Gurobi Logs, SVM, Production Planning

(2/7/2024)

Recap: A clustering problem

Input:

- n objects numbered $1, 2, \ldots, n$
- Desired number of clusters k, and a lower bound ℓ on the number of objects in a cluster
- A set D of pairs of dissimilar objects

Output:

- A partitioning of the objects into cluster $C_1, C_2, \ldots, C_k$

  Partitioning means:

  (i) $C_1 \cup C_2 \cup \cdots \cup C_k = \{1, 2, \ldots, n\}$, and

  (ii) $C_s \cap C_t = \varnothing$ for all $s \neq t$.

Goal:

- Minimize the total number of pairs $\{i, j\}$ where i and j are clustered in the same cluster, but are dissimilar (meaning, $\{i, j\} \in D$)

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijs} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if both objects } i \text{ and } j \text{ are put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(only for } i < j\text{)}$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\}\in D} \sum_{s\in K} x_{ijs} & \longleftarrow & \text{dissimilar pairs in the same cluster} \\
\text{s.t.} & \sum_{s\in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
 & \sum_{i\in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters}
\end{array}$$

How do we say: $x_{ijs} = \begin{cases} 1 & \text{if } y_{is} = 1 \text{ and } y_{js} = 1 \\ 0 & \text{otherwise.} \end{cases}$

i.e. We want $x_{ijs} = y_{is} y_{js}$

$$x_{ijs} \in \{0,1\} \qquad \forall i < j \in N\ , \forall s \in K$$

$$y_{is} \in \{0,1\} \qquad \forall i \in N\ , \forall s \in K$$

How to multiply binary variables

Claim

If x, y_1, y_2 satisfy the McCormick constraints

$$\begin{aligned} x &\leq y_1, \\ x &\leq y_2, \\ x &\geq 0, \\ x &\geq y_1 + y_2 - 1, \\ y_1, y_2 &\in \{0, 1\} \end{aligned}$$

then $x = y_1 y_2$. Note: x is not declared to be binary

Proof : Any point satisfying the above constraints has $y_1, y_2 \in \{0, 1\}$.

- If either y_1 or y_2 is 0, then the first 3 constraints imply that $x = 0$.
- The only remaining case is when both y_1 or y_2 is 1. In this case, the first constraint and the last one imply that $x = 1$.

Clustering Problem: Formulation 1^+

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad (\text{ for } i < j)$$

IP Formulation:

$$\begin{array}{llll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \ , s \in K & \\
& x_{ijs} \leq y_{is}, \ x_{ijs} \leq y_{js} & \forall i < j \in N \ , s \in K & \\
& \cancel{x_{ijs} \in \{0,1\}} \ \ x_{ijs} \geq 0 & \forall i < j \in N \ , s \in K & \\
& y_{is} \in \{0,1\} & \forall i \in N \ , s \in K &
\end{array}$$

Clustering Problem: Formulation 1^{++}

Decision variables

$$y_{is} = \begin{cases} 1 & \text{if } i \text{ in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad x_{ijs} = \begin{cases} 1 & \text{if both } i \text{ and } j \text{ are in } C_s \\ 0 & \text{otherwise.} \end{cases} \qquad (\text{ for } i < j)$$

IP Formulation:

$$\begin{array}{lrll}
\min & \sum_{\{i,j\} \in D} \sum_{s \in K} x_{ijs} & & \\
\text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\
& \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\
& x_{ijs} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \,, s \in K & \\
& \cancel{x_{ijs} \leq y_{is}}, \ \cancel{x_{ijs} \leq y_{js}} & \cancel{\forall i < j \in N} \,, \cancel{s \in K} & \\
& x_{ijs} \geq 0 & \forall i < j \in N \,, s \in K & \\
& y_{is} \in \{0, 1\} & \forall i \in N \,, s \in K &
\end{array}$$

Note: In an optimal solution, x_{ijs} is not guaranteed to be 1 when $y_{is} = 0$ or $y_{js} = 0$, but that's OK (obj function).

Clustering Problem: Formulation 2

Decision variables

$$y_{is} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if object } i \text{ is put in cluster } C_s \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if objects } i \text{ and } j \text{ are put in the same cluster} \\ 0 & \text{otherwise.} \qquad\qquad (\text{for } i < j) \end{cases}$$

IP Formulation:

$$\begin{array}{lrll} \min & \sum_{\{i,j\} \in D} z_{ij} & & \\ \text{s.t.} & \sum_{s \in K} y_{is} = 1 & \forall i \in N & \longleftarrow \quad \text{objects} \\ & \sum_{i \in N} y_{is} \geq \ell & \forall s \in K & \longleftarrow \quad \text{clusters} \\ & z_{ij} \geq y_{is} + y_{js} - 1 & \forall i < j \in N \ , \forall s \in K & \\ & z_{ij} \geq 0 & \forall i < j \in N & \\ & y_{is} \in \{0, 1\} & \forall i \in N, \ \forall s \in K & \end{array}$$

Comparing the formulations

- Size of the formulation:

	variables	constraints	nonzeros
Formulation 0	2,460	4,723	14,280
Formulation 1	2,460	7,063	16,620
Formulation 1^+	2,460	7,063	16,620
Formulation 1^{++}	2,460	2,383	7,260
Formulation 2	900	2,383	7,260

- Solution time:

	B& B nodes	Simplex iterations	Solution time
Formulation 0	40,560	11,210,558	327.78 seconds
Formulation 1	23,050	4,033,965	159.24 seconds
Formulation 1^+	4,715	842,274	32.87 seconds
Formulation 1^{++}	7,471	542,972	5.41 seconds
Formulation 2	4,392	369,175	3.98 seconds

Gurobi Output for Formulation 2

```
Gurobi log file for last model:
================================

900 variables, all binary
2383 constraints, all linear; 7260 nonzeros
40 equality constraints
2343 inequality constraints
1 linear objective; 226 nonzeros.

Gurobi 9.1.1: outlev=1
threads=4
Gurobi Optimizer version 9.1.1 build v9.1.1rc0 (linux64)
Thread count: 32 physical cores, 64 logical processors, using up
Optimize a model with 2383 rows, 900 columns and 7260 nonzeros
Model fingerprint: 0xae721739
Variable types: 0 continuous, 900 integer (900 binary)
Coefficient statistics:
Matrix range     [1e+00, 1e+00]
Objective range  [1e+00, 1e+00]
Bounds range     [1e+00, 1e+00]
RHS range        [1e+00, 1e+01]
```

(continued....)

```
Found heuristic solution: objective 81.0000000
Presolve removed 1662 rows and 554 columns
Presolve time: 0.00s
Presolved: 721 rows, 346 columns, 2274 nonzeros
Variable types: 0 continuous, 346 integer (346 binary)

Root relaxation: objective 0.000000e+00, 161 iterations, 0.00 se

* |  Nodes |   Current Node   |  Objective Bounds   |   Work
  |Exp Unex| Obj Depth IntInf|Incumbent BestBd Gap|It/Node Time

0    0       0.0   0    58     81.00  0.00  100%     -    0s
H    0       0                 34.00  0.00  100%     -    0s
0    0       0.0   0    85     34.00  0.00  100%     -    0s
0    0       1.5   0   127     34.00  1.50   95.6%   -    0s
0    0       1.5   0   124     34.00  1.50   95.6%   -    0s
0    2       1.5   0   121     34.00  1.50   95.6%   -    0s
*  271  239        17          32.00  9.04   71.7%  103   0s
H  494  297                    29.00 10.35   64.3%   93.5 0s
```

(continued....)

```
H  630  351                     28.00 11.21   59.9%   94.5 0s
*  633  335        18           27.00 11.21   58.5%   94.3 0s
H  691  316                     25.00 12.30   50.8%   95.0 0s
H  974  354                     24.00 13.91   42.0%   95.5 1s

Explored 4392 nodes (369175 simplex iterations) in 3.98 seconds

Optimal solution found (tolerance 1.00e-04)
Best objective 2.400e+01, best bound 2.400e+01, gap 0.0000%

369175 simplex iterations
4392 branch-and-cut nodes

Cutting planes:
Gomory: 3
MIR: 7
Zero half: 26
RLT: 128
BQP: 60
```

Solving IPs: computation time

- Consider the following LP formulation

$$
\begin{aligned}
\text{minimize} \quad & c^T x \\
\text{subject to} \quad & A^1 x \geq b^1, \\
& A^2 x = b^2, \\
& x \geq 0
\end{aligned}
$$

- The non-zeroes of this formulation is the number of nonzero entries in the matrices A^1 and A^2.
  - LPs are solved using either simplex or interior point algorithms,
  - In both cases one has to solve (many, many) linear equations
  - Computational burden per LP iteration typically grows with the number of non-zero entries of the constraint matrices A^1 and A^2
  - It also grows with the number of rows of A^1 and A^2.
- IP solution time depends on the number of B&B nodes and the LP solution time at each node.

Supervised binary classification

Another ML Example : Supervised binary classification

- We are given m objects and a description of their features.
- For the jth object let $a^j \in \mathbb{R}^n$ denote the associated feature vector.

  Example: $a^j \in \mathbb{R}^3$ could be: (measured in some scale)

  - a_1^j indicates the ellipticity of the object,
  - a_2^j: the length of its stem,
  - a_3^j is its color (in grayscale).
- Each objects belong to one of two classes.

  For example: It is the image of an apple or an orange
- We are interested in designing a classifier which, given a new object, will figure out the class that it belongs to.
- There are many ways of approaching this problem.

  (Ex: Decision trees, random forests, logistic regression, etc.)

Another ML Example : Supervised binary classification

- We are given m objects and a description of their features.
- For the jth object let $a^j \in \mathbb{R}^n$ denote the associated feature vector.

  Example: $a^j \in \mathbb{R}^3$ could be: (measured in some scale)

  - a_1^j indicates the ellipticity of the object,
  - a_2^j: the length of its stem,
  - a_3^j is its color (in grayscale).
- Each objects belong to one of two classes.

  For example: It is the image of an apple or an orange
- We are interested in designing a classifier which, given a new object, will figure out the class that it belongs to.
- There are many ways of approaching this problem.

  (Ex: Decision trees, random forests, logistic regression, etc.)

Linear classifiers for binary classification

- A linear classifier is defined by an n-dimensional coefficient vector $w \in \mathbb{R}^n$ and a number w_0.
- Given an object with feature vector $a \in \mathbb{R}^n$, the classifier declares it to be an apple if

$$\sum_{i=1}^{n} w_i a_i \geq w_0,$$

and an orange if

$$\sum_{i=1}^{n} w_i a_i < w_0.$$

- In words, a linear classifier makes decisions on the basis of a linear combination of the features of the object.
- Our objective is to use known objects to design a "good" linear classifier.

(Train on the known objects to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that would lead to a good linear classifier that you can use on new objects)

Linear classifiers for binary classification

- A linear classifier is defined by an n-dimensional coefficient vector $w \in \mathbb{R}^n$ and a number w_0.
- Given an object with feature vector $a \in \mathbb{R}^n$, the classifier declares it to be an apple if

$$\sum_{i=1}^{n} w_i a_i \geq w_0,$$

  and an orange if

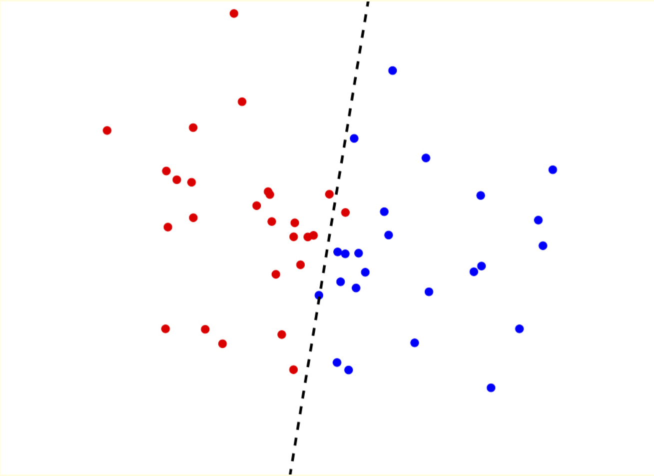
$$\sum_{i=1}^{n} w_i a_i < w_0.$$

- In words, a linear classifier makes decisions on the basis of a linear combination of the features of the object.
- Our objective is to use known objects to design a "good" linear classifier.

  (Train on the known objects to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that would lead to a good linear classifier that you can use on new objects)

Linear classifiers for binary classification

A linear classifier in $\mathbb{R}^2$:



- The coordinates of a point corresponds to its features.

Linear classifiers for binary classification

- How to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$?
- A reasonable approach would be to pick w and w_0 so that the classifier gives correct answer for all known objects (samples).
- Let $S = S^1 \cup S^2$ be the set of known samples with $S^2 = S \setminus S^1$
  - Let S^1 be the set of objects of type 1 (apples), and,
  - Let S^2 be the set of objects of type 2 (oranges)
- We are then looking for some $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that will satisfy:

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Note that the second set of constraints involves a strict inequality and therefore we cannot model it using linear programming.

Linear classifiers for binary classification

- How to pick $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$?
- A reasonable approach would be to pick w and w_0 so that the classifier gives correct answer for all known objects (samples).
- Let $S = S^1 \cup S^2$ be the set of known samples with $S^2 = S \setminus S^1$
  - Let S^1 be the set of objects of type 1 (apples), and,
  - Let S^2 be the set of objects of type 2 (oranges)
- We are then looking for some $w \in \mathbb{R}^n$ and $w_0 \in \mathbb{R}$ that will satisfy:

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Note that the second set of constraints involves a strict inequality and therefore we cannot model it using linear programming.

How to turn the "$<$" into a "$\le$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \ge w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \le w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \;=\; \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0\text{)}$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

$$\bar{w}^T a^j \ge \bar{w}_0, \qquad j \in S^1$$

$$\bar{w}^T a^j \le \bar{w}_0 - 1, \qquad j \in S^2.$$

- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.
  (Note: w and w_0 can have negative entries)

How to turn the "$<$" into a "$\leq$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \leq w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \; = \; \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0)$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

$$\bar{w}^T a^j \geq \bar{w}_0, \qquad j \in S^1$$
$$\bar{w}^T a^j \leq \bar{w}_0 - 1, \qquad j \in S^2.$$

- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.

(Note: w and w_0 can have negative entries)

How to turn the "$<$" into a "$\leq$"

- Notice that if for some choice of w and w_0 we have

$$w^T a^j \geq w_0, \quad \forall j \in S^1 \qquad \text{(apples)}$$

$$w^T a^j < w_0, \quad \forall j \in S^2 \qquad \text{(oranges).}$$

- Then $w^T a^j \leq w_0 - \epsilon$ for all $j \in S^2$, where

$$\epsilon \ = \ \min_{j \in S^2} (w_0 - w^T a^j) \qquad \text{(note: } \epsilon > 0\text{)}$$

- Therefore there exists some other choice $\bar{w}', \bar{w}_0'$, obtained by multiplying w and w_0 by a positive scalar $(1/\epsilon)$, that satisfies

$$\begin{aligned} \bar{w}^T a^j &\geq \bar{w}_0, & j \in S^1 \\ \bar{w}^T a^j &\leq \bar{w}_0 - 1, & j \in S^2. \end{aligned}$$

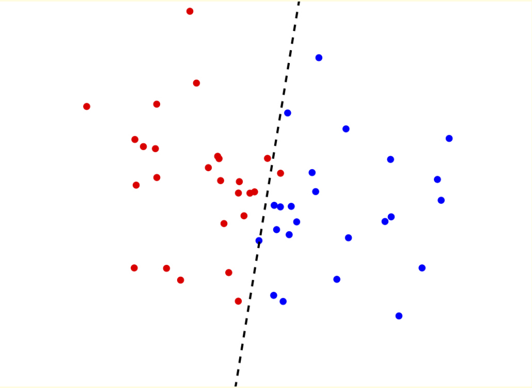
- Therefore, a linear classifier consistent with all available samples is a feasible solution to a linear programming problem.
  (Note: w and w_0 can have negative entries)

Support vector machine problem

- In practice a perfect linear classifier usually does not exist.

  i.e. the following system would be infeasible:

$$
\begin{aligned}
w^T a^j &\geq w_0, & j \in S^1 \\
w^T a^j &\leq w_0 - 1, & j \in S^2.
\end{aligned}
$$



Support vector machine problem

- In practice a perfect linear classifier usually does not exist.
  i.e. the following system would be infeasible:

$$
\begin{aligned}
&w^T a^j \geq w_0, && j \in S^1\\
&w^T a^j \leq w_0 - 1, && j \in S^2.
\end{aligned}
$$

- In this case, we look for a solution that minimizes total error:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in S} \delta_j \\
\text{subject to} \quad & w^T a^j + \delta_j \geq w_0, && j \in S^1\\
& w^T a^j - \delta_j \leq w_0 - 1, && j \in S^2\\
& \delta_j \geq 0 && j \in S^1 \cup S^2
\end{aligned}
$$

  where the variable δ_j measures the classification error of sample j.

- If all $\delta_j = 0$ in the optimal solution, then we have a perfect classifier.

Minimizing misclassified items instead of error

- Minimizing total error:

$$
\begin{array}{lll}
\text{minimize} & \sum_{j \in S} \delta_j & \\
\text{subject to} & w^T a^j + \delta_j \geq w_0, & j \in S^1 \\
& w^T a^j - \delta_j \leq w_0 - 1, & j \in S^2 \\
& \delta_j \geq 0 & j \in S^1 \cup S^2
\end{array}
$$

where the variable δ_j measures the classification error of sample j.

- Minimizing number of misclassified items:

$$
\begin{array}{lll}
\text{minimize} & \sum_{j \in S} z_j & \\
\text{subject to} & w^T a^j + M z_j \geq w_0, & j \in S^1 \\
& w^T a^j - M z_j \leq w_0 - 1, & j \in S^2 \\
& z_j \in \{0, 1\} & j \in S^1 \cup S^2
\end{array}
$$

where M is a large number (max allowed error) and variable z_j indicates if sample j is classified correctly or not.

Minimizing misclassified items instead of error

- Minimizing total error:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in S} \delta_j \\
\text{subject to} \quad & w^T a^j + \delta_j \geq w_0, & j \in S^1 \\
& w^T a^j - \delta_j \leq w_0 - 1, & j \in S^2 \\
& \delta_j \geq 0 & j \in S^1 \cup S^2
\end{aligned}
$$

  where the variable δ_j measures the classification error of sample j.

- Minimizing number of misclassified items:

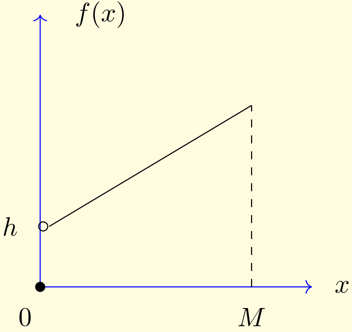
$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in S} z_j \\
\text{subject to} \quad & w^T a^j + M z_j \geq w_0, & j \in S^1 \\
& w^T a^j - M z_j \leq w_0 - 1, & j \in S^2 \\
& z_j \in \{0, 1\} & j \in S^1 \cup S^2
\end{aligned}
$$

  where M is a large number (max allowed error) and variable z_j indicates if sample j is classified correctly or not.

Using Big Ms: Lot-sizing Problem

Fixed Charges

- Economic activities frequently involve fixed and variable costs.
- Example: A production facility.
  - Fixed cost: if anything is produced at all (e.g., cost of starting up machines).
  - Variable cost: linear in the amount produced (e.g., cost of operating machines).



- In this case, the cost is

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ h + cx & \text{if } x > 0 \end{cases}$$

  (with $h, c > 0$).
- This is not a linear function.
- Not even a continuous function.

Modeling Fixed Charges

- We can hanle this using a binary variable $y \in \{0, 1\}$

$$y \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if } x > 0 \quad \text{(some production)} \\ 0 & \text{if } x = 0 \quad \text{(no production).} \end{cases}$$

- Then the total cost of production can now be written as

$$hy + cx.$$

- Let M be some upper bound on the value of variable x.

$$\begin{aligned} x &\leq My \\ y &\in \{0, 1\} \\ x &\geq 0 \end{aligned}$$

Note: $y = 1$, $x = 0$ is feasible but if minimizing with $h > 0$, it is fine!

- Linear programming relaxations of "big M" formulations tend to produce bad LP relaxations.
- One should choose the smallest possible "big M".

Modeling Fixed Charges

- We can hanle this using a binary variable $y \in \{0, 1\}$

$$y \overset{\text{interpretation}}{=} \begin{cases} 1 & \text{if } x > 0 \quad \text{(some production)} \\ 0 & \text{if } x = 0 \quad \text{(no production)}. \end{cases}$$

- Then the total cost of production can now be written as

$$hy + cx.$$

- Let M be some upper bound on the value of variable x.

$$\begin{aligned} & x \leq My \\ & y \in \{0, 1\} \\ & x \geq 0 \end{aligned}$$

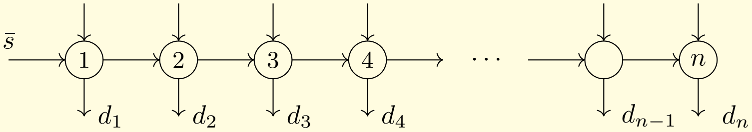
  Note: $y = 1$, $x = 0$ is feasible but if minimizing with $h > 0$, it is fine!

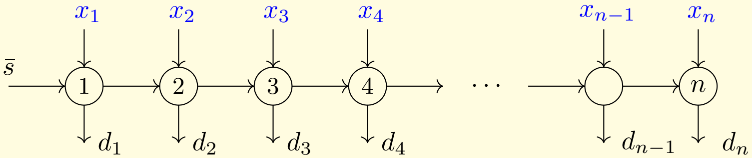
- Linear programming relaxations of "big M" formulations tend to produce bad LP relaxations.
- One should choose the smallest possible "big M".

Example: Uncapacitated lot-sizing problem

Production planning for a single item for an n-period horizon:

- Demand for the item is d_t for $t = 1, \ldots, n$.
- There is a fixed cost f_t of production in period t
- There is a production cost p_t per unit produced in period t
- There is a starting inventory of $\bar{s}$ units available at the beginning.
- There is a storage cost of h_t per unit in period t
- Find the minimum cost production plan to satisfy demand.





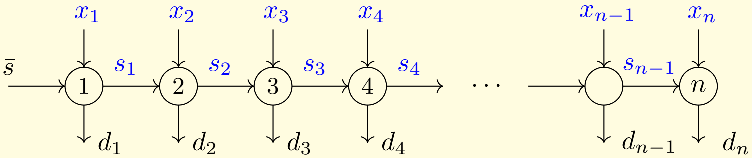
Decision variables:

- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$\begin{array}{lll}
\min & \displaystyle\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t & \\
\text{s.t.} & s_t = s_{t-1} + x_t - d_t & t \in \{1, \ldots, n\} \\
& x_t \leq M y_t & t \in \{1, \ldots, n\} \\
& s_0 = \bar{s}, & \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} & t \in \{1, \ldots, n\}
\end{array}$$

Question: How big should M be?



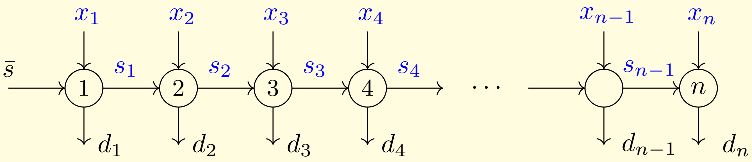
Decision variables:

- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$\begin{aligned}
\min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t && t \in \{1, \ldots, n\} \\
& x_t \leq M y_t && t \in \{1, \ldots, n\} \\
& s_0 = \bar{s}, \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} && t \in \{1, \ldots, n\}
\end{aligned}$$

Question: How big should M be?



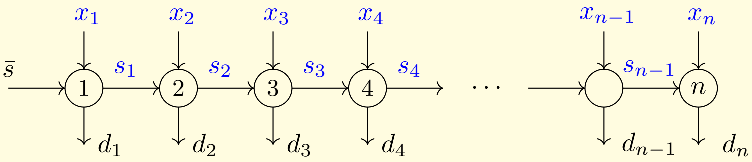
Decision variables:

- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0, 1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\
\text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t && t \in \{1, \ldots, n\} \\
& x_t \leq M y_t && t \in \{1, \ldots, n\} \\
& s_0 = \bar{s}, \\
& s_t,\ x_t \geq 0, \quad y_t \in \{0, 1\},\ x_t \in \mathbb{Z} && t \in \{1, \ldots, n\}
\end{aligned}
$$

Question: How big should M be?



Decision variables:

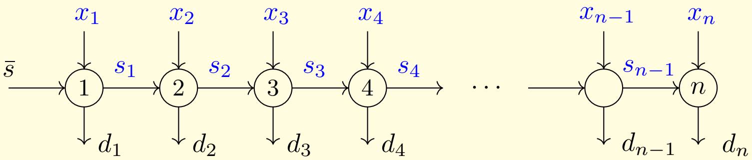
- $x_t \geq 0$ to denote the quantity produced in period t
- $y_t \in \{0,1\}$ to denote if production occurs in period t
- $s_t \geq 0$ to denote the stock at the end of period t

Formulation:

$$\begin{aligned} \min \quad & \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t \\ \text{s.t.} \quad & s_t = s_{t-1} + x_t - d_t && t \in \{1,\ldots,n\} \\ & x_t \leq M y_t && t \in \{1,\ldots,n\} \\ & s_0 = \bar{s}, \\ & s_t,\ x_t \geq 0, \quad y_t \in \{0,1\},\ x_t \in \mathbb{Z} && t \in \{1,\ldots,n\} \end{aligned}$$

Question: How big should M be?

Finding the smallest Big M



- How big should M be?

$$x_t \leq My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

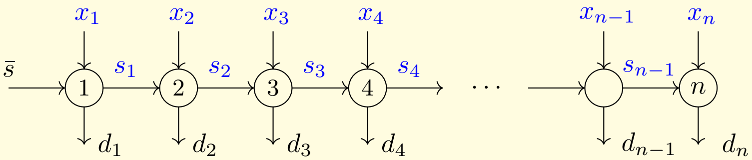
- The big Ms can be different for different t:

$$x_t \leq M^t y_t \qquad t \in T \longleftarrow \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Finding the smallest Big M



- How big should M be?

$$x_t \;\leq\; My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

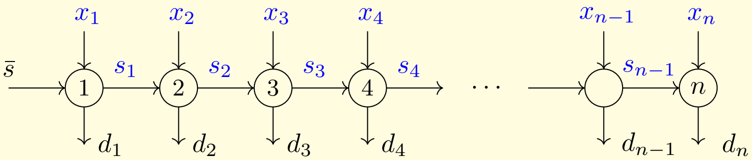
- The big Ms can be different for different t:

$$x_t \;\leq\; M^t y_t \qquad t \in T \;\longleftarrow\; \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Finding the smallest Big M



- How big should M be?

$$x_t \;\leq\; My_t \qquad t \in \{1,\ldots,n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

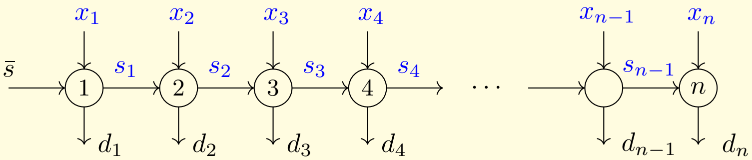
- The big Ms can be different for different t:

$$x_t \;\leq\; M^t y_t \qquad t \in T \;\longleftarrow\; \{1,\ldots,n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Finding the smallest Big M



- How big should M be?

$$x_t \ \leq \ My_t \qquad t \in \{1, \ldots, n\}$$

(M has to be an upper bound on $x_t \implies M = \sum_{s=1}^{n} d_s$.)

- The big Ms can be different for different t:

$$x_t \ \leq \ M^t y_t \qquad t \in T \ \longleftarrow \ \{1, \ldots, n\}$$

where

$$M^t = \sum_{i=t}^{n} d_i \qquad \leftarrow \text{use this}$$

Big Ms

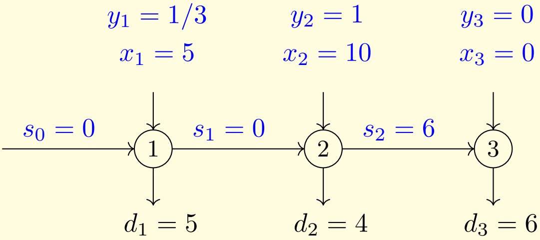
- Big M (meaning a very large number) is used in constraints of the form
$$x \le My$$
where $y \in \{0, 1\}$
- Variable y is forced to be 1, if $x > 0$.
- This helps model fixed costs and similar relations.
- If you use big Ms, then the number M should be at least as large as the largest value x can take.
- If you use big Ms, try to use the smallest possible number to obtain a better formulation.
- If you can formulate your problem without big Ms, do not use big Ms.

Another way to formulate the lot sizing problem

- We use variables x_t for quantity produced in t with upper bounds

$$x_t \;\leq\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{s=t}^{n} d_s$$

- Now consider a small instance with 3 time periods where
  - Fixed costs are [100, 20, 50] and all other costs are zero.
  - Demand is [5, 4, 6]
  - The optimal LP solution for this instance is



What can we do to fix this?

Another way to formulate the lot sizing problem

- We use variables x_t for quantity produced in t with upper bounds

$$x_t \;\leq\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{s=t}^{n} d_s$$

- We now disaggregate these x_t variables and use variables using
  - $q_{t,i} \geq 0$ to denote production in period t for demand in period $i \geq t$

Another way to formulate the lot sizing problem

- We use variables x_t for quantity produced in t with upper bounds

$$x_t \;\leq\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{s=t}^{n} d_s$$

- We now disaggregate these x_t variables and use variables using
  - $q_{t,i} \geq 0$ to denote production in period t for demand in period $i \geq t$

New formulation:

$$\min \;\; \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} h_t s_t$$

$$\begin{array}{lll}
\text{s.t.} & s_t = s_{t-1} + x_t - d_t & t \in T \\
& x_t = \sum_{i=t}^{n} q_{t,i} & t \in T \\
& q_{t,i} \leq d_i y_t & t \in T,\; i \geq t \\
& \sum_{t=1}^{i} q_{t,i} = d_i & i \in T \\
& s_0 = \bar{s}, s_t,\; x_t,\; q_{t,i} \geq 0, \;\; y_t \in \{0,1\},\; x_t \in \mathbb{Z} & t \in T
\end{array}$$

Extended formulations

- What we did is called building an extended formulation

  (i.e. use additional variables to formulate the same problem)

- The first formulation had

$$x_t \;\leq\; M^t y_t \qquad \text{where} \qquad M^t = \sum_{i=t}^{n} d_i \qquad i \in T$$

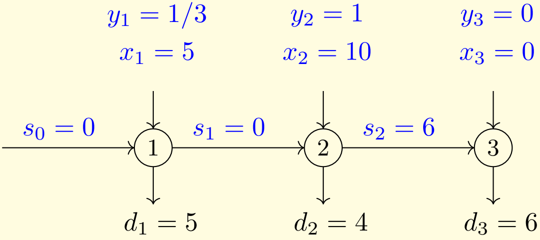
- The new formulation has additional $q_{t,i}$ variables

$$x_t = \sum_{i=t}^{n} q_{t,i}, \quad q_{t,i} \leq d_i y_t, \quad \sum_{t=1}^{i} q_{t,i} = d_i \quad i \in T$$

- Notice that if a solution (x, y, s, q) is feasible for new formulation LP, (x, y, s) is feasible for original formulation LP
- But some (x, y, s) is feasible for original LP are not feasible for new LP
- Now LPs might be harder to solve but you will need fewer B&B nodes.
- Computationally, the new formulation is faster to solve te integer to optimality.

Example

- Remember the optimal LP solution to the original formulation:



- There are no possible values for $q_{t,i}$ variables that will make this solution feasible for the second formulation because

$$x_t = \sum_{i=t}^{n} q_{t,i}, \quad q_{t,i} \le d_i y_t, \quad \sum_{i=1}^{t} q_{t,i} = d_i \quad t \in T$$

For period $t = 1$, this means

$$q_{1,1} \le \frac{1}{3}(5) \quad \text{and} \quad q_{1,1} = 1$$