Board notes from Friday February 18  Homework due Friday February 22, returned Monday February 25.
  Midterm Wednesday February 27

Start of Dynamic Programming


Not algorithms: Duality theory of linear programming. KKT theorem. Good guesses.

Per Wikipedia:

Algorism is what we now know as basic arithmetic with decimals. Arabian mathematicians documented new arithmetic methods and made many other contributions to decimal arithmetic.

The words algorism and algorithm come from the name al-Khwarizmi ("the one from Khwarizm" – possibly in present-day Uzbekistan), Persian astronomer and mathematician. He wrote a treatise in Arabic in 825 AD, "On Calculation with Hindu Numerals." Other Arabic words in mathematics and elsewhere: algebra, zero, alfalfa, alcohol, tariff, azimuth, zenith, cotton, sugar, spinach.

Although Gaussian elimination is named for the mathematician Carl Friedrich Gauss, the earliest presentation of it can be found in the important Chinese mathematical text The Nine Chapters on the Mathematical Art, dated approximately 150 B.C.E.

Now: dynamic programming, a highly versatile algorithm
Terms to be illustrated: stage, state, action, state-transition

Maximize \( \ln(x_1) + \ln(x_2) + \ln(x_3) \)
subject to \( x_1 + x_2 + x_3 = s_1 \)
\( x_i \geq 0 \)

Interpretation: we are back-packing in the Wind River range and we start with a supply \( s_1 \) of food. We will eat \( x_i \) on day \( i \). The (not too difficult) task is to choose appropriate \( x_i \)'s. The purpose of the exercise is to introduce and illustrate the following concepts from dynamic programming.

Stages: first \( x_3 \), then \( x_2 \), and last \( x_1 \). Think ahead, reason back.
States: \( x_i \) equals amount of food you still have at the start of stage \( i \).
Actions: \( x_i \) equals what you eat at stage \( i \).
State-transition: \( s_{i+1} = s_i - x_i \).
Value function: \( f^*(s_i) \) equals the utility achieved from stage \( i \) until the end.
Solution:

\[ f^*_3(s_3) = \text{maximum}_{x_3} \ln(x_3) \text{ subject to } x_3 \leq s_3 \]

\[ = \ln(s_3) \]

\[ x^*_3 = s_3 \]

End of computations for stage 3.

\[ f^*_2(s_2) = \text{maximum}_{x_2} \ln(x_2) + f^*_3(s_2 - x_2) \]

\[ = \text{maximum}_{x_2} \ln(x_2) + \ln(s_2 - x_2) \]

Differentiate w.r.t. \( x_2 \) and set equal to zero and solve for \( x_2 \).

\[ 0 = \frac{1}{x_2} - \frac{1}{s_2 - x_2} \]

\[ \frac{1}{x_2} = \frac{1}{s_2 - x_2} \]

\[ x_2 = s_2 - x_2 \]

\[ x^*_2 = \frac{s_2}{2} \]

\[ f^*_2(s_2) = \ln\left(\frac{s_2}{2}\right) + \ln\left(\frac{s_2}{2}\right) \]

\[ = 2 \ln\left(\frac{s_2}{2}\right) \]

\[ = 2 \ln(s_2) - 2 \ln 2 \]

End of computations for stage 2:

\[ f^*_1(s_1) = \text{maximum}_{x_1} \ln(x_1) + f^*_2(s_1 - x_1) \]

\[ = \text{maximum}_{x_1} \ln(x_1) + 2 \ln(s_1 - x_1) - 2 \ln 2 \]

Differentiate w.r.t. \( x_1 \) and solve

\[ 0 = \frac{1}{x_1} - \frac{2}{s_1 - x_1} \]

\[ 2x_1 = s_1 - x_1 \]

\[ x^*_1 = \frac{s_1}{3} \]

\[ f^*_1(s_1) = \ln\left(\frac{s_1}{3}\right) + 2 \ln\left(\frac{2}{3}s_1\right) - 2 \ln 2 \]

\[ = 3 \ln\left(\frac{s_1}{3}\right) \]

The exercise illustrated the following:

Stages, states, action, state transition, principle of optimality