Variable selection with BART

Rodney Sparapani
Associate Professor of Biostatistics
Medical College of Wisconsin

2024 ISBA World Meeting in Venice

The DART prior: BART with sparse variable selection

Linero 2018 *JASA*

- For variable selection with a Big P, specify a Dirichlet prior as $[s_1, ..., s_P] | \theta \overset{\text{prior}}{\sim} \text{D}(\theta/P, ..., \theta/P)$ that we call DART
- In the BART package, set the argument `sparse=TRUE` while the default is `sparse=FALSE` for uniform $s_j = P^{-1}$
- The prior parameter θ can be fixed or random: supplying a positive number will specify θ fixed at that value while the default, `theta=0`, specifies random
- The random θ prior is induced by $\theta/(\theta + \rho) \overset{\text{prior}}{\sim} \text{Beta}(a,\ b)$
- a defaults to 0.5 that can be over-ridden by the argument `a`
- b defaults to 1.0 that can be over-ridden by the argument `b`
- ρ can be specified by the argument `rho` that defaults to zero representing the value P; provide a value to over-ride

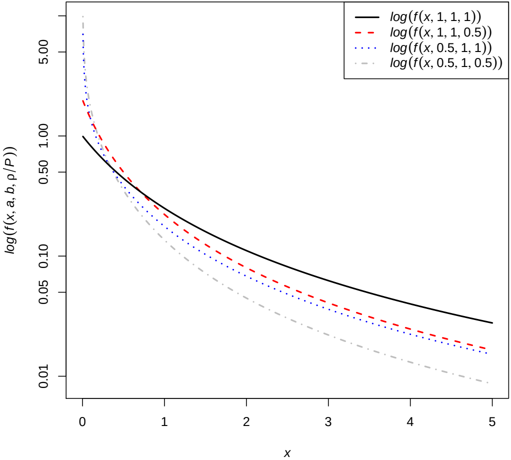
The DART prior

- The inducement of sparsity is controlled by the distribution of the arguments to the Dirichlet prior: θ/P
- Setting $a = 0.5$ is more sparse while $a = 1$ is less so
- If additional sparsity is desired, set ρ to a value smaller than P
- It can be shown that $\theta/P \sim F(a, b, \rho/P)$ where $F(.)$ is the Beta Prime distribution scaled by ρ/P
- The lesser sparse setting is $(a, b, \rho/P) = (1, 1, 1)$
- Sparsity is increased by reducing a : $(0.5, 1, 1)$ the default
  or by reducing ρ, e.g., $(1, 1, 0.5)$
  and even moreso by reducing both: $(0.5, 0.5, 1)$

The DART prior

The distribution of $\boldsymbol{\theta/P}$ and the sparse Dirichlet prior

Sparapani, Spanbauer and McCulloch 2021 *JSS*



Posterior computation for DART

- Posterior computation related to the Dirichlet sparse prior
- If a Dirichlet prior is placed on the variable splitting probabilities, $\boldsymbol{s}$, then its posterior samples are drawn via Gibbs sampling with conjugate Dirichlet draws
- The Dirichlet parameter is updated by adding the total variable branch count over the ensemble, $\boldsymbol{m_j}$, to the prior setting, $\frac{\boldsymbol{\theta}}{\boldsymbol{P}}$, i.e., $\left[\frac{\boldsymbol{\theta}}{\boldsymbol{P}} + \boldsymbol{m_1}, \ldots, \frac{\boldsymbol{\theta}}{\boldsymbol{P}} + \boldsymbol{m_P}\right]$ (Multinomial conjugacy)
- In this way, the Dirichlet prior induces a "rich get richer" variable selection strategy
- The sparsity parameter, $\boldsymbol{\theta}$, is drawn on a grid of values
- This draw only depends on $[\boldsymbol{s_1}, \ldots, \boldsymbol{s_P}]$
- BART R package: each variable's branch count is returned in the fit object: `varcount` and `varcount.mean`
- And the probabilities are returned too: `varprob` and `varprob.mean`

DART with grouped variables

Chipman, George et al. 2021
Computational approaches to Bayesian Additive Regression Trees
within the book *Computational Statistics in Data Science*

- We have P variables, but Q of them encode a grouped variable such as dummy indicators for a categorical variable (these are the first Q variables without loss of generality): $x_1, ..., x_Q$
- N.B. This applies to multiple grouped variables; however, for brevity, a single grouped variable will suffice
- The variable selection probabilities are $s = [s_1, ..., s_P]$
- There are two other probability collections of interest
- The collapsed probabilities, $p = \left[s_1 + \cdots + s_Q, s_{Q+1}, ..., s_P\right]$
- And the re-scaled probabilities $q = \left[\tilde{s}_1, ..., \tilde{s}_Q\right]$ where $\tilde{s}_j \propto s_j$ such that $\sum_{j=1}^{Q} \tilde{s}_j = 1$

DART with grouped variables

- Blindly using Dirichlet variable selection probabilities, then we arrive at the following
- $s|\theta \overset{\text{prior}}{\sim} \mathrm{D}_P\left(\theta/P, ..., \theta/P\right)$ where the subscript P is the order of the Dirichlet
- $p|\theta \overset{\text{prior}}{\sim} \mathrm{D}_{\widetilde{P}}\left(Q\theta/P, \theta/P, ..., \theta/P\right)$ where $\widetilde{P} = P - Q + 1$
- $q|\theta \overset{\text{prior}}{\sim} \mathrm{D}_Q\left(\theta/P, ..., \theta/P\right)$
- The problem: the distribution of p_1, the first element of p, puts more prior weight on the grouped variable than the others

DART with grouped variables

- The solution to the problem is trivial: re-scale $\boldsymbol{q}$ by $\boldsymbol{Q}^{-1}$ while naturally re-defining $\boldsymbol{p}$ and $\boldsymbol{s}$ as follows.

$$\begin{aligned}
\boldsymbol{p}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_{\widetilde{P}}\left(\theta/\widetilde{P}, ..., \theta/\widetilde{P}\right)\\
\boldsymbol{q}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_{Q}\left(Q^{-1}\theta/\widetilde{P}, ..., Q^{-1}\theta/\widetilde{P}\right)\\
\boldsymbol{s}|\theta &\overset{\text{prior}}{\sim} \mathrm{D}_{P}\left(Q^{-1}\theta/\widetilde{P}, ..., Q^{-1}\theta/\widetilde{P}, \theta/\widetilde{P}, ..., \theta/\widetilde{P}\right)\\
&\overset{\text{prior}}{\sim} \mathrm{D}_{P}\left((\boldsymbol{q}|\theta), (\boldsymbol{p}|\theta)\right)
\end{aligned}$$

- The BART3 R package's `gbart` function takes this approach automatically when you supply a data frame with the covariates where the categorical variables are factors (rather than supplying a matrix for the covariates)

Thompson Sampling Variable Selection (TSVS)

Liu and Rockova, JASA 2023

- A stochastic optimization approach to variable subset selection based on reinforcement learning with Thompson Sampling by an extension of BART with DART
- This is a multi-armed bandit (MAB) problem where each arm chosen corresponds to a variable selected
- For $t = 1, \ldots, T$ iteratively search for the optimal subset
- $y_i = f_t(x_i) + \epsilon_i$ where $f_t \overset{\text{prior}}{\sim}$ BART with DART
- Choose optimal $S_O(t) \subset \{1, ..., P\}$ important to the fit $f_t(x_i)$
- $p_j(t)$: variable inclusion probabilities
  $p_j(t) \overset{\text{ind}}{\sim} \text{Beta}\left(u_j(t),\ v_j(t)\right)$ where $j = 1, ..., P$
- $\gamma_j(t)$: a Bernoulli reward if x_j is chosen at step t
  $\gamma_j(t) \overset{\text{ind}}{\sim} \text{B}\left(p_j(t)\right)$
- $\pi_j(t)$: variable importance probability is the expected reward

$$\begin{aligned}\pi_j(t) &= \text{E}\left[\gamma_j(t)\right] = \text{E}\left[\text{E}\left[\gamma_j(t)|p_j(t)\right]\right] = \text{E}\left[p_j(t)\right] \\ &= u_j(t)/(u_j(t) + v_j(t))\end{aligned}$$

Multi-armed Bandits (MAB)

- MAB: Decide which of P arms to play at step t, given the outcome of the previous $t-1$ steps where $t = 1, ..., T$
- Goal: maximize sum of expected rewards and minimize regret
- Multi-play Scenario: At each step t, select a subset S_t of arms and receive binary rewards of all selected arms
- Reward: $\gamma_j(t) \overset{\text{ind}}{\sim} B(p_j(t))$
  N.B. this is Liu's notation: typically, it would be γ_{jt}, p_{jt}
- Maximize the sum of expected rewards over the drawn arms
- Optimal action: select arms $S_O(t) = \{j : \gamma_j(t) = 1\}$
- Regret, $\mathcal{R}(T)$: expected cumulative reward difference between the optimal drawing policy and the selected draws

$$E[\mathcal{R}(T)] = E\left\{ \sum_{t=1}^{T} \left(\sum_{j \in S_O} p_j(t) - \sum_{j \in S_t} p_j(t) \right) \right\}$$

Multi-armed Bandits (MAB)

- Global Reward, $R_C(S)$: a computational oracle regret minimizer when an oracle furnishes probabilities $p_j(t)$

$$R_C(S_t) = \sum_{i \in S_t} \log(C + \gamma_j(t))$$

$$r_p^C(S_t) = E[R_C(S_t)] = \sum_{i \in S_t} \left[p_j(t) \log\left(\frac{C+1}{C}\right) - \log\left(\frac{1}{C}\right)\right]$$

- Computational Oracle, S_O: $S_O = \arg\max_S r_p^C(S)$

$$S_O = \left\{ j : p_j(t) \geq \frac{\log(1/C)}{\log[1 + 1/C])} \right\}$$

Setting $C = (\sqrt{5} - 1)/2$ gives the median probability model

$$S_O = \{ j : p_j(t) \geq 0.5 \}$$

TSVS Algorithm for High Dimensions: Big P or Big N

Initialize parameters: you may need to experiment with those in red to get adequate performance especially M and T

- $\tilde{C} = \frac{\log(1/C)}{\log(1+C)/C}$ for some $0 < C < 1$ (typically, $\tilde{C} = 0.5$)
- L, length of DART chain burn-in discarded
- M, length of DART chain to keep
  N.B. typically, you have to run DART serially, i.e., NOT with parallel processing since the effective lengths of the chain in parallel would be $M/$`mc.cores` rather than M
- H, number of trees: typically, $H = 10$
- T, number of steps
- Prior settings: $u_j(0) > 0,\ v_j(0) > 0\ \text{ where } j = 1, \ldots, P$

TSVS Algorithm

```
For $t = 1, ..., T$
  a. For $j = 1, ..., P$, draw $p_j(t) \sim \text{Beta}\left(u_j(t-1),\ v_j(t-1)\right)$
  b. Set $S_t = \{j : p_j(t) \geq \tilde{C}\}$
  c. Fit DART model $f_t(x(t))$ with $x_j(t)$ where $j \in S_t$
  d. For $j = 1, ..., P$
       (i) If $j \notin S_t$, then set $\gamma_j(t) = 0$
           Else calculate reward $\gamma_j(t)$ from DART fit $f_t(.)$
      (ii) Set $u_j(t) = u_j(t-1) + \gamma_j(t)$
     (iii) Set $v_j(t) = v_j(t-1) + 1 - \gamma_j(t)$
      (iv) Calculate variable importance probability
           $\pi_j(t) = \frac{u_j(t)}{u_j(t)+v_j(t)}$
```

Trajectories of important covariates for $\pi_j(t)$ will exceed 0.5 by T

TSVS Algorithm: “Offline” for Big P

- N.B. there are no limits on P
- For example, TSVS can be used when $P >> N$
- Typically, $M = 1000$
- If $j \in S_t$, then set $\gamma_j(t) = 1$ when the corresponding `varcount` for the Mth draw is $m_{jM} > 0$
- Otherwise, set $\gamma_j(t) = 0$
- Liu and Rockova recommend $T = 500$, but our experience has been that $T = 20$ or 50 is often all that is needed

TSVS Algorithm: “Online” Big $N >> P$ with sharding

- Typically, $M = 10000$
- If $j \in S_t$, then set $\gamma_j(t) = 1$ when the corresponding `varcount.mean` for the M draws is $M^{-1} \sum_k m_{jk} = \overline{m}_j \geq 1$
- Otherwise, set $\gamma_j(t) = 0$
- Typically, $T = 100$
- The data set is partitioned into shards of size N/T and at each step you progress through the shards rather than the whole data set which is too big for DART to process efficiently
- However, due to the performance of TSVS, you may need to pass through the data set multiple times with bootstrapping
- So, you might consider B bootstrap passes through the data $T = A \times B$ with random shards of size N/B
- Typically, $B = 5$ and $A = 20$

Drug discovery classification: `NCItopo` data-frame

Chipman, George and McCulloch 2010 *Annals of Applied Stat*

- Anti-HIV drug screening process from the US National Cancer Institute (NCI) Developmental Therapeutics Program
- The discovery of new drugs depends on developing compounds with minimal toxic effects
- Statistical predictive toxicology relies on observed training data to learn the relationship between chemical structure features and toxicity response
- CEM cells are derived from T cells provided by a patient (with initals CEM) who suffered from acute lymphoblastic leukemia
- Compound structural features are provided by the Topological Information Indices (TII)
- TII are a common choice since they are easy to calculate; very sensitive to small changes in molecular structure; and do not depend on conformation of the molecule

Drug discovery classification: NCItopo data-frame

Chipman, George and McCulloch 2010 *Annals of Applied Stat*

- Potency: categorical response that measures whether a compound protects CEM cells from HIV-1 infection
- Compounds with Potency>0 are active
- There are $P = 260$ TII covariates
- 29,374 compounds were tested and 542 are active
- To load the NCItopo data-frame and get help

```
R> library(BART3)
R> data(NCItopo)
R>?NCItopo
```