{(k)})^T (x - x^{(k)})
  \]

- since $\hat{f}(x) \geq f(x)$ for all $x$, no trust region is needed
  - true objective at $\tilde{x}$ is better than convexified objective
  - true feasible set contains feasible set for convexified problem

- SCP sometimes called ‘convex-concave procedure’
Example (BV §7.1)

- given samples $y_1, \ldots, y_N \in \mathbb{R}^n$ from $\mathcal{N}(0, \Sigma_{\text{true}})$
- negative log-likelihood function is

$$f(\Sigma) = \log \det \Sigma + \text{Tr}(\Sigma^{-1}Y), \quad Y = (1/N) \sum_{i=1}^{N} y_i y_i^T$$

(dropping a constant and positive scale factor)

- ML estimate of $\Sigma$, with prior knowledge $\Sigma_{ij} \geq 0$:

$$\begin{align*}
\text{minimize} & \quad f(\Sigma) = \log \det \Sigma + \text{Tr}(\Sigma^{-1}Y) \\
\text{subject to} & \quad \Sigma_{ij} \geq 0, \quad i, j = 1, \ldots, n
\end{align*}$$

with variable $\Sigma$ (constraint $\Sigma \succ 0$ is implicit)
• first term in $f$ is concave; second term is convex
• linearize first term in objective to get

$$\hat{f}(\Sigma) = \log \det \Sigma^{(k)} + \text{Tr} \left( (\Sigma^{(k)})^{-1} (\Sigma - \Sigma^{(k)}) \right) + \text{Tr}(\Sigma^{-1}Y)$$
Numerical example

convergence of problem instance with \( n = 10, \ N = 15 \)
Alternating convex optimization

• given nonconvex problem with variable \((x_1, \ldots, x_n) \in \mathbb{R}^n\)

• \(I_1, \ldots, I_k \subset \{1, \ldots, n\}\) are index subsets with \(\bigcup_j I_j = \{1, \ldots, n\}\)

• suppose problem is convex in subset of variables \(x_i, i \in I_j\), when \(x_i, i \notin I_j\) are fixed

• alternating convex optimization method: cycle through \(j\), in each step optimizing over variables \(x_i, i \in I_j\)

• special case: bi-convex problem
  – \(x = (u, v)\); problem is convex in \(u (v)\) with \(v (u)\) fixed
  – alternate optimizing over \(u\) and \(v\)
Nonnegative matrix factorization

• NMF problem:

\[
\begin{align*}
\text{minimize} \quad & \| A - XY \|_F \\
\text{subject to} \quad & X_{ij}, Y_{ij} \geq 0
\end{align*}
\]

variables \( X \in \mathbb{R}^{m \times k} \), \( Y \in \mathbb{R}^{k \times n} \), data \( A \in \mathbb{R}^{m \times n} \)

• difficult problem, except for a few special cases (e.g., \( k = 1 \))

• alternating convex optimization: solve QPs to optimize over \( X \), then \( Y \), then \( X \ldots \)
Example

- convergence for example with \( m = n = 50, \ k = 5 \)
  (five starting points)