Math 6880/7875: Advanced Optimization
Regularization and relaxation

Akil Narayan[1]

[1]Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

Feburary 22, 2022

Regularization and relaxation

Two approaches aim to "improve" either the ?

- Regularization – augmenting the objective, typically to improve "quality" of solutions
- Relaxation – changing the objective, typically to make problem easier to solve

Regularization

The idea behind regularization

Consider the unconstrained optimization,

$$\min_{x \in \mathbb{R}^n} f(x).$$

(No serious changes if this a constrained problem instead.)

The motivating issues behind regularization are that the above problem

- may have many, spurious, solutions
- may be numerically difficult to solve due to sensitivity of f
- may have a solution that is 'unphysical' in practice

Regularization combats these issues by augmenting the objective,

$$\min_{x \in \mathbb{R}^n} f(x) + \lambda R(x),$$

where $\lambda > 0$ is a *regularization parameter*, and $R(x) \geqslant 0$ is the *regularization function*.

Frequently R is chosen to penalize 'bad' solution behavior.

The idea behind regularization

Consider the unconstrained optimization,

$$\min_{x \in \mathbb{R}^n} f(x).$$

(No serious changes if this a constrained problem instead.)

The motivating issues behind regularization are that the above problem

- may have many, spurious, solutions
- may be numerically difficult to solve due to sensitivity of f
- may have a solution that is 'unphysical' in practice

Regularization combats these issues by augmenting the objective,

$$\min_{x \in \mathbb{R}^n} f(x) + \lambda R(x),$$

where $\lambda > 0$ is a *regularization parameter*, and $R(x) \geqslant 0$ is the *regularization function*.

Frequently R is chosen to penalize 'bad' solution behavior.

Tikhonov regularization

Perhaps the simplest regularization is Tikhonov regularization:

$$\min_{x \in \mathbb{R}^n} f(x) + \lambda \|x\|_2^2.$$

This type of regularization is interpretable by considering extremes:

- $\lambda \downarrow 0$: the objective converges to f, and if $x_*(\lambda)$ denotes a solution to the above problem at a fixed λ, then with some assumptions on f one has that $x_*(\lambda) \to x_*(0)$.
- $\lambda \uparrow \infty$: the regularization term dominates f (assuming f is finite everywhere) and $x_*(\lambda) \to 0$

Tikhonov regularization is well-understood in several contexts.

Tikhonov regularization for least squares

In a regularized linear least squares problem, we have

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|x\|_2^2.$$

Note that this is also a linear least squares problem....

The effect of the regularization can be deduced by considering the normal equations:

$$(A^T A + \lambda I) x_*(\lambda) = A^T b.$$

The effect is to make the normal equations better conditioned. (E.g., if $A^T A$ is rank-deficient) Furthermore, one can show in this case that

$$\lim_{\lambda \downarrow 0} x_*(\lambda) = \arg\min \|x\|_2^2 \quad \text{subject to } \|Ax - b\|_2^2 \text{ is minimized.}$$

I.e., in the limit this regularization produces the minimal-norm least squares solution.

Tikhonov regularization for least squares

In a regularized linear least squares problem, we have

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|x\|_2^2.$$

Note that this is also a linear least squares problem....

The effect of the regularization can be deduced by considering the normal equations:

$$(A^T A + \lambda I) x_*(\lambda) = A^T b.$$

The effect is to make the normal equations better conditioned. (E.g., if $A^T A$ is rank-deficient) Furthermore, one can show in this case that

$$\lim_{\lambda \downarrow 0} x_*(\lambda) = \arg\min \|x\|_2^2 \quad \text{subject to } \|Ax - b\|_2^2 \text{ is minimized.}$$

I.e., in the limit this regularization produces the minimal-norm least squares solution.

Tikhonov regularization for least squares

In a regularized linear least squares problem, we have

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 + \lambda \|x\|_2^2.$$

Note that this is also a linear least squares problem....

The effect of the regularization can be deduced by considering the normal equations:

$$(A^T A + \lambda I) x_*(\lambda) = A^T b.$$

The effect is to make the normal equations better conditioned. (E.g., if $A^T A$ is rank-deficient) Furthermore, one can show in this case that

$$\lim_{\lambda \downarrow 0} x_*(\lambda) = \arg\min \|x\|_2^2 \quad \text{subject to } \|Ax - b\|_2^2 \text{ is minimized.}$$

I.e., in the limit this regularization produces the minimal-norm least squares solution.

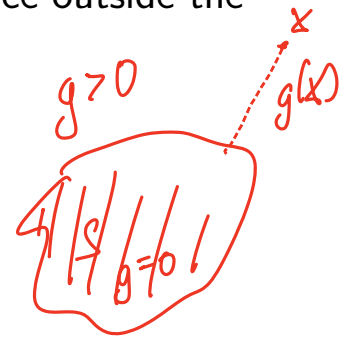
Penalty methods

Penalty methods are a type of regularization. Consider:

$$\min_{x \in S} f(x).$$

Instead of directly solving this constrained optimization problem, penalty methods convert it into an unconstrained problem by regularizing presence outside the feasible set:

$$\min_{x \in S} f(x) + \lambda g(x),$$



where $g(x)$ (typically) satisfies:

$$g(x) = 0, \ \ x \in S$$
$$g(x) > 0, \ \ x \notin S.$$

For example, $g(x) = \max\{0, -\mathrm{sdist}(x, S)\}$ is a common choice, where sdist is the signed distance function to S:

$$\mathrm{sdist}(x, S) = \begin{cases} \mathrm{dist}(x, \partial S), & x \in S \\ -\mathrm{dist}(x, \partial S), & x \notin S \end{cases}$$

Penalty algorithm outline

$$\min_{x \in S} f(x) + \eta g(x), \tag{1}$$

Penalty methods typically solve several unconstrained optimization problems:

1. Initialize initial guess x_0, penalty parameter $\eta > 0$, and amplification factor $M > 1$.
2. Solve (2) to obtain unconstrained solution $x(\eta)$
3. Set $\eta \leftarrow M\eta$
4. Set $x_0 \leftarrow x(\eta)$. Return to step 2.

The expectation is that as η increases, the solution to the unconstrained problem approaches the solution of the original constrained problem.

There is typically no guarantee that $x(\eta)$ for any given η is feasible.

Interior Point Methods, I

Interior Point Methods/Barrier Methods are a stronger type of penalty method approach: Iterates are forced to be feasible at each step.

$$\min_{x \in S} f(x) \quad \longrightarrow \quad \min_{x} f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x)), \tag{2}$$

where

$$x \in S \quad \longleftrightarrow \quad g_i(x) \leqslant 0, \quad i \in [P],$$

and this time we are interested in sending η_i to 0.

At a high level, we just chose a different function, $\log(-g_i(x))$, which is called a *logarithmic barrier function*.

However, the advantages are that this logarithmic barrier is much stronger than a signed distance function, and it ensures that solutions are feasible – i.e., that solutions are *interior* to S.

Interior Point Methods, I

Interior Point Methods/Barrier Methods are a stronger type of penalty method approach: Iterates are forced to be feasible at each step.

$$\min_{x \in S} f(x) \quad \longrightarrow \quad \min_{x} f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x)), \tag{2}$$

where

$$x \in S \quad \longleftrightarrow \quad g_i(x) \leqslant 0, \quad i \in [P],$$

and this time we are interested in sending η_i to 0.

At a high level, we just chose a different function, $\log(-g_i(x))$, which is called a *logarithmic barrier function*.

However, the advantages are that this logarithmic barrier is much stronger than a signed distance function, and it ensures that solutions are feasible – i.e., that solutions are *interior* to S.

Interior Point Methods, II

$$\min_x f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x)).$$

Setting the gradient of this function to zero results in

$$\nabla f + \sum_{i=1}^{P} \frac{-\eta_i}{g_i(x)} \nabla g_i(x) = 0.$$

The form above suggests introduction of dual/Lagrange variables:

$$\lambda_i := \frac{-\eta_i}{g_i(x)} \geqslant 0,$$

so we can rewrite the gradient as,

$$\nabla f + \sum_{i=1}^{P} \lambda_i \nabla g_i(x) = 0.$$

I.e., this is KKT stationarity.

Interior Point Methods, III

$$\min_{x \in S} f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x)),$$

$g_i(x) \leq 0$

with conditions

$$\nabla f + \sum_{i=1}^{P} \lambda_i \nabla g_i(x) = 0,$$
$$(-g_i(x))\lambda_i = \eta_i.$$

$\rightarrow -g_i(x)\left[\lambda_i - \frac{\eta_i}{-g_i(x)}\right] = 0$

The first condition is augmented Lagrangian stationarity, and the second condition is type of "perturbed" complementary slackness.

Interior point methods are numerical solvers for (x, λ) satisfying the above conditions.
A simple version uses Newton's method on the objective $f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x))$, subject to $(-g_i(x))\lambda_i = \eta_i$.

Again, iterates are forced to be feasible, and the barrier function promotes optima in the interior of S.

Like generic penalty methods, the problem is solved repeatedly, sending $\eta_i \downarrow 0$, reusing previous solutions as initial guesses.

Interior Point Methods, III

$$\min_{x \in S} f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x)),$$

with conditions

$$\nabla f + \sum_{i=1}^{P} \lambda_i \nabla g_i(x) = 0,$$

$$(-g_i(x))\lambda_i = \eta_i.$$

The first condition is augmented Lagrangian stationarity, and the second condition is type of "perturbed" complementary slackness.

Interior point methods are numerical solvers for (x, λ) satisfying the above conditions.
A simple version uses Newton's method on the objective $f(x) - \sum_{i=1}^{P} \eta_i \log(-g_i(x))$, subject to $(-g_i(x))\lambda_i = \eta_i$.

Again, iterates are forced to be feasible, and the barrier function promotes optima in the interior of S.

Like generic penalty methods, the problem is solved repeatedly, sending $\eta_i \downarrow 0$, reusing previous solutions as initial guesses.

Other regularization examples

- In neural networks, one regularizes learning by, e.g., an ℓ^2-type term involving the network weights + biases to combat overfitting.
- In similar statistical fitting problems, a sparsity-promoting term is added to encourage "simpler" models, i.e., data-fitting models with fewer active features. (E.g., LASSO/basis pursuit)
- In graph learning, a graph Laplacian regularization is employed to promote simplicity of the learned graph
- In (ill-posed) inverse problems, a regularization term is sometimes used to ensure some type of unique solution.
- In algorithms, regularization is used to make operations more stable. (Cf. Gauss-Newton vs. Levenberg-Marquardt)
- The nuclear norm of a matrix is often used in matrix completion in the context of recovery of low-rank, sparse matrices

Other regularization examples

- In neural networks, one regularizes learning by, e.g., an ℓ^2-type term involving the network weights + biases to combat overfitting.
- In similar statistical fitting problems, a sparsity-promoting term is added to encourage "simpler" models, i.e., data-fitting models with fewer active features. (E.g., LASSO/basis pursuit)
- In graph learning, a graph Laplacian regularization is employed to promote simplicity of the learned graph
- In (ill-posed) inverse problems, a regularization term is sometimes used to ensure some type of unique solution.
- In algorithms, regularization is used to make operations more stable. (Cf. Gauss-Newton vs. Levenberg-Marquardt)
- The nuclear norm of a matrix is often used in matrix completion in the context of recovery of low-rank, sparse matrices

Other regularization examples

- In neural networks, one regularizes learning by, e.g., an ℓ^2-type term involving the network weights + biases to combat overfitting.
- In similar statistical fitting problems, a sparsity-promoting term is added to encourage "simpler" models, i.e., data-fitting models with fewer active features. (E.g., LASSO/basis pursuit)
- In graph learning, a graph Laplacian regularization is employed to promote simplicity of the learned graph
- In (ill-posed) inverse problems, a regularization term is sometimes used to ensure some type of unique solution.
- In algorithms, regularization is used to make operations more stable. (Cf. Gauss-Newton vs. Levenberg-Marquardt)
- The nuclear norm of a matrix is often used in matrix completion in the context of recovery of low-rank, sparse matrices

Other regularization examples

- In neural networks, one regularizes learning by, e.g., an ℓ^2-type term involving the network weights + biases to combat overfitting.
- In similar statistical fitting problems, a sparsity-promoting term is added to encourage "simpler" models, i.e., data-fitting models with fewer active features. (E.g., LASSO/basis pursuit)
- In graph learning, a graph Laplacian regularization is employed to promote simplicity of the learned graph
- In (ill-posed) inverse problems, a regularization term is sometimes used to ensure some type of unique solution.
- In algorithms, regularization is used to make operations more stable. (Cf. Gauss-Newton vs. Levenberg-Marquardt)
- The nuclear norm of a matrix is often used in matrix completion in the context of recovery of low-rank, sparse matrices

Other regularization examples

- In neural networks, one regularizes learning by, e.g., an ℓ^2-type term involving the network weights + biases to combat overfitting.
- In similar statistical fitting problems, a sparsity-promoting term is added to encourage "simpler" models, i.e., data-fitting models with fewer active features. (E.g., LASSO/basis pursuit)
- In graph learning, a graph Laplacian regularization is employed to promote simplicity of the learned graph
- In (ill-posed) inverse problems, a regularization term is sometimes used to ensure some type of unique solution.
- In algorithms, regularization is used to make operations more stable. (Cf. Gauss-Newton vs. Levenberg-Marquardt)
- The nuclear norm of a matrix is often used in matrix completion in the context of recovery of low-rank, sparse matrices

$$(J^T J) \; x = J^T b$$
$$\uparrow$$
$$+\lambda I$$

Other regularization examples

- In neural networks, one regularizes learning by, e.g., an ℓ^2-type term involving the network weights + biases to combat overfitting.
- In similar statistical fitting problems, a sparsity-promoting term is added to encourage "simpler" models, i.e., data-fitting models with fewer active features. (E.g., LASSO/basis pursuit)
- In graph learning, a graph Laplacian regularization is employed to promote simplicity of the learned graph
- In (ill-posed) inverse problems, a regularization term is sometimes used to ensure some type of unique solution.
- In algorithms, regularization is used to make operations more stable. (Cf. Gauss-Newton vs. Levenberg-Marquardt)
- The nuclear norm of a matrix is often used in matrix completion in the context of recovery of low-rank, sparse matrices

Relaxation

Relaxation is an approximation strategy generally employed to make problems "easier" to solve.

We've already seen one example of relaxation from compressed sensing:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \text{ subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

$$\downarrow$$

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

In this particular case, we saw that under certain conditions on $\boldsymbol{A}$, the solution to these two problems are the same.

In general, relaxation methods *change* the optimization problem in a different one.

The hope is that the solution to the relaxed problem is "close" to the solution of the original problem.

Relaxation

Relaxation is an approximation strategy generally employed to make problems "easier" to solve.

We've already seen one example of relaxation from compressed sensing:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \text{ subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

$$\downarrow$$

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ subject to } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

In this particular case, we saw that under certain conditions on $\boldsymbol{A}$, the solution to these two problems are the same.

In general, relaxation methods *change* the optimization problem in a different one.

The hope is that the solution to the relaxed problem is "close" to the solution of the original problem.

There are "bad" relaxations

Not every sensible relaxation "works" as intended. Consider:

$$\min \|\boldsymbol{x}\|_1 \text{ subject to } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \text{ is minimized}$$

This is a regularization problem: the norm $\|\cdot\|_1$ is used as a regularizer, typically to promote sparsity of $\boldsymbol{x}$.

The $\|\cdot\|_1$ is generally harder to work (say than the $\|\cdot\|_2$) norm, e.g., since it's not differentiable everywhere.

One might be tempted to relax $\|\cdot\|_1$ to $\|\cdot\|_2$:

$$\min \|\boldsymbol{x}\|_2 \text{ subject to } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \text{ is minimized}$$

This problem is certainly easier to solve (it's a minimum norm least squares problem).

But the solution x to this relaxed problem is not (generally) sparse, and hence gives quite a different answer than the $\|\cdot\|_1$ problem.

There are "bad" relaxations

Not every sensible relaxation "works" as intended. Consider:

$$\min \|\boldsymbol{x}\|_1 \text{ subject to } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \text{ is minimized}$$

This is a regularization problem: the norm $\|\cdot\|_1$ is used as a regularizer, typically to promote sparsity of $\boldsymbol{x}$.

The $\|\cdot\|_1$ is generally harder to work (say than the $\|\cdot\|_2$) norm, e.g., since it's not differentiable everywhere.

One might be tempted to relax $\|\cdot\|_1$ to $\|\cdot\|_2$:

$$\min \|\boldsymbol{x}\|_2 \text{ subject to } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \text{ is minimized}$$

This problem is certainly easier to solve (it's a minimum norm least squares problem).

But the solution $\boldsymbol{x}$ to this relaxed problem is not (generally) sparse, and hence gives quite a different answer than the $\|\cdot\|_1$ problem.

General relaxation

Consider the constrained optimization problem,

$$\min_{x \in S} f(x).$$

A general relaxation of this problem identifies (i) a lower bound g for the function f, and/or (ii) a larger feasible set T:

$$\min_{x \in T} g(x),$$

where

$$f \leqslant g \text{ on } S, \qquad S \subseteq T.$$

Relaxation guarantees

$$\min_{x \in S} f(x) \quad \longrightarrow \quad \min_{x \in T} g(x)$$

How does a solution x_* of the original problem compare to a solution x_{**} of the relaxed problem?

- We always have that $f(x_*) \geqslant f(x_{**})$, so that x_* is an upper bound for the relaxed problem.
- A much stronger statement: Assume that $f(x) = g(x)$ for all $x \in S$. If the relaxed solution $x_{**} \in S$, then it is also optimal for the original problem.

Relaxation guarantees

$$\min_{x \in S} f(x) \quad \longrightarrow \quad \min_{x \in T} g(x)$$

How does a solution x_* of the original problem compare to a solution x_{**} of the relaxed problem?

- We always have that $f(x_*) \geqslant f(x_{**})$, so that x_* is an upper bound for the relaxed problem.
- A much stronger statement: Assume that $f(x) = g(x)$ for all $x \in S$. If the relaxed solution $x_{**} \in S$, then it is also optimal for the original problem.

Relaxations

Some typical examples of relaxations are

- Euclidean-type "norms"
  - Sparsity: $\|\cdot\|_0 \to \|\cdot\|_1$
  - Smoothness: $\|\cdot\|_1 \to \|\cdot\|_2$
- Matrix rank: $\mathrm{rank}(\cdot) \to \|\cdot\|_{\mathrm{NN}}$, the nuclear norm
- (Mixed) Integer linear programs: discrete $\to$ continuous
- Lagrangian relaxation methods

$$\|A\|_{NN} = \sum_{i=1}^{r} \sigma_i(A)$$

Lagrangian relaxation

Lagrangian relaxation aims to transfer "difficult to handle" constraints to the objective. Consider:

$$\min_{x \in S_1 \cap S_2} f(x),$$

where S_1 is "easy" to handle, and S_2 is "hard" to handle.

The theory is general and rich, but we'll specialize to linear programming and remove the "easy" constraints to state results:

$$\begin{aligned} &\text{Minimize } \boldsymbol{c}^T \boldsymbol{x} \\ &\text{Subject to } \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}. \end{aligned}$$

The Lagrangian Bounding Principle

$$\text{Minimize } \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{Subject to } \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}.$$

Let $\boldsymbol{x}_*$ be solution to the above problem.
The relaxed version of the problem is

$$\min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

We will need some extra notation to state results:

$$L(\lambda) := \min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

A simple result:

Lemma

For any $\lambda \geqslant 0$, $L(\lambda) \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$.

The Lagrangian Bounding Principle

$$\text{Minimize } \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{Subject to } \boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}.$$

Let $\boldsymbol{x}_*$ be solution to the above problem.
The relaxed version of the problem is

$$\min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

We will need some extra notation to state results:

$$L(\lambda) := \min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

A simple result:

Lemma

For any $\lambda \geqslant 0$, $L(\lambda) \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$.

The Lagrangian dual problem

$$L(\lambda) = \min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$
$$L(\lambda) \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$$

To make the Lagrangian bound as tight as possible, we might try to solve the following Lagrangian dual problem:

$$L_* = \max_{\lambda \geqslant 0} L(\lambda).$$

The definition of this problem immediately leads to the following.

Theorem (Weak duality)

$L_* \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$

The Lagrangian dual problem

$$L(\lambda) = \min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$
$$L(\lambda) \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$$

To make the Lagrangian bound as tight as possible, we might try to solve the following Lagrangian dual problem:

$$L_* = \max_{\lambda \geqslant 0} L(\lambda).$$

The definition of this problem immediately leads to the following.

Theorem (Weak duality)

$L_* \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$

Optimality certificates

The lower bound guarantee $L_* \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$ doesn't in general imply equality.

The difference between these two, $\boldsymbol{c}^T \boldsymbol{x}_* - L_*$ is the duality gap, roughly speaking is a fudge factor that we suffer when solving the dual problem relative to the primal one.

Even in the presence of a duality gap, the Lagrangian gives us a quantitative measure of how far we are from optimality:

$$\frac{\boldsymbol{c}^T \boldsymbol{x} - L(\lambda)}{L(\lambda)},$$

is a relative daulity gap measure, and is a quantitative prescription on how far from a solution we are.

Of course, such a measure is most useful when the duality gap vanishes.

Optimality certificates

The lower bound guarantee $L_* \leqslant \boldsymbol{c}^T \boldsymbol{x}_*$ doesn't in general imply equality.

The difference between these two, $\boldsymbol{c}^T \boldsymbol{x}_* - L_*$ is the duality gap, roughly speaking is a fudge factor that we suffer when solving the dual problem relative to the primal one.

Even in the presence of a duality gap, the Lagrangian gives us a quantitative measure of how far we are from optimality:

$$\frac{\boldsymbol{c}^T \boldsymbol{x} - L(\lambda)}{L(\lambda)}, \sim \frac{c^T x_* - L_*}{L_*} \; ?$$

is a relative daulity gap measure, and is a quantitative prescription on how far from a solution we are.

Of course, such a measure is most useful when the duality gap vanishes.

Optimality for linear programming

$$L(\lambda) = \min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

One of the more useful facts about linear programming is that the duality gap vanishes under mild assumptions.

Theorem (Optimality test)

Suppose $(\boldsymbol{x}, \lambda)$ is such that $\boldsymbol{x}$ is feasible (i.e., $\boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}$), and the pair satisifes the complementary slackness condition,

$$\lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = 0,$$

then $L(\lambda) = L_*$, and $\boldsymbol{x} = \boldsymbol{x}_*$.

The result above is a cornerstone of many linear programming solvers, which exploit duality in computations.

Typically the dual problem solvers utilize descent types of algorithms.

Optimality for linear programming

$$L(\lambda) = \min_x \boldsymbol{c}^T \boldsymbol{x} + \lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}).$$

One of the more useful facts about linear programming is that the duality gap vanishes under mild assumptions.

Theorem (Optimality test)

Suppose $(\boldsymbol{x}, \lambda)$ is such that $\boldsymbol{x}$ is feasible (i.e., $\boldsymbol{A}\boldsymbol{x} \leqslant \boldsymbol{b}$), and the pair satisifes the complementary slackness condition,

$$\lambda^T (\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = 0,$$

then $L(\lambda) = L_*$, and $\boldsymbol{x} = \boldsymbol{x}_*$.

The result above is a cornerstone of many linear programming solvers, which exploit duality in computations.

Typically the dual problem solvers utilize descent types of algorithms.

Related papers I

- Claire Boyer, Antonin Chambolle, Yohann De Castro, Vincent Duval, Frédéric de Gournay, and Pierre Weiss, *On Representer Theorems and Convex Regularization*, SIAM Journal on Optimization **29** (2019), no. 2, 1260–1281.
- Julianne Chung, Matthias Chung, and Dianne P. O'Leary, *Optimal regularized low rank inverse approximation*, Linear Algebra and its Applications **468** (2015), 260–269.
- Christian Clason, Carla Tameling, and Benedikt Wirth, *Convex Relaxation of Discrete Vector-Valued Optimization Problems*, SIAM Review **63** (2021), no. 4, 783–821.
- A. M. Geoffrion, *Duality in Nonlinear Programming: A Simplified Applications-Oriented Development*, SIAM Review **13** (1971), no. 1, 1–37.
- Per Christian Hansen, *Analysis of Discrete Ill-Posed Problems by Means of the L-Curve*, SIAM Review **34** (1992), no. 4, 561–580.
- Jorge Nocedal and S. Wright, *Numerical Optimization*, 2 ed., Springer Series in Operations Research and Financial Engineering, Springer-Verlag, New York, 2006.