13. Dual methods

- dual of convex problem with linear constraints
- differentiability of dual function
- dual decomposition
- rate control

Convex problem with linear constraints

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

($G \in \mathbf{R}^{m\times n}$, $A \in \mathbf{R}^{p\times n}$)

dual function

$$\begin{aligned} g(\lambda,\nu) &= \inf_{x\in\mathbf{dom}\, f} L(x,\lambda,\nu) \\ &= \inf_{x\in\mathbf{dom}\, f} \left(f(x) + (G^T\lambda + A^T\nu)^T x - h^T\lambda - b^T\nu\right) \\ &= -h^T\lambda - b^T\nu - f^*(-G^T\lambda - A^T\nu) \end{aligned}$$

$f^*(y) = \sup_{x\in\mathbf{dom}\, f}(x^T y - f(x))$ is the conjugate of f

this lecture: methods that solve dual by a first-order method

Dual feasibility

(λ, ν) is dual feasible if $\lambda \succeq 0$ and $L(x, \lambda, \nu)$ is bounded over $x \in \mathbf{dom}\, f$:

$$-G^T\lambda - A^T\nu \in \mathbf{dom}\, f^*, \qquad \lambda \succeq 0$$

- to solve dual problem when feasible set is not $\mathbf{R}_+^m \times \mathbf{R}^p$
  - make dual domain explicit ($e.g.$, when using gradient projection)
  - minimize L with a method that detects unboundedness, and use unbounded direction to generate cutting-plane for dual domain ($e.g.$, when using ACCPM)
- dual methods simplify if feasible set is $\mathbf{R}_+^m \times \mathbf{R}^p$

  some sufficient conditions:

  - $\mathbf{dom}\, f$ is bounded ($e.g.$, after addition of variable bounds)
  - f strongly convex ($e.g.$, after addition of a regularization term)

Subgradients of conjugate function

$$f^*(y) = \sup_{x \in \mathbf{dom}\, f} (x^T y - f(x))$$

from page 2–7, f^* is subdifferentiable at all points in $\mathbf{int}\,\mathbf{dom}\, f^*$

weak subgradient rule

if $\hat{x}$ maximizes $x^T\hat{y} - f(x)$ over $x \in \mathbf{dom}\, f$, then $\hat{x} \in \partial f^*(\hat{y})$

$$\begin{aligned} f^*(y) = \sup_{x \in \mathbf{dom}\, f} (x^T y - f(x)) \quad &\geq \quad \hat{x}^T y - f(\hat{x}) \\ &= \quad \hat{x}^T\hat{y} - f(\hat{x}) + \hat{x}^T(y - \hat{y}) \\ &= \quad f^*(\hat{y}) + \hat{x}^T(y - \hat{y}) \end{aligned}$$

Conjugate of closed functions

a convex function is **closed** if $\mathbf{epi}\, f$ is a closed set (see BV §A.3.3)

properties: for f closed and convex

- $f^{**} = f$: conjugate of conjugate of f is f (BV exercise 3.39)
- $y \in \partial f(x)$ if and only if $x \in \partial f^*(y)$

  proof: if $\hat{y} \in \partial f(\hat{x})$, then $x^T\hat{y} - f(x)$ reaches its maximum over x at $\hat{x}$; therefore $\hat{x} \in \partial f^*(\hat{y})$; this shows

  $$\hat{y} \in \partial f(\hat{x}) \quad \Longrightarrow \quad \hat{x} \in \partial f^*(\hat{y})$$

  reverse implication follows from $f^{**} = f$

Differentiability of conjugate

if f is closed and strictly convex, then f^* is differentiable on $\mathbf{int\,dom}\, f^*$

proof suppose $x_1, x_2 \in \partial f^*(\hat{y})$, and define $x_3 = (1/2)(x_1 + x_2)$

from previous page, $\hat{y} \in \partial f(x_1)$ and $\hat{y} \in \partial f(x_2)$; therefore

$$f^*(\hat{y}) = x_1^T\hat{y} - f(x_1), \qquad f^*(\hat{y}) = x_2^T\hat{y} - f(x_2)$$

and (by definition of f^*), $f^*(\hat{y}) \geq x_3^T\hat{y} - f(x_3)$

combining the three inequalities gives

$$f(\frac{1}{2}(x_1 + x_2)) \geq \frac{1}{2}(f(x_1) + f(x_2))$$

if f is strictly convex, this is only possible if $x_1 = x_2$

implication for dual methods

if f on page 13–2 is closed and strictly convex

- dual function is differentiable for $-G^T\lambda - A^T\nu \in \mathbf{int\,dom}\, f^*$
- dual gradient is given by

$$\begin{aligned}\nabla g_\lambda(\lambda, \nu) &= -h + G\,\nabla f^*(-G^T\lambda - A^T\nu) \\ \nabla g_\nu(\lambda, \nu) &= -b + A\,\nabla f^*(-G^T\lambda - A^T\nu)\end{aligned}$$

- to ensure differentiability of g, add small strictly convex term to f

Lipschitz continuity of dual gradient

from page 6–8: if f is strongly convex with constant μ, then

$$\|\nabla f^*(u) - \nabla f^*(v)\|_2 \leq \frac{1}{\mu}\|u - v\|_2$$

i.e., ∇f^* is Lipschitz continuous with constant $1/\mu$

implication for dual methods

if f on page 13–2 is strongly convex then ∇g is Lipschitz continuous with

$$L = \frac{\lambda_{\max}(G^TG + A^TA)}{\mu}$$

to ensure Lipschitz continuity of ∇g, add small strongly convex term to f

Dual decomposition

$$\begin{array}{ll} \text{minimize} & \sum_{j=1}^{r} f_j(x_j) \\ \text{subject to} & x_j \in \mathcal{C}_j, \quad j = 1, \ldots, r \\ & \sum_{j=1}^{r} G_j x_j \preceq h \end{array}$$

- objective and constraints are separable in r variables x_j
- last constraint is called a *complicating* (or *coupling*) constraint

dual function

$$g(\lambda) = \sum_{j=1}^{r} g_j(\lambda) - h^T\lambda, \qquad g_j(\lambda) \triangleq \inf_{x_j \in C_j} \left(f_j(x_j) + \lambda^T G_j x_j\right)$$

master problem

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{r} g_j(\lambda) - h^T\lambda \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

if $\lambda \succeq 0$ is the only constraint, this can be solved by (sub-)gradient projection, or Nesterov's accelerated gradient projection

subproblem

$$\begin{array}{ll} \text{minimize} & f_j(x_j) + \lambda^T G_j x_j \\ \text{subject to} & x_j \in C_j \end{array}$$

- optimal value is $g_j(\lambda)$
- if $\hat{x}_j$ solves the subproblem, then $-G_j\hat{x}_j$ is a subgradient of $-g_j$ at λ
- subproblems are independent and can be solved in parallel

Quadratic programming example

$$
\begin{array}{ll}
\text{minimize} & \sum_{j=1}^{r} (1/2)x_j^T P_j x_j + q_j^T x_j \\
\text{subject to} & A_j x_j \preceq b_j \\
& \sum_{j=1}^{p} G_j x_j \preceq h
\end{array}
$$

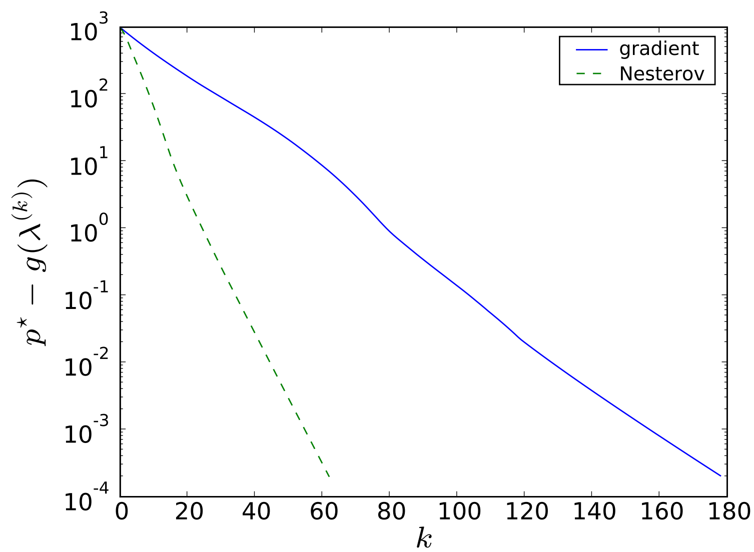
- $r = 10$; variables $x_j \in \mathbf{R}^{100}$; $A_j \in \mathbf{R}^{100\times 100}$, $G_j \in \mathbf{R}^{10\times 100}$
- $P_j \succ 0$; implies dual function has Lipschitz continuous gradient

subproblems are QPs

$$
\begin{array}{ll}
\text{minimize} & (1/2)x_j^T P_j x_j + (q_j + G_j^T \lambda)^T x_j \\
\text{subject to} & A_j x_j \preceq b_j
\end{array}
$$

gradient projection (lecture 5) and Nesterov's method (7-12)

- fixed step size (equal in the two methods)
- plot shows convergence of master problem



Network rate control

network flows

- n flows, with fixed routes, in a network with m links
- variable $f_j \geq 0$ denotes the rate of flow j
- flow utility is $U_j : \mathbf{R} \to \mathbf{R}$, concave, increasing

capacity constraints

- traffic t_i on link i is sum of flows passing through it
- $t = Rf$, where R is the routing matrix

$$R_{ij} = \begin{cases} 1 & \text{flow } j \text{ passes over link } i \\ 0 & \text{otherwise} \end{cases}$$

- link capacity constraint: $t \preceq c$

Rate control problem

$$\begin{array}{ll} \text{maximize} & U(f) = \sum_{j=1}^n U_j(f_j) \\ \text{subject to} & Rf \preceq c \end{array}$$

- convex problem
- dual decomposition gives decentralized method

Lagrangian (for minimizing $-U$)

$$\begin{aligned} L(f, \lambda) &= -U(f) + \lambda^T(Rf - c) \\ &= -\lambda^T c + \sum_{j=1}^n \left(-U_j(f_j) + (r_j^T\lambda) f_j\right) \end{aligned}$$

- λ_i is price (per unit flow) for using link i
- $r_j^T\lambda$ is the sum of prices along route j (r_j is jth column of R)

dual function

$$
\begin{aligned}
g(\lambda) &= -\lambda^T c + \sum_{j=1}^{n} \inf_{f_j}(-U_j(f_j) + (r_j^T\lambda) f_j) \\
&= -\lambda^T c - \sum_{j=1}^{n} (-U_j)^*(-r_j^T\lambda),
\end{aligned}
$$

dual rate control problem

$$
\begin{array}{ll}
\text{maximize} & -\lambda^T c - \sum_{j=1}^{n} (-U_j)^*(-r_j^T\lambda) \\
\text{subject to} & \lambda \succeq 0
\end{array}
$$

(Sub-)gradients of dual function

$$
\begin{aligned}
g(\lambda) &= -\lambda^T c - \sum_{j=1}^{n} \sup_{f_j}(U_j(f_j) - (r_j^T\lambda) f_j) \\
&= -\lambda^T c - \sum_{j=1}^{n} (-U_j)^*(-r_j^T\lambda),
\end{aligned}
$$

- subgradient (of negative dual)

$$
R\bar{f} - c \in \partial(-g)(\lambda) \quad \text{where} \quad \bar{f}_j = \operatorname{argmax}\left(U_j(f_j) - (r_j^T\lambda) f_j\right)
$$

  if U_j is strictly concave, this is a gradient

- $r_j^T\lambda$ is the sum of link prices along route j
- $c - R\bar{f}$ is vector of link capacity margins for flow $\bar{f}$

Dual decomposition rate control algorithm

given initial link price vector $\lambda \succ 0$ ($e.g.$, $\lambda = \mathbf{1}$).

repeat

1. sum link prices along each route: calculate $\Lambda_j = r_j^T \lambda$
2. optimize flows (separately) using flow prices:

$$f_j := \operatorname{argmax} \left(U_j(f_j) - \Lambda_j f_j \right)$$

3. calculate link capacity margins $s := c - Rf$
4. update link prices:

$$\lambda := (\lambda - ts)_+$$

   where t is the step size

Dual decomposition rate control algorithm

- decentralized:
  - links only need to know the flows that pass through them
  - flows only need to know prices on links they pass through
- prices converge to optimal; so do flows (since U is strictly concave)
- iterates can be (and often are) infeasible, $i.e.$, $Rf \not\preceq c$
(but we do have $Rf \preceq c$ in the limit)
- have upper bound $-g(\lambda)$ on optimal utility $U^\star$

Generating feasible flows

- define $\eta_i = t_i/c_i = (Rf)_i/c_i$
  - $\eta_i < 1$ means link i is under capacity
  - $\eta_i > 1$ means link i is over capacity
- define f^{feas} as

$$f_j^{\text{feas}} = \frac{f_j}{\max\{\eta_i \mid \text{flow } j \text{ passes over link } i\}}$$

- f^{feas} will be feasible, even if f is not
- finding f^{feas} is also decentralized

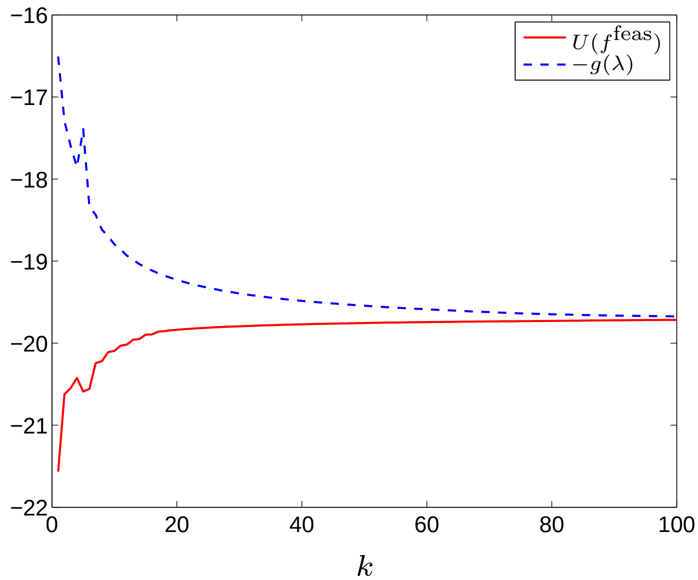
Example

- $n = 10$ flows, $m = 12$ links; 3 or 4 links per flow
- link capacities chosen randomly, uniform on $[0.1, 1]$
- $U_j(f_j) = \log f_j$ (can be argued to give proportionally fair flows)
- optimal flow as a function of price:

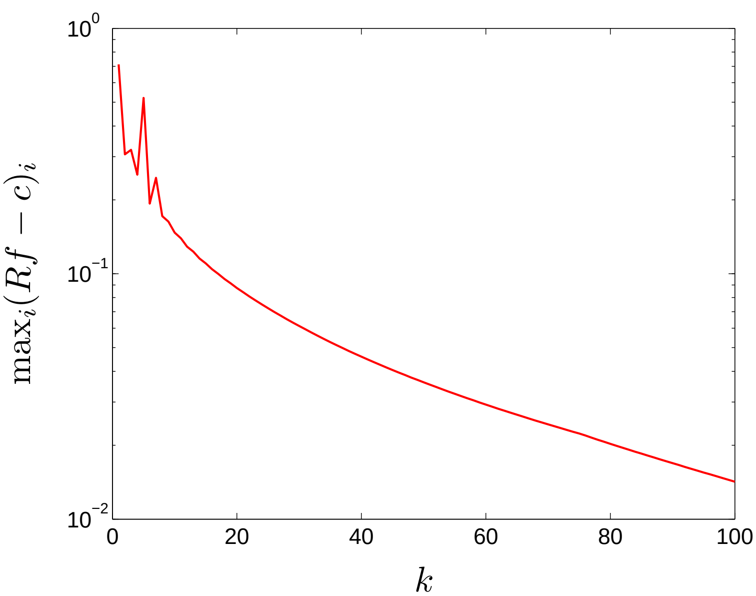
$$\bar{f}_j = \operatorname{argmax}(U_j(f_j) - \Lambda_j f_j) = 1/\Lambda_j$$

- initial prices: $\lambda = \mathbf{1}$
- constant stepsize $\alpha_k = 3$

Convergence of primal and dual objectives



Maximum capacity violation



References

- S. Boyd, course notes for EE364b, Convex Optimization II

  the rate control example is taken from EE364b lecture 9 and the notes on decomposition

- M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, *Layering as optimization decomposition: A mathematical theory of network architectures*, Proceedings IEEE (2007)

- D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods* (1989)

- L.S. Lasdon, *Optimization Theory for Large Systems* (1970)