Lecture 15: CS395T Numerical Optimization for Graphics and AI — Linear Programming (Simplex Method

Qixing Huang
The University of Texas at Austin
huangqx@cs.utexas.edu

Disclaimer

This note is adapted from

- Section 13 of *Numerical Optimization* by Jorge Nocedal and Stephen J. Wright. Springer series in operations research and financial engineering. Springer, New York, NY, 2. ed. edition, (2006)

1 Standard Form of a Linear Program

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq 0 \end{aligned} \tag{1}$$

We will talk about how to convert the following form into the standard form:

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\text{minimize}} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & A_{\text{eq}}\boldsymbol{x} = \boldsymbol{b}_{\text{eq}} \\ & A_{\text{ineq}}\boldsymbol{x} \geq \boldsymbol{b}_{\text{ineq}} \end{aligned} \tag{2}$$

2 Applications of Linear Programming in AI and Graphics

The particular formulations were described in class.

- Shape Matching
- Shape Reconstruction (via L1 minimization)
- Parameterization
- MAP Inference (Structural Prediction)
- Ranking from Pairwise Measurements
- Trajectory Planning

Additional applications were discussed in class.

3 Simplex Method

3.1 Introduction

In geometric terms, the feasible region is defined by all values of $\boldsymbol{x}$ such that

$$A\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{x} \geq 0$$

is a (possibly unbounded) convex polytope. There is a simple characterization of the extreme points or vertices of this polytope, namely an element $\boldsymbol{x} = (x_1, \cdots, x_n)^T$ of the feasible region is an extreme point if and only if the subset of column vectors A_i corresponding to the nonzero entries of $\boldsymbol{x}(x_i \neq 0)$ are linearly independent. In this context such a point is known as a **basic feasible solution (BFS)**.

It can be shown that for a linear program in standard form, if the objective function has a maximum value on the feasible region then it has this value on (at least) one of the extreme points. This in itself reduces the problem to a finite computation since there is a finite number of extreme points, but the number of extreme points is unmanageably large for all but the smallest linear programs.

It can also be shown that if an extreme point is not a maximum point of the objective function then there is an edge containing the point so that the objective function is strictly increasing on the edge moving away from the point. If the edge is finite then the edge connects to another extreme point where the objective function has a greater value, otherwise the objective function is unbounded above on the edge and the linear program has no solution. The simplex algorithm applies this insight by walking along edges of the polytope to extreme points with greater and greater objective values. This continues until the maximum value is reached or an unbounded edge is visited, concluding that the problem has no solution. The algorithm always terminates because the number of vertices in the polytope is finite; moreover since we jump between vertices always in the same direction (that of the objective function), we hope that the number of vertices visited will be small.

The solution of a linear program is accomplished in two steps. In the first step, known as Phase I, a starting extreme point is found. Depending on the nature of the program this may be trivial, but in general it can be solved by applying the simplex algorithm to a modified version of the original program. The possible results of Phase I are either that a basic feasible solution is found or that the feasible region is empty. In the latter case the linear program is called infeasible. In the second step, Phase II, the simplex algorithm is applied using the basic feasible solution found in Phase I as a starting point. The possible results from Phase II are either an optimum basic feasible solution or an infinite edge on which the objective function is unbounded below.

3.2 Simplex Tableau

A linear program in standard form can be represented as a tableau of the form

$$\begin{bmatrix} 0 & -\boldsymbol{c}^T & 0 \\ 0 & A & \boldsymbol{b} \end{bmatrix}$$

The first row defines the objective function and the remaining rows specify the constraints. (Note, different authors use different conventions as to the exact layout.) If the columns of A can be rearranged so that it contains the identity matrix of order p (the number of rows in A) then the tableau is said to be in canonical form. The variables corresponding to the columns of the identity matrix are called basic variables while the remaining variables are called nonbasic or free variables. If the values of the nonbasic variables are set to 0, then the values of the basic variables are easily obtained as entries in $\boldsymbol{b}$ and this solution is a basic feasible solution. The algebraic interpretation here is that the coefficients of the linear equation represented by each row are either 0, 1, or some other number. Each row will have 1 column with value 1 1, $p-1$ columns with coefficients 0, and the remaining columns with some other coefficients (these other variables represent our non-basic variables). By setting the values of the non-basic variables we ensure in each row that the value of the variable represented by a 1 in its column is equal to the b value at that row.

Conversely, given a basic feasible solution, the columns corresponding to the nonzero variables can be expanded to a nonsingular matrix. If the corresponding tableau is multiplied by the inverse of this matrix then the result is a tableau in canonical form.

Let

$$\begin{bmatrix} 0 & -\boldsymbol{c}_B^T & -\boldsymbol{c}_D^T & 0 \\ 0 & I & D & \boldsymbol{b} \end{bmatrix}$$

be a tableau in canonical form. Additional row-addition transformations can be applied to remove the coefficients $\boldsymbol{c}_B^T$ from the objective function. This process is called pricing out and results in a canonical tableau

$$\begin{bmatrix} 0 & \mathbf{0} & -\bar{\boldsymbol{c}}_D^T & z_B \\ 0 & I & D & \boldsymbol{b} \end{bmatrix}$$

where z_B is the value of the objective function at the corresponding basic feasible solution. The updated coefficients, also known as relative cost coefficients, are the rates of change of the objective function with respect to the nonbasic variables.

3.3 Pivot Operations

The geometrical operation of moving from a basic feasible solution to an adjacent basic feasible solution is implemented as a pivot operation. First, a nonzero pivot element is selected in a nonbasic column. The row containing this element is multiplied by its reciprocal to change this element to 1, and then multiples of the row are added to the other rows to change the other entries in the column to 0. The result is that, if the pivot element is in row r, then the column becomes the r-th column of the identity matrix. The variable for this column is now a basic variable, replacing the variable which corresponded to the r-th column of the identity matrix before the operation. In effect, the variable corresponding to the pivot column enters the set of basic variables and is called the entering variable, and the variable being replaced leaves the set of basic variables and is called the leaving variable. The tableau is still in canonical form but with the set of basic variables changed by one element.

3.4 Algorithm

Let a linear program be given by a canonical tableau. The simplex algorithm proceeds by performing successive pivot operations each of which give an improved basic feasible solution; the choice of pivot element at each step is largely determined by the requirement that this pivot improves the solution.

Entering variable selection

Since the entering variable will, in general, increase from 0 to a positive number, the value of the objective function will decrease if the derivative of the objective function with respect to this variable is negative. Equivalently, the value of the objective function is decreased if the pivot column is selected so that the corresponding entry in the objective row of the tableau is positive.

If there is more than one column so that the entry in the objective row is positive then the choice of which one to add to the set of basic variables is somewhat arbitrary and several entering variable choice rules such as Devex algorithm have been developed.

If all the entries in the objective row are less than or equal to 0 then no choice of entering variable can be made and the solution is in fact optimal. It is easily seen to be optimal since the objective row now corresponds to an equation of the form

$$z(\boldsymbol{x}) = z_B + \text{non-negative terms corresponding non-basic variables.}$$

Note that by changing the entering variable choice rule so that it selects a column where the entry in the objective row is negative, the algorithm is changed so that it finds the maximum of the objective function rather than the minimum.

Leaving variable selection

Once the pivot column has been selected, the choice of pivot row is largely determined by the requirement that the resulting solution be feasible. First, only positive entries in the pivot column are considered since this guarantees that the value of the entering variable will be non-negative. If there are no positive entries in the pivot column then the entering variable can take any non-negative value with the solution remaining feasible. In this case the objective function is unbounded below and there is no minimum.

Next, the pivot row must be selected so that all the other basic variables remain positive. A calculation shows that this occurs when the resulting value of the entering variable is at a minimum. In other words, if the pivot column is c, then the pivot row r is chosen so that

$$b_r/a_{rc}$$

is the minimum over all r so that $a_{rc} > 0$. This is called the minimum ratio test. If there is more than one row for which the minimum is achieved then a dropping variable choice rule can be used to make the determination.

3.5 Complexity of Simplex Algorithm

The simplex algorithm indeed visits all 2^n vertices in the worst case [1], and this turns out to be true for any deterministic pivot rule. However, in a landmark paper using a smoothed analysis, Spielman and Teng [2] proved that when the inputs to the algorithm are slightly randomly perturbed, the expected running time of the simplex algorithm is polynomial for any inputs – this basically says that for any problem there is a "nearby" one that the simplex method will efficiently solve, and it pretty much covers every real-world linear program you'd like to solve. Afterwards, Kelner and Spielman [3] introduced a polynomial time randomized simplex algorithm that truley works on any inputs, even the bad ones for the original simplex algorithm.

[1] https://en.wikipedia.org/wiki/Klee-Minty_cube
[2] https://arxiv.org/pdf/cs/0111050.pdf
[3] http://www.cs.yale.edu/homes/spielman/Research/SimplexStoc.pdf