Lesson 6 - Conditional Inference Trees and Random Forests

Erin M. Buchanan

02/14/2019

Language Topics Discussed

- Categories
- Category Structure
- Category Theories

Definition

- What is a category?
  - Category – group or organization of related things
  - Concept – a member of a category (i.e. the thing)
  - Animals: dog, cat, bird, fish

Category Structure

- Super-ordinate – more abstract (animal, mammal)
- Basic level names – dog
- Subordinate – specific category member (collie, beagle)

Category Theories

- Category formation
  - Based on the way we perceive the world
  - Cognitive economy – memory is organized to be efficient, avoiding lots of duplication
- Several ways we learn categories:
  - Feature list theory
  - Probabilistic theories
  - Prototype theories
  - Exemplar theories
  - Theory theories (rule based)

Feature List Theory

- Our semantic knowledge is based around a list of features that make up that concept
  - Defining features – essential to the meaning of words
  - Characteristic features – usually true of category members, but not always
- How might we test this theory?
  - Sentence verification task: judge if a sentence is true or false

Sentence Verification Task

- A dog is an animal.
- Stage 1: overall feature similarity is computed.
  - If you have lots of features overlapping, sentence RL is fast.
  - If you have no overlap, sentence RL is fast.
- Stage 2: consider the "defining" features.
  - These longer RL show the gray areas or fuzzy boundaries of categories.

Probabilistic Feature Model

- Core description - essential defining features of the concept
- Identification procedures used to identify instances of a category
- Features are weighted by saliency and probability

Issues

- "Defining" features
- Inter-correlated features – relationship between features not captured
- Procedural invariance – same question gives you different answers – it shouldn't with these theories
  - Is a robin a bird?
  - Is a bird a robin?

Family Resemblance Models

- Prototype theory versus exemplar theory
  - Prototype – an abstraction that is the best example of a category
  - Prototypes are likely a combination of experienced examples, but may not exist in real world
  - Exemplar theory – we compare information to a specific stored example
  - Instantiation principle – category includes detailed information about the range of instances
- These are very similar in their ideas, but the underlying core is distinction

Family Resemblance Models

- You decide that something is in the category by comparing to the prototype or exemplar
- Schema – a means for organizing knowledge
- Features are said to be schema fillers
- Sentence verification faster for prototypical members

Theory theories

- People represent categories as miniature theories that describe facts about those categories and how they relate
- Sort of like a dictionary
- Children do something like this

(more on these theories and semanticity over the next few weeks)

The Statistics

- Conditional Inference Trees: method of regression and classification based on binary recursive partitioning
  - First, assess the association of the IV with the DV and chose the one with the largest association
  - Second, the data is split into two subsets . . . if the IV is categorical, this is performed along categorical lines, while continuous data might be median split
  - Continue these associations and splits until no variables are related to the DV
  - Considered a “tree” because we are creating branches and leaves

Advantages

- As compared to other recursive partitioning and classification procedures, this procedure:
  - Variable selection is less biased (i.e., does not automatically pick a variable it can split the most)
  - Do not need to “prune” the tree
  - Shows you p-values for the splits

Permutation

- To obtain those p values, you use permutation
- Permutation means that you rearrange the data, calculate a statistic, and count how many times the effect was present in the rearranged data (want small)
- Note the differences from bootstrapping

Random Forests

- Random Forests show the importance of each variable, averaged over many conditional trees
- Akin to the idea of partial variance
- Both conditional inference trees and forests are useful when the data is sparse and is non-parametric, thus, reducing assumptions

Getting Started

```
#install.packages("party")
library(Rling)
library(party)
```

```
## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Dataset

- DV: category instances: make, have, cause
- IVs:
  - CrSem: semanticity of the actor (animate, inanimate)
  - CeSem: semanticity of the actee (animate, inanimate)
  - CdEv: semanticity of the event (mental, physical, social)
  - Neg: negation (negative or not)
  - Coref: yes (you did the thing to you) versus no (you did the thing to others)
  - Poss: possessive yes or no

Cleaning up the data

```
data(caus)

reduced_data = subset(caus,
                      Cx == "make_V" | Cx == "have_V" | Cx ==
reduced_data$Cx = droplevels(reduced_data$Cx)
```

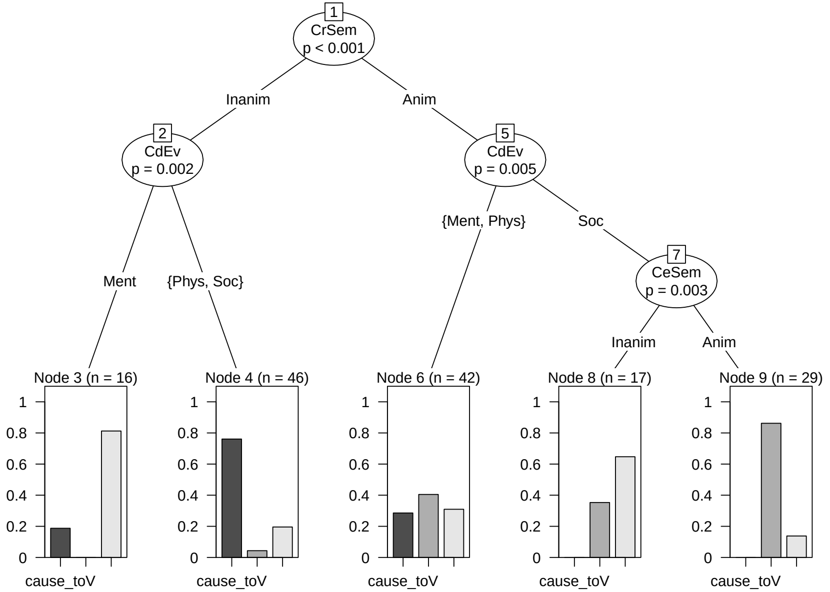
Getting the model started

```
#start with a random number generator
set.seed(549354)

#generate a tree
tree.output = ctree(Cx ~ CrSem + CeSem + CdEv + Neg + Coref
                    data = reduced_data)
```

Make a plot

```
plot(tree.output)
```



Interpretation

- Tree includes all possible splits that were significant at $p < .05$
- Ovals are the names of the variables with the best split
- The splits are shown on the branches
- Bottom bar chart helps show the number of DV instances in each split

Interpretation

- The first split is between inanimate and inanimate actors
- The next split is on the semanticity of the event
- On the left side, that split into mental versus physical and social
  - Make appears to be featurally comprised of mental inanimate events
  - Cause appears to be featurally comprised of physical or social inanimate events

Interpretation

- On the right side, we see another split, but into mental and physical versus social
  - Each verb is equally allocated to animate mental and physical events
- The animate actor social groupings, then split on the semanticity of the actee
  - Make has another feature set of animate actor, social action, and inanimate actee
  - Have is comprised of animate actor, social action, and animate actees

But how good is the model?

- Like log regression, you can tabulate the predicted probability and the actual outcome
- Add up the diagonal and divide by the total

```
outcomes = table(predict(tree.output), reduced_data$Cx)
outcomes
```

```
##
##             cause_toV have_V make_V
##   cause_toV        35      2      9
##   have_V           12     42     17
##   make_V            3      6     24
```

```
sum(diag(outcomes)) / sum(outcomes) * 100
```

```
## [1] 67.33333
```

Random Forests

```
forest.output = cforest(Cx ~ CrSem + CeSem + CdEv + Neg + C
                    data = reduced_data,
                    #grow a big forest
                    controls = cforest_unbiased(ntree = 100
                                                mtry = 2))
```

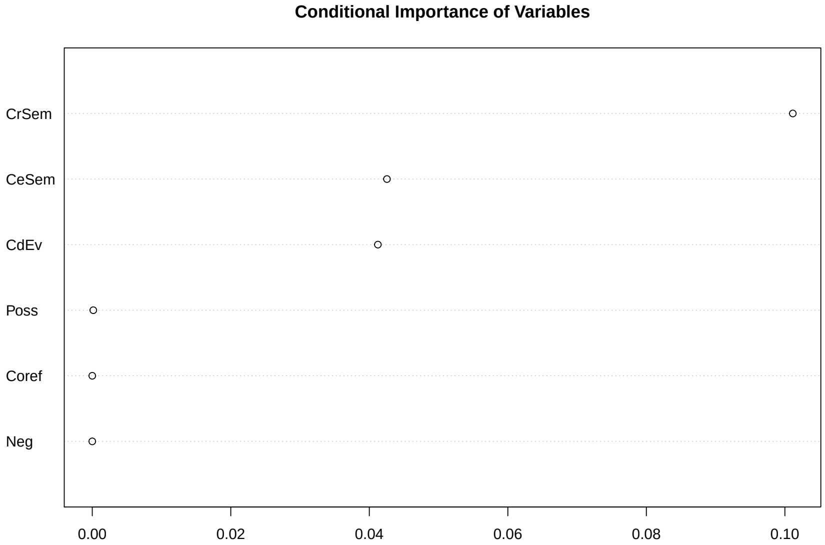
Variable Importance

```
forest.importance = varimp(forest.output,
                           conditional = T)
round(forest.importance, 3)
```

```
## CrSem CeSem  CdEv   Neg Coref  Poss
## 0.101 0.043 0.041 0.000 0.000 0.000
```

Variable Importance

```
dotchart(sort(forest.importance),
         main = "Conditional Importance of Variables")
```



Model Prediction

```
forest.outcomes = table(predict(forest.output), reduced_dat
forest.outcomes
```

```
##
##              cause_toV have_V make_V
##   cause_toV         39      2      9
##   have_V             8     43     17
##   make_V             3      5     24
```

```
sum(diag(forest.outcomes)) / sum(forest.outcomes) * 100
```

```
## [1] 70.66667
```

Summary

- We can begin to think about defining features of categories by looking at sentences (or other related category tasks).
- Using a conditional inference tree or random forest, we can see what might be defining or exemplars for each category.
- You can use these models with sparse data or potentials for interactions.