1 Administrivia

I'm running a reading group on advanced ML topics this semester. We haven't covered ML yet in this class, but if you're interested you're welcome to attend. You must read the papers and be prepared for a discussion, though (even if your contribution is to say "what is this"?). See my webpage for details. The group will continue in future semesters, and you'll continue to be welcome.

2 Linear Programming (LP)

- In the 1930's, the US military was trying to figure out how best to feed its troops. They had been just writing down menus and ordering lots of food, but there were occasional problems with malnourishment and there was always the worry that they might be paying far too much to feed the troops (when you're talking about feeding tens or hundreds of thousands of people, spending a few cents extra per adds up to a lot of dollars.)
- In this new age of scientific nutrition, they were interested in converting over to doing things in a more principled way.
- Specifically, they were faced with a problem of this form:
- Each soldier needs to get 2900 calories a day, needs 10 mg of iron, 1000 μg of vitamin A, 2 mg of vitamin B_6, 58 grams of protein, etc.
- Steak costs \$$v_s$/lb and provides c_s calories, i_s mg of iron, a_s mg of vitamin A, b_s mg of vitamin B_6, p_s grams of protein, etc.
- Broccoli costs \$$v_b$/lb and provides c_b cals, i_b mg of iron, a_b mg of vitamin A, etc.
- So on through all of the possible foods that they could purchase in bulk and provide to their troops.
- The question is: what should the daily diet be such that each soldier gets the necessary nutritional balance (calories, vitamins, sodium, protein, etc.) *at the minimum cost*?
- This became the canonical problem in **linear programming** (LP).
- Initially, they solved this with a lot of guesswork and estimation, and did pretty well: George Stigler estimated that the optimal diet would cost $39.93/year (in 1939 dollars, of course).
- In 1947 Dantzig developed the **simplex algorithm**, which solved the problem exactly. They applied it to Sigler's original version of the diet problem to get an optimal cost of $39.69/year. So Stigler had gotten within 24 *cents per year* of the correct value. Solution of this problem required nine equations in 77 unknowns and took 120 person-days of effort (they didn't have a real computer to work with, so they put nine clerks with desk calculators to run the algorithm by hand).

- The linear programming formulation of this problem looks like this. Let x_i be the amount of food i in the diet. Then you want to:

$$
\begin{array}{ll}
\text{Minimize} & \sum_{i \in \text{foods}} x_i v_i \\
\text{Subject to} & \sum_{i \in \text{foods}} x_i c_i \geq 2900 \\
& \sum_{i \in \text{foods}} x_i s_i \geq 1000 \\
& \sum_{i \in \text{foods}} x_i a_i \geq 2 \\
& \vdots \\
& x_i \geq 0 \ \ \forall i
\end{array}
$$

- This just says, in a formal way, that you have to meet all of the nutritional needs, but that in addition to that, you want to minimize cost.

- The first line is called the **objective function** (a.k.a., criterion, cost function, or optimization function). Much like in the search problems we've already talked about.

- Note that the variables under your control here are the x variables. These just represent the amount of each type of food that we'll be buying. Things like c_b and v_s are out of our control – they're fixed quantities or parameters to the system. So what you're trying to do is find a single vector of xs that minimizes the cost while, at the same time, meeting all your constraints. The dimension of the space you're searching through is defined by the number of x variables you have. I.e., $x_1 \ldots x_n \Rightarrow \mathbb{R}^n$

- The rest of the lines are called **constraints** (a.k.a., boundary conditions) – they constrain which solutions are possible.

- E.g., the last set of constraints says that you can't eat a *negative* amount of any food. (This should be plausible.) The earlier constraints say that you have to meet the nutritional requirements.

- Any x vector that meets all the constraints is called a **feasible vector** and the set of all feasible vectors is, clearly, the **feasible region**

- Note a couple of important points:

  – The criterion function is *linear* in x. This is why this is called *linear* programming. (The *programming* part came about historically – remember that this stuff was being figured out before we actually had programmable computers).

  – The constraints are *also* linear in x. This has important geometric consequences that we'll discuss shortly.

- Most of you have probably solved systems of linear equations in earlier classes, so this should sound vaguely familiar, but it turns out to be somewhat more complex than simple Gaussian elimination. How *would* you go about solving this kind of problem?

- Let's look at a very restricted version of this problem: *see the exercise slide*

- Next time, we'll talk about the simplex method in detail and describe the linear algebra that's necessary to solve this problem. For the moment, though, we'll focus on the geometry of it and some intuitive understanding.

- From the solution to the excercise, it's clear that the feasible region is bounded by line segments. It's unbounded to the upper right, in this case. In general, there are three things that can happen:

  - There could be no feasible region. In this case, the system is unsolvable – no solution exists.
  - The feasible region could be unbounded in one or more directions. If your cost function is decreasing in that direction, then the solution is unbounded – it's possible to achieve $-\infty$ cost. This usually doesn't happen for real problems.
  - The feasible region could be bounded on all sides.

- As a bit of terminology: in a plane, a linear subset is just called a **line**; in 3-space, a linear subset is a **plane**. What is a linear subset of an arbitrary n-dimensional space called? A: **hyperplane.**

- Similarly, in a plane, a set bounded by lines is called a **polygon**; in 3-space, a set bounded by planes is called a **polyhedron**. What is a set bounded by hyperplanes in n-space called? A: **polytope**.

- The region defined by a linear inequality is called a **halfspace**.