Uses of duality

Geoff Gordon & Ryan Tibshirani
Optimization 10-725 / 36-725

Remember conjugate functions

Given $f : \mathbb{R}^n \to \mathbb{R}$, the function

$$f^*(y) = \max_{x \in \mathbb{R}^n} \; y^T x - f(x)$$

is called its **conjugate**

- Conjugates appear frequently in dual programs, as

$$-f^*(y) = \min_{x \in \mathbb{R}^n} \; f(x) - y^T x$$

- If f is closed and convex, then $f^{**} = f$. Also,

$$x \in \partial f^*(y) \quad \Leftrightarrow \quad y \in \partial f(x) \quad \Leftrightarrow \quad x \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \; f(z) - y^T z$$

  and for strictly convex f, $\nabla f^*(y) = \operatorname{argmin}_{z \in \mathbb{R}^n} (f(z) - y^T z)$

Uses of duality

We already discussed two key uses of duality:

- For x primal feasible and u, v dual feasible,

$$f(x) - g(u, v)$$

  is called the **duality gap** between x and u, v. Since

$$f(x) - f(x^\star) \leq f(x) - g(u, v)$$

  a zero duality gap implies optimality. Also, the duality gap can be used as a stopping criterion in algorithms

- Under strong duality, given dual optimal $u^\star, v^\star$, any primal solution minimizes $L(x, u^\star, v^\star)$ over $x \in \mathbb{R}^n$ (i.e., satisfies stationarity condition). This can be used to **characterize** or **compute** primal solutions

Outline

- Examples
- Dual gradient methods
- Dual decomposition
- Augmented Lagrangians

(And many more uses of duality—e.g., dual certificates in recovery theory, dual simplex algorithm, dual smoothing)

Lasso and projections onto polyhedra

Recall the lasso problem:

$$\min_{x \in \mathbb{R}^p} \ \frac{1}{2}\|y - Ax\|^2 + \lambda \|x\|_1$$

and its dual problem:

$$\min_{u \in \mathbb{R}^n} \ \|y - u\|^2 \ \text{ subject to } \ \|A^T u\|_\infty \leq \lambda$$

According to stationarity condition (with respect to z, x blocks):

$$Ax^\star = y - u^\star$$

$$A_i^T u^\star \in \begin{cases} \{\lambda\} & \text{if } x_i^\star > 0 \\ \{-\lambda\} & \text{if } x_i^\star < 0 \\ [-\lambda, \lambda] & \text{if } x_i^\star = 0 \end{cases}, \quad i = 1, \ldots p$$

where $A_1, \ldots A_p$ are columns of A. I.e., $|A_i^T u^\star| < \lambda$ implies $x_i^\star = 0$

Directly from dual problem,

$$\min_{u \in \mathbb{R}^n} \|y - u\|^2 \text{ subject to } \|A^T u\|_\infty \leq \lambda$$

we see that

$$u^\star = P_C(y)$$

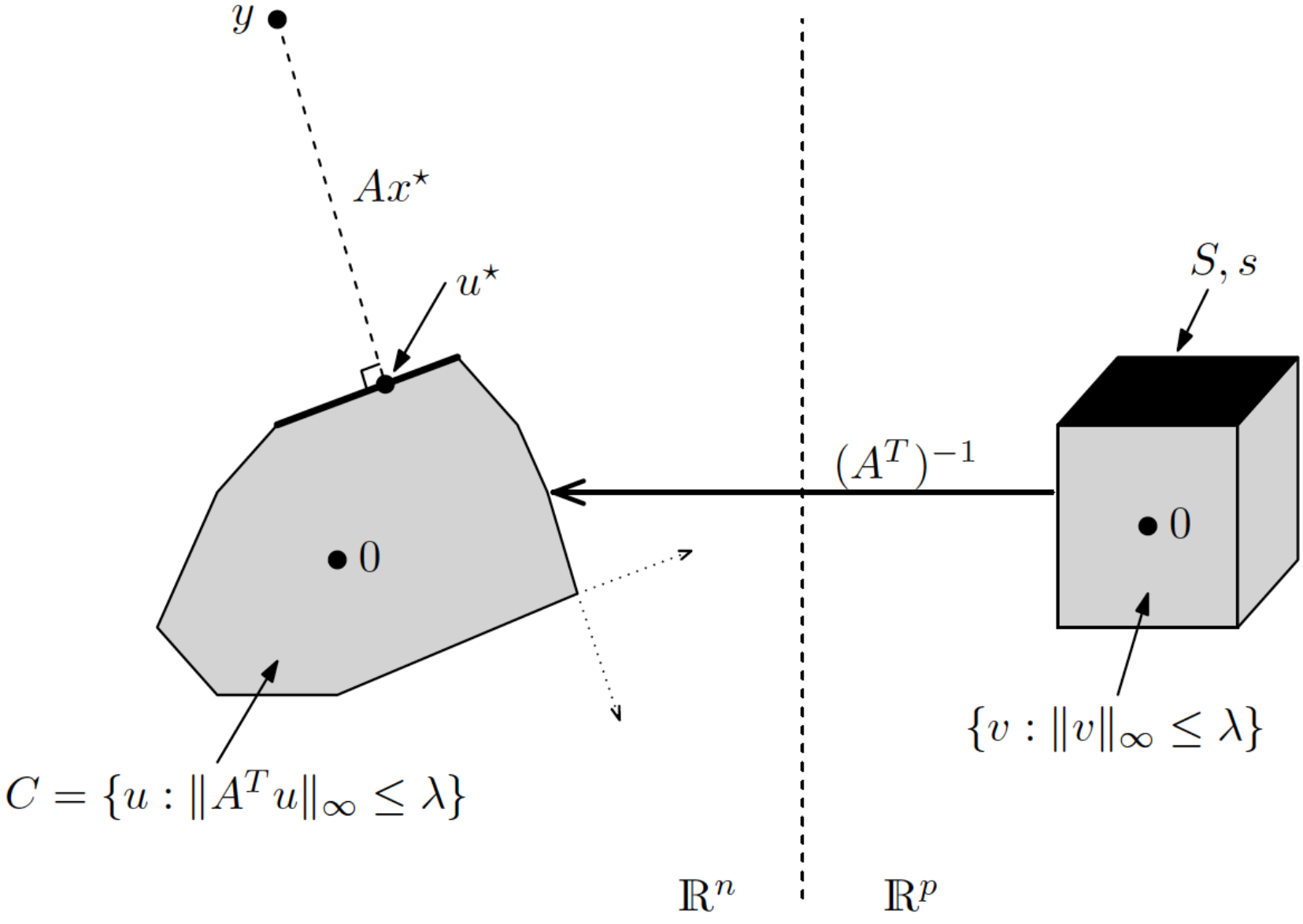
projection of y onto polyhedron

$$\begin{aligned} C &= \{u \in \mathbb{R}^n : \|A^T u\|_\infty \leq \lambda\} \\ &= \bigcap_{i=1}^{p} \{u : A_i^T u \leq \lambda\} \cap \{u : A_i^T u \geq -\lambda\} \end{aligned}$$

Therefore the lasso fit is

$$Ax^\star = (I - P_C)(y)$$

residual from projecting onto C


y
$Ax^\star$
$u^\star$
S, s
$(A^T)^{-1}$
0
0
$C = \{u : \|A^T u\|_\infty \leq \lambda\}$
$\{v : \|v\|_\infty \leq \lambda\}$
$\mathbb{R}^n$
$\mathbb{R}^p$

Consider the lasso fit $Ax^\star$ as a function of $y \in \mathbb{R}^n$, for fixed A, λ. From the dual perspective (and some geometric arguments):

- The lasso fit $Ax^\star$ is nonexpansive with respect to y, i.e., it is Lipschitz with constant $L = 1$:

$$\|Ax^\star(y) - Ax^\star(y')\| \leq \|y - y'\| \quad \text{for all } y, y'$$

- Each face of polyhedron C corresponds to a particular active set S for lasso solutions[1]
- For almost every $y \in \mathbb{R}^n$, if we move y slightly, it will still project to the same face of C
- Therefore, for almost every y, the active set S of the lasso solution is locally constant,[2] and the lasso fit is a locally affine projection map

[1,2] These statements assume that the lasso solution is unique; analogous statements exist for the nonunique case

Safe rules

For the lasso problem, somewhat amazingly, we have a **safe rule:**[3]

$$|A_i^T y| < \lambda - \|A_i\|\|y\|\frac{\lambda_{\max} - \lambda}{\lambda_{\max}} \quad \Rightarrow \quad x_i^\star = 0, \quad \text{all } i = 1, \ldots p$$

where $\lambda_{\max} = \|A^T y\|_\infty$ (the smallest value of λ such that $x^\star = 0$), i.e., we can eliminate features apriori, without solving the problem. (Note: this is **not an if and only if** statement!) Why this rule? Construction comes from lasso dual:

$$\max_{u \in \mathbb{R}^n} \; g(u) \; \text{ subject to } \; \|A^T u\|_\infty \leq \lambda$$

where $g(u) = (\|y\|^2 - \|y - u\|^2)/2$. Suppose u_0 is a dual feasible point (e.g., take $u_0 = y \cdot \lambda/\lambda_{\max}$). Then $\gamma = g(u_0)$ lower bounds dual optimal value, so dual problem is equivalent to

$$\max_{u \in \mathbb{R}^n} \; g(u) \; \text{ subject to } \; \|A^T u\|_\infty \leq \lambda, \; g(u) \geq \gamma$$

---

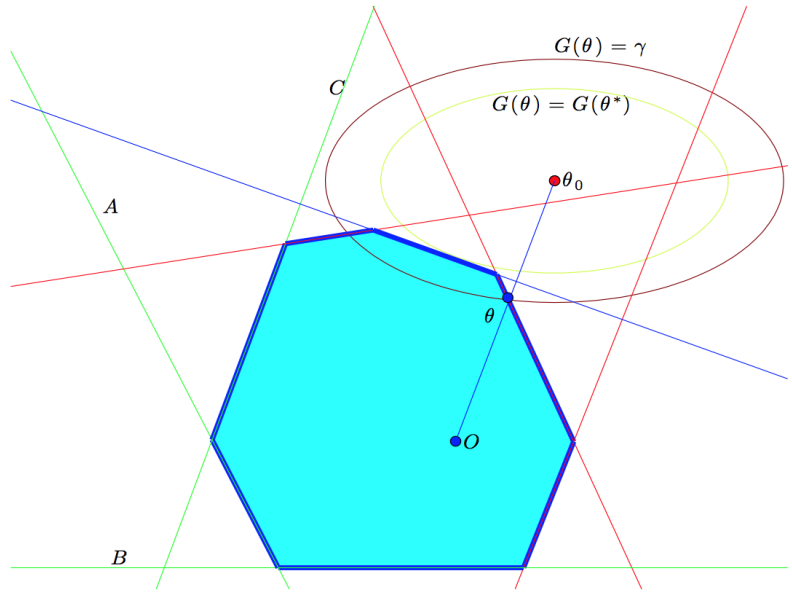
[3] L. El Ghaoui et al. (2010), *Safe feature elimination in sparse learning*. Safe rules extend to lasso logistic regression and 1-norm SVMs, only g changes

Now consider computing

$$m_i = \max_{u \in \mathbb{R}^n} |A_i^T u| \text{ subject to } g(u) \geq \gamma, \quad \text{for } i = 1, \ldots p$$

Note that

$$\begin{aligned} & m_i < \lambda \\ \Rightarrow \; & |A_i^T u^\star| < \lambda \\ \Rightarrow \; & x_i^\star = 0 \end{aligned}$$



4

Through another dual argument, we can explicitly compute m_i, and

$$m_i < \lambda \quad \Leftrightarrow \quad |A_i^T y| < \lambda - \sqrt{\|y\|^2 - 2\gamma} \cdot \|x\|$$

Substituting $\gamma = g(y \cdot \lambda / \lambda_{\mathsf{max}})$ then gives safe rule on previous slide

[4] From L. El Ghaoui et al. (2010), *Safe feature elimination in sparse learning*

Beyond pure sparsity

Consider something like a reverse lasso problem (also called 1-norm analysis):

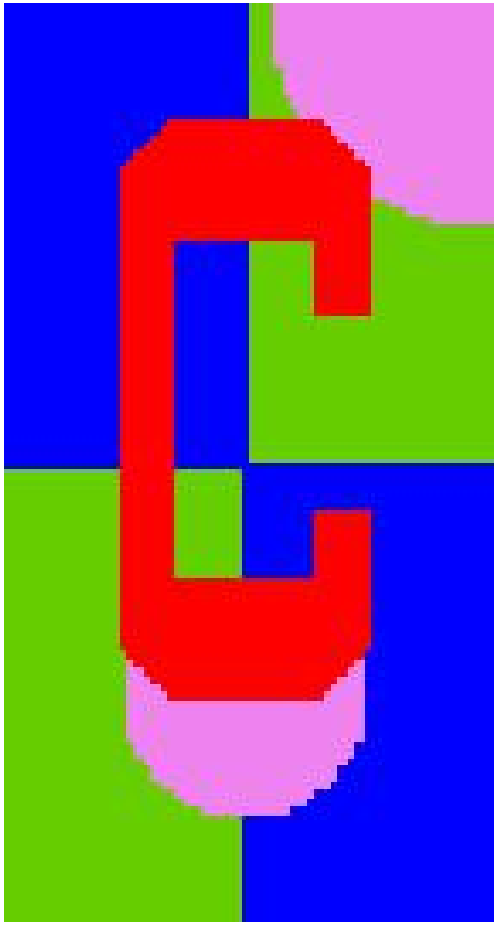
$$\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - x\|^2 + \lambda \|Dx\|_1$$

where $D \in \mathbb{R}^{m \times n}$ is a given penalty matrix (analysis operator). Note this cannot be turned into a lasso problem if $\mathrm{rank}(D) < m$

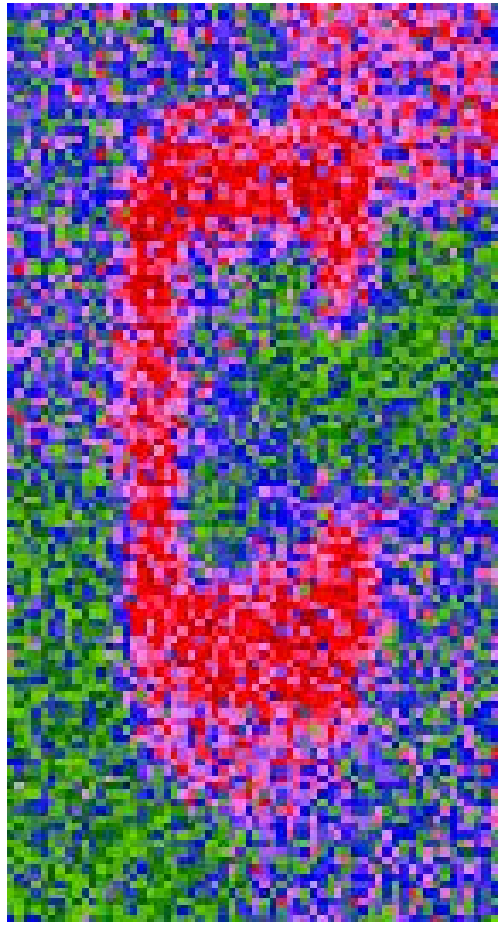
Basic idea: $Dx^\star$ is now sparse, and we choose D so that this gives some type of desired structure in $x^\star$. E.g., **fused lasso** (also called total variation denoising problems), where D is chosen so that

$$\|Dx\|_1 = \sum_{(i,j) \in E} |x_i - x_j|$$

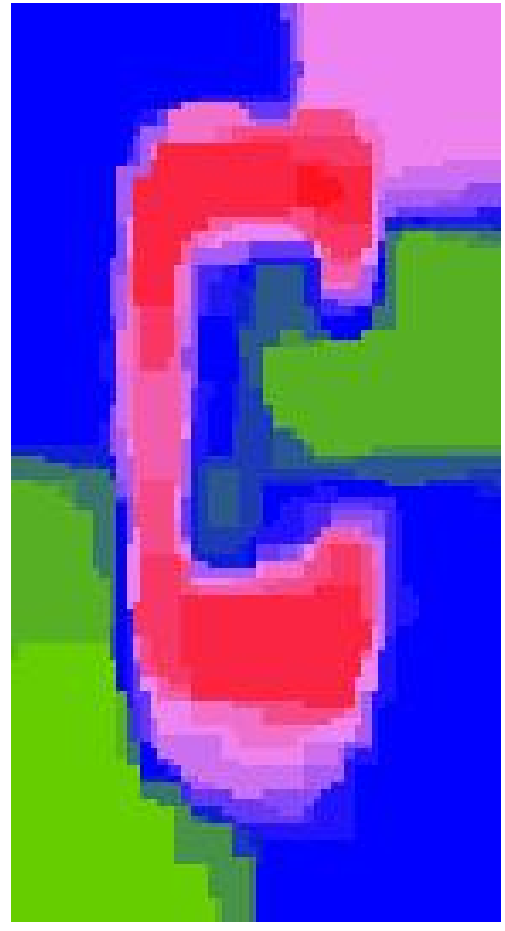
for some set of pairs E. In other words, D is incidence matrix for graph $G = (\{1, \ldots p\}, E)$, with arbitrary edge orientations



original image



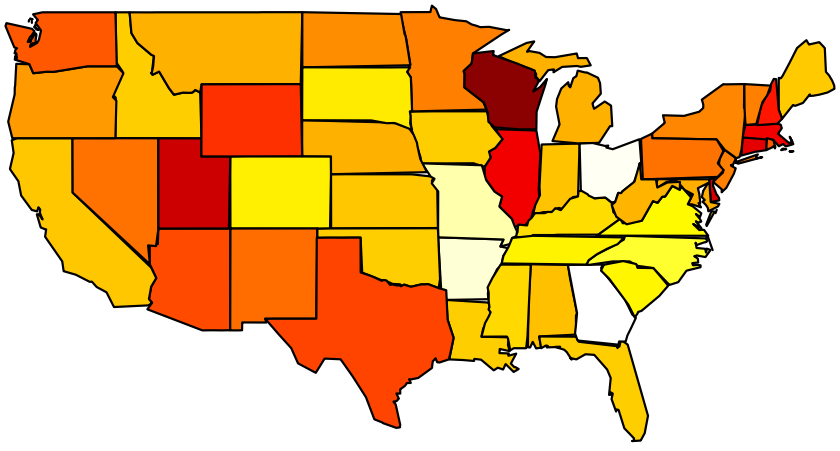
noisy version



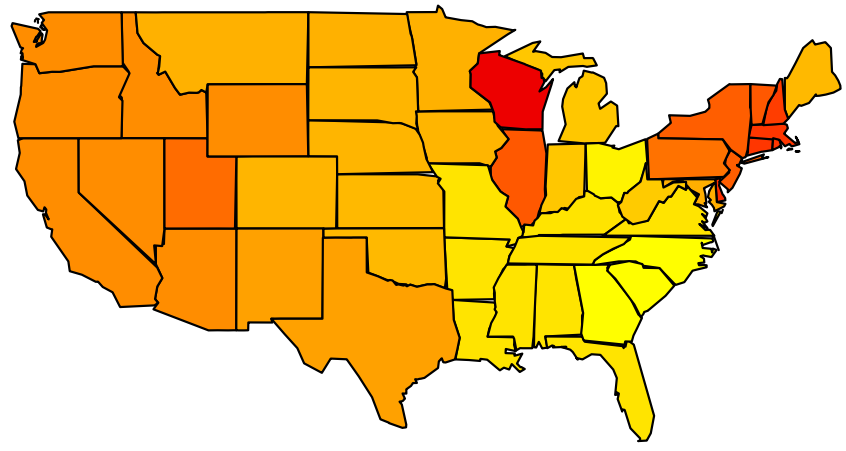
fused lasso solution

Here D is incidence matrix on 2d grid

For each state, we have log proportion of H1N1 cases in 2009 (from the CDC)



observed data



fused lasso solution

Here D is the incidence matrix on the graph formed by joining US states to their geographic neighbors

Using similar steps as in lasso dual derivation, here dual problem is:

$$\min_{u \in \mathbb{R}^m} \|y - D^T u\|^2 \text{ subject to } \|u\|_\infty \leq \lambda$$

and primal-dual relationship is

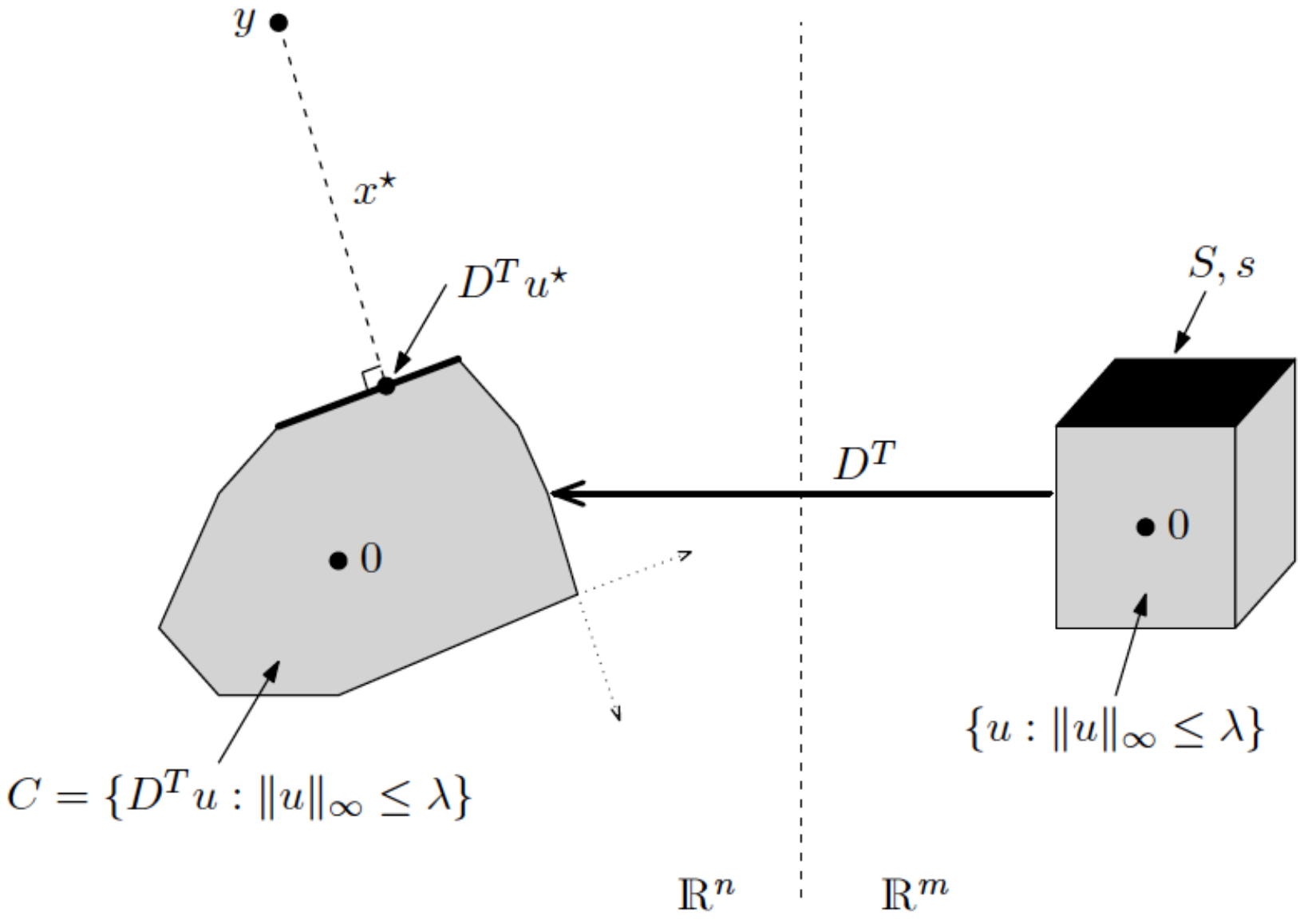
$$x^\star = y - D^T u^\star$$

$$u^\star \in \begin{cases} \{\lambda\} & \text{if } (Dx^\star)_i > 0 \\ \{-\lambda\} & \text{if } (Dx^\star)_i < 0 \\ [-\lambda, \lambda] & \text{if } (Dx^\star)_i = 0 \end{cases}, \quad i = 1, \ldots m$$

Clearly $D^T u^\star = P_C(y)$, where now

$$C = \{D^T u : \|u\|_\infty \leq \lambda\}$$

also a polyhedron, and therefore $x^\star = (I - P_C)(y)$


y
$x^\star$
$D^T u^\star$
S, s
D^T
0
0
$C = \{D^T u : \|u\|_\infty \leq \lambda\}$
$\{u : \|u\|_\infty \leq \lambda\}$
$\mathbb{R}^n$
$\mathbb{R}^m$

Same arguments as before show that:

- Primal solution $x^\star$ is Lipschitz continuous as a function of y (for fixed D, λ) with constant $L = 1$
- Each face of polyhedron C corresponds to a nonzero pattern in $Dx^\star$
- Almost everywhere in y, primal solution $x^\star$ admits a locally constant structure $S = \mathrm{supp}(Dx^\star)$, and therefore is a locally affine projection map

Dual is also very helpful for algorithmic reasons: it uncomplicates (disentagles) involvement of linear operator D with 1-norm

Prox function in dual problem now very easy (projection onto ∞-norm ball) so we can use, e.g., generalized gradient descent or accelerated generalized gradient method on the dual problem

Dual gradient methods

What if we can't derive dual (conjugate) in closed form, but want to utilize dual relationship? Turns out we can still use dual-based subradient or gradient methods

E.g., consider the problem

$$\min_{x \in \mathbb{R}^n} \; f(x) \; \text{ subject to } \; Ax = b$$

Its dual problem is

$$\max_{u \in \mathbb{R}^m} \; -f^*(-A^T u) - b^T u$$

where f^* is conjugate of f. Defining $g(u) = f^*(-A^T u)$, note that $\partial g(u) = -A \partial f^*(-A^T u)$, and recall

$$x \in \partial f^*(-A^T u) \quad \Leftrightarrow \quad x \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \; f(z) + u^T A z$$

Therefore the **dual subgradient method** (for minimizing negative of dual objective) starts with an initial dual guess $u^{(0)}$, and repeats for $k = 1, 2, 3, \ldots$

$$
\begin{aligned}
x^{(k)} &\in \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) + (u^{(k-1)})^T A x \\
u^{(k)} &= u^{(k-1)} + t_k (A x^{(k-1)} - b)
\end{aligned}
$$

where t_k are step sizes, chosen in standard ways

Recall that if f is strictly convex, then f^* is differentiable, and so we get **dual gradient ascent**, which repeats for $k = 1, 2, 3, \ldots$

$$
\begin{aligned}
x^{(k)} &= \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) + (u^{(k-1)})^T A x \\
u^{(k)} &= u^{(k-1)} + t_k (A x^{(k-1)} - b)
\end{aligned}
$$

(difference is that $x^{(k)}$ is unique, here)

In fact, f strongly convex with parameter $d \Rightarrow \nabla f^*$ Lipschitz with parameter $1/d$

Check: if f strongly convex and x is its minimizer, then

$$f(y) \geq f(x) + \frac{d}{2}\|y - x\|, \quad \text{all } y$$

Hence defining $x_u = \nabla f^*(u)$, $x_v = \nabla f^*(v)$,

$$f(x_v) - u^T x_v \geq f(x_u) - u^T x_u + \frac{d}{2}\|x_u - x_v\|^2$$
$$f(x_u) - v^T x_u \geq f(x_v) - v^T x_v + \frac{d}{2}\|x_u - x_v\|^2$$

Adding these together:

$$d\|x_u - x_v\|^2 \leq (u - v)^T (x_u - x_v)$$

Use Cauchy-Schwartz and rearrange, $\|x_u - x_v\| \leq (1/d) \cdot \|u - v\|$

Applying what we know about gradient descent: if f is strongly convex with parameter d, then dual gradient ascent with constant step size $t_k \leq d$ converges at rate $O(1/k)$. (Note: this is quite a strong assumption leading to a modest rate!)

Dual generalized gradient ascent and accelerated dual generalized gradient method carry through in similar manner

Disadvantages of dual methods:

- Can be slow to converge (think of subgradient method)
- Poor convergence properties: even though we may achieve convergence in dual objective value, convergence of $u^{(k)}, x^{(k)}$ to solutions requires strong assumptions (primal iterates $x^{(k)}$ can even end up being infeasible in limit)

Advantage: decomposability

Dual decomposition

Consider

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{B} f_i(x_i) \;\text{ subject to } Ax = b$$

Here $x = (x_1, \ldots x_B)$ is division into B blocks of variables, so each $x_i \in \mathbb{R}^{n_i}$. We can also partition A accordingly

$$A = [A_1, \ldots A_B], \;\text{ where } A_i \in \mathbb{R}^{m \times n_i}$$

Simple but powerful observation, in calculation of (sub)gradient:

$$x^+ \in \operatorname*{argmin}_{x \in \mathbb{R}^n} \sum_{i=1}^{B} f_i(x_i) + u^T A x$$

$$\Leftrightarrow \quad x_i^+ \in \operatorname*{argmin}_{x_i \in \mathbb{R}^{n_i}} f_i(x_i) + u^T A_i x_i, \quad \text{for } i = 1, \ldots B$$

i.e., minimization **decomposes** into B separate problems

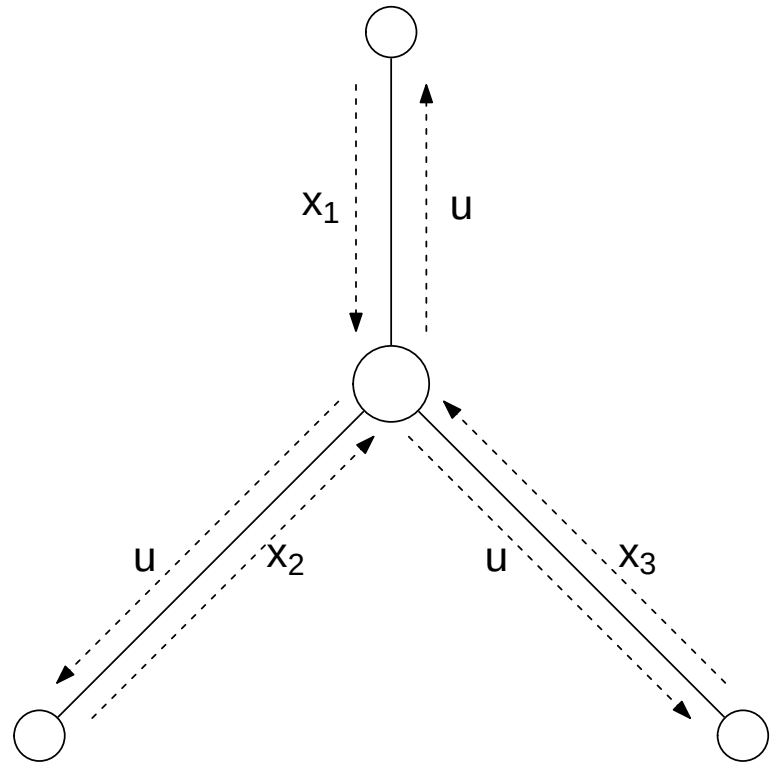
Dual decomposition algorithm: repeat for $k = 1, 2, 3, \ldots$

$$x_i^{(k)} \in \operatorname*{argmin}_{x_i \in \mathbb{R}^{n_i}} f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots B$$

$$u^{(k)} = u^{(k-1)} + t_k \Big(\sum_{i=1}^{B} A_i x_i^{(k-1)} - b \Big)$$

Can think of these steps as:

- **Broadcast:** send u to each of the B processors, each optimizes in parallel to find x_i
- **Gather:** collect $A_i x_i$ from each processor, update the global dual variable u



Example with inequality constraints:

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{B} f_i(x_i) \;\text{ subject to }\; \sum_{i=1}^{B} A_i x_i \leq b$$

Dual decomposition (projected subgradient method) repeats for $k = 1, 2, 3, \ldots$

$$x_i^{(k)} \in \operatorname*{argmin}_{x_i \in \mathbb{R}^{n_i}} f_i(x_i) + (u^{(k-1)})^T A_i x_i, \quad i = 1, \ldots B$$

$$v^{(k)} = u^{(k-1)} + t_k \Big(\sum_{i=1}^{B} A_i x_i^{(k-1)} - b \Big)$$

$$u^{(k)} = (v^{(k)})_+$$

where $(\cdot)_+$ is componentwise thresholding, $(u_+)_i = \max\{0, u_i\}$

Price coordination interpretation (from Vandenberghe's lecture notes):

- Have B units in a system, each unit chooses its own decision variable x_i (how to allocate its goods)
- Constraints are limits on shared resources (rows of A), each component of dual variable u_j is price of resource j
- Dual update:

$$u_j^+ = (u_j - ts_j)_+, \quad j = 1, \ldots m$$

  where $s = b - \sum_{i=1}^{B} A_i x_i$ are slacks
  - Increase price u_j if resource j is over-utilized, $s_j < 0$
  - Decrease price u_j if resource j is under-utilized, $s_j > 0$
  - Never let prices get negative

Augmented Lagrangian

Convergence of dual methods can be greatly improved by utilizing **augmented Lagrangian**. Start by transforming primal

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \; f(x) + \frac{\rho}{2}\|Ax - b\|^2 \\ &\text{subject to } Ax = b \end{aligned}$$

Clearly extra term $(\rho/2) \cdot \|Ax - b\|^2$ does not change problem

Assuming, e.g., A has full column rank, primal objective is strongly convex (parameter $\rho \cdot \sigma_{\min}^2(A)$), so dual objective is differentiable and we can use dual gradient ascent: repeat for $k = 1, 2, 3, \ldots$

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_{x \in \mathbb{R}^n} \; f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|^2 \\ u^{(k)} &= u^{(k-1)} + \rho(Ax^{(k-1)} - b) \end{aligned}$$

Note step size choice $t_k = \rho$, for all k, in dual gradient ascent. Why? Since $x^{(k)}$ minimizes $f(x) + (u^{(k-1)})^T Ax + \frac{\rho}{2}\|Ax - b\|^2$ over $x \in \mathbb{R}^n$,

$$\begin{aligned} 0 &\in \partial f(x^{(k)}) + A^T\Big(u^{(k-1)} + \rho(Ax^{(k)} - b)\Big) \\ &= \partial f(x^{(k)}) + A^T y^{(k)} \end{aligned}$$

This is exactly the **stationarity condition** for the original primal problem; can show under mild conditions that $Ax^{(k)} - b$ approaches zero (primal iterates approach feasibility), hence in the limit KKT conditions are satisfied and $x^{(k)}, u^{(k)}$ approach optimality

Advantage: much better convergence properties

Disadvantage: not decomposable (separability comprimised by augmented Lagrangian)

ADMM (Alternating Direction Method of Multipliers): tries for best of both worlds

References

- S. Boyd and N. Parikh and E. Chu and B. Peleato and J. Eckstein (2010), *Distributed optimization and statistical learning via the alternating direction method of multipliers*
- L. Vandenberghe, Lecture Notes for EE 236C, UCLA, Spring 2011-2012