An Introduction to Bayesian Nonparametrics

Introduction

- Last lecture discussed machine learning methods with large numbers of variables
- Machine learning methods can be used in non-linear and non-parametric problems where form of relationship between y and X is unknown
- $$y = f(X) + \varepsilon$$
- Don't know form of $f(.)$
- These two areas sound different but are similar: large numbers of parameters
- Lots of potential explanatory variables in a regression (want "machine" to sort through them all and find important ones)
- Lots of potential ways that $f(.)$ can be nonlinear (want "machine" to find specific ones)

Introduction

- Bayesian nonparametric methods have mostly been used in cross-sectional regression contexts, but recent interest in using nonparametric VARs for forecasting
- Why?
- Realization that existing choices for $f(.)$ such as linear, regime switching, structural break, TVP may not be rich enough
- Times of great turbulence (financial crisis, covid-19 pandemic) might be better to model nonparametrically
- They win many forecasting "horse races"
- Bayesian nonparametrics is a large and growing field, this lecture is only a brief introduction to two of the most popular methods
- BART: Bayesian additive regression trees
- Gaussian processes

BART: Bayesian Additive Regression Trees

- Idea: BART classifies observations into different groups each of which has same fitted/predicted value for y_t
- $$y_t = f(X_t) + \varepsilon$$
- BART figures out $f(.)$ using regression trees
- X might contain many variables (Big Data) or might contain few, but need machine learning method since many possible types of nonlinearity
- BART is Bayesian way of working with regression trees, non-Bayesian methods with names like "random forests" also very popular
- Many other Bayesian non-parametric or similar approaches (e.g. Dirichlet mixtures of Normals)

An Introduction to BART

- BART approximates each $f(X_t)$ as follows:

$$f(X_t) = \sum_{s=1}^{S} g_s(X_t | T_s, \mu_s),$$

- T_s are tree structures
- μ_s are tree-specific terminal nodes
- S denotes the total number of trees used.
- Dimension of μ_s is denoted by b_s which depends on the complexity of the tree
- Note that BART involves adding up different trees (this is the "A" in BART)
- But what is a regression tree?

Intuition of a Regression Tree

- Explain idea of BART for a single tree (suppress s subscripts)
- Conventional regression: For every value for X produces a fitted value for y
- BART does same thing, but in different way
- Splitting rule: Divides space of X into different intervals each of which has same fitted value for y (internal nodes, branches of tree)
- Fitted values are called terminal nodes (or leaves of the tree)

Intuition of a Regression Tree

- Terminal nodes depend on sets $\mathcal{A}_r$:

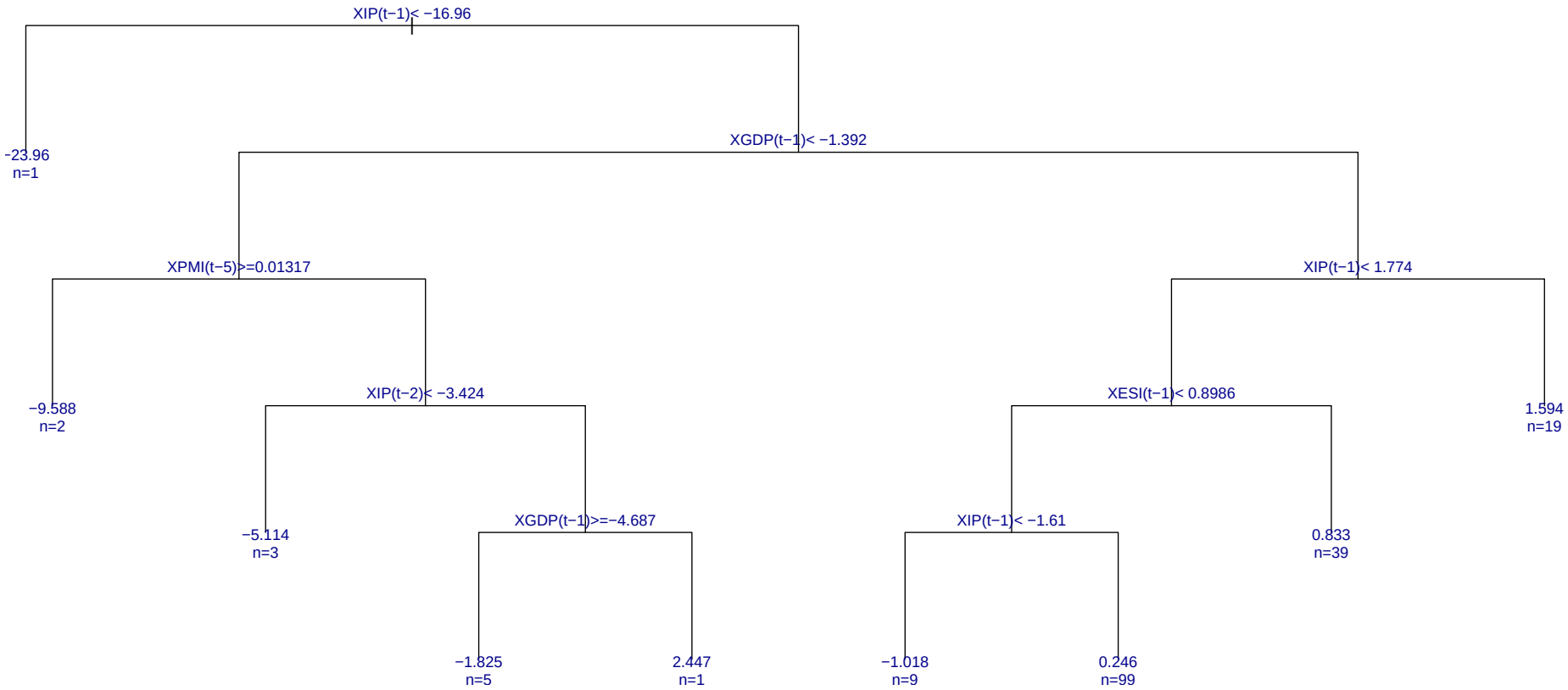
$$g(\boldsymbol{X}; T, \boldsymbol{\mu}) = \mu_r, \quad \text{if } \boldsymbol{X} \in \mathcal{A}_r, \quad r = 1, \ldots, b.$$

- Splitting rules take the form $\{\boldsymbol{X} \in \mathcal{A}_r\}$ or $\{\boldsymbol{X} \notin \mathcal{A}_r\}$
- Each splitting rule depends on whether an explanatory variable is above/below threshold
- Illustrate using a single tree using a specific data set, don't worry about data details (regression with 6 variables, XGDP is GDP growth, XIP industrial production growth, etc.)

Interpreting a Regression Tree

- Tree organized with condition (e.g. $XIP(t-1) < -16.96$) at top of every split
- Rule: If condition holds take the left branch, else take the right branch
- At the bottom are terminal nodes = leaves = fitted values for the dep. var.
- Example: rightmost leaf is 1.594
- Observations which have $XIP(t-1) >= -16.96$ and $XGP(t-1) >= 1.392$ and $XIP(t-1) >= 1.774$
- Fitted value for GDP growth for observations with last month's industrial production growth in interval $[-16.96, 1.774]$ and last month's GDP growth above -1.382 is 1.594.
- $n = 19$ means there are 19 observations that choose this node

Example of a Regression Tree



Properties of Regression Trees

- Everything in the tree is estimated by algorithm
- Values of terminal nodes
- Values of internal nodes (i.e. Choice of variables and thresholds in splitting conditions)
- The number of splits that occur (useful with correlated explanatory variables)
- Number of variables included is 6, but some never appear in internal nodes (loosely speaking, they are "insignificant")
- BART very flexible non-parametric (black box) algorithm
- Illustration is for one regression tree, BART adds up many gaining even more flexibility
- Empirically, often find adding many simple trees ("weak learners") works better than using one more complicated tree

Prior Shrinkage in BART

- Even a single regression tree can over-fit the data
- In theory, can fit perfectly, each observation gets own leaf set equal to actual value of y for that observation
- With regression trees need to avoid this through "regularization" which, for the Bayesian, means a prior
- I will provide informal description and motivation of the standard BART prior (see readings for details)
- Complete, more technical reading, is Chipman, George and McCulloch (2010, Annals of Applied Statistics), "BART: Bayesian Additive Regression Trees"
- Includes recommendations for default prior hyperparameter values which are commonly used

Prior Shrinkage in BART: The Number of Leaves on the Tree

- Trees with too many leaves (terminal nodes) can overfit
- Number of leaves depends on number of branches (internal nodes)
- let d = depth of tree (number of times a node is a splitting rule instead of a terminal node)
- Prior prob. that a node will split (not a terminal node) is:

$$\underline{a}(1+d)^{-\underline{b}}$$

- Decreasing in d implies deep trees unlikely (prior belief that tree is weak learner)
- Prior hyperparameters $\underline{a}$ and $\underline{b}$ can be chosen to reflect prior beliefs
- Chipman et al (2010) provide default recommendations $\underline{a} = 0.95$ and $\underline{b} = 2$ that are widely used

Prior Shrinkage in BART: Internal and Terminal Nodes

- Regression tree involves splitting rules (e.g. $XIP(t-1) < -16.96$) which involve both a variable ($XIP(t-1)$) and a value (-16.96)
- Prior over variables: Uniform (each variable equally likely to be in internal node)
- Prior over values: Uniform over the range of possible values variable can take (e.g. different percentiles of distribution of the variable)
- Chipman et al recommend scaling dependent variable to lie in interval $[-0.5, 0.5]$, hence expect terminal nodes (fitted values) to lie in interval
- Prior for terminal node (μ) is

$$N(0, \underline{\sigma_\mu})$$

- where $\underline{\sigma_\mu}$ is prior hyperparmeter
- Default recommendation $\frac{1}{4\sqrt{S}}$

Prior Shrinkage in BART: Summary

- Default BART prior of Chipman et al (2010) is automatic (the researcher does not have to choose prior hyperparameters)
- Default choices have worked well in wide variety of statistical applications
- What about S (number of trees)?
- Can treat as unknown parameter (but increases computational burden)?
- Chipman et al. "fast and expedient" advice is to set S large (e.g. $S = 200$) but experiment with a few different choices just to make sure results are robust to choice of S
- They say: "Our experience has been that as S is increased, starting with $S = 1$, the predictive performance of BART improves dramatically until at some point it levels off and then begins to very slowly degrade for large values of S."

Computation in BART

- I will provide intuitive description of the MCMC algorithm used to do Bayesian inference with BART
- But there are excellent, easy to use, BART packages in R (as easy as running a standard regression model, no coding required)
- All you need to do is download R + RStudio and click through one of the many online tutorials available
- R is as easy to use as Matlab and have similar structures so if you know one you are halfway to knowing the other
- You can treat both BART model and BART computation as black boxes where all is done automatically (the "machine" does it all)
- I would encourage you to look into these R packages if you wish to use BART in future research

Informal Description of Computation in BART

- Metropolis-Hastings (M-H) algorithm used with BART
- Shares some similarities to MC^3
- Remember M-H algorithms involve a candidate generating density and an acceptance probability
- Candidate draws are taken and then accepted with a certain probability (else they are rejected)
- Acceptance probability for BART is easy
- Key component is $p(y|X, T_1, \mu_1, .., T_S, \mu_S)$
- But this is just a Normal density (given a draw of the tree structures and terminal nodes, you have a draw of $f(X)$ and thus just use form of nonlinear regression model)
- I will say no more about acceptance probabilities

Generating Candidate Trees

- Remember the random walk M-H algorithm: candidate values of parameters generated based on taking one step away from current parameter draw
- Rememer MC^3: candidate models drawn by adding/deleting one explanatory variable from the current model draw
- M-H for BART has similar intuition: Take current draw of the tree and generate candidate by either
- Grow (split current leaf into two leaves)
- Prune (collapse adjacent leaves into one)
- Change (change splitting rule in an interior node)
- Swap (swap the splitting rules between two interior nodes)
- Swapping step sometimes left out (as in code provided in computer tutorial)

Generating Candidate Nodes

- Given tree structure, need to generate nodes
- Won't provide details or derivation, but conditional posteriors turn out to be Normal so easy to do
- Conditional posterior for error variance also standard

Summary: MCMC for BART

- I have sketched basic ideas of a basic MCMC algorithm for BART
- Details in Chipman et al paper
- Many variants on this algorithm proposed in the literature to speed it up or get MCMC to converge faster
- Many good BART packages written by statisticians allow you to use BART without much programming skills
- Here is a link which discusses one of them (bartMachine):
- https://towardsdatascience.com/a-primer-to-bayesian-additive-regression-tree-with-r-b9d0dbf704d
- Kapelner and Bleich "bartMachine: Machine Learning with Bayesian Additive Regression Trees" has full description

Brief Introduction to Gaussian Processes

- Gaussian processes (GPs) are alternative nonparametric approach
- BART classifies, GP smooths (similar to frequentist kernel methods)
- Begin with nonlinear regression model:

$$y_t = f(\boldsymbol{x}_t) + \varepsilon_t.$$

- f is unknown and ε_t is Normal
- Idea of GP: Let $\boldsymbol{f} = (f(\boldsymbol{x}_1), .., f(\boldsymbol{x}_T))'$ – all the points on the regression line — be a $T-$ *vector* of unknown parameters
- Another Big Data problem: equivalent to regression with a dummy variable for each observation
- How to estimate these T unknown parameters? Use Bayesian prior to avoid over-fitting

The GP Prior

- The prior is:

$$\boldsymbol{f} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}),$$

- Key thing in GP prior is prior covariance $\boldsymbol{K}$
- $T \times T$ "kernel" matrix with typical element $k(\boldsymbol{x}_t, \boldsymbol{x}_\tau)$
- Many choices for the kernel are possible

The Gaussian kernel

- One popular one is Gaussian kernel:

$$k(\boldsymbol{x}_t, \boldsymbol{x}_\tau) = \xi \times \exp\left(-\frac{\phi}{2}||\boldsymbol{x}_t - \boldsymbol{x}_\tau||^2\right),$$

- ξ, ϕ denoting the hyperparameters of the kernel (we estimate these)
- Idea: similar $\boldsymbol{x}_t$ and $\boldsymbol{x}_\tau$ imply similar $f(\boldsymbol{x}_t)$ and $f(\boldsymbol{x}_\tau)$
- Kernel measures distance between $\boldsymbol{x}_t$ and $\boldsymbol{x}_\tau$
- Degree of smoothness of the function depends on ϕ.
- Note that if $\boldsymbol{x}_t = \boldsymbol{x}_\tau$, then $\text{Var}(f(\boldsymbol{x}_t)) = \xi$. This allows us to see that ξ controls the variance of the function f

Posterior for GP Regressions

- Posterior results easy: Normal prior and Normal likelihood function
- Equivalent to linear regression model with explanatory variables being one dummy variable for each observation
- Standard textbook results for Bayesian linear regression model hold (conditional on ξ, ϕ)
- MCMC algorithm draws from $\boldsymbol{f}$ given ξ, ϕ and ξ, ϕ given $\boldsymbol{f}$
- Latter involves only 2 parameters so several (simple) methods possible

Summary

- Bayesian nonparametric methods growing in popularity
- Largely in research fields other than economics, but increasingly in economics
- Have performed well in forecasting horse races
- This lecture covers two of the model popular methods
- BART: classification algorithm
- GP: smoothing algorithm