Causal Inference I

Mixtape Session

Roadmap

Background Material for Causal Forests

- A Brief History of Trees
- Fundamental Machine Learning Concepts
- Parameters
- Data
- Unconfoundedness and common support assumptions

Estimation

- Causal forest algorithm
- Understanding Honesty in Causal Forests
- Final comments

Key Challenges in Causal Inference

- Confounding: Hidden biases that can skew results.
- Selection bias: Non-random assignment to treatment.
- Measurement error: Inaccuracies in data collection.

If we can address the problem using covariates, then we may be in a situation to use causal forests (but not all problems fit this scenario)

Causal Forests: Bridging Machine Learning & Causal Inference

Causal forests are . . .

- A powerful tool for estimating heterogeneous treatment effects using tree-based methods.
- Combining the strengths of random forests with the principles of causal inference to uncover nuanced relationships.
- Addressing challenges in observational data: leveraging the unconfoundedness assumption and advanced modeling techniques.

Athey and Wager: Pioneering Work on Causal Forests

- Susan Athey & Stefan Wager's foundational work: "Estimation and Inference of Heterogeneous Treatment Effects using Random Forests" in 2019 JASA
- Addressed challenges in traditional methods: Bias-variance trade-off and overfitting.
- Introduced causal trees as building blocks for causal forests, leveraging bootstrapping and honest splitting.
- We'll take a leisurely walk through it so no one feels left behind

Origins of Decision Trees: The 1960s

- The foundational concepts of decision trees emerged in the 1960s.
- Initially explored in cognitive psychology to model human decision processes.
- Used in medical decision-making to aid doctors in diagnosing diseases based on a hierarchical structure of symptoms and outcomes.
- These early trees were simplistic and manually constructed, but they paved the way for algorithmic tree-building methods in the subsequent decades.
- The concept gained traction as it offered a visual and interpretable way to make decisions based on multiple criteria.

Early Beginnings

- Ross Quinlan's ID3 in the 1980s: pioneering tree algorithm.
- ID3 uses an information-theoretic approach: It selects the attribute that provides the best split (maximizing information gain) at each node, building the tree iteratively.
- Real-world example: Diagnosing medical conditions. ID3 could be applied to a dataset where symptoms are attributes and diseases are classes. The tree would guide medical professionals by asking about the most informative symptoms first, helping narrow down potential diagnoses.

Classification and Regression Trees (CART)

- Introduced by Breiman et al. in 1986.
- CART allows for the creation of binary trees for both classification (categorizing data into classes) and regression (predicting numerical values).
- Uses a ”greedy” approach: At each step, it selects the best split based on a specific criterion (like Gini impurity for classification or mean squared error for regression) without concern for future decisions.
- Real-world example: Predicting housing prices. Using a dataset with features like house size, location, and age, CART can be used in regression mode to predict the price of a house based on these attributes.

Ensemble Methods and Random Forests

- Leo Breiman introduced Random Forests in 2001.
- Random Forests are an ensemble method: They combine predictions from multiple decision trees to produce a more robust and accurate result.
- The method introduces randomness in two ways: By bootstrapping samples for training each tree and by selecting a random subset of features at each split.

Ensemble Methods and Random Forests Example

- Real-world example: Credit scoring.
- Banks use Random Forests to predict the likelihood of a loan applicant defaulting.
- The model takes into account various factors like income, employment history, and credit score, aggregating insights from multiple trees to assess risk.

Boosting and Gradient Boosted Trees

- Boosting introduced by Schapire in 1990; later refined by Freund; is an ensemble technique that focuses on reducing bias by giving more weight to misclassified instances in subsequent models.
- Gradient Boosted Trees introduced by Friedman in the late 1990s and early 2000s builds trees sequentially, where each tree tries to correct the errors made by the previous ones and uses gradient descent to minimize the loss function.
- Both methods aim to improve prediction accuracy by combining weak learners (models slightly better than random guessing) to form a strong learner (a model with high predictive accuracy).

Boosting and Gradient Boosted Trees Example

- Real-world example: Customer churn prediction.
- Telecom companies use Gradient Boosted Trees to predict which customers are likely to terminate their services.
- By analyzing data like call patterns, customer complaints, and billing information, the model identifies high-risk customers, helping companies take proactive retention measures.

Modern Use and Software Development

- Trees are staples in machine learning toolkits.
- Libraries: scikit-learn (Python), rpart and randomForest (R).
- Popular for interpretability and visualization.

Decision Trees

- Flowchart-like structure used for making decisions.
- Decisions made by asking a series of questions.
- Comprises nodes (questions), branches (answers), and leaves (decisions/outcomes).

Random Forests

- An ensemble of decision trees.
- Uses bootstrapped samples to build individual trees.
- Aggregates predictions: majority vote or averaging.

Leaves in Trees

- Leaves are the terminal nodes of trees.
- In causal forests, leaves estimate treatment effects.
- Allows capturing heterogeneous effects across data segments.

Regularization

- A technique to prevent overfitting.
- In trees: limit depth, minimum samples in a leaf, randomness in feature selection.
- Ensures the model generalizes well to new data.

Benefits of Ensemble Methods

- Strength in numbers: multiple trees reduce variance.
- Can capture complex, non-linear relationships.
- Improved accuracy and robustness.

Decision Trees in Causal Inference

- Which brings us to now – causal inference
- Transition to causal inference frameworks in the 2000s.
- Scholars like Susan Athey, Stefan Wager and Guido Imbens developed Causal Trees and Forests.

Origins of Causal Forests

- Birthed from the intersection of causal inference and machine learning.
- Preceded by methods like propensity score matching, regression discontinuity, and instrumental variables.
- These methods did not explore or identify the HTEs though making them primarily used to summarize treatment effects at the highest level (e.g., ATE, ATE, LATE)
- This led to a need for more flexible, non-parametric methods to estimate heterogeneous treatment effects, particularly when tech firms began more intense targeting (e.g., recommendation systems)

Overfitting in Propensity Score Estimation

- Estimating propensity scores often involves logistic regression with many covariates.
- Overfitting can arise with a large set of covariates relative to sample size or with high-degree interactions.
- Overfitted models yield scores too close to 0 or 1, complicating matching.

Nearest Neighbor Matching & Common Support

- Common support ensures overlap in propensity score distributions.
- Yet, nearest neighbor matching might pair units not very close in propensity scores.
- Especially problematic when untreated units greatly outnumber treated ones.

Quality of Matches

- Common support as measured by propensity scores doesn't guarantee high-quality matches.
- This is because common support should be a concept we are thinking of as holding, not in the propensity score, but in the actual stratification of the data using the dimensions of the covariates
- Units with similar propensity scores might differ on key covariates given the curse of dimensionality kicks in very quickly as we increase covariates with high dimensions.
- Matching without replacement can lead to lower-quality subsequent matches.

Sensitivity to Specification

- Recall that we are *estimating* the propensity score; we don't know the truth even if we know the covariates to use for estimation
- Match quality and causal estimates can be sensitive to propensity score model specification.
- Minor model changes can lead to different matched samples and different matched samples means we have variation in treatment effect estimation that is due to these matching irregularities
- Highlights the importance of robustness checks.

Challenges of High-Dimensional Covariate Spaces

- Modern datasets often have a vast number of covariates, making the ”curse of dimensionality” a prominent concern.
- Even with a few covariates, the curse can arise, but it’s especially pronounced in high-dimensional settings.
- While the assumption of unconfoundedness requires controlling for all confounders, in practice, ensuring this in high dimensions is challenging.
- The ”kitchen sink” approach, adding numerous covariates to control for confounding, can introduce its own problems.
- Causal forests offer a flexible way to address these challenges, capturing complex relationships without the need for overly restrictive linear specifications.

Heterogeneous Treatment Effects

- Which brings us to causal forests – they moved beyond the aggregate causal parameters into more nuanced ones based HTEs while still contending with the problems of high dimensional data
- In high-dimensional settings, treatment effects can vary across many dimensions.
- Causal forests enabled new insights into how treatment effects manifest in different segments of the data, but as it was data driven, it was less prone to arbitrary groupings by researchers

Adaptively Identifying Key Differences

- Imagine sorting people based on their features, like age or political views.
- Some groups show very different responses to the same treatment.
- Causal forests help pinpoint and focus on these groups, revealing where the treatment works best (or worst).

Regularization

- Random forests introduce randomness via bootstrapping and random subsets of predictors.
- This randomness acts as a form of regularization.
- Helps prevent overfitting in high-dimensional settings.

Key Assumptions for Causal Inference

- **Unconfoundedness**: Given observed covariates, treatment assignment is independent of potential outcomes.
- Meaning: After accounting for observed characteristics, there's no systematic difference between treatment and control groups affecting the outcome.
- **Common Support**: For every combination of covariates, there's a positive chance of being in either treatment or control.
- Meaning: We always have data to compare treated and untreated outcomes for individuals with similar characteristics.

Parameters, data requirements and assumptions

- **Parameters**: What are the research questions and parameters to have in mind for this method?
- **Data requirement**: What kinds of datasets should you be thinking of as particularly useful?
- **Assumptions**: What are the underlying causal assumptions?

Motivating the Importance of HTEs

- Heterogeneous Treatment Effects (HTEs) are the parameters and play a pivotal role in targeting interventions effectively.
- Applications:
  - → Medicine: Identifying individuals most benefited by specific treatments.
  - → Marketing: Retaining subscribers likely to respond positively to specific campaigns.
  - → Political Messaging: Targeting potential voters or donors effectively.
- Common thread: Aiming to cause a desired behavior or attitude change. Who should be the target?

Lift vs. Individual Treatment Effect

- **Lift**: An intuitive measure representing the difference between expected outcomes in treatment and control groups. Commonly used in practical applications to gauge the effect of an intervention.
- **Individual Treatment Effect (ITE)**: A more formal term in causal inference representing the effect of treatment on an individual.
- It is very common to see these terms used interchangeably because Lift is a industry term (particularly common at Amazon), but ITE is causal inference jargon within academia

Defining Lift and Its Importance

- Lift is the difference between the expected outcome in the treatment and the control (I call it the simple difference in mean outcomes in my book)
  - → If outcome is dichotomous: Lift = Probability (desired behavior in treatment) - Probability (desired behavior in control).
  - → If outcome is continuous: Lift = Mean outcome in treatment - Mean outcome in control.
- Example: If 55% in treatment voted and 50% in control did, lift = 5 percentage points.
- Example: For donations, if treatment average is $10.00 and control is $7.50, lift = $2.50.
- It's a calculation; it isn't causal until we make assumptions about treatment assignment mechanism (i.e., randomization, parallel trends)

What to do with these HTEs

- **Model Training**: Use historical or experimental data to train a model estimating HTEs based on individual characteristics.
- **Prediction**: Apply the trained model to new individuals (outside the original dataset) to predict their expected treatment effects.
- **Intervention Selection**: Identify individuals who are predicted to benefit the most from the intervention.
- **Maximize Effectiveness**: Strategically deliver the intervention to those individuals to maximize overall impact.
- **Continuous Learning**: As more data becomes available, refine and retrain the model to enhance prediction accuracy and intervention effectiveness.

Finding Heterogeneous Treatment Effects

While many strategies don’t prioritize causal effects, certain models are designed to optimize for them.

To comprehend these, a dive into theoretical foundations is essential.

Let’s do a quick review at a high level now

Objective of HTEs

Recall that we do not require in causal inference that treatment effects be constant across people; they can be *heterogenous* meaning different responses to the same intervention are possible (and likely)

Our aim with causal forests is to measure the individual causal effect, represented as:

$$\delta_i = Y_i^1 - Y_i^0$$

.

This is framed, in other words, within the Rubin causal model, which posits a strict definition of causality which is one of its strengths – clarify of parameter definitions

The Challenge in Causal Inference

This task is **impossible** though by the same Rubin causal model

We cannot simultaneously observe an individual's outcomes in both the treatment and control conditions due to a "switching equation" that moves from potential outcomes to realized outcomes

$$Y_i = D_i Y_i^1 + (1 - D_i) Y_i^0$$

This is what Holland (1986) means by "the fundamental problem of causal inference" – history creates counterfactuals making individual causal effects inacessible

Approaching the Challenge

So what can we do?

A common solution involves comparing outcomes between two large, similar groups—one exposed to an intervention or treatment, the other a control.

This helps determine the average treatment effect (ATE),

$$E[Y^1|D=1] - E[Y^0|D=0]$$

Under some situations this can be estimated with data (i.e., randomization) and others it cannot be

Estimating Individual Treatment Effects

Under assumptions we define later, we will assume there is some randomization in the data and we have found it

To pinpoint effects for specific individuals, algorithms identify 'similar others' in the dataset.

Their outcomes serve as proxies, helping estimate potential outcomes for individual targets.

Big idea: this is about estimating "strata specific" or "partitioned" ATEs which are called the conditional ATE or CATE (more later)

Understanding ”Similar Others”

- ”Similar others” refers to units (or individuals) in the dataset with comparable values on observed characteristics.
- Think of it as a concept akin to the nearest neighbor in methods like k-nearest neighbors (k-NN).
- Based on some distance metric (often Euclidean), we identify observations close to the target observation in the covariate space.
- In the context of HTEs, these units serve as proxies to help infer potential outcomes for a particular individual under a different treatment condition.

Why HTEs Over CATEs and Vice Versa?

- **HTEs (Heterogeneous Treatment Effects)**:
  - → *Tailored Interventions*: Identify specific subgroups that benefit the most or least from an intervention.
  - → *Resource Allocation*: Direct resources more efficiently by targeting those with the highest expected lift.
  - → *Uncover Mechanisms*: Understand the underlying factors that drive different responses to treatment.

- **CATEs (Conditional Average Treatment Effects)**:
  - → *Broad Overview*: Get a general sense of the treatment's effectiveness within predefined strata.
  - → *Easier Interpretation*: Less complex than HTEs, making them more accessible for broader audiences.
  - → *Stability*: Pre-defined strata can lead to more stable and consistent estimates, especially with smaller sample sizes.

Conditional Average Treatment Effect (CATE)

- Definition: $\tau(x) = \mathbb{E}[Y^1 - Y^0 | X = x]$
- Represents the difference in potential outcomes for treated vs. untreated units, conditioned on covariate values x.
- Allows us to capture heterogeneous treatment effects across different subgroups.
- But causal forests move beyond this CATE concept into a different kind of HTE concept

Understanding Heterogeneous Treatment Effects

- So the essence of identifying HTEs is about comparing ”like with like” but in the context of ”like in the treatment” with ”like in the control”
- HTE method stratifies or groups data based on combinations of covariates (e.g., race, gender, political affiliation) or what you may hear described as ”dimension”
- Within each stratum, the method calculates the difference in outcomes between the treatment and control groups – so not the average overall, but the average within some strata
- This difference provides an estimate of the average treatment effect (ATE) specific to that subgroup

Simple Breakdown of Stratified ATE

1. **Stratification**: Partition data based on combinations of covariates. Each represents a unique combination like raceXgenderXpolitical.
2. **Calculate Outcomes**: Within each stratum, compute the average outcome for treatment and control groups.
3. **Compute ATE**: Subtract control group average from treatment group average for each stratum.
4. **Generalization**: Uncover patterns, understand which covariates influence the treatment effect most.

From Stratified ATE to Causal Forests

- Causal forests are an evolution of the stratified ATE concept. Instead of pre-defined strata, they search for the best splits in data to identify HTEs.
- While stratified ATE focuses on average effects within fixed covariate combinations, causal forests dynamically determine where the treatment effect varies the most.
- The forest ”learns” from the data which covariate splits are most relevant in predicting heterogeneous effects.
- The result? A more nuanced understanding of where and how the treatment effect differs across subgroups.

Data requirements

- Before diving into the causal assumptions, let’s look at the data requirements
- What are the types of data you should have in your mind for this method?
- We’ll review

1. Sample Size

- Larger datasets are preferable for robust HTE estimates.
- While causal forests can work with smaller datasets, precision might be compromised with many covariates.
- More data provides better granularity and helps in identifying meaningful heterogeneous effects.

2. Number & Type of Covariates

- Can handle a large number of both discrete and continuous covariates.
- Continuous covariates are split based on observed distribution and outcome relationship.
- As dimensionality increases, more data is required for meaningful splits.

3. Handling Missing Data

- Handle missing data before input: consider imputation or other methods.
- Ensure missingness doesn’t introduce bias into treatment effect estimates.
- Some implementations can handle missing data but best to address beforehand.

4. Balance of Treatment & Control

- Aim for a good balance between treatment and control groups.
- Extreme imbalances can affect the precision and robustness of estimates.

5. Common Support

- Ensure sufficient representation in both treatment and control for all subgroups.
- Sparse cells (specific covariate combinations) can pose challenges for accurate estimates.

6. Variability & Granularity in Covariates

- Causal forests require variability in covariates for meaningful splits.
- Consider the granularity: age as continuous vs. binned can influence tree structures.

Assumption of Unconfoundedness

- Unconfoundedness: Treatment is independent of potential outcomes (i.e., as good as random) for units with identical covariate values or “strata”.

$$(Y^1, Y^0) \perp\!\!\!\perp D \mid X$$

- Causal forests are in the “branch” of causal inference that assumes unconfoundedness (like DML, propensity scores, regression and matching)
- But their strength is in their adaptability to large dimensions which makes them powerful tools under this assumption.

Common Support in High Dimensions

- After unconfoundedness, we need “common support” – non-empty cells for the entire stratification of the data based on the covariates in treatment and control
- Unconfoundedness says we are allowed to use covariates for causal inference, but common supports says it’s actually possible to do it because we have units in treatment and control for all dimensions of X
- In high-dimensional settings, ensuring common support becomes challenging though (slide after next for more)

Example: Breakdown of Common Support

- Consider two binary covariates, sex (male/female) and age (adult/child).
- Even if sex and age individually have overlap, the joint distribution (e.g., sex=0, age=1) might lack overlap.
- With more covariates, gaps in joint distribution become more probable: the ”curse of dimensionality.”

Table 1: Stratified sample with common support

Table: Counts and Titanic survival rates by strata and first class status.

	First class		All other classes		
Strata	Obs	Mean	Obs	Mean	Total
Male adult	175	0.326	1,492	0.188	1,667
Female adult	144	0.972	281	0.626	425
Male child	5	1	59	0.407	64
Female child	1	1	44	0.613	45
Total observations	325		1,876		2,201

Table 2: Stratified sample without common support

Table: Counts and Titanic survival rates by strata and first class status.

	First class		All other classes		
Strata	Obs	Mean	Obs	Mean	Total
Male adult	175	0.326	1,492	0.188	1,667
Female adult	144	0.972	281	0.626	425
Male child	5	1	59	0.407	64
Female child	0	n/a	44	0.613	44
Total observations	324		1,876		2,200

A Word of Caution

- As we delve into causal forests, be mindful of the term ”partition” and its usage.
- It aligns closely with ”stratification,” but remember the adaptive, data-driven nature of partitioning in this context.
- Terminology can sometimes be a barrier, but understanding the underlying concepts bridges the gap.

Strata vs. Partition

- Historically, in many statistical contexts, we use ”strata” and ”stratification” to refer to dividing data into subsets based on certain criteria.
- In the literature on causal forests and decision trees, the term ”partition” is more commonly used.
- This terminology traces back to the computer science and machine learning origins of decision trees, where data is ”partitioned” into subsets during the tree-building process.

Why the Difference?

- ”Stratification” often implies a deliberate, predefined division of data based on known, important criteria.
- ”Partitioning” in trees is more dynamic and adaptive, with divisions made based on data-driven decisions to optimize a specific objective.
- While they conceptually overlap, the terms have different historical and contextual connotations.

Traditional Stratified Analysis vs. Causal Forests

- **Stratified Analysis:** Divides data into strata based on covariates and estimates ATE within each stratum.
- **Causal Forests:** Builds decision trees to specifically search for subgroups with differing treatment effects, focusing on heterogeneity.

Searching for HTEs: What does it mean?

- Not just looking for places where the treatment has an effect.
- Specifically looking for places where the treatment effect differs from one subgroup to another.
- This is the essence of **heterogeneous** treatment effects.

How does Causal Forest Achieve This?

- During tree-building, the split criteria isn't just about predicting the outcome (like in traditional regression trees).
- It's about finding splits that maximize the difference in treatment effects between the resulting subgroups.
- In other words, the tree is built to find heterogeneity in treatment effects, not just to predict the outcome.

Optimizing Over Differences in HTEs

- Consider two potential splits:
  - → Split A: ATE in subgroup 1 is 5%, and ATE in subgroup 2 is 6%.
  - → Split B: ATE in subgroup 1 is 2%, and ATE in subgroup 2 is 9%.
- While the ATEs in Split A are closer to each other, Split B has a larger difference, indicating greater heterogeneity.
- Causal forests would favor Split B as it reveals a more pronounced difference in treatment effects.

Benefits of Searching for HTEs

- Allows for more targeted interventions by identifying specific groups that benefit most.
- Provides richer insights into the mechanisms of treatment effects.
- Offers a nuanced understanding compared to a one-size-fits-all ATE approach.

Strata-Specific ATEs

- Black men aged 45yo Democrat: ATE = 45
- Black women aged 45yo Republican: ATE = 45
- White men 40yo Democrat: ATE = 10
- White women aged 40yo Republican: ATE = 45

Traditional Approach vs. Causal Forests

- Traditional approach: Focus on strata-specific ATEs, leading to separate treatment effects for each distinct group.
- Causal forests: Prioritize finding splits in the data that maximize the difference in treatment effects between subgroups.

Causal Forests Optimization

- It doesn’t merely compute the average treatment effects for predetermined strata.
- Instead, it searches for partitions in the data where the treatment effect is most heterogeneous.
- The goal is to identify where the treatment effect differs the most, not just the average effect within known subgroups.

Outcome of Causal Forests on the Example

- Causal forests might recognize two dominant patterns:
  - → ATE of 45 for the majority of the groups.
  - → ATE of 10 for White men 40yo Democrat.
- Instead of treating each stratum separately, causal forests might group Black men aged 45yo Democrat, Black women aged 45yo Republican, and White women aged 40yo Republican together due to similar ATEs.

Estimation Process

- Within each leaf, treatment effects are estimated by comparing outcomes of treated and control units.
- This difference in means provides the localized average treatment effect for units in that leaf.
- These localized estimates are then aggregated across the forest to provide an overall estimate.

Roadmap

Background Material for Causal Forests

- A Brief History of Trees
- Fundamental Machine Learning Concepts
- Parameters
- Data
- Unconfoundedness and common support assumptions

Estimation

- Causal forest algorithm
- Understanding Honesty in Causal Forests
- Final comments

Data

For causal forests to perform well, you typically need a large number of observations (large N) because the method benefits from having a lot of data to find nuanced treatment effects. However, having a variety of covariates can also be beneficial as it allows the algorithm to find more refined subgroups of interest.

Causal Forests: Estimating HTEs

- Causal forests find natural splits to estimate HTEs, not strata-specific ATEs.
- Focuses on regions where treatment effects vary the most.
- Still needs overlap (common support) within splits to make valid comparisons.
- Without overlap in a region, causal inferences are unreliable.

Importance of Common Support in Causal Forests

- Causal forests adaptively partition data based on treatment effect variability.
- Within each partition, valid comparisons require data from both treatment and control groups.
- Absence of common support in a region means no reliable causal inference for that region.
- In summary: Causal forests are adaptive but still rely on foundational causal inference assumptions for validity.

Causal Forests: Addressing the Challenge

- Causal forests partition the covariate space, estimating effects within partitions.
- Allows for local treatment effect estimation where there's overlap.
- Highlights regions where common support might be violated.
- Before going into the algorithm, let me review a high level idea

Regularization in Causal Forests

- First, regularization, one of the things I usually associate with machine learning predictive analytic techniques
- Regularization is crucial to prevent overfitting in machine learning.
- In tree-based methods, regularization isn't a direct term but is achieved through various techniques.
- In causal forests, regularization is accomplished a particular way, but first I want to just say it's purpose and lead us into that

Regularization Techniques

- **Tree Pruning**: Simplify the model by removing some leaves after a tree is fully grown (more later).
- **Minimum Leaf Size**: Prevent overly granular splits by setting a minimum number of observations in a leaf (more later!)
- **Maximum Depth**: Limit how deep a tree can grow. (more later!!)
- **Bagging**: Average predictions over multiple trees. (more later!!!)
- **Random Feature Selection**: At each split, only consider a random subset of features. (more later !!!!)

Honesty as Regularization

- In causal forests, the ”honesty” principle acts as a form of regularization
- Data is split into tree-building and outcome estimation sets which is going to be how avoidijng overfitting are accomplished.
- This separation ensures the model doesn’t overfit when estimating treatment effects.
- So keep this in mind as we dive now into the algorithm

The Causal Forest Algorithm

- Combines multiple causal trees to estimate heterogeneous treatment effects.
- Each tree is built using a bootstrapped sample, introducing variability and reducing overfitting.
- Random subsets of covariates (”feature bagging”) are chosen at each split, preventing domination by a few strong predictors.
- Treatment effects are aggregated across trees, stabilizing estimates and improving accuracy.

1. Objective Function for Splits

- Causal trees prioritize identifying differences in treatment effects over just reducing variance.
- Splits are made by maximizing the differences in estimated treatment effects between subgroups.
- Unlike traditional regression trees, which focus on outcome prediction, causal trees home in on treatment effect heterogeneity.

2. Minimizing Noise and Overfitting

- In finite samples, random variability is inevitable.
- ”Honesty” in tree-building ensures we don’t overfit to the noise.
- Data is split into tree-building and treatment effect estimation sets.
- This separation helps to ensure observed differences are genuine, not just artifacts of the data.

Honesty in Causal Forests

- Traditional decision trees use the same data to decide on splits and estimate outcomes.
- This can lead to ”noise fitting” or ”overfitting” where the model learns not just the underlying patterns but also the random noise present in the data.
- ”Honesty” was introduced to separate the tree-building process from outcome estimation.

Avoiding Noise Fitting

- By splitting the data into two sets, one for building the tree and the other for estimating treatment effects, the method avoids being overly influenced by noise.
- The term ”honesty” implies that the method provides a more genuine or unbiased estimate of the treatment effect.
- It’s a safeguard against the tree being too optimistic in its predictions or identifying spurious patterns.

Origins and Implications

- The term ”honesty” in this context originates from the work of Athey and Imbens, among others, emphasizing the need for unbiasedness in causal effect estimation.
- It’s a departure from traditional machine learning which often optimizes for prediction accuracy at the risk of overfitting.
- In causal inference, overfitting can lead to biased treatment effect estimates, hence the emphasis on honesty to ensure robustness and reliability.

Example: Overfitting in Medical Data

- **Scenario**: Researchers are analyzing medical data to predict which patients are most likely to develop a specific illness based on a variety of health metrics (e.g., blood pressure, cholesterol, etc.).
- **Traditional Tree**: Using a regular decision tree, the model identifies a very specific subgroup of patients: those with blood pressure in a narrow range (121-123 mm Hg), cholesterol level of precisely 205 mg/dL, and who had exactly 3 doctor visits last year.
- **Outcome**: This subgroup, although specific, shows a very high rate of illness in the training data.

Pitfalls of Overfitting

- **Issue**: This subgroup might just be a random cluster in the training data – maybe by sheer chance, a few patients with those exact metrics got sick, but it doesn't reflect a broader, generalizable trend.
- **Result**: When the model is tested on new data, the predictions for this subgroup are highly inaccurate due to the model being too tailored to the noise of the training data.
- **Honest Approach**: By separating tree-building from outcome estimation, an honest tree wouldn't be swayed by such random clusters and would be more robust to the inherent randomness in any dataset.

Honesty is a Guard Against Overfitting

- **Conventional Issue**: Decision trees often capture noise by tailoring the subgroups too closely to the training data's outcomes.
- **Overfitting Manifestation**: A subgroup in the training data might not reflect a broader, generalizable trend. This can lead to poor predictions on new data.
- **Honest Approach**:
  - → Use one sample to determine tree structure based on covariates.
  - → Use a separate, unbiased sample to estimate outcomes for the defined subgroups.
- **Benefit**: This separation ensures the model doesn't overfit to the idiosyncrasies of a single dataset, resulting in more robust and reliable predictions.

3. Pruning and Minimum Leaf Size

- To prevent over-segmentation, trees can be pruned. Less informative branches can be trimmed.
- Setting a minimum leaf size ensures each terminal node has a sufficiently large sample.
- This reduces the influence of random noise on estimated treatment effects.

Understanding Over-Segmentation

- **Definition**: Over-segmentation occurs when data is divided into excessive, often tiny, subgroups that don't capture meaningful patterns but rather capture noise.
- **Simple Example**: Imagine a classroom where students are grouped by height to predict their test scores. Over-segmentation would mean creating groups for every centimeter increase in height, even when such minute differences don't meaningfully correlate with scores.
- **In Context of First Sample**: When building the tree with the first sample, there's a risk of over-segmentation as the model tries to find the "best" splits. This is why pruning and setting minimum leaf sizes are crucial.

Over-Segmentation and Common Support

- Over-segmentation can lead to tiny subgroups with limited data, producing unreliable estimates.
- It can also result in subgroups without both treatment and control observations, violating the common support assumption.
- Pruning and setting minimum leaf sizes address these issues, ensuring more reliable and robust treatment effect estimates.

4. Multiple Trees & Bagging

- A random forest averages results over many trees.
- Each tree is built on a bootstrapped sample and uses a random subset of predictors.
- By averaging over many trees, random forests reduce variance and provide a more robust estimate of treatment effects.

What is Bagging?

- ”Bagging” stands for Bootstrap AGGregatING.
- It involves taking multiple random samples (with replacement) from the dataset and building a separate decision tree for each sample.
- For instance, if our dataset has 1000 rows, we might create 10 bootstrapped samples. Each sample might contain duplicates of some rows and might omit some others.
- Each of these samples will be used to build a separate tree.

Why Bagging?

- Trees built on different samples will show variability.
- A single decision tree can be sensitive to small changes in data (high variance).
- By averaging predictions over multiple trees, bagging reduces this variance.
- Think of it as consulting multiple experts and then averaging their opinions.

Random Subsets of Predictors

- In addition to bootstrapped samples, each tree in a random forest considers only a random subset of predictors at each split.
- If we have 10 predictors, a single tree might only look at 3 or 4 of them for deciding a split.
- This introduces further variability among trees and ensures that the forest doesn’t overly rely on any one predictor.

Combining Predictions: Understanding Averaging

- After building many trees, the random forest aggregates their predictions.
- For estimating treatment effects, this means averaging the predicted values.
- Averaging helps in reducing the noise and provides a more stable and consistent estimate.

Concrete Example: Advertising Campaign Impact on Sales

- Imagine a company launching a large advertising campaign.
- The ”treatment” is the exposure to the campaign. The outcome is the increase in sales.
- Different consumers might respond differently based on their previous purchase history, demographics, etc.

Concrete Example: Advertising Campaign Impact on Sales

- We bootstrap our consumer data to create diverse samples. One bootstrapped sample might end up with more young consumers, while another might have more elderly consumers due to sampling with replacement.
- Each tree in our causal forest focuses on different attributes. One tree might split based on age groups, another on previous purchase frequency.
- By averaging the predictions from all trees, the company can better identify which consumers are most likely to increase their purchases after being exposed to the campaign, estimating the HTEs.

Implication

- The decision on meaningful HTEs is a mix of:
  - → Statistical criteria of the tree algorithm.
  - → Variability in the data.
  - → Regularization techniques applied.
- While slight differences are everywhere, causal forests identify systematic differences, not just noise.

What Makes it Honest?

- So let's return to the honesty principle from earliers
- Overfitting is a concern: How to ensure we're not identifying random quirks as significant treatment effects?
- Solution: Split training data into two subsamples: a splitting subsample and an estimating subsample.
- Build the causal tree using the splitting sample.
- Apply the tree to the estimating sample, and determine treatment effects within each leaf.
- Future cases use these estimates when the model is applied.

Added Benefits of Honest Causal Trees

- Athey and colleagues demonstrate that treatment effect estimates from honest causal trees are asymptotically normal.
- Asymptotically normal: As sample size grows indefinitely, treatment effect estimate follows a normal distribution.
- Key implications:
  - → Enables calculation of variance and 95% confidence intervals.
  - → Recognizes inherent uncertainty in predictions.
  - → Potential to target anyone with an expected treatment effect statistically significantly above zero.
- Practical value: Appreciating the uncertainty and making informed decisions in interventions.

1. Data Splitting

- So let’s review now: the data is divided into two distinct parts:
  - → **Splitting Sample**: Used to structure the tree.
  - → **Estimation Sample**: Used to compute treatment effects.
- This split ensures a clear distinction between the model’s structure and its performance evaluation.

2. Tree Building with the Splitting Sample

- The tree's structure (i.e., where to make splits based on covariates) is decided using the splitting sample.
- At this stage, the algorithm is **not** evaluating treatment effects.
- It's focusing on defining the "framework" based on covariates and potential treatment interactions.
- **Framework Explained**: The "framework" refers to the foundational structure of the tree which sets up the subgroups based on covariates without yet assigning or evaluating treatment effects to these groups.

3. Treatment Effect Estimation with the Estimation Sample

- So we have the framework setting up subgroups based on covariates, but what?
- We apply this pre-constructed tree to the estimation sample – a kind of “moving over” from the training data to the estimation sample.
- As each data point lands in a leaf node, the treatment effect within that leaf is computed.
- This is done by calculating the difference in outcomes (e.g., means, medians) between the treated and control groups within the leaf.
- Recall what unconfoundedness means – randomization *within* covariate strata, and leaves *are* covariate strata

4. The Intuition Behind Honesty

- The splitting sample sets the ”stage” or ”framework.”
- The estimation sample acts as the ”actors” that play out within that framework.
- By separating these roles, the model’s structure isn’t biased by the treatment effects it later evaluates.
- This approach guards against overfitting and provides a more genuine estimate of treatment effects.

Implication of Honesty

- Achieves a more ”honest” treatment effect estimate.
- Mitigates the risk of the model being overly optimistic about its own performance.
- The tree structure is determined without ”peeking” at treatment effect outcomes, minimizing the likelihood of identifying false patterns.
- Let’s walk through a numerical example together to see if we understand

What the Algorithm IS Doing

- At each node, it evaluates potential split points for every variable, not necessarily every unique value.
- For continuous variables like GPA, it might consider natural breakpoints or where there are significant changes in the treatment effect.
- The split criterion, e.g., maximizing the difference in treatment effects, determines the ”best” split.
- Once the best split is chosen, data divides accordingly, creating two child nodes.
- The process continues recursively, now evaluating potential splits at the child nodes.

Bagging in the Context

- Bagging involves creating multiple datasets by sampling with replacement from the original dataset.
- A separate tree is built for each of these bootstrapped datasets.
- Final prediction (or estimated treatment effect) is an average of outputs from all these trees.
- By leveraging the diversity in these bootstrapped samples, bagging reduces the variance in predictions.

What the Algorithm is NOT Doing

- It does **not** make a split, then backtrack and try another split at the same node.
- It does **not** randomly choose a split without evaluating its quality.
- It does **not** use the same data for both determining the best split and estimating the treatment effect, thanks to the ”honesty” principle.

Summarizing the Process

- The algorithm's objective is to find splits that best segregate data based on the difference in treatment effects.
- This involves thorough evaluations at each node, ensuring the best possible decision at every step.
- While it might sound computationally intensive, modern computing power and algorithmic optimizations make this feasible. However, the process can take time, especially with large datasets or numerous variables.

4. Treatment Effect Estimation with the Estimation Sample

- Apply the tree structure to the estimation sample.
- For each individual in a leaf, compute the difference in wages between those who underwent the job training and those who didn’t.
- This difference provides the treatment effect for individuals in that specific leaf.

5. Intuitive Takeaway

- The tree structure was decided without knowing the exact treatment effect in the estimation sample.
- By keeping the structure and effect estimation separate, we minimize the risk of ”chasing” random patterns in the data.
- This ensures a robust and ”honest” estimate of treatment effects.

Variable Importance

- Causal forests can rank variables by their importance in determining heterogeneous treatment effects.
- Variables that frequently lead to splits have a higher importance score.
- Provides insights into which covariates play crucial roles in treatment effect heterogeneity.

Hyperparameters and Their Importance

- Like other machine learning methods, causal forests have hyperparameters that can be tuned.
- Examples include the number of trees, depth of trees, and minimum samples in a leaf.
- Proper tuning is essential to ensure model performance and prevent overfitting.

Role of Cross-Validation

- To validate the performance of causal forests, out-of-sample validation is crucial.
- Cross-validation involves partitioning the data into subsets, training on some and validating on others.
- Helps in hyperparameter tuning and assessing the robustness of treatment effect estimates.

Software Tools and Practical Implementation

- Multiple software packages are available for causal forests, including in R and Python.
- 'grf' in R is a popular choice for causal forests.
- Proper data preprocessing, feature engineering, and post-estimation diagnostics are crucial for successful implementation.

Practical Applications of Causal Forests

- Healthcare: Estimating the effects of treatments or interventions on patient outcomes.
- Economics: Understanding the impact of policy changes or financial strategies.
- Marketing: Evaluating the effectiveness of advertising campaigns on different demographics.
- Social Sciences: Studying the influence of educational programs or societal interventions.
- These applications underscore the versatility and utility of causal forests in real-world decision-making.

Strengths and Weaknesses of Causal Forests

- **Advantages**:
  - → Captures complex, non-linear relationships.
  - → Robust to high-dimensional covariate spaces.
  - → Offers insights into heterogeneous treatment effects.
- **Limitations**:
  - → Requires a good deal of computational resources.
  - → Assumption of unconfoundedness needs to hold (imo this is most likely being blurred by practitioners; would they use propensity scores? Probably not)
  - → Might not perform well with very sparse data.

Extensions and Innovations

- **Generalized Random Forests**: Expands causal forests to other statistical tasks, like quantile regression.
- **Adaptive Causal Trees**: Dynamic algorithms that adjust to evolving data streams.
- **Instrumental Variable Forests**: Incorporates instrumental variables to handle unobserved confounding.
- Continuous developments ensure that causal forest methods remain at the forefront of causal inference techniques.

Data Preparation for CRF

- Understand the dataset: Ensure you have both treatment and control groups.
  - → *“What’s on the LHS? The RHS?* - JohnDinardo”
- Pre-process the data: Clean and handle missing values.
- Split the data: Create training and test datasets to evaluate the model’s performance.

CRF Review: Data Preparation

- Ensure both treatment and control groups are present.
- Clean and handle missing values.
- Split the dataset for training and testing to evaluate model performance.

CRF Review: The Bagging Procedure

- Bootstrap the data (Bagging) to create multiple bootstrapped samples drawn (with replacement) from the dataset.
- A tree is constructed for each sample, capturing diverse data characteristics.

CRF Review: Tree Construction

- Trees segregate data based on covariates to maximize differences in treatment effects
- But not in the same dataset; two datasets – a type of ”training dataset” and a type of ”estimation dataset” (as opposed to a prediction dataset)
- ”Honesty” principle ensures separate samples for tree structure and treatment effect estimation.
- Iteratively evaluates potential split points for every variable.

CRF Review: Treatment Effect Estimation

- Applied on the estimation sample using the pre-constructed tree.
- Treatment effects computed within leaf nodes by calculating differences in outcomes between treated and control groups.

CRF Review: Hyperparameters and Evaluation

- Number of trees (iterations) can be tuned for optimal performance.
- Model evaluation done using test dataset to assess predictive and causal inference accuracy.
- Iterative process: hyperparameter tuning and cross-validation for optimal results.

Interpreting CRF Results

- Assess variable importance: Understand which features most influence causal effects.
- Uncover nuanced insights: Recognize patterns that traditional methods might miss.
- Visualize results: Use plots and graphs to represent causal relationships.

Limitations and Challenges

- Overfitting: Ensure the model doesn’t overly adapt to the training data.
- Hyperparameter tuning: Crucial for model performance and accurate causal inference.
- Model generalizability: Ensure findings can be applied to other, similar datasets.

Concluding Thoughts on CRF

- Versatility of CRF: Applicable in various fields and research domains.
- Encourage exploration: Challenge researchers to explore and apply CRF in their work.
- Continuous learning: As with all models, stay updated with advancements and improvements in CRF methodology.

Conclusion and Key Points

- Causal forests elegantly combine machine learning with causal inference.
- They address challenges of high dimensionality and heterogeneous effects.
- A powerful tool, but like all methods, they have their strengths and limitations.
- With continuous research, the potential and applications of causal forests are ever-expanding.

Temporary page!

LaTeX was unable to guess the total number of pages correctly.
was some unprocessed data that should have been added to
page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus pag
away, because LaTeX now knows how many pages to expect fo
document.