17 Constrained Optimization

best frequently has constraints

nutrition

groceries $\vec{g} \geq 0$

prices $\vec{p}$

price $\min \vec{g} \cdot \vec{p}$

minimum requirements $\vec{m}$

nutrition value $N$

$N \cdot \vec{g} \geq \vec{m}$

defines linear program, LP

price may be a function of quantity, not linear

quadratic objective, quadratic program, QP

general case mathematical program

portfolios, routing airplanes, running a factory

program as plan, not computer program, but can be same

electrical networks [Dennis, 1958]


flow control [Low et al., 2002]

layering [Chiang et al., 2007]

sorting

variables $\vec{x}$, objective minimize $f(\vec{x})$, constraints $\vec{c}(\vec{x})$

max = -min

slack variables to convert inequality to equality

$$c(\vec{x}) \geq 0$$ (17.1)

replace with

$$c(\vec{x}) - s = 0$$

$$s \geq 0$$ (17.2)

combinatorial $x$ equals 1 or -1 can be relaxed as algebraic constraint $(x^2 - 1)^2 = 0$

L1 norm

$$|\vec{x}|_1 = \sum_i |x_i|$$ (17.3)
17.1 Lagrange Multipliers

compressed sensing, sparsity
non-differentiable
[Schmidt et al., 2007]
\[ (x)_+ = \max(x, 0) \]
\[ (x)_- = \max(-x, 0) \]

\[ |x| = (x)_- + (x)_+ \] (17.4)

can be relaxed

\[ |x| \approx |x|_\alpha \]
\[ = \frac{1}{\alpha} \left[ \log(1 + e^{-\alpha x}) + \log(1 + e^{\alpha x}) \right] \] (17.5)

\[ \frac{d|x|_\alpha}{dx} = \frac{1}{1 + e^{-\alpha x}} - \frac{1}{1 + e^{\alpha x}} \] (17.6)

\[ \frac{d^2|x|_\alpha}{dx^2} = \frac{2\alpha e^{\alpha x}}{(1 + e^{\alpha x})^2} \] (17.7)

minimize for increasing \( \alpha \)

17.1 LAGRANGE MULTIPLIERS

single equality constraint \( c(\vec{x}) = 0 \)
step in direction \( \vec{d} \) to minimize \( f \) while satisfying the constraint

\[ 0 = c(\vec{x} + \vec{\delta}) \]
\[ \approx c(\vec{x}) + \nabla c \cdot \vec{\delta} \]
\[ = \nabla c \cdot \vec{\delta} \] (17.8)

step also minimizes \( f \)

\[ 0 > f(\vec{x} + \vec{\delta}) - f(\vec{x}) \]
\[ \approx f(\vec{x}) + \nabla f \cdot \vec{\delta} - f(\vec{x}) \]
\[ = \nabla f \cdot \vec{\delta} \] (17.9)

if \( \nabla c(\vec{x}) \) and \( \nabla f(\vec{x}) \) aligned not possible to find a direction, hence \( \vec{x} \) is a local minimizer
define *Lagrangian*

\[ \mathcal{L} = f(\vec{x}) - \lambda c(\vec{x}) \] (17.10)

solve for
\[ 0 = \nabla \mathcal{L} = \nabla f - \lambda \nabla c \]  
\hspace{1cm} (17.11)

multiple constraints
linear combination

\[ \nabla f(\bar{x}) = \sum_i \lambda_i \nabla c_i(\bar{x}) \]  
\hspace{1cm} (17.12)

\[ f(\bar{x}) = \sum_i \lambda_i c_i(\bar{x}) \]  
\hspace{1cm} (17.13)

solving gives \( \bar{x}(\bar{\lambda}) \), substitute into constraints to find \( \bar{\lambda} \)

inequality constraint

\[ 0 \leq c(\bar{x} + \delta) \approx c(\bar{x}) + \nabla c \cdot \delta \]  
\hspace{1cm} (17.14)

if constraint not active \( (c > 0) \), can just do gradient descent \( \delta = -\alpha \nabla f \)
for an active constraint \( \nabla f \cdot \delta < 0 \) and \( \nabla c \cdot \delta \geq 0 \)
define half-planes
no intersection if point in same direction \( \nabla f = \lambda \nabla c \)
same condition, but now \( \lambda \geq 0 \)

\section*{17.2 Optimality}

first-order condition
equality constraints \( c_i(\bar{x}), i \in \mathcal{E} \)
inequality constraints \( c_i(\bar{x}), i \in \mathcal{I} \)
inactive constraint \( \lambda_i = 0 \)
complementarity: \( \lambda_i c_i = 0 \): Lagrange multiplier only non-zero when constraint is active, otherwise reduces to gradient descent

\[ \nabla_{\bar{x}} \mathcal{L}(\bar{x}, \bar{\lambda}) = 0 \]
\[ c_i(\bar{x}) = 0 \quad (i \in \mathcal{E}) \]
\[ c_i(\bar{x}) \geq 0 \quad (i \in \mathcal{I}) \]
\[ \lambda_i \geq 0 \quad (i \in \mathcal{I}) \]
\[ \lambda_i c_i(x) = 0 \]  
\hspace{1cm} (17.15)

Karush-Kuhn-Tucker (KKT) conditions
necessary, not sufficient
second order condition: positive definite Lagrangian Hessian
sensitivity
replace \( c(x) = 0 \) with \( c(x) = \epsilon \)
minimizer \( \vec{x} \) goes to \( \vec{x}_\epsilon \)

\[
f(\vec{x}_\epsilon) - f(\vec{x}) \approx \nabla f \cdot (\vec{x}_\epsilon - \vec{x})
= \lambda \nabla c \cdot (\vec{x}_\epsilon - \vec{x})
\approx \lambda (c(\vec{x}_\epsilon) - c(\vec{x}))
= \lambda \epsilon
\]

\[
\frac{df}{d\epsilon} = \lambda
\tag{17.16}
\]

shadow prices: change in utility per change in constraint
\( \vec{x} \) primal \( \lambda \) dual
multi-objective
Pareto
not possible to improve one constraint without making others worse
defines Pareto frontier
can combine in multi-objective function with relative weights

17.3 SOLVERS

17.3.1 Penalty

penalty combine

\[
\mathcal{F} = f(\vec{x}) + \frac{\mu}{2} \sum_i c_i^2(\vec{x})
\tag{17.17}
\]

\[
\frac{\partial \mathcal{F}}{\partial x_j} = \frac{\partial f}{\partial x_j} + \mu \sum_i c_i \frac{\partial c_i}{\partial x_j}
\tag{17.18}
\]

\[
\mathcal{L} = f(\vec{x}) - \sum_i \lambda_i c_i(\vec{x})
\tag{17.19}
\]

\[
\frac{\partial \mathcal{L}}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_i \lambda_i \frac{\partial c_i}{\partial x_j}
\tag{17.20}
\]

effectively taking \( c_i = -\lambda_i / \mu \)
solving a different problem
driven to 0 as \( \mu \rightarrow \infty \)
small \( \mu \) may be unbounded
large \( \mu \) may be ill-conditioned
nonsmooth penalty

\[
\mathcal{F} = f(\vec{x}) + \mu \sum_{i \in E} |c_i(\vec{x})| + \mu \sum_{i \in I} [c_i(\vec{x})]_-
\tag{17.21}
\]
can be exact for large \( \mu \) [Nocedal & Wright, 2006]
non-differentiable
approximate (17.5)

### 17.3.2 Augmented Lagrangian

augmented Lagrangian

\[
\mathcal{L} = f(\bar{x}) - \sum_i \lambda_i c_i(\bar{x}) + \frac{\mu}{2} \sum_i c_i^2(\bar{x}) \tag{17.22}
\]

\[
\frac{\partial \mathcal{L}}{\partial x_j} = \frac{\partial f}{\partial x_j} - \sum_i \lambda_i \frac{\partial c_i}{\partial x_j} + \mu \sum_i c_i \frac{\partial c_i}{\partial x_j} \tag{17.23}
\]

\[
\lambda_i^* = \lambda_i - \mu c_i
\]

\[
c_i = (\lambda_i - \lambda_i^*)/\mu
\]

vanishes much faster, as Lagrange multiplier estimates converge

\[
\lambda_i^{(n+1)} = \lambda_i^{(n)} - \mu c_i
\]

minimize \( \bar{x} \), update \( \lambda \), increase \( \mu \)

### 17.3.3 Interior Point

interior point
basis largest, most efficient solvers
directly solve system of equations

\[
\min_{\bar{x}} f(\bar{x})
\]

\[
c_i(\bar{x}) = 0 \quad (i \in \mathcal{E})
\]

\[
c_i(\bar{x}) - s_i = 0 \quad (i \in \mathcal{I})
\]

\[
s_i \geq 0 \tag{17.24}
\]

KKT conditions, perturb from boundary

\[
\nabla f - \sum_i \lambda_i \nabla c_i(\bar{x}) = 0
\]

\[
c_i(\bar{x}) = 0 \quad (i \in \mathcal{E})
\]

\[
c_i(\bar{x}) - s_i = 0 \quad (i \in \mathcal{I})
\]

\[
\lambda_i s_i = \mu \quad (i \in \mathcal{I}) \tag{17.25}
\]

iterate Newton step on system, decrease \( \mu \)
same as barrier method

\[
\min_{\bar{x}, s_i} f(x) - \mu \sum_i \log s_i \quad (i \in \mathcal{I})
\]

\[
c_i(\bar{x}) = 0 \quad (i \in \mathcal{E})
\]
\[ c_i(\bar{x}) - s_i = 0 \quad (i \in \mathcal{I}) \]  
(17.26)

KKT condition for \( s_i \)

\[ \frac{1}{s_i} - \lambda_i = 0 \]  
(17.27)

\[ \lambda_i s_i = \mu \]  
(17.28)

### 17.4 SELECTED REFERENCES


Unusually clear coverage of a field full of unusually opaque books.

### 17.5 PROBLEMS

(17.1) Given a point \((x_0, y_0)\), analytically find the closest point on the line \(y = ax + b\) by minimizing the distance \(d^2 = (x_0 - x)^2 + (y_0 - y)^2\) subject to the constraint \(y - ax - b = 0\).

(17.2) Consider a set of \( N \) nodes that has each measured a quantity \( x_i \). The goal is to find the best estimate \( \bar{x} \) by minimizing

\[ \min_\bar{x} \sum_{i=1}^{N} (\bar{x} - x_i)^2, \]  
(17.29)

however each node \( i \) can communicate only with nodes \( j \) in its neighborhood \( j \in \mathcal{N}(i) \). This can be handled by having each node obtain a local estimate \( \bar{x}_i \), and introducing a consistency constraint \( c_{ij} = \bar{x}_i - \bar{x}_j = 0 \) \( \forall j \in \mathcal{N}(i) \).

(a) What is the Lagrangian?

(b) Find an update rule for the estimates \( \bar{x}_i \) by evaluating where the gradient of the Lagrangian vanishes.

(c) Find an update rule for the Lagrange multipliers by taking a Newton root-finding step on their associated constraints.

(17.3) Sorting can be written in terms of a permutation matrix \( P \) as \( \bar{s} = P \cdot \bar{u} \), where \( \bar{u} \) is a vector of unsorted numbers, \( \bar{s} \) are the sorted numbers, and each row and column of \( P \) has one 1 and the rest of the elements are 0. Defining the vector \( \bar{n} \) to be \( \{1, 2, \ldots\} \), sorting can be done by maximizing \( \bar{n} \cdot \bar{s} \). Solve this as a constrained optimization for a vector of random numbers.