Proximal Gradient Descent

Jianyong Sun

Xi'an Jiaotong University

jy.sun@xjtu.edu.cn

Overview

Outline

- Proximal gradient descent
- Convergence analysis
- ISTA, matrix completion
- Acceleration

Decomposable functions

Suppose

$$\min_x f(x) = g(x) + h(x)$$

- g is convex and differentiable, $\mathrm{dom}(g) = \mathbb{R}^n$
- h is convex, not necessarily differentiable

If f were differentiable, then gradient descent update would be

$$x^+ = x - t \cdot \nabla f(x)$$

Recall motivation: minimize quadratic approximation to f around x, replace $\nabla^2 f(x)$ by $\frac{1}{t} I$

$$x^+ = \operatorname*{argmin}_z \underbrace{f(x) + \nabla f(x)^T (x - z) + \frac{1}{2t}\|x - z\|_2^2}_{\tilde{f}_t(z)}$$

Decomposable functions

In our case f is not differentiable, but $f = g + h$, g is differentiable, Why do not we make quadratic approximation to g, leave f alone?

I.e. update

$$\begin{aligned} x^+ &= \underset{z}{\operatorname{argmin}}\, \tilde{g}(z) + h(z) \\ &= \underset{z}{\operatorname{argmin}}\, g(x) + \nabla g(x)^T (x - z) + \frac{1}{2t}\|x - z\|_2^2 + h(z) \\ &= \underset{z}{\operatorname{argmin}}\, \frac{1}{2t}\|z - (x - t \cdot \nabla g(x))\|_2^2 + h(z) \end{aligned}$$

Here

$\frac{1}{2t}\|z - (x - t \cdot \nabla g(x))\|_2^2$	stay close to gradient update for g
$h(z)$	also make h small

Proximal gradient descent

Define proximal mapping

$$\mathrm{prox}_t(x) = \operatorname*{argmin}_z \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

Proximal gradient descent: choose initial point $x^{(0)}$ repeat

$$x^{(k)} = \mathrm{prox}_{t_k}\left(x^{(k-1)} - t_k \nabla g\left(x^{(k-1)}\right)\right), k = 1, 2, \cdots$$

To make this update step look familiar, can rewrite it as

$$x^{(k)} = x^{(k-1)} - t_k \cdot G_{t_k}\left(x^{(k-1)}\right)$$

where G_t is the generalized gradient of f

$$G_t(x) = \frac{x - \mathrm{prox}_t(x - t\nabla g(x))}{t}$$

What good did this do?

You have a right to be suspicious, $\cdots$, may look like we just swap one minimization problem for another

Key point is that $\text{prox}_t(\cdot)$ can be computed analytically for a lot of important functions h. Note

- Mapping $\text{prox}_t(\cdot)$ doesn't depend on g at all, only on h
- Smooth part g can be complicated, we only need to compute its gradients

Convergence analysis will be in terms of number of iterations of the algorithm. Keep in mind that each iteration evaluates $\text{prox}_t(\cdot)$ once, this can be cheap or expensive, depending on h

Example: ISTA

Given $y \in \mathbb{R}^n, X \in \mathbb{R}^{n\times p}$, recall lasso criterion

$$f(\beta) = \underbrace{\frac{1}{2}\|y - X\beta\|_2^2}_{g(\beta)} + \underbrace{\lambda\|\beta\|_1}_{h(\beta)}$$

Prox mapping is now

$$\text{prox}_t(\beta) = \underset{z}{\text{argmin}}\, \frac{1}{2t}\|\beta - z\|_2^2 + \lambda\|z\|_1 = S_{\lambda t}(\beta)$$

where $S_\lambda(\beta)$ is the soft-thresholding operator

$$[S_\lambda(\beta)]_i = \begin{cases} \beta_i - \lambda & \text{if } \ \beta_i > \lambda \\ 0 & \text{if } \ -\lambda \le \beta_i \le \lambda \\ \beta_i + \lambda & \text{if } \ \beta_i < -\lambda \end{cases}$$

Example: ISTA

Recall $\nabla g(\beta) = -X^T(y - X\beta)$, hence proximal gradient update is

$$\beta^+ = S_{\lambda t}(\beta + tX^T(y - X\beta))$$

Often called the iterative soft-thresholding algorithm (ISTA)[1]. Very simple algorithm to compute a lasso solution.

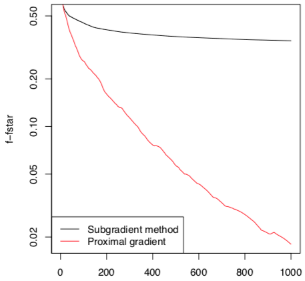


Figure: Example of proximal gradient (ISTA) vs. subgradient convergence rate

[1]Beck and Teboulle (2008), "A fast iterative shrinkage-thresholding algorithm for linear inverse problem"

Convergence analysis

With criterion $f(x) = g(x) + h(x)$, we assume

- g is convex and differentiable, $\mathrm{dom}(g) = \mathbb{R}^n$ and ∇g is Lipschitz continuous with constant $L > 0$
- h is convex, $\mathrm{prox}_t(x)$ can be evaluated.

Theorem

Proximal gradient descent with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)}) - f^\star \leq \frac{\|x^{(0)} - x^\star\|_2^2}{2tk}$$

Proximal gradient descent has convergence rate $O(1/k)$ or $O(1/\epsilon)$

Same as gradient descent! But remember, this counts for the number of iterations, not operations.

Backtracking line search

Similar to gradient decent, but operates on g. We fix a parameter $0 < \beta < 1$. At each iteration, start with $t = 1$, and while

$$g(x - tG_t(x)) > g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}\|G_t(x)\|_2^2$$

shrink $t = \beta t$, Else perform prox gradient update.

Under same assumptions, we get the same rate

Theorem

Proximal gradient descent with backtracking line search satisfies

$$f(x^{(k)}) - f^\star \le \frac{\|x^{(0)} - x^\star\|_2^2}{2t_{\min} k}$$

where $t_{\min} = \min\{1, \beta/L\}$

Example: matrix completion

Given a matrix $Y \in \mathbb{R}^{m\times n}$, and only observe entries $Y_{ij}, (i,j) \in \Omega$. Suppose we want to fill in missing entries (e.g. for a recommender system), so we solve a matrix completion problem:

$$\min_{B\in\mathbb{R}^{m\times n}} \frac{1}{2} \sum_{(i,j)\in\Omega} (Y_{ij} - B_{ij})^2 + \lambda \|B\|_{\mathrm{tr}}$$

Here $\|B\|_{\mathrm{tr}}$ is the trace (or nuclear) norm of B

$$\|B\|_{\mathrm{tr}} = \sum_{i=1}^{r} \sigma_i(B)$$

where $r = \mathrm{rank}(B)$ and $\sigma_1(X) \geq \cdots \geq \sigma_r(X) \geq 0$ are the singular values.

Define P_Ω, projection operator onto observed set

$$[P_\Omega(B)]_{ij} = \begin{cases} B_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$$

Then the criterion is

$$f(B) = \underbrace{\frac{1}{2}\|P_\Omega(Y) - P_\Omega(B)\|_F^2}_{g(B)} + \underbrace{\lambda\|B\|_{\text{tr}}}_{h(B)}$$

Two ingredients needed for proximal gradient descent

- Gradient calculation $\nabla g(B) = -(P_\Omega(Y) - P_\Omega(B))$
- Prox function

$$\text{prox}_t(B) = \underset{Z\in\mathbb{R}^{m\times n}}{\text{argmin}} \frac{1}{2t}\|B - Z\|_F^2 + \lambda\|Z\|_{\text{tr}}$$

Claim: $\text{prox}_t(B) = S_{\lambda t}(B)$, matrix soft-thresholding at the level λ. here $S_\lambda(B)$ is defined by

$$S_\lambda(B) = U\Sigma_\lambda V^T$$

where $B = U\Sigma V^T$ is an SVD, and Σ_λ is diagonal with

$$(\Sigma_\lambda)_{ii} = \max\{\Sigma_{ii} - \lambda, 0\}$$

Why? Note that $\text{prox}_t(B) = Z$ where Z satisfies

$$0 \in Z - B + \lambda t \cdot \partial\|Z\|_{\text{tr}}$$

Fact: if $Z = U\Sigma V^T$, then

$$\partial\|Z\|_{\text{tr}} = \{UV^T + W : \|W\|_{op} \le 1, U^T W = 0, WV = 0\}$$

where $\|A\|_{op} = \max\{\|Au\|_2 : \|u\|_2 = 1\}$, Now plug in $Z = S_{\lambda t}(B)$ and check that we can get 0.

Hence proximal gradient update step is

$$B^{+} = S_{\lambda t}\Big(B + t(P_{\Omega}(Y) - P_{\Omega}(B))\Big)$$

Note that $\nabla g(B)$ is Lipschtz continuous with $L = 1$, so we can choose fixed step size $t = 1$. Update step is now

$$B^{+} = S_{\lambda}(P_{\Omega}(Y) + P_{\Omega}^{\perp}(B))$$

where $P_{\Omega}^{\perp}(B)$ projects onto unobserved set, $P_{\Omega}(B) + P_{\Omega}^{\perp}(B) = B$

This is the soft-impute algorithm[2]

[2]Mazumder et al. (2011) "Spectral regularization algorithm for learning large incomplete matrices"

Special cases

Proximal gradient descent also called composite gradient descent or generalized gradient descent

Why "general"? This refers to the several special cases, when minimizing $f = g + h$

- $h = 0 \rightarrow$ gradient descent
- $h = I_C \rightarrow$ projected gradient descent
- $g = 0 \rightarrow$ proximal minimization algorithm

Therefore these algorithms all have $O(1/\epsilon)$ convergence rate.

Projected gradient descent

Given closed, convex set $C \in \mathbb{R}^n$

$$\min_{x \in C} g(x) \Longleftrightarrow \min_{x} g(x) + I_C(x)$$

where $I_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$ is the indictor function of C. Hence

$$\begin{aligned} \mathrm{prox}_t(x) &= \operatorname*{argmin}_{z} \frac{1}{2t}\|x - z\|_2^2 + I_C(z) \\ &= \operatorname*{argmin}_{z \in C} \|x - z\|_2^2 \end{aligned}$$

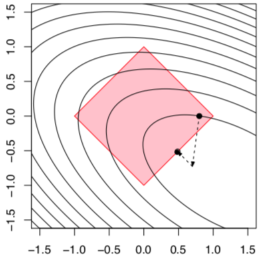
I.e. $\mathrm{prox}_t(x) = P_C(x)$, projection operator onto C

Projected gradient descent

Therefore proximal gradient update step is

$$x^+ = P_C(x - t\nabla g(x))$$

i.e. perform usual gradient update and then project onto C, called projected gradient descent.



Note: projected subgradient method works too.

Proximal minimization algorithm

Consider for h convex (not necessarily differentiable)

$$\min_x h(x)$$

Proximal gradient update step is just

$$x^+ = \operatorname*{argmin}_z \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

Called proximal minimization algorithm. Faster than subgradient method, but not implementable unless we know *prox* in closed form.

What happens if we can't evaluate prox?

Theory for proximal gradient, with $f = g + h$, assumes that prox function can be evaluated, i.e. assumes that minimization

$$\mathrm{prox}_t(x) = \underset{z}{\operatorname{argmin}} \frac{1}{2t}\|x - z\|_2^2 + h(z)$$

can be done exactly. In general, not clear what happens if we just minimize this approximately.

But, if you can precisely control the errors in approximating the prox operator, then you can recover the original convergence rate[3]

In practice, if prox evaluation is done approximately, then it should be done to decently high accuracy.

[3] Schimidt et al. (2011), "Convergence rates of inexact proximal gradient methods for convex optimization"

Acceleration

Turns out we can accelerate proximal gradient descent in order to achieve the optimal $O(1/\sqrt{\epsilon})$ convergence rate. Four ideas (three acceleration methods) by Nesterov

- 1983: original acceleration idea for smooth function
- 1988: another acceleration idea for smooth function
- 2005: smoothing techniques for nonsmooth functions, coupled with original acceleration idea
- 2007: acceleration idea for composite functions[4]

We will follow Beck and Teboulle (2008), extension of Nesterov (1983) to composite functions[5]

[4]Each step uses entire history of previous steps and makes two prox calls

[5]Each step uses information from two last steps and makes one prox call.

Accelerated proximal gradient method

Our problem, as before

$$\min_x g(x) + h(x)$$

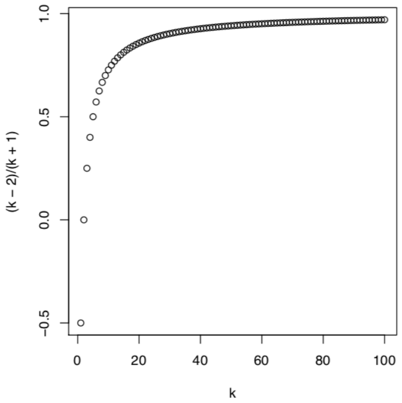
where g convex, differentiable, and h convex. Accelerated proximal gradient method: choose initial point $x^{(0)} = x^{(-1)} \in \mathbb{R}^n$, repeat

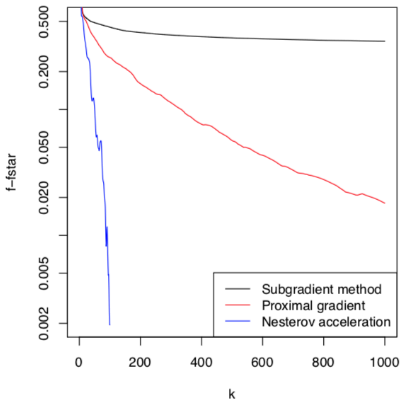
$$\begin{aligned} v &= x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)}) \\ x^{(k)} &= \text{prox}_t(v - t_k \nabla g(v)) \end{aligned}$$

for $k = 1, 2, \cdots$,

- First step $k = 1$ is just usual proximal gradient update
- After that $v = x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)})$ carries some "momentum" from previous iterations
- $h = 0$ gives accelerated gradient method.

Accelerated proximal gradient method





Note: accelerated proximal gradient is not a descent method ('Nesterov ripples')

Convergence Analysis

As usual, we are minimizing $f(x) = g(x) + h(x)$, assuming

- g is convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇g is Lipschitz continuous with constant $L > 0$
- h is convex, prox function can be evaluated.

Theorem

Accelerated proximal gradient method with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)}) - f^\star \leq \frac{2\|x^{(0)} - x^\star\|_2^2}{t(k+1)^2}$$

Achieves the optimal rate $O(1/k^2)$ for first-order methods! i.e. a rate of $O(1/\sqrt{\epsilon})$

Backtracking line search

Fix $\beta < 1$, $t_0 = 1$, at iteration k, start with $t = t_{k-1}$, and while

$$g(x^+) > g(v) + \nabla g(v)^T (x^+ - v) + \frac{1}{2t}\|x^+ - v\|_2^2$$

shrink $t = \beta t$, and let $x^+ = \mathrm{prox}_t(v - t\nabla g(v))$. Else keep x^+

Under same assumptions, we get the same rate

Theorem

Accelerated proximal gradient method with backtracking line search satisfies

$$f(x^{(k)}) - f^\star \leq \frac{2\|x^{(0)} - x^\star\|}{t_{\min}(k+1)^2}$$

where $t_{\min} = \min\{1, \beta/L\}$.

FISTA

Recall lasso problem

$$\min_{\beta} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

and ISTA (iterative soft-thresholding algorithm)

$$\beta^{(k)} = S_{\lambda t_k}\Big(\beta^{(k-1)} + t_k X^T(y - X\beta^{(k-1)})\Big), k = 1, 2, \cdots$$

$S_\lambda(\cdot)$ being vector soft-thresholding. Applying acceleration gives us FISTA (F is for Fast)[6] for $k = 1, 2, 3, \cdots$

$$\begin{aligned} v &= \beta^{(k-1)} + \frac{k-2}{k+1}(\beta^{(k-1)} - \beta^{(k-2)}) \\ \beta^{(k)} &= S_{\lambda t_k}\Big(v + t_k X^T(y - Xv)\Big) \end{aligned}$$

[6]Back and Teboulle (2008) actually call their general acceleration technique (for general g, h) FISA, which may be somewhat confusing

Is acceleration always useful?

Acceleration can be a very effective speedup tool ... but should it always be useful?

In practice, the speedup of using acceleration is diminished in the presence of warm starts. I.e. suppose want to solve lasso problem for tuning parameter values

$$\lambda_1 > \lambda_2 > \cdots > \lambda_T$$

- When solving for λ_1, initialize $x^{(0)} = 0$, record solution $\hat{x}(\lambda_1)$
- When solving for λ_j, initialize $x^{(0)} = \hat{x}(\lambda_{j-1})$, the recorded solution for λ_{j-1}

Over a fine enough grid of λ values, proximal gradient descent can often perform just as well without aceleration.

Sometimes backtracking and acceleration can be disavantageous! Recall matrix completion problem, the proximal gradient update is

$$B^+ = S_\lambda\Big(B + t(P_\Omega(Y) - P_\Omega^\perp(B))\Big)$$

where S_λ is the matrix soft-thresholding operator ... requires SVD

- One backtracking loop evaluates generalized gradient $G_t(x)$, i.e. evaluates $\mathrm{prox}_t(x)$, across various values of t. For matrix completion, this means multiple SVDs...
- Acceleration changes argument we pass to prox: $v - t\nabla g(v)$ instead of $x - t\nabla g(x)$. For matrix completion (and $t = 1$)

$$B - \nabla g(B) = \underbrace{P_\Omega(Y)}_{sparse} + \underbrace{P_\Omega^\perp(B)}_{\text{low rank}} \Rightarrow \text{fast SVD}$$

$$V - \nabla g(V) = \underbrace{P_\Omega(Y)}_{sparse} + \underbrace{P_\Omega^\perp(V)}_{\text{not necessarily low rank}} \Rightarrow \text{slow SVD}$$

Duality

Duality

- Duality in linear programs
- Lagrangian
- Lagrange dual function
- Lagrange dual problem
- Examples
- Weak and strong duality

Duality in linear programs

Suppose we want to find lower bound on the optimal value in convex problem

$$B \le \min_{x \in C} f(x)$$

E.g., consider the following simple LP

$$\begin{aligned} \min_{x,y} \quad & x + y \\ \text{subject to} \quad & x + y \ge 2 \\ & x, y \ge 0 \end{aligned}$$

What is a lower bound? Easy, take $B = 2$.

Duality in linear programs

Try again:

$$\begin{aligned} \min_{x,y} \quad & x + 3y \\ \text{subject to} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned}$$

$$\begin{array}{rr} & x + y \geq 2 \\ + & 2y \geq 0 \\ = & x + 3y \geq 2 \end{array}$$

Lower bound $B = 2$

More generally:

$$\begin{aligned} \min_{x,y} \quad & px + qy \\ \text{subject to} \quad & x + y \geq 2 \\ & x, y \geq 0 \end{aligned}$$

$$\begin{aligned} a + b &= p \\ a + c &= q \\ a, b, c &\geq 0 \end{aligned}$$

Lower bound $B = 2a$, for any a, b, c satisfying above

Duality in linear programs

What's the best we can do? Maximize our lower bound over all possible a, b, c:

$$\begin{array}{llll}
\min_{x,y} & px + qy & \max_{a,b,c} & 2a \\
\text{subjec to} & x + y \geq 2 & \text{subject to} & a + b = p \\
 & x \geq 0 & & a + c = q \\
 & y \geq 0 & & a, b, c \geq 0 \\
\text{primal LP} & & \text{dual LP} &
\end{array}$$

Note: the number of dual variables is the number of primal constraints.

Duality in linear programs

Try another one:

$$\begin{array}{llll}
\min_{x,y} & px + qy & \max_{a,b,c} & 2c - b \\
\text{subjec to} & x \geq 0 & \text{subject to} & a + 3c = p \\
 & y \leq 1 & & -b + c = q \\
 & 3x + y = 2 & & a, b \geq 0 \\
\text{primal LP} & & \text{dual LP} &
\end{array}$$

Note: in the dual problem, c is unconstrained.

Duality in linear programs

Given $c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, G \in \mathbb{R}^{r\times n}, h \in \mathbb{R}^r$

Primal Linear Programs (LP)

$$\begin{aligned} \min_{x} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & Gx \leq h \end{aligned}$$

Its Dual LP is

$$\begin{aligned} \max_{a,b} \quad & -b^T u - h^T v \\ \text{subject to} \quad & -A^T u - G^T v = c \\ & v \geq 0 \end{aligned}$$

Duality in linear programs

Explanation # 1: for any u and $v \geq 0$, and x primal feasible,

$$u^T(Ax - b) + v^T(Gx - h) \leq 0, \text{i.e.,}$$

$$(-A^T u - G^T v) \geq -b^T u - h^T v$$

So if $c = -A^T u - G^T v$, we get a bound on primal optimal value.

Explanation # 2: for any u and $v \geq 0$, and x primal feasible

$$c^T x \geq c^T x + u^T(Ax - b) + v^T(Gx - h) := L(x, u, v)$$

So if C denotes primal feasible set, $f^\star$ primal value, then for any u and $v \geq 0$,

$$f^\star \geq \min_{x \in C} L(x, u, v) \geq \min_{x \in \mathbb{R}^n} L(x, u, v) := g(u, v)$$

Duality in linear programs

In other words, $g(u, v)$ is a lower bound on $f^\star$ for any u and $v \geq 0$.

Note that

$$g(u, v) = \begin{cases} -b^T u - h^T v & \text{if } \; c = -A^T u - G^T v \\ -\infty & \text{otherwise} \end{cases}$$

Now we can maximize $g(u, v)$ over u and $v \geq 0$ to get the tightest bound, and this gives exactly the dual LP as before.

This last perspective is actually completely general and applies to arbitrary optimization problems (even nonconvex ones).

Lagrangian

Consider general minimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \le 0, i = 1, \cdots, m \\ & \ell_j(x) = 0, j = 1, \cdots, r \end{aligned}$$

Need not to be convex, but of course we will pay special attention to convex case

We define the Lagrangian as

$$L(x, u, v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x)$$

New variables $u \in \mathbb{R}^m, v \in \mathbb{R}^r$ with $u \ge 0$

Lagrangian

Important property: for any v and $v \geq 0$,

$$f(x) \geq L(x,u,v) \text{ at each feasible } x$$

why? For feasible x

$$L(x,u,v) = f(x) + \sum_{i=1}^{m} u_i \underbrace{h_i(x)}_{\geq 0} + \sum_{j=1}^{r} v_j \underbrace{\ell_j(x)}_{=0} \leq f(x)$$

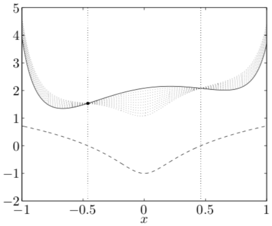


Figure: Solid line is f, dashed line is h, feasible set $\approx [-0.46, 0.46]$, each dotted line shows $L(x,u,v)$ for difference $u \geq 0$ and v (reproduced from B & V)

Lagrange dual function

Let C denote primal feasible set, $f^\star$ denote primal optimal value. Minimizing $L(x,u,v)$ over all $x \in \mathbb{R}^n$ gives a lower bound

$$f^\star \ge \min_{x\in C} L(x,u,v) \ge \min_{x\in\mathbb{R}^n} L(x,u,v) := g(u,v)$$

We call $g(u,v)$ the Lagrange dual function, and it gives a lower bound on $f^\star$ for any u and $v \ge 0$, called dual feasible u, v

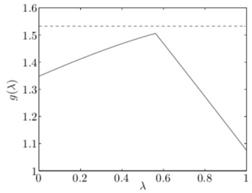


Figure: Solid line is $g(\lambda)$, dual variable λ is our u, dashed horizontal line is f^* (reproduced from B & V)

Quadratic programming

Consider quadratic programming (QP)

$$\begin{array}{rl} \displaystyle\min_{x\in\mathbb{R}^n} & \frac{1}{2}x^TQx + c^Tx \\ \text{subject to} & Ax = b, x \geq 0 \end{array}$$

where $Q \succ 0$. Lagrangian

$$L(x,u,v) = \frac{1}{2}x^TQx + c^Tx - u^Tx + v^T(Ax - b)$$

Lagrange dual function

$$g(u,v) = \min_{x\in\mathbb{R}^n} L(x,u,v) = -\frac{1}{2}(c - u + A^Tv)^TQ^{-1}(c - u + A^Tv) - b^Tv$$

For any $u \geq 0$ and any v, this is a lower bound on primal optimal value $f^\star$

Quadratic programming

Same problem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \tfrac{1}{2}x^T Q x + c^T x \\ \text{subject to} \quad & Ax = b, x \geq 0 \end{aligned}$$

but $Q \succeq 0$. Lagrangian

$$L(x, u, v) = \frac{1}{2}x^T Q x + c^T x - u^T x + v^T (Ax - b)$$

Lagrange dual function

$$g(u, v) = \begin{cases} -\tfrac{1}{2}(c - u + A^T v)^T & Q^+(c - u + A^T v) - b^T v \\ & \text{if } \; c - u + A^T v \perp \text{null}(Q) \\ -\infty & \text{otherwise} \end{cases}$$

where Q^+ denotes generalized inverse of Q. For any $u \geq 0, v$, and $c - u + A^T v \perp \text{null}(Q)$, $g(u, v)$ is a nontrivial lower bound on $f^\star$

Quadratic programming

If choose $f(x)$ to be quadratic in 2 variables, subject to $x \geq 0$, dual function $g(u)$ is also quadratic i n2 variables, also subject to $u \geq 0$

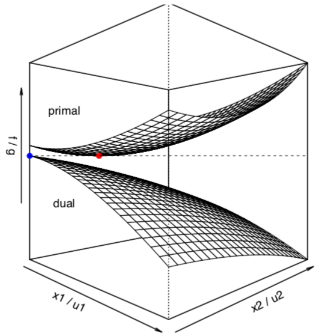


Figure: Dual function $g(u)$ provides a bound on $f^\star$ for every $u \geq 0$, largest bound this gives us: turns out to be exactly $f^\star$, coincidence?

Lagrange dual problem

Consider primal problem

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & h_i(x) \le 0, i = 1, \cdots, m \\ & \ell_j(x) = 0, j = 1, \cdots, r \end{array}$$

Our constructed dual function $g(u, v)$ satisfies $f^\star \ge g(u, v)$ for all $u \ge 0$ and v. Hence best lower bound is given by maximizing $g(u, v)$ over all dual feasible u, v, yielding Lagrange dual problem

$$\begin{array}{ll} \max\limits_{u\in\mathbb{R}^m, v\in\mathbb{R}^r} & g(u, v) \\ \text{subject to} & u \ge 0 \end{array}$$

Key property, called weak duality: if dual optimal value $g^\star$, then

$$f^\star \ge g^\star$$

Note that this always holds (even if primal problem is nonconvex).

Lagrange dual problem

Another key property: the dual problem is a convex optimization problem (as written, it is a concave maximization problem)

Again, this is always true (even when primal problem is not convex)

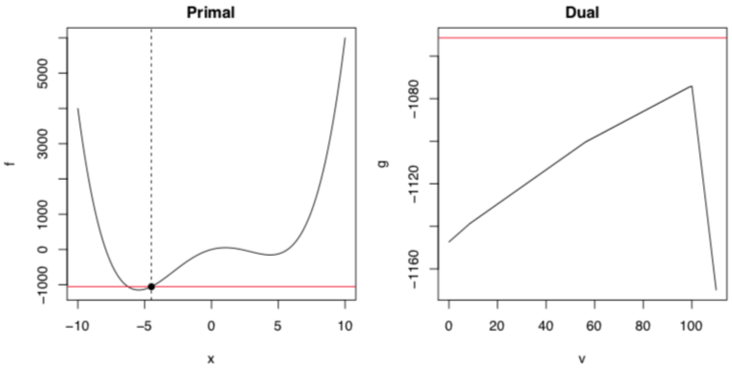
By definition

$$\begin{aligned} g(u,v) &= \min_x \left\{ f(x) + \sum_{i=1}^m u_i h_i(x) + \sum_{j=1}^r v_j \ell_j(x) \right\} \\ &= \underbrace{-\max \left\{ -f(x) - \sum_{i=1}^m u_i h_i(x) - \sum_{j=1}^r v_j \ell_j(x) \right\}}_{\text{pointwise maximum of convex functions in } (u,v)} \end{aligned}$$

i.e. g is concave in (u,v), and $u \geq 0$ is a convex constraint, hence dual problem is a concave maximization problem.

Example: nonconvex quartic minimization

Define $f(x) = x^4 - 50x^3 + 100x$ (nonconvex), minimize subject to constraint $x \geq -4.5$



Dual function g can be derived explicitly (via closed-form equation for roots of a cubic equation). Form of g is quite complicated, and would be hard to tell whether or not g is concave, but it must be!

Strong duality

In some problems, we have observed that actually

$$f^\star = g^\star$$

which is called strong duality.

Slater's condition: if the primal is a convex problem (i.e. f and $h_1, \cdots, h_m$ are convex, $\ell_1, \cdots, \ell_r$ are affine, and there exists at least one strictly feasible $x \in \mathbb{R}^n$, meaning

$$h_1(x) < 0, \cdots, h_m(x) < 0 \qquad \text{and} \qquad \ell_1(x) = 0, \cdots, \ell_r(x) = 0$$

then strong duality holds.

This is a pretty weak condition. It can be further refined: need strict inequalities only over functions h_i that are not affine.

For LPs

- Easy to check that the dual of the dual LP is the primal LP
- Refined version of Slater's condition: strong duality holds for an LP if it is feasible
- Apply same logic to its dual LP: strong duality holds if it is feasible
- Hence strong duality holds for LPs, except when both primal and dual are infeasible.

In other words, we pretty much always have strong duality for LPs.

Summary

Given a minimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & h_i(x) \leq 0, i = 1, \cdots, m \\ & \ell_j(x) = 0, j = 1, \cdots, r \end{aligned}$$

we defined the Lagrangian

$$L(x,u,v) = f(x) + \sum_{i=1}^{m} u_i h_i(x) + \sum_{j=1}^{r} v_j \ell_j(x)$$

and the Lagrange dual function:

$$g(u,v) = \min_x L(x,u,v)$$

and the subsequent dual problem is

$$\begin{aligned} \max_{u,v} \quad & g(u,v) \\ \text{subject to} \quad & u \geq 0 \end{aligned}$$

Properties

Important properties

- Dual problem is always convex, i.e. g is always concave (even if primal problem is not convex)
- The primal and dual optimal values, $f^\star$ and $g^\star$, always satisfy weak duality $f^\star \geq g^\star$
- Slater's condition: for convex primal, if there is an x such that

$$h_1(x) < 0, \cdots, h_m(x) < 0 \qquad \text{and} \qquad \ell_1(x) = 0, \cdots, \ell_r(x) = 0$$

  then strong duality holds. Can be further refined to strict inequalities over the noonaffine $h_i, i = 1, \cdots, m$

Questions?