Mathematical representation of the drought decision model

Trisha Shrum

January 13, 2017

1 Indices

Time is indexed by t, units in years. Variables with this index are allowed to vary by year.

Insurance decision is indexed by i, where $i = 1$ corresponds to the purchase of insurance and $i = 0$

2 Data

- Grid base data (what is this?): `grid_base.RData`, called in `load.R` script
- Insurance base data (what is this?): `insurance_base.RData`, called in `load.R` script
- NOAA Index (be more specific): `noaaIndex.RData`, called in `load.R` script
- Coops (what is this?): `coops.RData`, called in `load.R` script
- MLRA zone data (be more specific): `mlra=readOGR("data","mlra_v42")` called in `load.R` script. *I don't know what this function means*

3 Profit Model

$$\pi_t = R_t - C_t \tag{1}$$

3.1 Revenues

Base sales:

- Function: `CalculateExpSales`
- Inputs: `herd, calf.sell, wn.wt, p.wn`

$$R_{b,t}(h, \phi, w_t, p_t) = q_t(h, \phi, w_t)p_t \tag{2}$$

$$q_t = \phi h * w_t \tag{3}$$

Parameters:

- q_t: Quantity of calves sold (pounds)
- ϕ: Average percentage of calves sold, `calf.sell`
- h: herd size (number of cows, does not include calves), `herd`
- w_t: calf weight at weaning in year t (pounds), `wn.wt`
- p_t: price of calves at weaning in year t ($/pound), `p.wn`

3.2 Costs

Total Cost Function:

$$C_t = baseC_t + iprem + adapt \tag{4}$$

3.2.1 Base costs

- Function: `CalculateBaseOpCosts`
- Inputs: `herd, cow.cost`

$$baseC_t = \gamma h \tag{5}$$

Parameters:

- h: herd size (number of cows, does not include calves), `herd`
- γ: base operating costs ($/cow), `cow.cost`

4 Drought Adaptation

Drought Adaptation Options:

1. Buy Feed
2. Rent Pasture
3. Sell Pairs and Replace

4. Do Nothing

How much adaptation is needed? Depends on the length of adaptation action (λ = `days.act`) and the intensity of the drought (α = `forage.potential`). Different drought adaptation actions are scaled slightly differently to account for fixed costs (e.g., trucking) and variables costs (e.g., days of pasture rental).

- Function: `CalculateDaysAction`
  - Description: Calculate the number of days rancher pays for a drought adaptation action. NOTE: This function assumes that the actions take place only in one year.
  - Inputs: `act.st.yr` (year the action starts), `act.st.m` (month the action starts), `act.end.yr` (year the action ends), `act.end.m` (month the action ends)
  - Outputs: Number of days drought adaptation action takes place (days) (`days.act`)
  - Limitation: Not equipped to handle drought adaptation in multiple years. Currently only works for the first year (bug identified).

$$\lambda = \texttt{days.act} = 30 * (\texttt{act.end.m} - \texttt{act.end.m}) \tag{6}$$

- Function: `CalculateAdaptationIntensity`
  - Description: Takes forage potential and an adaptation intensity factor to provide a scalar of drought action. If forage potential is above 1 (no drought), then this variable goes to 0 (no adaptation). The variable has a maximum of 1, which assumes that drought actions are parameterized at full forage replacement for the full herd.
  - Inputs: `adpt.intensity.factor` $= \psi$ (parameter that scales adaptation actions to reflect actual adaptation behavior. Currently defaults to 1 which assumes a one-to-one ratio of drops in forage percentage to need for forage replacement.), `forage.potential` $= \alpha$ (the percentage of average forage produced in a year based on rainfall. See forage potential functions.)
  - Output: `drght.act.adj` (scales action to account for forage potential's deviation from the norm.)

$$\beta = \texttt{drght.act.adj} = \begin{cases} \min\{1, (1-\alpha) * \psi)\} & \text{if } \alpha \leq 0 \\ 0 & \text{else.} \end{cases} \tag{7}$$

4.1 Drought Adaptation Option: Buy Feed

4.1.1 Costs

In addition to base costs, $C_{b,t}$, with the buy feed adaptation option, we add the cost of buying feed according to the following function:

- Function: `CalculateFeedCost`
  - Description: Calculating the costs of purchasing additional feed
  - Inputs: `hay.ration` (hay ration assuming no grazing (pounds/head/day) *Source needed*), `p.hay` (price of hay ($/ton). user input), `oth.ration` (ration of non-hay feed (pounds/head/day) *Source needed*, `p.oth` (price of other feed ($/ton). User input. Does not come into play since the model assumes only feeding hay), β = `intens.adj` (drought intensity adjustment, eq. 7), λ = `days.act` (days adaptation action (days), eq. 6), `herd` = h (size of herd (head of cows, does not include calves))
  - Outputs: `cost.feed` (additional costs to feed the herd over the remainder of the season ($/year))

$$C_{f,t} = \beta\lambda h \left(\frac{ration_{hay}}{2000} * p_{hay} + \frac{ration_{oth}}{2000} * p_{oth} \right) \tag{8}$$

4.1.2 Revenues

Revenues are unchanged from the base level. (eq. 2)

$$R_{f,t} = R_{b,t} \tag{9}$$

4.2 Drought Adaptation: Rent Pasture

4.2.1 Costs

In addition to base costs, we add the cost of renting pasture according to the following function:

- Function: `CalculatePastureRentCost`
  - Description: Calculates the costs of renting pasture and trucking pairs
  - Inputs: m = `n.miles` (distance to rented pasture (miles)), p_{truck} = `truck.cost` (trucking cost per loaded mile ($/mile/truck)), p_{rent} = `past.rent` (price of renting pasture per animal unit month, where an animal unit is a cow/calf pair ($/pair/month)), β = `intens.adj` (portion of herd moving to rented pasture), λ

= `days.act` (days on rented pasture(days), C_{fixed} = `oth.cost` (all other non-rental, non-trucking costs ($)), w_{max} = `max.wt` (maximum weight per truck (pounds)), w_cow = `cow.wt` (average cow weight (pounds)), w_{calf} = `calf.wt` (average 'current' weight of calves before trucking to rented pasture (pounds)), h =`herd` (size of herd (head of cows, does not include calves))

- Output: `cost.rentpast` (total costs of using renting pasture including transport costs on top of normal operating costs ($/year))
- Assumptions: Only cows are trucked back home. Fixed costs cover transaction costs (time, etc.) of arranging pasture rental.

Number of trucks needed to transport portion of herd (pairs) to rented pasture:

$$n_{to} = \lceil \beta h * \lceil w_{max}/(w_{cow} + w_{calf}) \rceil \rceil \tag{10}$$

Number of trucks needed to transport portion of herd (cows only) back to home pasture:

$$n_{from} = \lceil \beta h * \lceil w_{max}/w_{cow} \rceil \rceil \tag{11}$$

Cost of hiring trucks:

$$C_{trucks} = m * p_{truck}(n_{to} + n_{from}) \tag{12}$$

Cost of renting pasture:

$$C_{rent} = \beta \lambda h \frac{p_{rent}}{30} \tag{13}$$

Total cost of ranching operation with drought adaptation through rental pasture:

$$C_{r,t} = C_{trucks} + C_{rent} + C_{fixed} + C_{base} \tag{14}$$

4.2.2 Revenues

Due to losses during the stress of trucking cows and calves, the revenues are lower than normal.

- Function: `CalculateRentPastRevenue`
  - Description: Calculates calf sale revenues after trucking pairs to rented pastures
  - Inputs: `calf.loss` (additional calf deaths due to transport stress (head of calves)), `calf.wt.adj` (adjustment for calf weaning weights (%)), `calf.sell` (average percentage of calves sold (%)), `wn.wt` (average weight at weaning (pounds)), `p.wn` (expected sale price of calves ($/pound)), `herd` (size of herd (head of cows, does not include calves)), β = `intens.adj` (portion of herd moving to rented pasture)

- Outputs: `rev.rentpast` (change in revenue due to mortality and weight loss from trucking to rented pasture)

Number of calves sold after accounting for calf mortality in transport calf.sales.rent ¡- herd * intens.adj * calf.sell - calf.loss calf.sales.home ¡- herd * (1 - intens.adj) * calf.sell

Selling weight after accounting for weight loss due to transport stress sell.wt.rent ¡- expected.wn.wt * (1 + calf.wt.adj)

Expected calf sale revenues rev.rentpast ¡- p.wn * (calf.sales.rent * sell.wt.rent + calf.sales.home * expected.wn.wt)

5 Insurance Model

- Function: `insMat`
  - Description: Generates a matrix representing insurance premium payments and indemnities for a specified grid cell over a five-year interval.
  - Inputs: `tgrd` (target grid cell), `yyr` (starting year), `clv` (coverage level), `acres` (insured acres), `pfactor` (land productivity factor), `insPurchase` (a matrix representing insurance allocation to two-month intervals, with rows written in the format [mm,amt])
  - Outputs: a 5 x n? matrix with insurance premium payments (column ?) and indemnities (column ?) for a specified grid cell over a five-year interval.
- Function: `droughtCalculator`
  - Inputs: `yy` (year of interest), `clv` (RMA coverage level. Accepted values are 0.7, 0.75, 0.8, 0.85, 0.9), `acres` (insured acres), `pfactor` (productivity factor of grazing land), `insPurchase` (a m? x n? matrix of intervals from 1-11 for which insurance is purchased. For example, purchases for the April-May and May-June intervals at 50% protection each would be entered as 'rbind(c(3,0.5),c(5,0.5))' Consecutive intervals are not allowed.)
  - Outputs (list): `prem_noSbdy` (total premium with subsidy), `prem_wSbdy` (total premium without subsidy), `prodPrem` (premium paid by producer), `indemrate` (indemnity rate (stack, by month)), `indemnity` (indemnity (stack, by month)), `indemtot` (total indemnity)
  - Requirements: Insurance allocation for consecutive intervals is not permitted. Insurance must be allocated for at least two intervals. Insurance allocation intervals must range from 1-11. Insurance allocation may not exceed 60% per interval. Insurance allocation must sum to 100%.

Coverage Subsidies (`covsub`):

Coverage Level	Subsidy Rate
70%	59%
75%	59%
80%	55%
85%	55%
90%	51%

$$sbdy = \begin{cases} 0.59, & \text{if } \texttt{clv} \leq 0.75 \\ 0.55, & \text{if } 0.75 < \texttt{clv} < 0.90 \\ 0.51, & \text{if } \texttt{clv} \geq 0.90 \end{cases} \tag{15}$$

Insurance Purchase (`insPurchase`, `insp`):

Default:Excel model defaults $\texttt{ins} = \begin{bmatrix} 3 & 0.5 \\ 5 & 0.5 \end{bmatrix}$

Option "autoSelect.insurance":

- Function: `insAlloc`
  - Description: Automates range insurance allocation to two-month RMA intervals using a grid cell/COOP site's forage potential weights. Returns a matrix formatted as the 'insPurchase' input for function 'insMat'. Allocation for chosen two-month intervals is roughly proportional to the relative value of each interval's forage potential weight. Adjustments to allocation percentages are automatically made if a selection is invalid for one or more intervals, either too high (>60%) or too low (10%). User-specified min/max allocation percentages falling within this range may also be substituted by setting the 'max.alloc' and 'min.alloc' arguments.
  - Inputs: `fpwt` (A vector of monthly forage potential weights for the target site. Monthly intervals are averaged to two-month intervals to match RMA insurance selections.), `niv` (number of two-month intervals to insure), `by.rank` (if TRUE (default), ranks forage potential weights by interval in descending order and selects the 'niv' most highly ranked non-consecutive intervals to insure. If FALSE, selects the combination of 'niv' non-consecutive two-month intervals with the highest average forage potential weights.), `max.alloc` (maximum interval allocation, 0.6), `min.alloc` (minimum interval allocation, 0.1).
  - Outputs:
- Helper Function: `dcInfo` (extracts drought calculator information from a grid cell), inputs: `dc` (drought calculator output), `tgrd` (target grid cell id)

6 Limitations

Currently not equipped to handle multi-year droughts. This can be changed.