# Lecture 1: Quantile Regression and Linear Programming

Mandatory readings:

Koenker (2005): 1, 2, 6, 8.8

Bertsimas and Tsitsiklis (1997): 1.1-1.8, 2.1-2.6, 3.1, 3.2, 3.5, 3.6.

## 1.0 Introduction: Quantile Regression

Median regression is as important a tool for regression analysis as is ordinary linear regression (OLS), and is a more robust estimator. In addition, quantile regression on multiple quantiles offers a more complete description of the conditional distribution of the outcome variable.

That being said, OLS has remained the mainstay of regression analysis. Main hurdles facing quantile regressions are difficulties of interpretation (e.g., in program evaluation) and less transparent computations (we don't exactly get a closed form the way we do with OLS). These notes attempt to overcome and hopefully trivialize these difficulties.

## 1.1 Definitions and Interpretations

Let $\rho_\tau$ be the check function

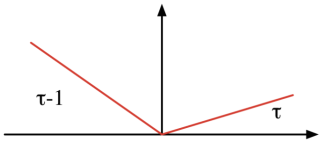

**Figure 1** *The check function is the loss function used in quantile regression.*

$$\begin{aligned}\rho_\tau(u) &= u \cdot (\tau - \mathbf{1}\{u < 0\}) \\ &= \tau \cdot u_+ + (1-\tau) \cdot u_-,\end{aligned}$$

where $u_+ = \max\{u, 0\}$ and $u_- = \max\{-u, 0\}$. The $\tau^{\text{th}}$ quantile of $\{Y_1, ..., Y_n\}$ may be obtained as

$$\min_\xi \sum_{i=1}^n \rho_\tau(Y_i - \xi).$$

As with the average and linear (least-squares) regression, we may want to parametrize the quantile in terms of some explanatory variables $x_i$, $i = 1, ..., n$. **Quantile regression** imposes a linear form on the quantile solution

$$\min_{\beta} \sum_{i=1}^{n} \rho_{\tau}(Y_i - x_i^T \beta). \tag{1}$$

Note that for $\tau = 0.5$, this problem is simply regression under the $\ell_1$ loss, or median regression,

$$\min_{\beta} \sum_{i=1}^{n} \left| Y_i - x_i^T \beta \right| .$$

We do not investigate the robustness properties of quantile regression –resulting from the use of an $\ell_1$ instead of, say, $\ell_2$ loss– but they are studied in Koenker (2005) and further motivate median regression in particular as a competitor to OLS.

Another important point which is not further explored in this note but which was discussed in class is that, as in the OLS case, one must be careful when interpreting the least-squares regression coefficient on the treatment variable in a causal framework. In fact, some interpretative pitfalls appear to be more beguiling in the quantile regression than in the OLS case.[1]

## 1.2 Modern Introduction to Quantile Regression

### 1.2.0 Modern Introduction to Least-Squares Regression

Modern treatments of least-squares regression are centered around the best linear predictor (BLP) in population. The BLP is easily shown to be the *de facto* target –that which is consistently estimated– of the least-squares estimate of the regression function, and thus affords interpretation of the least-squares estimand in the ubiquitous case of misspecification.

In other words, juvenile or antiquated treatments of least-squares regression may present it as estimating the regression function $E[Y|X]$ where the pair $X, Y$ is distributed according to some unknown law $F$. But if the regression function is not well specified as linear, i.e., if it is not the case that $E[Y|X] = X\beta$ for some $\beta$, then our linear estimate $X\hat{\beta}$ cannot consistently estimate it, even asymptotically. More to the point, we are left not knowing what we are estimating!

The natural question is then: are we targeting –consistently estimating– an object as close as possible to the desired regression function $E[Y|X]$ in some precise, rigorous

[1] It is easy to think of the average effect as impacting that moment of the population. Quantile effects should be interpreted similarly, but it may be tempting –and wrong– to conclude that the effect at the $\tau^{\text{th}}$ quantile is the effect on the subjects at the $\tau^{\text{th}}$ quantile "before" treatment, or in the control group.

sense? The answer is a resonant yes, and a modern treatment of least-squares regression is built around this robust identification perspective.

The intuition of this approach is captured by a triad of statistical objects:

1. The **regression function**,

$$E[Y|X] = \arg\min_{\tilde{Y}} E\left[\left(Y - \tilde{Y}\right)^2\right], \tag{2}$$

   where $\tilde{Y}$ is an unrestricted function of $X$ and $X, Y \sim F$, with $X$ a vector and $Y$ a scalar. This is the object we are hoping to get as close to as possible but may not estimate precisely using least-squares linear regression, even asymptotically, if it is not itself linear. *The regression function is what we wish we could estimated in the limit.*

2. The **best linear predictor in sample**,

$$\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \arg\min_{\tilde{\mathbf{Y}}=\mathbf{X}\beta,\ \beta\in\mathbb{R}^p} \left\|\mathbf{Y} - \tilde{\mathbf{Y}}\right\|_2^2, \tag{3}$$

   where $\tilde{\mathbf{Y}}$ is restricted to be a linear function of $\mathbf{X}$, $\mathbf{X} \in \mathbb{R}^{n\times p}$, $\mathbf{Y} \in \mathbb{R}^p$, and $n$ is the sample size. *This is what we have.* It yields a regression coefficient estimate

$$\hat{\beta}_{OLS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}.$$

3. The **best linear predictor in population**,

$$E^*[Y|X] = \arg\min_{\tilde{Y}\ \text{linear}} E\left[\left(Y - \tilde{Y}\right)^2\right], \tag{4}$$

   where $\tilde{Y}$ is restricted to be a linear function of $X$. *This is what we do estimate in the limit.* In particular, $E^*[Y|X] = X\beta_{OLS}$ and

$$\hat{\beta}_{OLS} \to \beta_{OLS}.$$

Answering our earlier prompt, we are getting the "closest" linear approximation to the nonlinear function we desire, and are close precisely in the sense of the unweighted $\ell_2$ distance: the best linear predictor in sample targets the best linear predictor in population, and

$$E^*[Y|X] = \arg\min_{\tilde{Y}=X\beta,\ \beta\in\mathbb{R}^p} E\left[\left(E[Y|X] - X\beta\right)^2\right]. \tag{5}$$

That is, the object we target using OLS is the best linear approximation for the regression function in the mean squared error sense.

### 1.2.1 Modern Introduction to Quantile Regression

Subsection 1.2.0 answered the question of how to interpret of the least-squares regression estimate under misspecification. The analogous question, in the case of quantile regression, is obviously of equal importance.

We have a quantile regression coefficient estimate

$$\hat{\mathbf{Y}}(\tau) = \arg \min_{\tilde{\mathbf{Y}}=\mathbf{X}\beta} \sum_{i=1}^{n} \rho_\tau(\tilde{\mathbf{Y}}_i - \mathbf{X}_i^T \beta), \tag{6}$$

and a corresponding population estimand

$$X\beta(\tau) = \arg \min_{\tilde{Y}=X\beta} E\left[\rho_\tau(Y - \tilde{Y})\right]. \tag{7}$$

If the quantile regression function $Q_Y(\tau)$ is indeed linear, then (7) is $Q_\tau(Y)$, and since (6) provides a consistent estimate of (7), it provides a consistent estimate of the true quantile regression function $Q_\tau(Y)$.

However, the ubiquitous case is again that in which $Q_Y(\tau)$ is nonlinear, i.e., the case of misspecification in linear quantile regression.

We are thus left anew with the existential question: what is the sense, if any, in which we are getting the best linear approximation $X\beta$ to the quantile regression function $Q_Y(\tau)$? Is there a quantile regression analog to (5)?

This question was answered in a very elegant paper by Angrist, Chernozhukov, and Fernández-Val (2006).

They find that quantile regression indeed has an interpretation as a best linear approximation to the nonlinear object of interest, and that it minimizes the distance to the object of interest according to a specific weighted $\ell_2$ norm. We give their result.

Define the quantile regression process,

$$\beta(\tau) := \arg \min_{\beta \in \mathbb{R}^p} E\left[\rho_\tau\left(Y - X^T\beta\right)\right]. \tag{8}$$

Of course (8) does tell us that the linear quantile regression estimate minimizes prediction loss according to the check function, but that is an unsatisfying answer. We would like an analog to (5). We want to know if, and in what sense, the quantile regression estimate approximates the true, potentially nonlinear, quantile regression function.

The following theorem is the central result of Angrist, Chernozhukov, and Fernández-Val (2006).

**Theorem 1** *Suppose that the conditional density $f_Y(y|X)$ exists a.s., the $E[Y]$, $E[Q_\tau(Y|X)]$, and $E\|X\|$ are finite, and that $\beta(\tau)$ solves (8) uniquely. Then*

$$\beta(\tau) = \arg\min_{\beta\in\mathbb{R}^p} E\left[\omega_\tau(X,\beta)\cdot\left(X^T\beta - Q_\tau(Y|X)\right)^2\right], \tag{9}$$

*where*

$$\omega_\tau(X,\beta) = \int_0^1 (1-u)\cdot f_Y\left(u\cdot X^T\beta + (1-u)\cdot Q_\tau(Y|X)\middle| X\right) du \geq 0. \tag{10}$$

The weight (10) is a weighted average of $f_Y(y|X)$ from $y = Q_\tau(Y|X)$ to $y = X^T\beta$, with weight decreasing linearly. To get intuition for the weights, consider the case in which $f_Y(y|X)$ is very flat near $Q_\tau(Y|X)$ and $\triangle_\tau(X,\beta)$ is small. Then the weights $\omega_\tau(X,\beta)$ are approximately proportional to the density of the outcome variable evaluated at the estimated quantile $f_Y\left(Q_\tau(Y|X)\middle| X\right)$. The intuitive reading is that, according to this loss, it is more important to get the slope of the quantile regression function right for $X$'s that generate $Y$'s more likely to be near their true conditional quantile. In that sense, we can think of the "effective weights" as $\omega_\tau(X,\beta)f_X(X)$ where is the density of $X$.

Theorem 1 almost delivers the equivalent of (5), but not quite since the weights $\omega_\tau(X,\beta)$ depend on $\beta$ and (9) is not a weighted $\ell_2$ norm for the difference $X^T\beta - Q_\tau(Y|X)$. They provide a corollary affording us precisely that interpretation.

**Corollary 1**

*Suppose that the conditional density $f_Y(y|X)$ exists and is bounded a.s., $E[Y]$, $E[Q_\tau(Y|X)^2]$, and $E\|X\|^2$ are finite, and $\bar\beta(\tau)$ is the solution of (9). Then*

$$\bar\beta(\tau) = \beta(\tau) = \arg\min_{\beta\in\mathbb{R}^p} E\left[\bar\omega_\tau(X)\cdot\triangle_\tau^2(X,\beta)\right],$$

*where*

$$\bar\omega_\tau(X) = \int_0^1 (1-u)\cdot f_Y\left(u\cdot X^T\bar\beta(\tau) + (1-u)\cdot Q_\tau(Y|X)\middle| X\right) du.$$

That is, if we fixed the weights $\omega_\tau(X,\beta)$ at the value of the solution $\bar\beta$ (in fact the result holds for any $\bar\beta$) then quantile regression may be interpreted as the best linear approximation to the quantile regression function according to the weighted $\ell_2$ norm,

$$X\beta(\tau) = \arg\min_{\tilde Y \text{ linear}} \left\|\tilde Y - Q_\tau(Y|X)\right\|_{\bar\omega_\tau},$$

where $\|Z\|_{\bar{\omega}_\tau} = E\left[\bar{\omega}_\tau(X) \cdot Z^2\right]$, $\bar{\omega}_\tau(X) = \omega_\tau(X, \bar{\beta})$ , and $\bar{\beta}$ is the solution to (8). We have our best linear predictor interpretation! Although the weighting scheme may not be particularly intuitive in practice, the important conclusion is that under misspecification $X\hat{\beta}(\tau)$ is targeting a regression function which is as close as possible to $Q_\tau(Y|X)$, with respect to a reasonable notion of distance.

Angrist, Chernozhukov, and Fernández-Val (2006) show that once you cast the population quantile regression problem in the $\ell_2$ framework, you recuperate traditional OLS results based on orthogonal projections such as residual regression and omitted variable bias.

In a very nice and important application of their result, Angrist, Chernozhukov, and Fernández-Val (2006) derive a sandwich formula for the robust asymptotic covariance covariance of $\hat{\beta}(\tau)$, $\tau \in (0,1)$.

✹

We give, for completion, the limit distribution of $\hat{\beta}_\tau$ evaluated at a single, given quantile $\tau \in (0,1)$ and asymptotically consistent for $\beta_0$.

**Theorem 2 (Van der Vaart, 1998, Theorem 5.23)**

*Under technical conditions, if the data are i.i.d. and* $\hat{\beta} \xrightarrow{P} \beta_0$ , *then*

$$\sqrt{n}\,(\beta - \beta_0) \xrightarrow{d} N(0, H^{-1}\Omega H^{-1}),$$

*where* $\Omega = E\left[\left(\frac{\partial \rho_\tau}{\partial \beta}(Z, \beta_0)\right)\left(\frac{\partial \rho_\tau}{\partial \beta}(Z, \beta_0)\right)^T\right]$ *and* $H = \frac{\partial^2}{\partial \beta^2} E\left[\rho_\tau(Z, \beta)\right]\Big|_{\beta_0}$.

Verifying that the technical assumptions hold for quantile regression is a minor affaire. In any case, the gradient and Hessian need to be computed explicitly to give a closed form solution for the asymptotic variance.

The first derivative of $\rho_\tau$ is

$$\frac{\partial \rho_\tau}{\partial \beta}(Z, \beta) = X\left(\mathbf{1}\{y - X^T\beta < 0\} - \tau\right).$$

Thus we have

$$\Omega = E\left[XX^T\left(\mathbf{1}\{y - X^T\beta < 0\} - \tau\right)^2\right].$$

If the linear model is exact, then $E\left[\mathbf{1}\{y - X^T\beta < 0\}\middle| X\right] = \tau$, thus by expanding the square and using iterated expectations, one gets $\Omega = E\left[XX^T\right]\tau(1-\tau)$.

The Hessian is readily obtained. Observe that, defining[2] the error $u := y - X^T\beta_0$, we can write $y - X^T\hat{\beta} = u - X^T(\hat{\beta} - \beta_0)$. Thus

$$\begin{aligned}
\frac{\partial^2}{\partial\beta^2}E\rho_\tau(Z,\beta) &= \frac{\partial}{\partial\beta}E\left[XE\left[\left(\mathbf{1}\{y - X^T\beta < 0\} - \tau\right)\big|\, X\right]\right] \\
&= E\left[X\frac{\partial}{\partial\beta}\left(\int_{-\infty}^{X^T(\beta-\beta_0)} f_{u|X}(t)dt - \tau\right)\right] \\
&= E\left[XX^Tf_{u|X}(X^T(\beta-\beta_0))\right],
\end{aligned}$$

evaluating at $\beta = \beta_0$ yields the Hessian

$$H = E\left[XX^Tf_{u|X}(0)\right] =^A E\left[XX^T\right]f_u(0),$$

where the second equality $=^A$ relied in the additional assumption: $A)$ $u|X \sim u$.

We may obtain an estimate $\hat{f}_u(0)$ either by kernel density estimation or directly. The density estimation is an unpalatable aspect of this method, it motivates inspecting alternatives for developing confidence intervals, such as the inversion of regression rankscore tests investigated in Lecture 2.

### 1.1.0 Quantile Regression as the Solution of a Linear Program

We will find it instructive to consider the quantile regression problem explicitly as a linear program. First, it will make more immediate the computational questions. Second, and perhaps most importantly, it will provide intuition for some of the (otherwise sometimes opaque) properties of the estimator.

We use matrix notation, where $Y, \varepsilon \in \mathbb{R}^n$, $X \in \mathbb{R}^{n\times p}$, $\beta \in \mathbb{R}^p$. The quantile regression problem is then written as

$$\min_{\beta} \tau\mathbf{1}_n^Tu + (1-\tau)\mathbf{1}_n^Tv \tag{11}$$

$$\text{s.t. } [X, -I, I]\begin{bmatrix}\beta\\ u\\ v\end{bmatrix} = Y \tag{12}$$

$$u, v \geq 0. \tag{13}$$

The feasible set of the quantile regression problem, $\left\{\beta : (X,-I,I)(\beta,u^T,v^T)^T = Y,\ u,v \geq u\right\}$

[2] Two aspects of this "trick" are important: the $u$'s have sample homologues $\hat{u}$, and the integration is up to 0 at $\beta = \beta_0$.

is a polyhedron (defined below). In fact, the feasible set of any linear program is a polyhedron. As we will see below, the solution of a linear program will always be attained at a vertex of this feasible polyhedron (think about it, easy!).

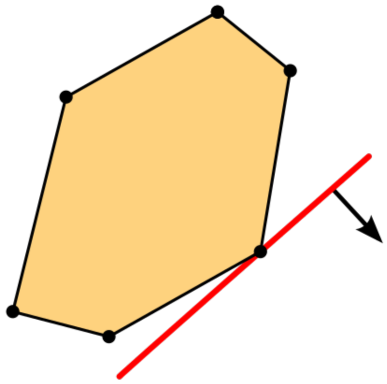

**Figure 2** *The solution of a linear program is attained at the vertex of the polyhedron describing its feasible set.*

Consequently, a deep, theorist's understanding of quantile regression requires an equally deep understanding of the nature of such a solution. We will uncover a rich relation between the algebraic formulation of a solution and its geometry. Furthermore, we will find that the geometry informs algorithms for finding solutions to quantile regression and other linear programs.

## 1.2 Linear Programming

Any optimization problem of the following form is a linear program,

$$\min \; c^T x$$

subject to

$$A_1 x \geq b_1$$

$$A_2 x \leq b_2$$

$$A_3 x = b_3$$

$$x_i \geq 0, \;\; i \in I$$

$$x_j \leq 0, \;\; j \in J,$$

for $x, c \in \mathbb{R}^n$, $A_l \in \mathbb{R}^{m_l \times n}$ and $b \in \mathbb{R}^{m_l}$ for $l = 1, 2, 3$, and $I, J \subset \{1, ..., n\}$

Many problems which may not seem to take this form at first sight are in fact linear programs. Linearly constrained programs with piecewise linear convex objective functions (Bertsimas and Tsitsiklis, p.15) are good examples. See exercises in Problem Set 3 for other examples.

Any linear program in the following form is a linear program in **standard form**,

$$\min c^T x$$

subject to

$$Ax = b$$

$$x \geq 0.$$

Any linear program may be rewritten in standard form. See exercises in Problem Set 3.

### 1.2.2 Geometry of Polyhedron and Characterization of Solutions

As observed above (and shown below), the solution of an LP is attained at the vertex of its feasible polyhedron.[3] We want to understand the properties of these vertices. For a polyhedron $P$, we say that a vector $x \in P$ is an **extreme point** of $P$ if we cannot find two vectors $y, z \in P$, both different from $x$, and a scalar $\lambda \in [0, 1]$, such that $x = \lambda y + (1-\lambda)z$. This definition is geometrically obvious.

Our study of polyhedrons as the feasible sets of linear programs will be facilitated if we can define this object of interest in the language of linear programming. Call a vector $x \in P$ a **vertex** of $P$ if there exists some $c$ such that $c^T x < c^T y$ for all $y \in P$, $y \neq x$. This is likewise geometrically obvious, and it should be transparent that it describes the same object: a vector $x \in P$ is an extreme point if and only if it is a vertex.

Extreme points and vertices give us two very geometric definitions of the object of interest. But one objective of our study is to tie the geometric interpretation to the algebraic interpretation. This can be done, for a general polyhedron, by considering its basic feasible solutions. For a polyhedron $P$ defined by linear equality and inequality constraints, a vector $x \in P$ is a **basic feasible solution** if all constraints are satisfied, all equality constraints are active[4], out of the constraints that are active at $x$, there are $n$ of them that are linearly independent. The relation between the geometric and algebraic definitions is given by the following result.

[3] It may not be unique if a face of the polytope with vertex $x^*$ is perpendicular to $c$. Since our motivation is quantile regression, we think of this as a probability zero event and do not treat it in these notes.

[4] We say that a constraint is active if it holds with equality.

**Theorem 3** *Let $P$ be a nonempty polyhedron and let $x^* \in P$. Then, the following are equivalent: $x^*$ is a vertex, $x^*$ is an extreme point, $x^*$ is a basic feasible solution.*

Proof: Bertsimas and Tsitsiklis (1997) p. 51.

✹

Since any linear program may be written in standard form, we can limit our study to polyhedrons defined as

$$P = \{x \in \mathbb{R}^n : Ax = b,\ x \geq 0\}.$$

This will deliver an even more geometrically and linear algebraically attractive formulation of basic feasible solutions. We will work under the assumption that the rows of $A \in \mathbb{R}^{m \times n}$ are linearly independent, which requires $m \leq n$.

We have a good geometric intuition for the vertices (they are...well...vertices!), and we would like to use the beautiful standard form formulation of polytopes to elicit an intuitive algebraic formulation of vertices. Obviously, a point inside of $P$ or of a sub-polyhedron (such as a face or edge) of $P$ will not be a vertex.

Any polyhedron may be defined as the intersection of half spaces defined by (nonredundant) linear inequalities. Then, by requiring $n$ such linear inequalities to hold with equality, we pin down a vertex. The same intuition carries over for a polyhedron in standard form; think of the polyhedron in the $n-m$ dimensional subspace $\{x \in \mathbb{R}^n : Ax = b\}$ defined by the intersection of the halfspaces $\{x : x_i \geq 0\}$, $i = 1, ..., n$. To pin down a vertex of the polyhedron, we need $n - m$ of these inequalities to hold with equality. This intuition is confirmed by the following theorem, which is central to the theory of linear programming and to the construction of exterior point algorithms for solving linear programs.

**Theorem 4** *Consider the constraints $Ax = b$ and $x \geq 0$ and assume that the matrix $A$ has linearly independent rows. A vector $x \in \mathbb{R}^n$ is a basic solution if and only if we have $Ax = b$, and there exist indices $B(1)$,...,$B(m)$ such that:*
*(a) The columns $A_{B(1)}, ..., A_{B(m)}$ are linearly independent,*
*(b) If $i \neq B(1), ..., B(m)$, then $x_i = 0$.*

Proof: Bertsimas and Tsitsiklis (1997) p. 53.

If $x$ is a basic feasible solution, then $x_{B(1)}, ..., x_{B(m)}$ are called **basic variables**. The others are called nonbasic. The columns $A_{B(1)}, ..., A_{B(m)}$ are called basic columns. We say they form a basis (see the "active basis" of the simple method below) and define the basis matrix

$$B = \left[A_{B(1)}, ..., A_{B(m)}\right].$$

The basic variables are determined as the unique solution of $Bx_B = b$,

$$x_b = B^{-1}b.$$

This is key. This is what will allow us to think of a solution of a linear program as $x_B$, a parameter in a continuous space, but also as a choice of basis –by their index $(B(1), ..., B(m)) \subset \{1, ..., n\}$. This highlights the discrete nature (!) of the solution space of a linear program.

### 1.2.3 Simplex Method

The development of the simplex method uses the geometry of linear programming. In developing the simplex method, we will always refer to the linear program in standard form,

$$\min\ c^T x$$

subject to

$$Ax = b,$$

$$x \geq 0,$$

where $A \in \mathbb{R}^{m \times n}$, $m \leq n$.

The idea at the core of the simplex method is a beautiful one. Since we know that the optimal solution of a "well-behaved" linear program will be at a vertex (of which which there are finitely many) and the problem is thus effectively discretized, we find a way of moving from vertex to vertex, along edges, in a direction that always reduces cost $c^T x$ and brings us closer to the solution.

The quantities informing our moves along, and choice of, edges are extracted from the optimality conditions of the linear program. We can break down the tasks for constructing a step of the simplex method in *three conceptually distinct parts.* First, we need to be able to move a point along edges of a polyhedron. Second, we need to recognize when to stop moving along the edge –so we don't exit the polyhedron. Third, we need to be able to pick edges to move along in an "optimal" way, so to reduce the total number of steps of the algorithm before completion.

We construct a step of the simplex algorithm: suppose we are at a vertex $x$ and we wish to move along an edge to a neighboring vertex $x'$ such that $c^T x \geq c^T x'$. First, we need to be able to **move along an edge** –without considerations for optimality at the moment. The key idea here is that moving along an edge is equivalent to moving to another vertex, which is again equivalent to exchanging one (exiting) basic variable for one (entering) nonbasic variable. That is, we are moving from $x$ to $x + \theta d$ where $x = (x_B, 0)$ (reorder without loss of generality) and $d$ is constrained to have value $d_i = 0$ for $i \notin B \cup \{j\}$ and $d_i = 1$ for $i = j$, the index of the entering variable. Then, as we increase $\theta$ away from zero, $x + \theta d$ must remain feasible in terms of the equality constraint. Hence, it must be that

$$\begin{gathered} A(x+\theta d) = b \\ \Leftrightarrow Ax + \theta Ad = b \\ \Leftrightarrow Ad = 0 \\ \Leftrightarrow 0 = \sum_{i=1}^{n} A_i d_i = \sum_{i=1}^{m} A_{B(i)} d_{B(i)} + A_j = B d_B + A_j \\ \Leftrightarrow d_B = -B^{-1} A_j. \end{gathered}$$

We have a direction vector! It is called the $j^{\text{th}}$ basic direction.

Notice that we have only used one of the two feasibility criteria to figure out in *which directions* we can move. The remaining inequality constraint will tell us *how far* we can move along any given edge, i.e., in any basic direction. That is, we need to **stop moving along an edge once we've hit another vertex**. Geometrically, it is clear that this is the same as hitting some hyperplane corresponding to a $x_i \geq 0$ constraint for some $i \in B$ –remember, the equality condition can never be violated by increasing $\theta$. That is, we want the largest $\theta$ for which all $x_i$'s are still nonnegative. If $d_i \geq 0$, then $x_i' = x_i + \theta d_i \geq 0$ for all $\theta \geq 0$, and the non-negativity constraint will never bind in that dimension -in particular, if $d \geq 0$, then $\theta^* = \infty$. Further recall that $x_j$ is increasing in $\theta$ and the only other entries that are changing are the basic variables. Therefore,

$$\theta^* = \min_{\{i=1,\ldots,m | d_{B(i)} < 0\}} \left( -\frac{x_{B(i)}}{d_{B(i)}} \right),$$

and now we know when to stop!

We have said how to move along an edge, and how to stop moving once we've reached another vertex; now we know how to move from vertex to vertex along the edge between them. The last thing to elucidate is how to **pick the edge, or direction, that is the "most" cost reducing**, in some practical sense. A natural guide would be the value of the rate of cost change in the candidate direction (basically, the gradient). The change in cost when moving in the $j^{\text{th}}$ basic direction for a unit change in $\theta$ is

$$\begin{aligned} c^T d &= c_j + c_B^T d_B \\ &= c_j - c_B^T B^{-1} A_j. \end{aligned}$$

We call

$$\bar{c}_j := c_j - c_B^T B^{-1} A_j$$

the **reduced cost** of the variable $x_j$.

The reduced cost is a reasonable criteria for picking a direction. That is, we can pick our entering variable to be $x_{j^*}$ for $j^* = \arg\min_j \bar{c}_j$. Of course, a big rate of cost decrease does not guarantee a large decrease –the vertex could be short– and an alternative way of choosing could be $j^* = \arg\min_j \bar{c}_j \theta_j^*$, there $\theta_j^*$ is the length of the move in the $j^{\text{th}}$ basic direction. Now we know in which direction to go!

**Remark** *The choice of direction, often called pivoting rule, has been treated cursorily but is of great practical importance, and the best rule may depend on the specific application. We have also ignored degeneracy problems, assuming our linear program was well behaved. We have also ignored questions of computational complexity.*

✹

The main computation hurdle is the inversion of the basis matrix $B$ at each step of the simplex method. However, the matrix to invert at any given step is only slightly different from the one inverted in the previous step. More precisely, for $B$ the basis matrix of the previous step with the $l^{\text{th}}$ basis vector leaving and the $j^{\text{th}}$ nonbasic vector entering to form the new basis matrix $\bar{B}$, we have

$$B = \left[A_{B(1)}, ..., A_{B(m)}\right], \ \bar{B} = \left[A_{B(1)}, ..., A_{B(l-1)}, A_j, A_{B(l+1)}, ..., A_{B(m)}\right].$$

Hence it seems we should be able to use the previous inversion result $B^{-1}$ to reduce the computational burden of calculating $\bar{B}^{-1}$.

On can indeed obtain $\bar{B}^{-1}$ from $B^{-1}$ via elementary row operations. Observe that

$$\begin{aligned} B^{-1}\bar{B} &= \left[B^{-1}A_{B(1)}, ..., B^{-1}A_{B(l-1)}, B^{-1}A_j, B^{-1}A_{B(l+1)}, ..., B^{-1}A_{B(m)}\right] \\ &= \left[e_1, ..., e_{l-1}, u, e_{l+1}, ..., e_m\right], \end{aligned}$$

where $u = -d = B^{-1}A_j$ and $e_i$ is the $i^{\text{th}}$ Euclidian basis vector. For each entry of $u_i$ of $u$, $i \neq l$, we can set it to zero by adding the $l^{\text{th}}$ row times $-u_i/u_l$ to the $i^{\text{th}}$ row. That is equivalent to left multiplying $B^{-1}\bar{B}$ by

$$Q_{(i)} = I - u_i/u_l \cdot e_i e_l^T$$

and dividing the $l^{\text{th}}$ row by $u_l$, which can be done by left-multiplying by a diagonal matrix that has all ones on the diagonal, but for $1/u_l$ in the $l^{\text{th}}$ entry. Call that matrix $Q_{(l)}$. Then, by construction,

$$Q_{(1)}Q_{(2)} \cdots Q_{(m)} B^{-1}\bar{B} = I.$$

Consequently, we obtain

$$\bar{B}^{-1} = Q_{(1)} \cdots Q_{(m)} B^{-1}$$

using only the available $B^{-1}$ and elementary row operations.

For more details, see Bertsimas and Tsitsiklis (1997) subsection 3.3. Further note that, in industrial applications, it standard to maintain and iterate on an LU decomposition rather than the explicit inverse.

### 1.2.4 Duality

**Example: Poutine** *Duality in linear programming is nicely captured by stories of buyers and sellers doing their own maximization but having to trade at the resulting market price.*

*Suppose you are trying to consume the ideal vector of nutrients*

$$b = (b_1 \text{ fiber},\ b_2 \text{ iron},\ b_3 \text{ joy})$$

*by making a poutine, which has potatoes, gravy, and cheese curds as its ingredients. Then the quantity $x$ of each ingredient which you will purchase must satisfy an equality of the form $Ax = b$, i.e.,*

$$\begin{pmatrix} \text{fiber/potato} & \text{fiber/gravy} & \text{fiber/cheese curd} \\ \text{iron/potato} & \text{iron/gravy} & \text{iron/cheese curd} \\ \text{joy/potato} & \text{joy/gravy} & \text{joy/cheese curd} \end{pmatrix} \begin{pmatrix} \text{qtty potatoes} \\ \text{qtty gravy} \\ \text{qtty cheese curds} \end{pmatrix} = \begin{pmatrix} b_1 \text{ fiber} \\ b_2 \text{ iron} \\ b_3 \text{ joy} \end{pmatrix}.$$

*Our poutine aficionado is of course cost minimizing, so he wants to make his poutine while minimizing a $c^T x$ objective of the form*

$$(\text{cost potatoes},\ \text{cost gravy},\ \text{cost cheese curds}) \begin{pmatrix} \text{qtty potatoes} \\ \text{qtty gravy} \\ \text{qtty cheese curds} \end{pmatrix}.$$

*That is, making a poutine is tantamount to solving a linear program in standard form,*

$$\min\ c^T x$$

*subject to*

$$Ax = b$$

$$x \geq 0.$$

*A seller offers the desired nutrients directly, and wants to price them. That is, he needs to pick a price vector*

$$p = \begin{pmatrix} \text{price fiber} \\ \text{price iron} \\ \text{price joy} \end{pmatrix}$$

*to maximize $p^T b$, the returns he will make with the consumer –who has a fixed nutrients consumption. The buyer has the outside option of making his own poutine to get the nutrients, so the seller has a $A^T p \leq c$ constraint on his price vector,*

$$\begin{pmatrix} \text{fiber/potato} & \text{iron/potato} & \text{joy/potato} \\ \text{fiber/gravy} & \text{iron/gravy} & \text{joy/gravy} \\ \text{fiber/cheese curd} & \text{iron/cheese curd} & \text{joy/cheese curd} \end{pmatrix} \begin{pmatrix} \text{price fiber} \\ \text{price iron} \\ \text{price joy} \end{pmatrix} \leq \begin{pmatrix} \text{cost potatoes} \\ \text{cost gravy} \\ \text{cost cheese curds} \end{pmatrix}.$$

*That is, the seller solves a linear program*

$$\max\ p^T b$$

*subject to*

$$A^T p \leq c.$$

*We will find that the two linear programs are in fact dual to each other. The intuitive consequence is that by pricing directly the nutrients under a constraint of no arbitrage, the profit maximization exercise of the seller delivers the exact same total cost as the cost minimization exercise of the the buyer,*

$$p^{*T} b = c^T x^*.$$

*This will be a general conclusion of linear programing duality.*

Given a linear program, which we take to be our primal problem (P), we can derive its dual problem explicitly using its Lagrangian. Consider the standard form problem

$$\min \ c^T x$$

subject to

$$Ax = b \qquad\qquad \text{(P)}$$

$$x \geq 0.$$

We can relax this problem by replacing the hard constraint with a penalty and thus obtain its Lagrangian for which the minimization is

$$g(p) := \min_{x \geq 0} \ c^T x + p^T(b - Ax),$$

where $p$ is the vector of Lagrangian parameters. The first observation is that, for any $p$, the objective of the relaxed problem yields a lower bound on the primal, i.e.,

$$g(p) = \min_{x \geq 0} \ c^T x + p^T(b - Ax) \leq c^T x^* + p^T(b - Ax^*) = c^T x^*, \tag{14}$$

where $x^*$ is the solution of the primal.

Therefore, the tightest lower bound is is given by

$$\max_{p \in \mathbb{R}^m} g(p) \qquad\qquad \text{(D)},$$

which we refer to as the dual problem (D).

As observed in the example above, an important result here will be that the optimal cost in both the primal and the dual problems are the same.

We can do a little more work on the formulation of (D) to shrink the set of feasible $p$'s and concentrate out $x$. Note that

$$\begin{aligned} g(p) &= \min_{x \geq 0} \left[ c^T x + p^T(b - Ax) \right] \\ &= p^T b + \min_{x \geq 0} \left[ c^T x - p^T Ax \right], \end{aligned}$$

but

$$\min_{x \geq 0} \left( c^T - p^T A \right) x = \begin{cases} 0, & \text{if } c^T - p^T A \geq 0, \\ -\infty, & \text{otherwise.} \end{cases}$$

Hence we can restrict the support to $p$'s such that $c^T - p^T A \geq 0$, and for all those $p$'s it holds that $\left(c^T - p^T A\right) x$, so we can concentrate out $x$. The dual problem (D) may be rewritten as

$$\max \ p^T b$$

subject to

$$p^T A \leq c^T.$$

This gives the dual for the primal in standard form. But to each primal his dual, and if we repeated the same cxercises for different linear constraints on linear combinations and variables, we would collect the following duality relationships, reproduced from Table 4.1 of Bertsimas and Tsitsiklis (1997).

| PRIMAL | minimize | maximize | DUAL |
|---|---|---|---|
| constraints | $\geq b_i$ | $\geq 0$ | variables |
|  | $\leq b_i$ | $\leq 0$ |  |
|  | $= b_i$ | free |  |
| variables | $\geq 0$ | $\leq c_j$ | constraints |
|  | $\leq 0$ | $\geq c_j$ |  |
|  | free | $= c_j$ |  |

✹

The inequality (14) is a special case of weak duality.

**Theorem 5: Weak Duality** *If $x$ is a feasible solution to the primal problem and $p$ is a feasible solution to the dual problem, then*

$$p^T b \leq c^T x.$$

The equality of the dual and primal objectives, at optimality, is a deeper and stronger result.

**Theorem 6: Strong Duality** *If a linear programming problem has an optimal solution, so does its dual, and the respective costs are equal.*

Proof Sketch

By the equivalence of linear programming formulations, it suffices to consider a linear program in standard form,

$$\min \ c^T x$$

subject to

$$Ax = b,$$

$$x \geq 0,$$

with $A$ having full full row rank. If we are at an optimal primal solution with active basis $B$, then $x^* = x_B = B^{-1}b$, and all reduced costs are greater and or equal to zero,

$$c^T - c_B^T B^{-1} A \geq 0_n. \tag{15}$$

Now consider the dual problem

$$\max \ p^T b$$

subject to

$$p^T A \leq c^T,$$

and the candidate solution[5] $\tilde{p} = c_B^T B^{-1}$. Note that (5) implies it is feasible. Finally, note that

$$\tilde{p}^T b = c_B^T B^{-1} b = c_B^T x_B = c^T x^*.$$

□

Remark the the argument above makes a nice use of the simplex algorithm as a proof method.

The main characterization of the relation between the primal optimal solution $x^*$ and the dual optimal solution $p^*$ is called complementary slackness.

**Theorem 7: Complementary Slackness** *Let $x$ and $p$ be feasible solutions to the primal and the dual problems, respectively. The vectors $x$ and $p$ are optimal solutions for the two respective problems if and only if*

$$p_i \left( a_i^T x - b_i \right) = 0, \ \forall \ i, \tag{16}$$

$$(c_j - p^T A_j) x_j = 0, \ \forall \ j, \tag{17}$$

*where $a_i$ is the $i^{\text{th}}$ row of $A$ and $A_j$ is the $j^{\text{th}}$ column of $A$.*

Note that neither weak duality, strong duality, or complementary slackness requires that the linear program be in standard form.

The following example gives a physical and geometric flavor to duality.

[5] The candidate does not come out of nowhere. In the expression (5) for the reduced costs, the piece $c_B^T B^{-1}$ may be directly interpreted as a marginal cost, which suggests it has the allure of a dual variable.

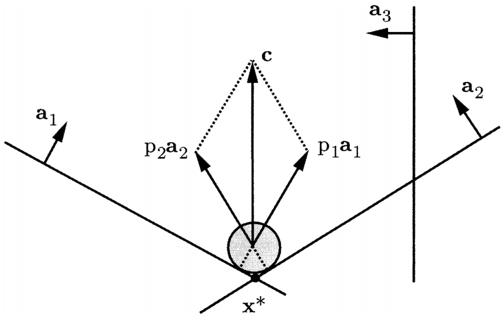

**Figure 3** *Mechanical analogy for linear programming duality and complementary slackness.*

**Example** *Consider a solid ball constrained to lie inside a polyhedron defined by inequality constraints of the form $a_i^T x_i \geq b_i$. The ball is pulled down by gravity and ends up in the lowest corner $x^*$ of the polyhedron, see Figure 3. This corner is an optimal solution to the problem*

$$\min\ c^T x$$

*subject to*

$$a_i^T x_i \geq b_i,\ \forall\ i,$$

*where $c$ is a vertical vector pointing upwards. Note that the problem is not in standard form.*

*The gravity vector must be counteracted by a vector of opposing forces that is exactly its negative, and the latter must be a linear combinations of forces normal to the walls against which the ball presses. That is,*

$$c = \sum_i p_i a_i = \sum_{i \in \mathcal{W}} p_i a_i, \tag{18}$$

*for some $p$, where $\mathcal{W}$ is the set of indexes of walls against which the ball presses, i.e., active constraints. Therefore, we must have $p_i = 0$ whenever $a_i^T x > b_i$, and it must hold for all $i$ that $p_i(b_i - a_i^T x^*) = 0$. By (18) it must hold for all $j$ that $c_j - p^T A_j = 0$, and thus $\left(c_j - p^T A_j\right) x_j = 0, \forall j$. Invoking complementary slackness, we can deduce that $p$ is the solution to the dual problem*

$$\max\ p^T b$$

*subject to*

$$A^T p = c,$$

$$p \geq 0.$$

✹

### 1.2.5 Column Generation and the Dual Revised Simplex Algorithm

Column and row generation as an application. Semi-infinite linear programming. Motivation for dual simplex.

✹

An intuitive treatment of the dual revised simplex algorithm is more subtle than in the primal case. It involves conceptual objects the tenor of which is less tangible; throughout the steps of the algorithm, we maintain a primal active basis even though it corresponds to a primal solution which is not primal feasible, and speak of maintaining the optimality –positive reduced costs, which is tantamount to dual feasibility– of this infeasible primal solution.

A coherent interpretation nevertheless emerges; since the feasibility condition of the dual problem corresponds to the nonnegativity of the reduced cost, we gather that whilst *the primal simplex algorithm maintains primal feasibility and converges to primal optimality, the dual simplex maintains primal optimality, and converges to primal feasibility.* Under this interpretation, it is sensible to entertain the notion of a primal basic solution –albeit unfeasible– throughout iterations of the dual simplex.

Recall the primal and dual standard form formulations,

$$\min \; c^T x$$

subject to

$$Ax = b$$

$$x \geq 0,$$

and

$$\max \xi^T b$$

subject to

$$A^T \xi \leq c.$$

The dual simplex algorithm proceeds, as in the primal case, by moving from a primal basis to another but, as opposed to the primal case, increasing the objective function value at each iteration. In order to speak of the "direction" and "length" of such moves, we first describe how the quantities of interest are updated at each iteration.

Throughout, we maintain and update a primal basis $B$, to which corresponds a primal variable $x_B = A_B^{-1}b$, a dual variable $\xi_B = A_B^{-T}c_B$, and a set of binding basic dual constraints

$$A_B^T\xi = A_B^T A_B^{-T} c_B = c_B.$$

Let the $r^{\text{th}}$ basis vector be the one leaving the primal basis –we treat the selection of $r$ below– and let $p = B(r)$ be its index in the full primal variable vector. Then the change in the $r^{\text{th}}$ dual basic constraint –the $p^{\text{th}}$ dual constraint– is

$$t := A_p^T\bar{\xi} - A_p^T\xi,$$

where $\xi$ is the current dual solution, and $\bar{\xi}$ the new dual solution.

Consequently $A_p^T\bar{\xi} - t = A_p^T\xi$ and the remaining primal basis elements remain, their corresponding dual constraints do not change and still bind, $A_j^T\bar{\xi} = A_j^T\xi = c_j$ for all $j \in B\backslash\{p\}$. Putting this together in matrix form, we have $A_B\bar{\xi} - e_r t = A_B^T\xi$, and the updating formula for the dual variable is

$$\bar{\xi} = \xi + A_B^{-T}e_r t.$$

From this, we immediately obtain the update formula for the reduced costs, which is the constraint for the dual feasibility,

$$\begin{aligned}
\bar{\mathbf{c}} &= c - A^T A_{\bar{B}}^{-T} c_{\bar{B}} = c - A^T\bar{\xi} \\
&= c - A^T(\xi + A_B^{-T}e_r t) \\
&= c - A^T A_B^{-T} c_B - A^T A_B^{-T} e_r t \\
&= \mathbf{c} - A^T A_B^{-T} e_r t.
\end{aligned}$$

For $j \in B\backslash\{p\}$,

$$\bar{\mathbf{c}}_j = \mathbf{c}_j - A_j^T A_B^{-T} e_r t = \mathbf{c}_j - e_{B^{-1}(j)}^T e_r t = \mathbf{c}_j = 0, \tag{19}$$

the reduced cost does not change and remains zero.[6] Note the abuse of notation, $j$ is the

[6] Notice that we had to switch notation around a bit. Now $\mathbf{c}$ stands for the reduced cost, $\bar{\mathbf{c}}$ stands for

$B^{-1}(j)^{\text{th}}$ entry of the basis, $B(B^{-1}(j)) = j$. Further,

$$\bar{\mathbf{c}}_p = \mathbf{c}_p - A_p^T A_B^{-T} e_r t = 0 - e_r^T e_r t = -t, \tag{20}$$

and

$$\bar{\mathbf{c}}_j = \mathbf{c}_j - A_j^T A_B^{-T} e_r t, \tag{21}$$

for all $j \notin B$.

We can also get a very nice update formula for the dual objective function,

$$\begin{aligned}\bar{Z} &= b^T \bar{\xi} \\ &= b^T \left(\xi + A_B^{-T} e_r t\right) \\ &= b^T \xi + b^T A_B^{-T} e_r t \\ &= Z + x_B^T e_r t \\ &= Z + x_p t.\end{aligned}$$

It follows that the $p$ which we choose to leave the basis, in order to increase the value of the dual objective function, must have a negative primal basic solution entry $x_p < 0$. This is exactly the sense in which we make the primal infeasible solution “more feasible”.

We know how to select a $p$. Given that choice, we need to find the value of $t$. The key to finding the right value of $t$ is that dual feasibility must be maintained. We know from (20) that this requires $t$ to be positive, $t > 0$. From (21), we find that

$$\bar{\mathbf{c}}_j \geq 0$$

$$\Leftrightarrow \mathbf{c}_j - A_j^T A_B^{-T} e_r t \geq 0$$

$$\Leftrightarrow t \leq \frac{\mathbf{c}_j}{A_j^T A_B^{-T} e_r} > 0 \text{ if } A_j^T A_B^{-T} e_r > 0, \tag{22}$$

$$\text{and } t \geq \frac{\mathbf{c}_j}{A_j^T A_B^{-T} e_r} < 0 \text{ if } A_j^T A_B^{-T} e_r < 0. \tag{23}$$

The idea is that we decrease $t$ until the first nonbasic dual constraint binds and we collect the index $q$ of the binding constraint. This is the index of the nonbasic variable entering the primal basis. We say that the solution value $\theta_D = t$ for which the first nonbasic dual constraint binds is the dual step length. Only the inequality (23) affects the dual step

the reduced cost in the next period, and there is no quantity associated with $\bar{c}$. Of course, $c$ remains the cost vector of the objective function.

size, thus

$$q \in \arg\max_{j \in \mathcal{F}} \left\{ \frac{\mathbf{c}_j}{A_j^T A_B^{-T} e_r} \right\} \text{ and } \theta_D = \frac{\mathbf{c}_q}{A_q^T A_B^{-T} e_r},$$

where $\mathcal{F} = \{j \notin B \mid A_j^T A_B^{-T} e_r < 0\}$.

A step of the dual simplex algorithm is achieved by having $p$ leave and $q$ enter the primal active basis, with the dual variable $\xi$, dual objective $Z$, and reduced costs $\mathbf{c}$ updated according to the derived formulae.

I may then update the inverse $A_{\bar{B}}^{-1}$ using basic row operations as described above, and likewise obtain $x_{\bar{B}} = A_{\bar{B}}^{-1} b$.

## 1.3 Quantile Regression Through the Lens of Linear Programming

Some of the properties of the quantile regression estimator arise naturally and intuitively as corollaries of linear programming theory. A good example is the perplexing observation that, for a quantile regression of $Y$ on $X$ with $X \in \mathbb{R}^{n \times p}$, the fitted regression hyperplane will systematically lie on $p$ observations, or support vectors. See Figure 4 for an illustration.

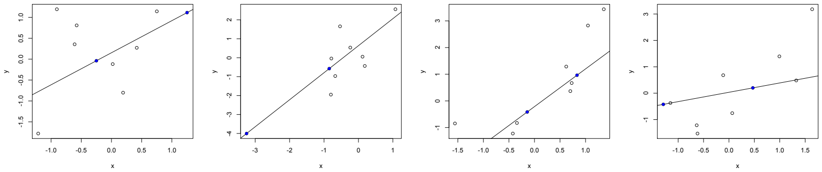

**Figure 4** *In the regression of Y on a constant vector and a vector of regressors, the fitted line always goes through two points.*

It turns out that this fact can be intuitively characterized using the theory of linear programming duality.

Analogy with OLS is fruitfill. Having $p$ points laying on the fitted hyperplane is, in a heuristic sense, the quantile regression analog of the OLS phenomenon that the fitted residuals are always orthogonal to the column span of the covariates: $X^T \hat{\varepsilon}$. This seemingly puzzling fact is a direct and geometric instance of the Pythagoeran theorem, and the linear algebra underpinning OLS delivers the full algebraic and geometric intuition of this orthogonality result.[7] One can argue that linear programming is to quantile regression what linear algebra is to least-squares regression. Understanding the linear programming structure of quantile regression unlocks many insights about the quantile regression method and estimator.

[7] https://upload.wikimedia.org/wikipedia/commons/8/87/OLS_geometric_interpretation.svg

### 1.3.1 The Key Identity

If the fitted regression hyperplane for the quantile regression of $Y$ on $X$ with $X \in \mathbb{R}^{n \times p}$ must pass through $p$ observations, these $p$ points determine the hyperplane and it must be that we can compute the fitted regression coefficient $\hat{\beta}_\tau$ using these $p$ observations and ignoring the $n-p$ remaining observations.

**Exercise** *Give a strictly geometric argument to deduce the formula for the hyperplane $L = \{x \in \mathbb{R}^p \ : \ y = x\beta\}$ in terms of $x_1, ..., x_p$, where $x_1, ..., x_p \in L$ and no subset of $l$ points lie on a $(l-2)$-dimensional hyperplane.*

We give a somewhat heuristic argument deriving the formula for $\hat{\beta}_\tau$ in terms of $p$ observations, or support vectors, which lie exactly on the fitted quantile regression hyperplane. We vaguely but intuitively recognize, in the selection of $p$ support vectors determining the optimal solution of our linear program, the notion of an optimal active basis. However, as can be deduced by glancing at (11)-(13), an active basis in the primal quantile regression linear program consists of *columns* the design matrix $X$. We therefore have to look at the *dual* quantile regression problem to get solutions discretized in terms of active bases made of *rows* of $X$, which is to say of regression observations.[8] What will then allow us to express the primal optimal solution in terms of the dual active basis is of course the set of identities detailing the relationship between primal and dual variables and conditions, the complementary slackness formulas (16) and (17).

The argument goes as follows. The dual formulation of the standard form quantile regression linear program is

$$\max \xi^T Y$$

subject to

$$\begin{pmatrix} X^T \\ -X^T \\ I \\ -I \end{pmatrix} \xi \leq \begin{pmatrix} 0_p \\ 0_p \\ \tau \cdot \mathbf{1}_n \\ (1-\tau) \cdot \mathbf{1}_n \end{pmatrix}.$$

The set of inequalities precisely define a polyhedron in terms of half spaces, as illustrated

[8] The standard form problem (P) has a dual (D) with a “tall” matrix, but the dual of the quantile regression problem simplifies to

$$\max \xi^T Y$$

subject to

$$X^T \xi = 0_p$$
$$(\tau - 1) \cdot \mathbf{1}_n \leq \xi \leq \tau \cdot \mathbf{1}_n,$$

which is similar to a primal formulation with a “wide” linear connstraint matrix, but with a slightly different constraints on the variables, suggesting we recuperate the notion of an active basis in this specific dual. Furthermore, consult exercise 4.

in Figure 2. Assuming the problem is feasible and bounded, we know that a solution must be attained at a vertex of the polyhedron, where $n$ conditions bind. Note that the first $2p$ inequality constraint amount to the equality condition $X^T\xi = 0$, thus accounting for $p$ degrees of freedom of $\xi$. That is to say, $n - p$ of the linear inequality constraints

$$\begin{pmatrix} I \\ -I \end{pmatrix} \xi \leq \begin{pmatrix} \tau \cdot \mathbf{1}_n \\ (1-\tau) \cdot \mathbf{1}_n \end{pmatrix}$$

must bind. Assuming our data was drawn from a law uniformly continuous with respect to Lebesgue measure, the problem is non-degenerate with probability one, and we may conclude that $n + p$ of the linear inequality must not bind. Hence, we have $\xi_i \leq \tau$ and $\xi_i \geq 1 = \tau$ for $i = 1, ..., n$, but it must be that $n + p$ of those are strict inequality. More importantly, it must be that

$$\xi_{i_j} < \tau \text{ and } \xi_{i_j} > 1 = \tau, \text{ for some } \{i_1, ...i_p\} \subseteq \{1, ..., n\}.$$

Then complementary slackness implies that

$$u_{i_j} = v_{i_j} = 0, \text{ for } j = 1, ..., p.$$

The primal equality constraint (12) thus implies that $x_{i_j}\beta = y_{i_j}$ for $j = 1, ..., p$. In matrix notation, allowing $B = \{i_1, ..., i_p\}$ to index the rows of $X$ and entries $Y$, and letting $X_B$ and $Y_B$ be the corresponding submatrix and subvector, we have that

$$X_B\hat{\beta}_\tau = Y_B,$$

where $X_B$ is now a $p \times p$ matrix, and thus

$$\hat{\beta}_\tau = X_B^{-1}Y_B.$$

## 1.4 A Taylor-Made Simplex for Quantile Regression

We have developed the primal simplex algorithm, a workhorse of linear optimization, which applies to linear programs in standard form. Since any linear program may be formulated in standard form (Bertsimas and Tsisiklis, 1997), the primal simplex may be used to solve any linear program.

However, a candid application of the primal simplex to the standard form representation of the quantile regression linear program delivers an unnecessarily sluggish procedure. The main burden on the procedure arises from the formulation of the regression and error variables, $\beta$ and $\varepsilon = Y - X\beta$, respectively, in terms of positive and negative parts. That is, we write

$$\beta_j = \beta_{+,j} - \beta_{-,j},\ \varepsilon_i = u_i - v_i,$$

where $\beta_{+,j}, \beta_{-,j}, u_i - v_i \geq 0$, $j = 1, ..., p$, $i = 1, ..., n$. The representation of the error terms in positive and negative parts is necessary in order to express the objective function as a linear function, while the such representation of the regression coefficients is only required in order to express the quantile regression linear program in standard form.

This raises the question of whether we can tailor a simplex to the specificities of the quantile regression problem and deliver a better computational performance. It is indeed possible; such an algorithm was delivered by Barrodale and Roberts (1973). Their algorithm is a primal simplex with a specific choice of pivot rule.[9]

### 1.4.0 The Quantile Regression Linear Program

In order to develop a tailor-made algorithm for quantile regression, we need to get a sense of the specificities of the problem –in the hope that we can exploit them in the design of the algorithm.

The quantile regression linear program in standard form is written

$$\min \tau \mathbf{1}_n^T u + (1-\tau)\mathbf{1}_n^T v$$

subject to

$$[X, -X, I, -I] \begin{pmatrix} \beta_+ \\ \beta_- \\ u \\ v \end{pmatrix} = Y$$

$$\beta_+, \beta_-, u, v \geq 0.$$

Three insights about the the quantile regression linear program motivate the algorithm of Barrodale and Roberts (1973).

### Insight 1: Starting Solution

First, we always have available a starting basic feasible solution. Simply set the regression coefficients to zero and fit the errors to the outcome variable,

[9] The choice of a pivot rule is where the art in the design of simplex algorithms lies. It came out of our discussion of the primal simplex simplex that any variable with positive reduced cost is a “reasonable” candidate for entering the basis. A rule for picking specifically which variable enters the basis is called a pivot rule.

$$\beta_{+,j} = \beta_{-,j} = 0,$$

$$u_i = Y_i,\ v_i = 0,\ \text{if } Y_i \geq 0,$$

$$u_i = 0,\ v_i = -Y_i,\ \text{if } Y_i < 0,$$

for $j = 1, ..., p$, and $i = 1, ..., n$.

### Insight 2: Stability of $\beta$-Full Solutions

There is a specific and rigorous sense in which the notion of $\beta_{+,j}$'s and $\beta_{-,j}$'s as basic elements is superficial, and it stems directly from their artificial introduction for the purpose of reformulating the quantile regression linear program in standard form. We first consider this intuitively; under a maximum reduced cost pivot rule, every time a $\beta_{+,j}$ exists the basis (i.e., $\beta_{+,j}$ is set to 0), it's to let $\beta_{-,j}$ enter the basis (i.e., $\beta_{-,j}$ becomes greater than zero). This is a consequence of the artificial nature of the non-negativeness restriction on the regression coefficient; for any given $j$, $\beta_{+,j}$ and $\beta_{-,j}$ keep track of the sign of $\beta_j$, but $\beta_j$ either represented by $\beta_{+,j}$ or $\beta_{-,j}$ never leaves the basis. Hence we *effectively* never pivot on any $\beta_j$.

We now consider this in rigorous terms. We define feasible solutions in which every regression coefficient is an active variable.

**Definition** *A basic feasible solution to the quantile linear program in standard form is a $\beta$-**full solution** if $p = \text{rank}(X)$ of the $\beta_{+,j}$'s and $\beta_{-,j}$'s are nonzero.*

Note that we assume that the $\beta$-full solution is obtained along a simplex moving in directions of positive reduced costs, so it is implied that only one of $\beta_{+,j}$ and $\beta_{-,j}$ may be nonzero at a time for any given $j$.

**Claim** *Once a simplex moving in directions of positive reduced cost attains a $\beta$-full solution, any solution of the subsequent iterations of the simplex is a $\beta$-full solution.*

This claim has an important practical implication. Since the $\beta_j$'s are effectively never involved in the pivoting of the simplex, we can "ignore" them. To be precise, we can run a simplex only moving $u_i$'s and $v_i$'s in and out of the basis and never pivot on the $\beta_i$'s, treating them as real variables (which can be positive or negative).

### Insight 3: Change-of-Sign Pivots are Cheap

The computationally demanding part of the simplex algorithm is the maintenance of the inverse basic matrix, $B^{-1}$. In subsection 1.2.3, we noticed that the matrices to invert in subsequent pivots differed only by a column, and building on this observation we detailed how this maintenance may be done at the cost of $m$ elementary row operations at each pivot. Although this “revision” of the simplex algorithm attenuates its computational burden, maintenance of $B^{-1}$ remains the main computational cost of the procedure.

However, when an pivot constitutes in a sign change, i.e. when a $u_i$ leaves and $v_i$ enters or vice versa, or when a $\beta_{+,j}$ leaves and $\beta_{-,j}$ enters or vice versa, the matrices to be inverted in subsequent pivots are even more similar. It is easy to verify that if $B$ and $\bar{B}$ are basic matrices before and after a sign change, then

$$\bar{B} = BD_{(s)},$$

where $D_{(s)} = \mathrm{diag}(\mathbf{1}_n - 2e_s)$ is a diagonal matrix with the $s^{\text{th}}$ diagonal entry replaced by a -1 and the $s$ indexes the variable which changed sign. Then,

$$\bar{B}^{-1} = \left(BD_{(i)}\right)^{-1} = D_{(i)}B^{-1},$$

thus $\bar{B}^{-1}$ may be obtained from $B^{-1}$ by simply negating a row.

We will therefore want to distinguish between two types of pivots according to the following definition.

**Definition** *We speak of a* ***change-of-sign pivot*** *when a $u_i$ leaves the basis and $v_i$ enters or vice versa, or when a $\beta_{+,j}$ leaves the basis and $\beta_{-,j}$ enters or vice versa, and we speak of a* ***change-of-variable*** *pivot otherwise.*

In practice, the inverse may be maintained in, say, an LU decomposition form, but the same general intuition applies.

### 1.4.1 The Modified Simplex Algorithm

The modified simplex algorithm of Barrodale and Roberts (1973) exploits the aforementioned three features of the quantile regression linear program. Obviously, it uses the available basic feasible solution described in the first insight as a starting solution. In accordance with the second insight, their procedure tries to get *quickly* to a *good* $\beta$-full solution and from then on “ignores” the $\beta_{+,j}$ and $\beta_{-,j}$’s when carrying out simplex pivots.

The most insightful ingredient of the algorithm comes from its clever exploitation of the third insight. As mentioned earlier, one must chose a pivot rule when designing a simplex algorithm, and that choice is in a sense more a matter of art than science. Indeed, a pivot rule typically only relies on local information; e.g., the reduced costs for each direction, or maybe the total change in the objective function for a single step in each direction. One cannot, with this information, select directions that will provably make for a shorter path

–one involving fewer pivots– to the optimum. One local criteria which is less frequently encountered in the design of simplex algorithms, but which is entirely legitimate, is to favor computationally cheaper pivots. The third insight indicates precisely which ones those are.

We can now give the modified simplex algorithm. It has two phases. The first delivers a $\beta$-full solution, the second runs a primal simplex which never attempts to pivot on regression coefficients. The key characteristic is that *in both phases, (cheap) change-of-sign pivots are carried out as long as the last variable to enter the basis in an (expensive) change-of-variable pivot still has positive reduced cost.*

### Phase 1: Constructing a Good $\beta$-Full Solution

In phase 1, only regression coefficients may enter the basis on change-of-variable pivots. Phase 1 begins with the starting solution described in the first insight. Then, an iteration goes as follows,

- Find the nonbasic $\beta_{+,j}$ or $\beta_{-,j}$ with the largest reduced cost, say it is $\beta_{s^*,j^*}$
- Augment $\beta_{s^*,j^*}$ until its reduced cost is less than or equal to zero
  - *This translates or tilts the fitted regression line, and every time it crosses an observation, a residual changes sign, which corresponds to a change-of-sign pivot on a residual pair $(u_i, v_i)$ for some $i$*
- Start over, moving on to the next $\beta_{+,j}$ or $\beta_{-,j}$ with the greatest reduced cost

After $p$ complete iterations, all of which involve a single change-of-variable pivot introducing a $\beta_{+,j}$ or $\beta_{-,j}$ into the basis and a finite number of change-of-sign pivots, we reach a $\beta$-full solution, and the first phase is completed.

### Phase 2: Primal Simplex Ignoring $\beta$

In phase 2, only regression residual variables may enter the basis on change-of-variable pivots. We again leverage the third insight by taking paths with "as many change-of-sign pivots as possible". An iteration goes as follows,

- Find nonbasic $u_i$ or $v_i$ with the largest reduced cost, say it is $u_{i^*}$
- Augment $u_{i^*}$ until its reduced cost is less than or equal to zero
  - *This –also– tilts the fitted regression line as we interpolate between and move from one exiting support vectors*[10] *to a new one and, every time the fitted*

[10] Recall that support vectors are observations whose residual variables $u_i$ and $v_i$ are both nonbasic.

*line crosses an observation, a residual changes sign, which corresponds to a change-of-sign pivot on a residual pair $(u_i, v_i)$ for some $i$*

- Start over, moving on to the next $u_i$ or $v_i$ with the greatest reduced cost

The second phase stops when reduced costs corresponding to all $u_i$'s and $v_i$'s are nonnegative. The full algorithm then stops.

### 1.4.2 Analysis of the Algorithm

It has been observed in practice that phase 1 delivers very good solutions. In fact, the algorithm sometimes terminates at the end of phase 1 because the first $\beta$-full solution is optimal. Barrodale and Roberts (1973) display such a case in a toy example. Obviously, one could get to a $\beta$-full solution faster by doing $p$ change-of-variables pivots from the start. However, at little additional cost –by only adding change-of-sign pivots– one gets a good, even sometimes optimal, candidate solution.

Nevertheless, to the extent that we think of phase 2 as the main algorithm for fitting the quantile regression coefficients by moving between sets of $p$ support vectors, we should think of phase 1 as a preprocessing step delivering quickly a good starting $\beta$-full solution. Then, alternatives for preprocessing, especially when dealing with large data sets, could be entertained; taking an OLS solution and declaring the $p$ observations with smallest OLS residuals as quantile regression support vectors could be reasonable, likewise obtaining a candidate solution from running quantile regression on a subset of the data would deliver a starting $\beta$-full solution.

For geometric intuition of phase 2, Koenker suggests looking at Edgeworth's dual plot ; it delivers the polytope we are implicitly running the simplex on.

The algorithm applies to median regression, but was extended to the case of a general quantile by Koenker and d'Orey (1987).

### 1.4.3 Comments on Original Paper

Some statements in the Barrodale and Roberts (1973) paper have entered the quantile regression vernacular even though their rigorous interpretation may not be immediate. The development and analysis of this section help articulate them in perhaps more accessible terms. The authors state that with their simplex, they "*discovered how to pass through several neighboring vertices in a single iteration*". This should of course not be understood to mean that they somehow travel multiple vertices in a single pivot, which is nonsensical. This precisely means, in the language characterized above, that for each unique change-of-variable pivot, they make multiple change-of-sign pivots –and they count one iteration per change-of-variable pivot.

They describe their "main" modification as choosing the vector $u_i$ or $v_i$ to leave the basis by picking that which "*causes the maximum reduction in the objective function.*" This suggests that the pivot is defined as introducing in the basis the vector with the greatest reduced cost, and making room for it by taking out of the basis the vector which will deliver the solution –e.g., in the case of a $\beta$-full solution, that implied by the support vectors corresponding to the candidate basis– with the lowest objective function value. This does not define a pivot, in particular it will not in general deliver a feasible solution. What they mean by an iteration is precisely one pass of the procedure described in phase 1 and 2 above, and what they mean by the "vector leaving the basis" is the last vector to leave the basis in the sequence of change-of-sign pivots following a change-of-variable pivots and terminating when the reduced cost of the variable which entered at the change-of-variable pivot becomes negative.

## 1.5 Interior Point Methods

Interior point methods. Ellipse.

## Exercises

**Exercise 1** Show that the problem of finding the minimum loss function value for quantile regression with interval-censored dependent variables is a linear program and give its explicit formulation.

**Exercise 2** Give the asymptotic variance formula for a fixed dimension kernel SVM.

**Exercise 3** Give a general class of problems for which the key identity of subsection 1.3.1 holds.

**Exercise 4** Give an algorithm for the revised simplex method for linear programs with equality constraints and bounded (upper- and lower-bounded) variables. Code this algorithm and apply it to the dual of the quantile regression linear program. Note how this is a primal simplex applied to a dual problem.

**Exercise 5** Make an example where you display quantile crossing. Figure out and program a non-crossing quantile regression method and apply it to your example.

**Exercise 6** Consider the case of a panel with very many individual fixed effects. Can you add constraints to make estimation faster?

**Exercise 7** Produce a representation such as Figure 2 for a problem in standard form.

## Research Questions

**Question 1** Consider the equivalent of a rank test for SVM.

**Question 2** Give a reoptimization preocedure for row augmentation in the primal problem in standard form. Observe that the rank of $A$ increases!

## References

Angrist, Joshua, Victor Chernozhukov, and Iván Fernández-Val. "Quantile regression under misspecification, with an application to the US wage structure." *Econometrica* 74, no. 2 (2006): 539-563.

Bertsimas, Dimitris, and John N. Tsitsiklis. *Introduction to linear optimization.* Vol. 6. Belmont, MA: Athena Scientific, 1997.

Barrodale, Ian, and F. D. K. Roberts. "Solution of an overdetermined system of equations in the l 1 norm [F4]." *Communications of the ACM* 17, no. 6 (1974): 319-320.

Koberstein, Achim. "The dual simplex method, techniques for a fast and stable implementation." *Unpublished doctoral thesis*, Universität Paderborn, Paderborn, Germany (2005).

Koenker, Roger. *Quantile regression.* No. 38. Cambridge university press, 2005.

Koenker, Roger, and Gilbert Bassett Jr. "Regression quantiles." *Econometrica: journal of the Econometric Society* (1978): 33-50. Harvard

Koenker, Roger W., and Vasco d'Orey. "Algorithm AS 229: Computing regression quantiles." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 36, no. 3 (1987): 383-393.