

07-380 Notes

Pre-reading: Linear Programming

Carnegie Mellon University
School of Computer Science

Contents

1	Linear Programming	1
1.1	The Diet Problem	1
1.2	The Form a Solver Wants	1
1.3	Fitting the Diet Problem into the Form	2
1.4	The Graphical Representation	4
2	Vectors, Matrices, and Dimensions	4
2.1	Notation	4
2.2	The Dimension Check for Matrix Multiplication	5
3	The Dot Product	5
3.1	Definition	5
3.2	Sign of the Dot Product	6
4	Lines and Half-Planes from Dot Products	7
4.1	Where $\mathbf{c}^\top \mathbf{x}$ Equals 0, 1, and 2	7
4.2	A Linear Equality: $\mathbf{a}^\top \mathbf{x} = b$	8
4.3	A Linear Inequality: $\mathbf{a}^\top \mathbf{x} \leq b$	8
5	Summary	10

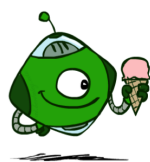
1 Linear Programming

1.1 The Diet Problem

We are trying to eat healthy by finding the optimal amount of food to purchase. The restaurant sells stir-fry by the ounce and boba by the fluid ounce, and it publishes the cost and nutrition facts per unit of each:

Food	Cost	Calories	Sugar	Calcium
Stir-fry (per oz)	1	100	3	20
Boba (per fl oz)	0.5	50	4	70

Our doctor has also given us a few nutrition goals to meet:



Healthy Squad Goals

$$\begin{aligned}
 2000 &\leq \text{Calories} \leq 2500 \\
 \text{Sugar} &\leq 100 \text{ g} \\
 \text{Calcium} &\geq 700 \text{ mg}
 \end{aligned}$$

Illustration: Ketrina Yim / UC Berkeley AI

What is the cheapest way to stay “healthy” with this menu? That is, how much stir-fry and how much boba should we buy?

We will use two **variables** for the two amounts we get to choose:

$$x_1 = \text{ounces of stir-fry}, \quad x_2 = \text{fluid ounces of boba}.$$

Everything else in the tables is a fixed number.

This is the **diet problem**, and it will be our running example for linear programming. In this pre-reading we will not try to figure out what the answer is. Instead, we will focus on how to write the problem down and on the linear algebra needed to read what we wrote.

1.2 The Form a Solver Wants

The diet problem has constraints that must be satisfied, and it also has a cost. We want the values of the variables that satisfy the constraints *and* have the lowest cost. This is an **optimization** problem. If we can write an optimization problem in the following form,

$$\begin{array}{ll} \min_{\mathbf{x}} & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} & A\mathbf{x} \preceq \mathbf{b} \end{array}$$

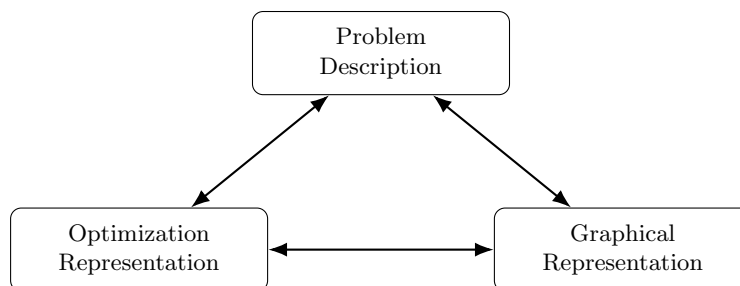
then we can hand it to a **linear programming solver**, a standard piece of software that will find the optimal $\mathbf{x}$ for us. A problem that can be written this way is called a **linear program**.

Reading the form line by line:

- “s.t.” is short for **such that**. The first line says what we are minimizing, and the second line says the conditions that must hold while we do it.
- $\mathbf{x}$ is the vector of **variables**, the amounts we get to choose.
- $\mathbf{c}^\top \mathbf{x}$ is the **objective function**, the cost we want to minimize. The vector $\mathbf{c}$ holds the cost of one unit of each variable.
- $A\mathbf{x} \preceq \mathbf{b}$ is the set of **constraints**. The matrix A and the vector $\mathbf{b}$ hold the numbers that define the limits.

We will define the transpose $\mathbf{c}^\top$, the product $A\mathbf{x}$, and the sizes of all of these objects in Sections 2 and 3. For now, the point is that there is one specific form that we are aiming for.

In this course, we want to be able to move between three representations of the same problem: the problem description in words, the optimization representation above, and a graphical representation that we can draw when there are only two variables.



This pre-reading covers the first two representations and the math behind the third. In lecture, we will put all three together.

1.3 Fitting the Diet Problem into the Form

Let's take the diet problem from its description in words to the optimization representation, one piece at a time.

Objective: The cost of our order is one dollar per ounce of stir-fry plus half a dollar per fluid ounce of boba:

$$\text{cost} = 1 x_1 + 0.5 x_2.$$

Constraints: Each row of the nutrition table tells us how much of that nutrient is in one unit of each food, so the total amount of a nutrient in our order is a similar expression in x_1 and x_2 . Each Healthy Squad Goal then becomes an inequality:

$$\begin{aligned} \text{Calories:} \quad & 100 x_1 + 50 x_2 \geq 2000 \\ & 100 x_1 + 50 x_2 \leq 2500 \\ \text{Sugar:} \quad & 3 x_1 + 4 x_2 \leq 100 \\ \text{Calcium:} \quad & 20 x_1 + 70 x_2 \geq 700 \end{aligned}$$

Notice that the calorie goal has both a lower and an upper limit, so it contributes two inequalities.

Pointing all the inequalities the same way: The form we are aiming for has only $\leq$ in it, but two of our constraints are $\geq$. To flip a $\geq$ into a $\leq$, we multiply both sides by -1 , which negates every number in the inequality:

$$\begin{aligned} -100 x_1 - 50 x_2 &\leq -2000 \\ -20 x_1 - 70 x_2 &\leq -700 \end{aligned}$$

Putting it all together, the diet problem in **scalar form** is:

$$\begin{aligned} \min_{x_1, x_2} \quad & 1 x_1 + 0.5 x_2 \\ \text{s.t.} \quad & -100 x_1 - 50 x_2 \leq -2000 \\ & 100 x_1 + 50 x_2 \leq 2500 \\ & 3 x_1 + 4 x_2 \leq 100 \\ & -20 x_1 - 70 x_2 \leq -700 \end{aligned}$$

Replacing the specific numbers with symbols: A solver does not know anything about stir-fry. It only sees the numbers, so let's name them. Write the cost per unit as c_1 and c_2 , and write the number in constraint m that multiplies variable n as $a_{m,n}$, with the limit for constraint m written as b_m :

$$\begin{array}{ll} \min_{x_1, x_2} & c_1 x_1 + c_2 x_2 \\ \text{s.t.} & a_{1,1} x_1 + a_{1,2} x_2 \leq b_1 \\ & a_{2,1} x_1 + a_{2,2} x_2 \leq b_2 \\ & a_{3,1} x_1 + a_{3,2} x_2 \leq b_3 \\ & a_{4,1} x_1 + a_{4,2} x_2 \leq b_4 \end{array} \quad \begin{array}{c|ccc} m & a_{m,1} & a_{m,2} & b_m \\ \hline 1 & -100 & -50 & -2000 \\ 2 & 100 & 50 & 2500 \\ 3 & 3 & 4 & 100 \\ 4 & -20 & -70 & -700 \end{array}$$

The negative signs from flipping the inequalities now live inside the numbers $a_{m,n}$ and b_m . The formulation itself has no minus signs in it.

Collecting the symbols into vectors and a matrix: Finally, we stack the variables into a vector $\mathbf{x}$, the costs into a vector $\mathbf{c}$, the constraint coefficients into a matrix A (one row per constraint, one column per variable), and the limits into a vector $\mathbf{b}$:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \quad A = \begin{bmatrix} -100 & -50 \\ 100 & 50 \\ 3 & 4 \\ -20 & -70 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix}.$$

With these, the diet problem is exactly

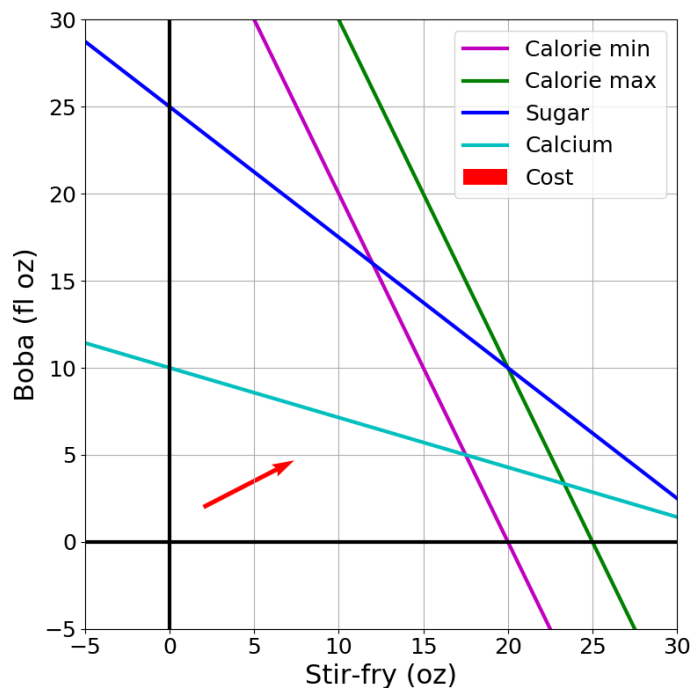
$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \preceq \mathbf{b} \end{aligned}$$

and we can pass $\mathbf{c}$, A , and $\mathbf{b}$ to a linear programming solver.

Every expression we wrote along the way, the cost and each constraint, is a **linear combination** of x_1 and x_2 : some number times x_1 , plus some number times x_2 . That is where the “linear” in linear programming comes from.

1.4 The Graphical Representation

When there are only two variables, we can draw the problem. The plot below puts x_1 (stir-fry) on the horizontal axis and x_2 (boba) on the vertical axis. Each of the four constraints from Section 1.3 shows up as a line, and the cost shows up as a direction, drawn as the red arrow.



Why does a constraint like $3x_1 + 4x_2 \leq 100$ turn into a line? Which side of the line is allowed? Why is the cost an arrow, and what does the arrow’s direction mean? The rest of this pre-reading builds up the linear algebra that answers these questions, starting with sizes of vectors and matrices, then the dot product, and then the lines and half-planes that dot products define in the plane.

2 Vectors, Matrices, and Dimensions

2.1 Notation

We follow the course notation guide:

- **Scalars** are plain lowercase letters, such as x_1 , c_2 , or b_m . Uppercase letters such as N and M are also scalars when they count something.

- **Vectors** are bold lowercase letters, such as $\mathbf{x}$ or $\mathbf{c}$. A vector $\mathbf{x} \in \mathbb{R}^N$ has N real-valued entries, $\mathbf{x} = [x_1, x_2, \dots, x_N]^\top$. By default, a vector is a **column vector**, which is why we write the transpose $^\top$ on the row of entries.
- **Matrices** are uppercase letters, such as A . A matrix $A \in \mathbb{R}^{M \times N}$ has M rows and N columns, and its entry in row m and column n is $a_{m,n}$.

The **transpose** of a column vector is the same entries laid out as a row. If

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{then} \quad \mathbf{x}^\top = [x_1 \quad x_2].$$

In the diet problem, $N = 2$ is the number of variables and $M = 4$ is the number of constraints, so

$$\mathbf{x} \in \mathbb{R}^2, \quad \mathbf{c} \in \mathbb{R}^2, \quad A \in \mathbb{R}^{4 \times 2}, \quad \mathbf{b} \in \mathbb{R}^4.$$

A vector in $\mathbb{R}^N$ is also a matrix with N rows and one column, so we say that $\mathbf{x}$ is $N \times 1$. This is the view we need for the next subsection.

2.2 The Dimension Check for Matrix Multiplication

Matrix multiplication is only defined when the sizes line up. The product of an $M \times K$ matrix and a $K \times N$ matrix is an $M \times N$ matrix:

$$[M \times N] = [M \times K][K \times N].$$

The **inner** dimensions, the two copies of K , must match. The **outer** dimensions, M and N , give the size of the result.

Some examples:

- In the diet problem, A is 4×2 and $\mathbf{x}$ is 2×1 . The inner dimensions match ($2 = 2$), so $A\mathbf{x}$ is defined and its size is 4×1 .
- Two column vectors in $\mathbb{R}^3$ cannot be multiplied directly: $[3 \times 1][3 \times 1]$ has inner dimensions 1 and 3, which do not match.
- A 1×3 row times a 3×1 column is defined, and the result is 1×1 . A 1×1 matrix is just a scalar.

With the sizes in hand, we can read the constraint line $A\mathbf{x} \preceq \mathbf{b}$ of the diet problem. The left side $A\mathbf{x}$ is 4×1 and the right side $\mathbf{b}$ is 4×1 , and the inequality compares them entry by entry:

$$A\mathbf{x} = \begin{bmatrix} -100x_1 - 50x_2 \\ 100x_1 + 50x_2 \\ 3x_1 + 4x_2 \\ -20x_1 - 70x_2 \end{bmatrix} \preceq \begin{bmatrix} -2000 \\ 2500 \\ 100 \\ -700 \end{bmatrix} = \mathbf{b}.$$

These are exactly the four scalar constraints from Section 1.3, stacked. Each entry of $A\mathbf{x}$ is a dot product, which is the subject of the next section.

3 The Dot Product

3.1 Definition

Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$, their **dot product** is the sum of the products of matching entries:

$$\mathbf{u}^\top \mathbf{v} = u_1v_1 + u_2v_2 + \dots + u_Nv_N.$$

We write the dot product as $\mathbf{u}^\top \mathbf{v}$ because it is a matrix multiplication: $\mathbf{u}^\top$ is $1 \times N$, $\mathbf{v}$ is $N \times 1$, the inner dimensions match, and the result is 1×1 , a scalar.

Notation: In this course, we always write the dot product as $\mathbf{u}^\top \mathbf{v}$. Writing $\mathbf{u}\mathbf{v}$ does not pass the dimension check ($[N \times 1][N \times 1]$), and we avoid $\mathbf{u} \cdot \mathbf{v}$ and $\mathbf{u} \times \mathbf{v}$ because they are easy to confuse with other kinds of multiplication.

The order does not matter: $\mathbf{u}^\top \mathbf{v} = \mathbf{v}^\top \mathbf{u}$, since each term $u_n v_n$ is the same either way.

Example: With $\mathbf{u} = [2, 3]^\top$ and $\mathbf{v} = [4, -1]^\top$,

$$\mathbf{u}^\top \mathbf{v} = \begin{bmatrix} 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \end{bmatrix} = (2)(4) + (3)(-1) = 8 - 3 = 5.$$

Back to the diet problem: The objective is a dot product,

$$\mathbf{c}^\top \mathbf{x} = c_1 x_1 + c_2 x_2 = 1 x_1 + 0.5 x_2,$$

which is the scalar cost we wrote in Section 1.3. Each constraint is also a dot product. If we let $\mathbf{a}_m$ be the vector of numbers in row m of A , so that $\mathbf{a}_m = [a_{m,1}, a_{m,2}]^\top$, then constraint m is

$$\mathbf{a}_m^\top \mathbf{x} = a_{m,1} x_1 + a_{m,2} x_2 \leq b_m.$$

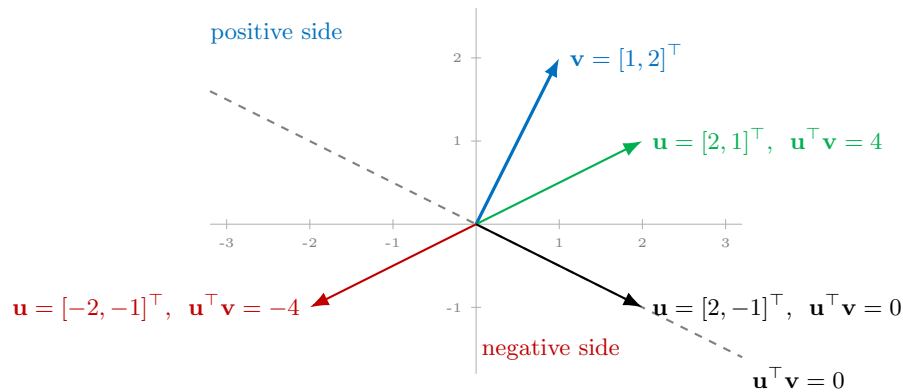
For example, the sugar constraint has $\mathbf{a}_3 = [3, 4]^\top$ and $b_3 = 100$. The product $A\mathbf{x}$ is just these four dot products stacked into a column:

$$A\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{x} \\ \mathbf{a}_2^\top \mathbf{x} \\ \mathbf{a}_3^\top \mathbf{x} \\ \mathbf{a}_4^\top \mathbf{x} \end{bmatrix}.$$

3.2 Sign of the Dot Product

The dot product of two vectors in $\mathbb{R}^2$ tells us something about the directions the vectors point. Fix $\mathbf{v} = [1, 2]^\top$ and try a few different $\mathbf{u}$:

$$\begin{array}{ll} \mathbf{u} = [2, 1]^\top : & \mathbf{u}^\top \mathbf{v} = (2)(1) + (1)(2) = 4 \\ \mathbf{u} = [2, -1]^\top : & \mathbf{u}^\top \mathbf{v} = (2)(1) + (-1)(2) = 0 \\ \mathbf{u} = [-2, -1]^\top : & \mathbf{u}^\top \mathbf{v} = (-2)(1) + (-1)(2) = -4 \end{array}$$



The pattern in the picture holds in general:

- $\mathbf{u}^\top \mathbf{v} > 0$ when $\mathbf{u}$ points roughly the same way as $\mathbf{v}$ (the angle between them is less than 90°).

- $\mathbf{u}^\top \mathbf{v} = 0$ when $\mathbf{u}$ is **perpendicular** to $\mathbf{v}$.
- $\mathbf{u}^\top \mathbf{v} < 0$ when $\mathbf{u}$ points roughly the opposite way from $\mathbf{v}$ (the angle between them is more than 90°).

So the line perpendicular to $\mathbf{v}$ through the origin splits the plane into a positive side, the side $\mathbf{v}$ points toward, and a negative side. Every vector $\mathbf{u}$ on the line itself has $\mathbf{u}^\top \mathbf{v} = 0$.

To develop a better understanding of the sign of the dot product, feel free to use this [Desmos](#) link. Dragging the point $\mathbf{u}$ around will change the value of $\mathbf{u}^\top \mathbf{v}$. Try to make it large and positive, then negative, then exactly zero, and notice where $\mathbf{u}$ sits relative to $\mathbf{v}$ when the dot product is zero.

4 Lines and Half-Planes from Dot Products

In this section, $\mathbf{x} = [x_1, x_2]^\top$ is a point that can be anywhere in the plane, and the other vector in each dot product is fixed. We will look at what the dot product looks like as $\mathbf{x}$ moves.

4.1 Where $\mathbf{c}^\top \mathbf{x}$ Equals 0, 1, and 2

Fix the cost vector $\mathbf{c} = [1, 2]^\top$. Then $\mathbf{c}^\top \mathbf{x} = x_1 + 2x_2$ assigns a number to every point $\mathbf{x}$ in the plane. For example,

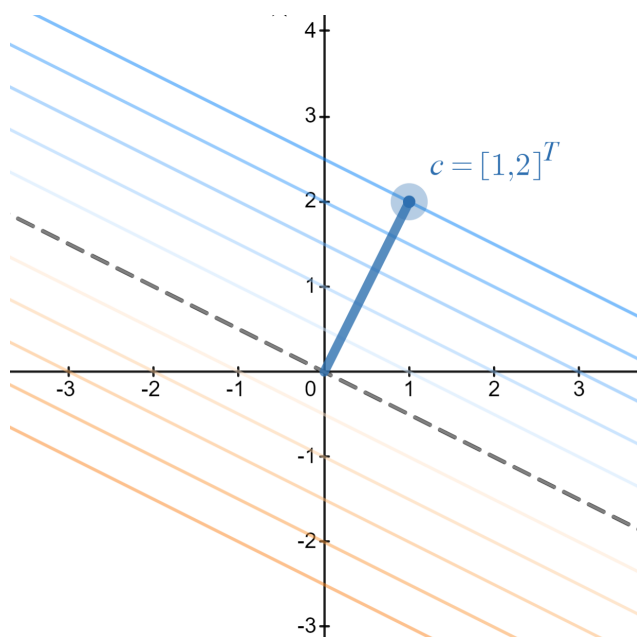
$$\mathbf{c}^\top \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 3, \quad \mathbf{c}^\top \begin{bmatrix} 2 \\ -1 \end{bmatrix} = 0, \quad \mathbf{c}^\top \begin{bmatrix} -4 \\ 2 \end{bmatrix} = 0, \quad \mathbf{c}^\top \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0.$$

Where $\mathbf{c}^\top \mathbf{x} = 0$: By Section 3.2, these are exactly the points $\mathbf{x}$ that are perpendicular to $\mathbf{c}$. They form a line through the origin, perpendicular to $\mathbf{c}$. The three zero-valued points above, $[2, -1]^\top$, $[-4, 2]^\top$, and $[0, 0]^\top$, all lie on this line.

Where $\mathbf{c}^\top \mathbf{x} = 1$, and where $\mathbf{c}^\top \mathbf{x} = 2$: Checking a few more points,

$$\mathbf{c}^\top \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1, \quad \mathbf{c}^\top \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 1, \quad \mathbf{c}^\top \begin{bmatrix} 2 \\ 0 \end{bmatrix} = 2, \quad \mathbf{c}^\top \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 2.$$

The points where $\mathbf{c}^\top \mathbf{x} = 1$ also form a line, parallel to the zero line, and the points where $\mathbf{c}^\top \mathbf{x} = 2$ form another parallel line beyond it. Since $\mathbf{c}^\top \mathbf{x}$ is positive on the side that $\mathbf{c}$ points toward, these lines are shifted from the origin in the direction of $\mathbf{c}$. The lines where $\mathbf{c}^\top \mathbf{x} = -1$ and $\mathbf{c}^\top \mathbf{x} = -2$ are shifted the same way in the opposite direction.



In the figure, $\mathbf{c} = [1, 2]^T$ is the thick arrow. The dashed line is where $\mathbf{c}^T \mathbf{x} = 0$. The blue lines are where $\mathbf{c}^T \mathbf{x}$ is $1, 2, 3, \dots$, and the orange lines are where it is $-1, -2, -3, \dots$, with darker colors for values further from zero. Every one of these lines is perpendicular to $\mathbf{c}$. The relationship between the vector $\mathbf{c}$ and these lines is the thing to hold onto: the lines are perpendicular to $\mathbf{c}$, and the value of $\mathbf{c}^T \mathbf{x}$ grows as we move in the direction $\mathbf{c}$ points.

To develop a better understanding of this relationship, feel free to use these Desmos links. In [LP: Cost at points](#), moving the vector $\mathbf{c}$ changes the value of $\mathbf{c}^T \mathbf{x}$ at each of the marked points; try to make all of the values positive, then all negative, then make some of them zero. In [LP: Zero cost](#) and [LP: Cost contours](#), moving $\mathbf{c}$ moves the lines; notice that they stay perpendicular to $\mathbf{c}$ no matter where you drag it.

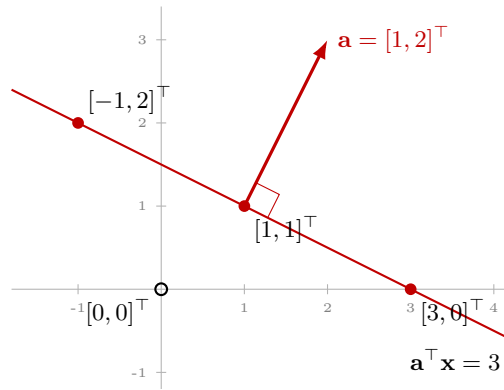
4.2 A Linear Equality: $\mathbf{a}^T \mathbf{x} = b$

Now we use the letters that we will use for constraints. Fix a vector $\mathbf{a} = [a_1, a_2]^T$ and a number b . The points $\mathbf{x}$ that satisfy

$$\mathbf{a}^T \mathbf{x} = b, \quad \text{that is,} \quad a_1 x_1 + a_2 x_2 = b,$$

form a line. This is the same picture as Section 4.1 with new names: the line is perpendicular to $\mathbf{a}$, and the number b says how far the line is shifted from the origin. When $b = 0$, the line passes through the origin. Larger positive b moves the line further in the direction $\mathbf{a}$ points, and negative b moves it the other way.

Example: With $\mathbf{a} = [1, 2]^T$ and $b = 3$, the line is $x_1 + 2x_2 = 3$. The points $[3, 0]^T$, $[1, 1]^T$, and $[-1, 2]^T$ are all on the line, since each gives $\mathbf{a}^T \mathbf{x} = 3$. The origin is not on the line, since $\mathbf{a}^T [0, 0]^T = 0 \neq 3$.



We have drawn $\mathbf{a}$ starting from a point on the line rather than from the origin, to make it easy to see that $\mathbf{a}$ is perpendicular to the line. Where we draw a vector does not change the vector; only its direction and length matter.

4.3 A Linear Inequality: $\mathbf{a}^\top \mathbf{x} \leq b$

Keep the same $\mathbf{a}$ and b , and change the equality to an inequality. The points $\mathbf{x}$ that satisfy

$$\mathbf{a}^\top \mathbf{x} \leq b, \quad \text{that is,} \quad a_1 x_1 + a_2 x_2 \leq b,$$

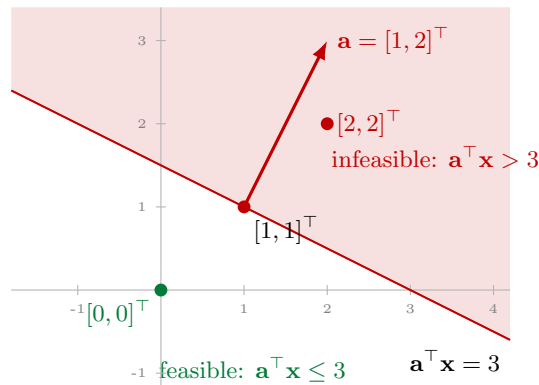
form a **half-plane**: everything on one side of the line $\mathbf{a}^\top \mathbf{x} = b$, including the line itself. We call the points that satisfy the inequality **feasible** and the points that violate it **infeasible**.

Which side is feasible? Start at any point on the line, where $\mathbf{a}^\top \mathbf{x} = b$ exactly, and take a step in the direction of $\mathbf{a}$. By Section 3.2, moving in the direction of $\mathbf{a}$ increases $\mathbf{a}^\top \mathbf{x}$, so after the step $\mathbf{a}^\top \mathbf{x} > b$ and the inequality is violated. Stepping in the opposite direction decreases $\mathbf{a}^\top \mathbf{x}$ and keeps the inequality satisfied. So:

The vector $\mathbf{a}$ points into the infeasible side of the constraint $\mathbf{a}^\top \mathbf{x} \leq b$.

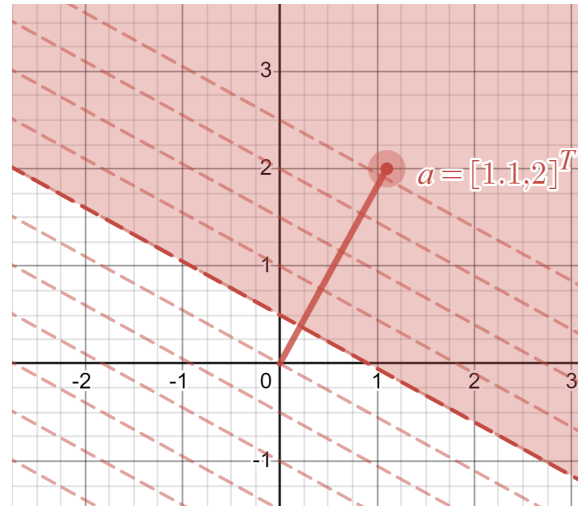
This holds for every value of b , including negative b . The number b only moves the boundary line; the vector $\mathbf{a}$ alone determines which side is infeasible.

Example: With $\mathbf{a} = [1, 2]^\top$ and $b = 3$, the constraint is $x_1 + 2x_2 \leq 3$. The origin is feasible, since $\mathbf{a}^\top [0, 0]^\top = 0 \leq 3$. The point $[2, 2]^\top$ is infeasible, since $\mathbf{a}^\top [2, 2]^\top = 6 > 3$. The point $[1, 1]^\top$ is on the boundary and is feasible, since $3 \leq 3$.



What about $\geq$? A constraint $\mathbf{a}^\top \mathbf{x} \geq b$ is the same as $(-\mathbf{a})^\top \mathbf{x} \leq -b$ after multiplying both sides by -1 , just as we did in Section 1.3. So it is a half-plane too, and the vector pointing into its infeasible side is $-\mathbf{a}$. In this course we will always convert to $\leq$ first, so that the rule “ $\mathbf{a}$ points into the infeasible side” applies to every constraint we write.

To develop a better understanding of a single linear constraint, feel free to use this [Desmos](#) link. Dragging $\mathbf{a}$ changes both the direction of the boundary line and which side is shaded, and the slider for b moves the line. Try a negative value of b and confirm that $\mathbf{a}$ still points into the shaded (infeasible) side.



In the figure, $\mathbf{a} = [1.1, 2]^\top$ and the shaded region is the infeasible side of $\mathbf{a}^\top \mathbf{x} \leq b$. The dashed lines are where $\mathbf{a}^\top \mathbf{x}$ takes on other constant values, parallel to the boundary and perpendicular to $\mathbf{a}$.

5 Summary

Expression	What it is
$\mathbf{u}^\top \mathbf{v}$	A scalar, $u_1 v_1 + \dots + u_N v_N$. Positive when $\mathbf{u}$ and $\mathbf{v}$ point roughly the same way, zero when they are perpendicular, negative when they point roughly opposite ways.
$\mathbf{c}^\top \mathbf{x}$	A number attached to every point $\mathbf{x}$. The points where $\mathbf{c}^\top \mathbf{x} = 0$ form the line through the origin perpendicular to $\mathbf{c}$; the points where it equals 1, 2, $\dots$ form parallel lines shifted in the direction $\mathbf{c}$ points.
$\mathbf{a}^\top \mathbf{x} = b$	A line perpendicular to $\mathbf{a}$, shifted from the origin by an amount set by b .
$\mathbf{a}^\top \mathbf{x} \leq b$	A half-plane bounded by that line. The vector $\mathbf{a}$ points into the infeasible side.

In lecture, we will stack several constraints $\mathbf{a}_m^\top \mathbf{x} \leq b_m$ into $A\mathbf{x} \preceq \mathbf{b}$, combine them with the objective $\mathbf{c}^\top \mathbf{x}$, and look at the resulting picture for the diet problem.