Chapter 25
Conditional Inference Trees and Random Forests

Natalia Levshina

Abstract This chapter discusses popular non-parametric methods in corpus linguistics: conditional inference trees and conditional random forests. These methods, which allow the researcher to model and interpret the relationships between a numeric or categorical response variable and various predictors, are particularly attractive in 'tricky' situations, when the use of parametric methods (in particular, regression models) can be problematic, for example, in the situations of 'small *n*, large *p*', complex interactions, non-linearity and correlated predictors. For illustration, the chapter discusses a case study of T and V politeness forms in Russian based on a corpus of film subtitles.

25.1 Introduction

Conditional inference trees (CITs) and conditional random forests (CRFs) are gaining popularity in corpus linguistics. They have been fruitfully used in models of linguistic variation, where the task is to find out which linguistic and extralinguistic factors determine the use of near-synonyms (e.g. *let*, *allow* or *permit*), alternating syntactic constructions (e.g. the double-object vs. *to*-dative) or sociolinguistic variants (e.g. the type of /r/ used by speakers of a particular dialect). The methods have been implemented in a user-friendly way in the packages `party` (Hothorn et al. 2006b; Strobl et al. 2007) and `partykit` (Hothorn and Zeileis 2015), which has contributed to the popularity of the methods.

CITs belong to the family of recursive partitioning methods, which involve the following basic steps.

Electronic Supplementary Material The online version of this chapter (https://doi.org/10.1007/978-3-030-46216-1_25) contains supplementary material, which is available to authorized users.

N. Levshina (✉)
Max Planck Institute for Psycholinguistics, Nijmegen, The Netherlands
e-mail: natalia.levshina@mpi.nl

M. Paquot, S. Th. Gries (eds.), *A Practical Handbook of Corpus Linguistics*,
https://doi.org/10.1007/978-3-030-46216-1_25

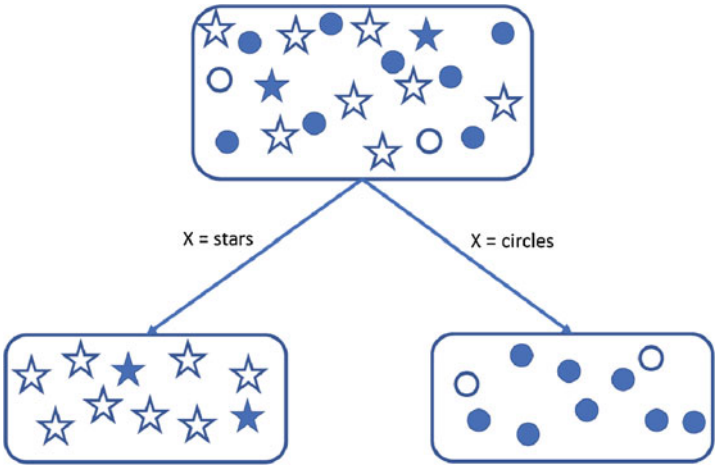

Fig. 25.1 Binary partitioning of a dataset

Step 1. Select the predictor which helps best to distinguish between different values of the response variable, using some statistical criterion.
Step 2. Make a split in this variable, splitting the data in several data sets. Most algorithms use binary partitioning, although non-binary splits have also been implemented.
Step 3. Repeat Steps 1 and 2 recursively until no further splits can be made, based on certain pre-defined criteria.

Figure 25.1 illustrates the main idea behind binary partitioning. Imagine a set of objects (circles and stars) that can be white or blue. The shape allows one to separate the white objects from the blue ones by splitting the entire data set into two subsets, one with stars and the other with circles. The goal is to achieve maximal purity (or minimal impurity) in the terminal nodes. In other words, we would like to have as few blue objects in the bottom left-hand node as possible, and as few white objects in the bottom right-hand node as possible. In our case, 80% of the objects in the node with stars are white, and 80% of the objects in the node with circles are blue. This means that splitting the data according to shape allows us to predict the colour correctly in 80% of the cases.

Random forests, including CRFs, represent an ensemble method, by which many individual trees are 'grown', and their predictions are averaged. Each tree is based on a random sample of n observations from the original dataset, usually with replacement, and on a random sample of k predictors from all predictors in the model. Random forests usually produce more accurate predictions than single trees.

CITs and CRFs have a number of advantages in some situations when the use of traditional approaches, such as regression analysis (see Chap. 21), may be inappropriate (see more details in Sect. 25.2). In addition, CITs allow one to interpret high-order interactions, which involve more than two predictors, in a very

convenient and intuitive way. However, the methods also have their pitfalls (see Sect. 25.2.6).

25.2 Fundamentals

25.2.1 Types of Data

Although most applications of these methods in corpus linguistics involve categorical response variables and predominantly categorical predictors, CITs and CRFs can model the relationships between the response variable and predictors at any scale of measurement. Also, one can fit models with multivariate response variables and censored data.[1] Traditionally, models with numeric response variables are called regression trees, and models with categorical response variables are referred to as classification trees.

CITs and CRFs can be particularly useful in the situations of small n, large p. This may be the case in many subfields of corpus linguistics, where data are small and costly, e.g. analysis of spoken data (e.g. Tagliamonte and Baayen 2012), multilingual data (e.g. Levshina 2016) or data from less well documented languages. For such datasets, logistic regression models would not be suitable. There are no fixed rules regarding the minimum number of observations, but it is possible to have a relatively high number of predictors. As an extreme example, one can mention the study of Strobl et al. (2008), where CRFs are used to investigate a binding property of 310 amino acid sequences depending on 105 predictors. A multiple regression analysis of such data would be inappropriate.

CRFs can be used in situations where predictors are highly intercorrelated. This happens quite often in corpus-linguistic research. For instance, one and the same underlying theoretical construct, such as transitivity in Hopper and Thompson's (1980) sense, can be represented by several correlated semantic and morphosyntactic variables. Another example is entrenchment of a word, which can be operationalized as frequency, dispersion and contextual diversity of a word in a corpus (e.g. Baayen 2010). In such cases, a regression analysis may fail due to multicollinearity issues.

In addition, CITs and CRFs represent an attractive alternative when some of the regression assumptions are not met, e.g. the assumptions of homoscedasticity (equal variability of the response variable across the range of values of the predictors) and non-linearity (lack of direct proportionality) of the relationship between a predictor and the outcome. This is particularly convenient if one wants to keep the original scale of the response variable rather than perform transformations (see, however,

[1] Censored data are characterised by the presence of some observations that only specify an interval instead of an exact value, such as age older than 80 years or any size of clothes larger than XXL.

Chap. 23 on generalized additive models, which represent a tool for modelling non-linear relationships).

25.2.2 The Assumptions

There are no traditional assumptions that should be met when fitting a CIT or a CRF, such as constant variance, non-linearity or normally distributed errors. However, a word of caution should be said about the independence of observations. At the moment of writing, the only working method for dealing with dependent observations seems to be including the grouping factor (e.g. the speaker or text IDs) on a par with all other predictors.[2] This has been done, for instance, by Tagliamonte and Baayen (2012), who added the individual speakers as a covariate in their CITs and CRFs. This method cannot be regarded as a perfect solution, unfortunately, as will be shown in Sect. 25.2.6.

25.2.3 Research Questions

The research questions that can be answered with the help of CITs and CRFs are the same as the ones that are answered with the help of regression analysis. A typical question is which linguistic factors help to predict the use of particular linguistic variants. Examples are variation of *was/were* in York English (Tagliamonte and Baayen 2012), transitivity patterns with the verb *give* in South Asian Englishes (Bernaisch et al. 2014), fore- and backclipping (e.g. *technology* > *tech*, but *racoon* > *coon*, cf. Lohmann 2013) and positioning of adverbial concessive clauses (Wiechmann and Kerz 2012). Most of this work has been done on English data. Some exceptions are Baayen et al. (2013), who investigate Russian aspectual prefixes and suffixes, and Levshina's (2016) study of causative constructions in 15 European languages. Usually, the task is to find the contextual (corpus) variables that are associated with the choice between two or more linguistic variants. CITs are particularly useful in multifactorial probabilistic grammar, when one investigates interactions between contextual variables and the variables representing language varieties, as in the study of several morphosyntactic alternations in World Englishes by Szmrecsanyi et al. (2016).

Similar to regression modelling, one can use these methods for explanation and prediction (including classification). CITs can be particularly useful for explanation and interpretation, whereas CRFs are usually better in prediction.

[2] A solution with random effects has been implemented in the package REEMtree, but it is available only for longitudinal data.

25.2.4 The Algorithms

25.2.4.1 The CIT Algorithm

The method is based on testing the null hypothesis that the distribution of the response variable D(Y) is equal to the conditional distribution of the response variable given some predictor D(Y|X). The global null hypothesis says that this holds for all predictors (Hothorn et al. 2006a). In order to test the hypothesis, one uses permutation (reshuffling) of the labels in the response variable Y. By permuting Y, its association with the predictor X is broken. As a result of this permutation, the null distribution is derived directly from the data. In contrast, traditional tests impose assumptions on the distribution of the data, so that the null distribution of a test statistic can be derived analytically.

As an illustration, consider the dative alternation in English. Some imaginary data are provided in Table 25.1. We investigate the dependence between the response (i.e. the use of the double object dative and the *to*-dative) and a number of predictors, including the information status of the Recipient (given or new). A permuted version of the response is shown in the third column of the table.

Based on this information, one computes a statistic that involves the difference in the association between response Y and predictor X before and after the permutation. The greater this difference, the stronger the association between Y and X. The package party offers two types of test statistics: c_{quad} (the default) and c_{max}.The difference in the results can be observed if there are categorical variables with more than two values (see Hothorn et al. 2006a or 2006b for the details). In most situations, there is no need to modify the default settings. One could then compare the statistics from different covariates and choose the largest one as the best candidate for the next split. However, this is not a very good idea when the predictors are on different measurement scales. In order to make the statistics comparable across different variables, the *p*-values are normally used (the default option). For the global null hypothesis test, the *p*-values are adjusted (a simple Bonferroni correction is used by default).

There are several options for computing the *p*-values. By default, the algorithm returns the asymptotic *p*-values because the test statistics used in the algorithm are

Table 25.1 Permutation of the response variable: an example

Example ID	Observed response Y (dative variant)	Permuted response Y (dative variant)	Observed predictor X (information status of Recipient)
1	To-dative	DO-dative	New
2	DO-dative	DO-dative	New
3	To-dative	To-dative	Given
101	DO-dative	DO-dative	New
102	To-dative	To-dative	Given
103	DO-dative	To-dative	Given
...	...	...	...

shown to tend to well-known distributions. More precisely, a χ^2 distribution in the case of the test statistic c_{quad} and a multivariate normal distribution in the case of c_{max} (Hothorn et al. 2006b). Alternatively, one can approximate the p-values by using a Monte-Carlo method and simulate the distribution of the test statistic by randomly reshuffling the data. The number of permutations can be specified, as well.

After the predictor has been selected, the next step is splitting the selected covariate into two disjoint sets. For variables with many possible splits, the algorithm computes a statistic for every possible binary split into subsets A and not-A, similar to how it was done during the process of variable selection. Next, the split with the maximal test statistic is chosen. The procedure is then repeated recursively until certain criteria are met, which are the following:

- minimum criterion for a split, which equals 1 - α. If the global null hypothesis cannot be rejected at a certain level of statistical significance α (by default, 0.05), no further splits are made. This parameter performs two functions: as the significance criterion (and therefore it should not be changed without a very good reason, since it strikes a balance between Type I and Type II errors), and a hyperparameter (i.e. a parameter defined before running the algorithm) determining the size of a tree;
- the minimum number of cases in a node before a split. If there are fewer cases than required, no split will be made;
- the minimum number of cases in a node after a split.

Importantly, one can obtain the values predicted by the model for the observations in the original dataset or for new data. How are these predictions obtained? As an example, consider a white star in the imaginary data displayed in Fig. 25.1. The algorithm 'puts' all stars in the left-hand terminal node. In that case, the predicted category (i.e. white or blue) will be determined by the majority vote. Since the node contains more white objects than blue ones, the predicted colour is white. The predicted probability will be 0.8 (or 80%) because the white objects represent 80% from the entire number of objects in this node (eight out of ten). In case of regression trees the mean value of the response variable in all observations in the relevant terminal node is computed. To find out how to compute the median or another statistic of interest, see `?party::treeresponse`.

25.2.4.2 The CRF Algorithm

A CRF is an ensemble of multiple CITs. The algorithm uses resampling with or without replacement to create a random sample for each tree. Importantly, only a sample of candidate predictors is randomly drawn for each individual CITs. Since only a restricted number of predictors is selected for every individual tree, each variable has the chance to appear in different contexts with different covariates. This may better reflect its potentially complex effect on the response variable (Strobl

et al. 2009). If some variable only matters in a very specific type of contexts, it gets a chance to tell its story when the stronger variables are out of the competition. This is particularly important in situations of multicollinearity, where predictors are highly intercorrelated.

To obtain the predicted values from a CRF, one needs to aggregate the results from the individual trees. To predict the response value for an individual observation with particular values of the predictors, the algorithm combines the information about all relevant observations that have the same properties as the observation of interest, in all trees. The predicted value for a given observation is then the average value of the response variable in all those observations (in case of regression trees with numeric response variables) or the most popular category (in case of classification trees with categorical response variables).

Importantly, one can distinguish between predicted values for the training samples and those for the out-of-bag (OOB) samples. Recall that certain data points are left out during the bootstrap sampling or subsampling before a tree is fitted. The OOB samples can be used to assess the predictive performance of that specific model since they were not used to build the model.

Importantly, CRFs also provide a linguist with the so-called conditional variable importance scores, which show how important each variable is, taking into account all others and their interactions. To compute this measure for a predictor, the algorithm averages the results from many trees and measures the decrease in prediction power if one randomly permutes the predictor. If a predictor is associated with the response strongly, there will be a substantial decrease in prediction accuracy. More exactly, a conditional permutation method is applied, as described by Strobl et al. (2008). For example, if Y is the response, and X and Z are categorical predictors, the dependence between Y and X is measured within the levels of Z. Let us revisit the example with the dative alternation. Imagine we are testing two predictors: information status of the Recipient (given or new) and the semantics of the Recipient (animate or inanimate). In order to compute the conditional importance of X (information status) given Z (semantics) in an individual tree, we will reshuffle the value of X within the levels of Z, as shown in Table 25.2. Next, we will test how much worse the individual conditional tree with the permuted data will predict the observed scores of the response variable in comparison with the original data. The conditional importance score of X for the entire forest is computed as an average over all trees. One can use the OOB samples or the training data for this task.

25.2.5 CITs and CRFs Compared with Other Recursive Partitioning Methods

The methods described in this chapter belong to a large family of recursive partitioning methods used for regression and classification. Other approaches include

Table 25.2 Conditional permutation scheme: an example

ID of observation	Response	Observed X (information status of Recipient)	Permuted X|Z	Predictor Z (semantics of Recipient)
1	To-dative	New	Given	Animate
2	DO	New	New	Animate
3	To-dative	Given	New	Animate
101	DO	New	Given	Inanimate
102	To-dative	Given	Given	Inanimate
103	DO	Given	New	Inanimate
...	...	...	...	...

CART (Classification And Regression Trees), CHAID (CHi-squared Automatic Interaction Detector), and QUEST (Quick, Unbiased and Efficient Statistical Tree). An overview of these methods can be found in Loh (2014).

One of the main differences of CITs from the other approaches is that the *p*-values are used as a stopping criterion and for choosing the next split (see Sect. 25.2.4.1). This enables one to compare the predictors at difference scales of measurement. The other well-known methods use other criteria, e.g. minimization of impurity in the nodes (the so-called Gini impurity measure) in CART. One of the consequences of using a statistical significance testing criterion is the decrease in statistical power as more and more splits are made in the data. As a result, overfitting is avoided. This approach is quite convenient in practice because it does not require pruning, i.e. removal of the low branches that do not contribute to the predictive power of a tree (see Kuhn and Johnson 2016: 177–178). However, statistical hypothesis tests are not directly related to the predictive performance of the model and 'cleanliness' of the nodes. This should be kept in mind when comparing the results of a CIT with results of a different recursive partitioning method.

Another important difference is that the CIT algorithm separates the steps of variable selection and making of a split, whereas most other methods merge this in one step. As a result, the variables with multiple splits (e.g. categorical variables with many different values) do not have advantages in comparison with the variables with few splits. The same holds for predictors with missing values.

As for CRF, the best known alternative is probably Breiman's (2001) random forests (see R package `randomForest`). An important distinctive feature of CRFs is that one can compute the conditional variable importance measures, which have some advantages in comparison with non-conditional ones (Strobl et al. 2008). Namely, they do not have a bias towards correlated predictors. If X is a real predictor and Z is a spurious one, and X and Z are correlated, the importance value of Z will increase if a traditional non-conditional method is used. This undesirable effect can be avoided if one uses the conditional permutation schema as described in Sect. 25.2.4.2. A disadvantage of conditional variable importance, however, is that it is computationally intensive.

Another difference between CRFs and other ensemble methods concerns the computation of predicted values. In many popular random forest algorithms, the predicted values are an aggregation of predictions from each individual tree (i.e. the mean predicted value or the most popular predicted outcome). In contrast, CRFs make predictions by retaining information about individual observations in each tree (see Sect. 25.2.4.2). As a result, terminal nodes with many observations play a greater role because they provide more data points for this calculation.

The conditional inference approaches do not always outperform the other approaches in terms of prediction or explanation. For example, CART-based random forests sometimes provide better predictive power than CRFs (Kuhn and Johnson 2016: 200–201). One can also obtain similar results when using alternative, less computationally intensive methods, such as CART. The conditional inference framework has been preferred in linguistic studies for two main reasons. First, it provides realistic estimates of variable importance of highly correlated predictors. Second, it avoids overfitting the data. A systematic comparison of the available methods and their practical advantages for corpus linguistics remains a task for future research.

25.2.6 Situations When the Use of CITs and CRFs May Be Problematic

There is no universal statistical method that can be used in all circumstances. CITs and CRFs are no exception to this rule. There are several cases where one might prefer to use a different method or perform additional checks.

Paradoxically, although CITs can be more successful than regression models in tricky situations, e.g. with non-linear patterns and high-order interactions, which involve more than two predictors, CITs may be quite useless in very simple situations, i.e. when the relationships between the response and the predictors are **linear** (i.e. directly proportional) and **additive** (i.e. there are no interactions). As an illustration, consider Fig. 25.2. This tree represents an attempt of a CIT to analyse the relationships between a numeric response variable *y* and predictors *x1* and *x2*, which were randomly generated from the normal distribution. The response variable was created using the following linear regression formula:

$$(1) \quad y = 0.7 + 5.4\,x1 - 2.8\,x2 + \varepsilon$$

where ε contained random normally distributed errors. The predictors do not interact and the relationship between them and the response variable is linear. One can see that the tree presentation is not very helpful in modelling these simple relationships. It creates an illusion of interactions that are not present in the data. Moreover, the numeric predictors are split multiple times, which masks the linear relationships between them and the response variable. A multiple regression model would be a much better choice in such cases (see Chap. 21).

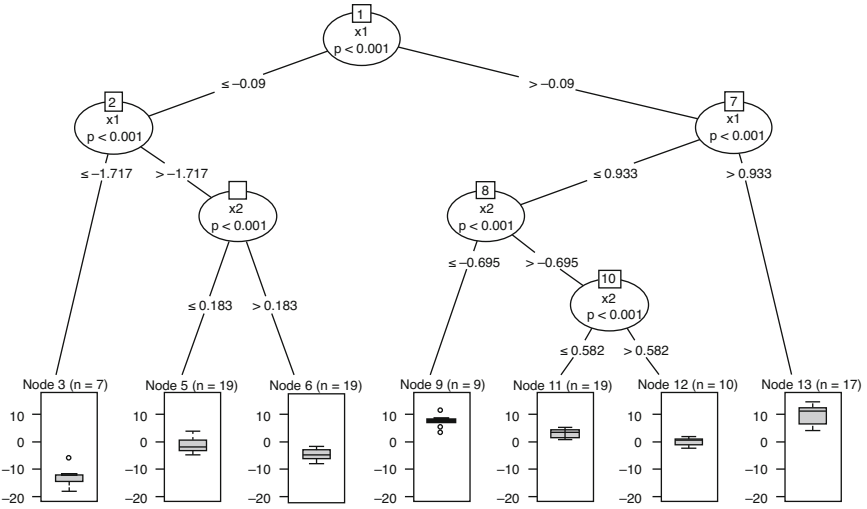

Fig. 25.2 A conditional inference tree model of data with a simple linear additive relationship between two continuous predictors and a continuous response variable

Table 25.3 Distribution of imaginary corpus data of differential argument marking

Role = Subject		
	Response = marked	**Response = unmarked**
Semantics = animate	5	20
Semantics = inanimate	20	5
Role = object		
	Response = marked	**Response = unmarked**
Semantics = animate	19	6
Semantics = inanimate	5	20

Moreover, CITs can run into problems when some predictors have a **strong crossover interaction**. As an illustration, consider Table 25.3, which contains imaginary corpus counts of different subjects and objects in a language with differential subject and object marking. The response variable reflects the presence of case marking: marked or unmarked. There are two interacting predictors of case marking: the semantic class (animate or inanimate) and syntactic role (subject or object). If the argument is a subject, it is usually unmarked when it is animate and marked when it is inanimate. As for objects, it is the other way round, which is why we can speak of a near-perfect crossover interaction.

When one fits a logistic regression model by using the `lrm` function in the `rms` package (Harrell Jr 2017), one obtains the coefficients shown in Table 25.4. The interaction term is highly significant. Figure 25.3, which visualizes the interaction with the help of the package `visreg` (Breheny and Burchett 2016), demonstrates that the effect of semantics is almost perfectly reverse for subjects and objects.

Table 25.4 The coefficient table of a logistic regression model based on the data in Table 25.3. Positive coefficients: increased likelihood of the marked form; negative coefficients: increased likelihood of the unmarked form

Term	Coefficient	SE	Wald Z	P-value
Intercept	−1.39	0.5	−2.77	0.0056
Semantics = inanimate	2.77	0.71	3.92	<0.0001
Role = object	2.54	0.69	3.71	0.0002
Semantics = inanimate: Role = object	−5.31	0.98	−5.4	<0.0001

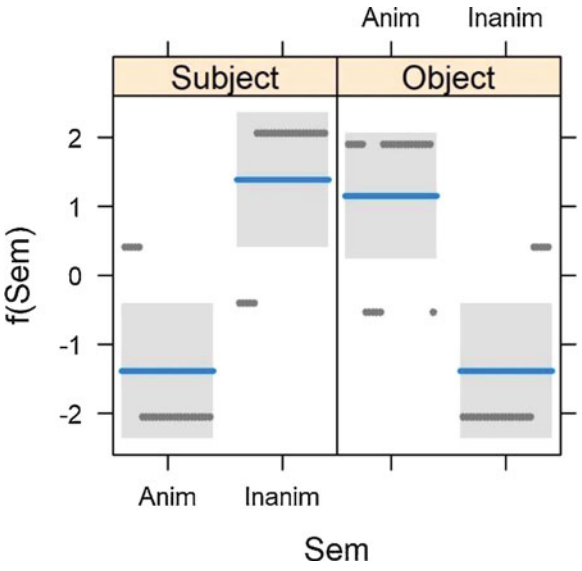

Fig. 25.3 Cross-over interaction between semantics and syntactic role in imaginary corpus data of differential argument marking

In such cases, CITs may be unable to identify any splits. Figure 25.4 displays the result of a CIT analysis with the default settings. It only contains a bar plot with the marginal proportions of the response categories. No splits are made. The method thus fails to uncover the underlying relationships between the predictors and the response.

Moreover, simple CITs may be **unstable** even when small changes in the data are made (Strobl et al. 2009; Kuhn and Johnson 2016: 174), especially if the model contains numerous correlated predictors. This is why individual trees need to be complemented by a random forest analysis, which aggregates the results from many individual trees (see an example in Sect. 25.3).

CRFs have their pitfalls, as well. One of them is **skewed response**. Random forests of any kind do not predict very well when the response is very skewed (although other methods, e.g. logistic regression, can experience problems, as well). For example, if the proportions of two outcomes are 98% and 2%, it is possible to

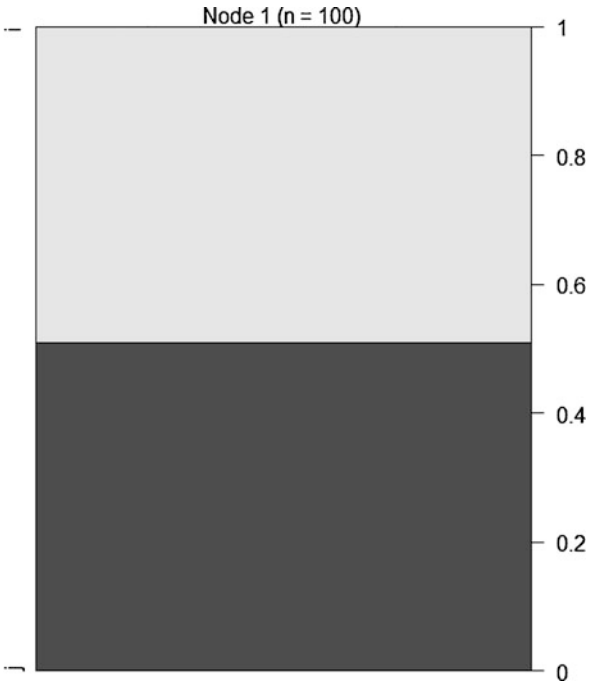

Fig. 25.4 A bar plot representing the tree model with zero splits for the data in Table 25.3

achieve 98% classification accuracy just by assigning the more frequent category. It will be difficult to find the predictors that perform better than this (Berk 2006). As a result, the predicted frequency of the rare category may be zero, i.e. the model will not discriminate between the categories. In such situations, one can try to 'upsample' the small category and/or 'downsample' the large one in order to make the distribution more balanced. Note, however, that the interpretation of the *p*-values in the trees becomes problematic in that case because the statistical power of the independence tests will change in comparison with the original sample.

As mentioned in Sect. 25.2.2, both CITs and CRFs have a problem with **dependent observations**, e.g. multiple examples from the same author, text or corpus segment. In such situations, one normally uses mixed-effects regression modelling (see Chap. 22). Some studies have treated the grouping factor as one of the predictors (e.g. Tagliamonte and Baayen 2012). However, this is not a perfect solution. First, CITs are based on the permutation framework which measures the association between the response and a predictor by permuting the response variable, without taking into account the levels of the grouping factor (see Sect. 25.2.4.1). Therefore, the information about the dependence of the data points is lost. Moreover, it is not clear how to use the same model with the grouping factor as a covariate on new data, where the grouping categories (e.g. texts or subcorpora) will be different. The model will therefore lose generalizability. Finally, as Baayen et al. (2013) demonstrate, the recursive partitioning methods do not perform very well when the grouping factor has many levels. In such cases, a mixed-effects model is the best choice.

Finally, one can have computational problems when growing CRFs and computing conditional variable importance on **very large datasets**. One can run out of memory, or it may take the algorithm a very long time to complete the computations. Moreover, there is danger that the trees may become too large and overfit the data. In that case, it is sometimes recommended to increase the minimum criterion, e.g. from 0.95 to 0.99.

From all this, it follows that CITs and CRFs should be applied in tandem in order to counterbalance their strengths and weaknesses, and preferably in combination with other methods, most importantly, fixed-effects or mixed-effects (generalized) linear regression models (see Chaps. 21 and 22).

Representative Study 1

Tagliamonte, S., and Baayen, R.H. 2012. Models, forests and trees of York English: Was/were variation as a case study for statistical practice. *Language Variation and Change* 24(2):135–178. doi:https://doi.org/10.1017/S0954394512000129**.**

Research question
The study investigates the factors that determine variation between *was* and *were* in plural past tense existential constructions in York English, as in *There was/were a lot of people*.

Data
The data come from a corpus of spoken York English at the turn of the twenty-first century. All plural past tense existential constructions (e.g. *There was/were* + PL noun) were extracted. The dataset contains 489 tokens from 83 individuals. Nine covariates are tested, which represent such social factors as the sex, age and education level of the speakers, and such linguistic factors as polarity, type of determination and proximity of the copula to its referent.

Methods
The paper utilizes fixed-effects and mixed-effects logistic models, as well as CITs and CRFs, where the speakers' IDs are added as a covariate.

Results
The CRFs provides the best prediction. The most important predictors are the speaker's age, polarity, type of determination and proximity. There is also very substantial interspeaker variation. The CIT reveals that linguistic and social factors play a role only for a subset of speakers. For that subset, the non-standard form *was* is more likely to occur in affirmative contexts than in negative ones. Further, the differentiation by age is only relevant in affirmative contexts. Younger speakers are more likely to use *was* than the older speakers.

Representative Study 2

Szmrecsanyi, B., Grafmiller, J., Heller, B., and Röthlisberger, M. 2016. Around the world in three alternations: Modeling syntactic variation in varieties of English. *English World-Wide* 37(2):109–137. doi: https://doi.org/10.1075/eww.37.2.01szm**.**

Research questions
This study focuses on three alternations in different geographic varieties of English: the dative alternation, the genitive alternation and variation in particle placement. The lects include two native varieties (Great Britain and Canada) and two non-native varieties (India and Singapore). The research questions are as follows:

1. Do the varieties of English share a core probabilistic grammar?
2. Is there a split between the native and non-native varieties of English?
3. Do the alternations under study differ in terms of their probabilistic sensitivity to variety effects?

Data
The authors extract tokens of the alternations from the relevant components of the International Corpus of English. In addition, some frequency information was collected from the GloWbE corpus. Relevant predictors were coded manually.

Methods
For each of the three alternations, Szmrecsanyi et al. modelled the data using conditional inference trees. The varieties of English were tested as a categorical variable. Next, they performed conditional random forest analysis and computed the conditional variable importance scores of the predictors. In addition, in order to interpret the results of the CFR analysis of the particle placement, predicted probabilities of the alternating variants were computed for different language varieties.

Results
The results are as follows:

1. The directions of the effects of the contextual variables are stable across the varieties in all three alternations, although there are quantitative differences with regard to the effect size.
2. As for the second research question, the data provide no conclusive results.
3. The particle placement alternation exhibits the most robust variety effects, and the genitive alternation the least. The authors explain this finding by

(continued)

the fact that the particle placement alternation is more tightly associated with specific lexical items (i.e. verb slots), which makes cross-varietal indigenization effects more likely.

25.3 A Practical Guide with R

25.3.1 T/V Forms in Russian: Theoretical Background and Research Question

This case study is a part of a larger project on European T and V politeness forms (Levshina 2017), which represent different degrees of politeness in addressing the Hearer, e.g. French *tu* and *vous*, German *du* and *Sie*, Russian *ty* and *vy*, usually accompanied by a corresponding verb form. This cross-linguistic study is based on the parallel corpus of film subtitles called ParTy.[3] The observations come from several films of different genres (see below). Since standard English has no politeness distinctions in the second person, the translators of subtitles to languages with the T/V distinction have to choose between these forms. This may be a difficult thing to do because the norms of T/V use are multifactorial and fluid.

According to Brown and Gilman (1960), the politeness forms in European languages can be described in terms of two dimensions: power and solidarity. Power means the ability of one person to control the behaviour of another one. In a one-to-one interaction, the participant with greater power addresses the participant with less power using T, while the participant with less power uses V. The power dimension used to play an important role earlier, but has been gradually replaced by solidarity semantics, with T for intimate communication, e.g. between family members and friends, and V for formal communication. Virtually any characteristic, e.g. gender, age, hobbies, political beliefs and even physical appearance (e.g. dreadlocks or tattoos) can be a basis for the perception of solidarity. See Levshina (2017) for an overview of these and more recent ideas. In what follows, we will discuss which factors influence the use of the Russian T and V forms, which are *ty* (second person singular) and *vy* (second person plural), respectively.

[3] Available at https://github.com/levshina/ParTy-1.0. Accessed 22 May 2019.

25.3.2 Data: Film Subtitles

The data for the present study come from online subtitles of nine popular films of different genres. The films are displayed in Table 25.5. The meta-information about the year and genres is taken from the International Movies Database.[4]

The data set for the study was created as follows. First, 228 interactive contexts with the pronouns *you* or *yourself* were identified in the English data. All plural references were excluded. Their translations were found in ten other languages, including Russian. It is important to mention that one should speak about T/V forms, rather than about T/V pronouns because there are many examples when the verb form is the only clue that helps us to distinguish between T and V. Consider an example in (2) from Russian, where the first T form has no explicit pronominal subject, while the second one has both the pronoun and the verb form.

(2) *Duma-ješ,* *ona* *tut?* *Ty* *ošiba-eš-sja.*
think-PRES.2SG she here you.NOM be.mistaken-PRES.2SG-REFL[5]
"Do you really think she's here? You're mistaken."

25.3.3 Variables

The film situations with *you* or *yourself* were coded for 16 variables, which are presented in Table 25.6.

Table 25.5 Films represented in the data set

Film	Year	Genres
Avatar	2009	Action, adventure, fantasy
Black Swan	2010	Drama, thriller
Bridge of Spies	2015	Drama, history, thriller
Frozen	2013	Animation, adventure, comedy
Inception	2010	Action, adventure, sci-fi
Spectre	2015	Action, adventure, thriller
The Grand Budapest Hotel	2014	Adventure, comedy, crime
The Imitation Game	2014	Biography, drama, thriller
The Iron Lady	2011	Biography, drama, history

[4] www.imdb.com. Accessed 22 May 2019.

[5] The abbreviations stand for the following. 2SG: second person singular; NOM: Nominative case; PRES: Present tense; REFL: Reflexive.

Table 25.6 Variables for the case study of T/V forms

Name	Meaning	Values
Relational (dyadic) variables		
Rel_Age	Whether the hearer is older or younger than the speaker	"Same", "Older" or "Younger"
Rel_Power	Whether there is power asymmetry between the participants in general or in the given situation, e.g. a parent and a child, a general and a soldier, a boss and his/her employee	"Greater" (the hearer has power over the speaker), "less" (the speaker has power over the hearer) or "equal"
Rel_Class	The social class difference in the dyad	"Higher" (the hearer belongs to a higher social class than the speaker), "lower" (the hearer belongs to a lower social class than the hearer) or "equal"
Rel_Sex	The sex of the speaker and the hearer	"F_F" (female speaker and female hearer), "F_M" (female speaker and male hearer), "M_F" (male speaker and female hearer) and "M_M" (male speaker and male hearer)
Rel_Circle	The social circle to which the speaker and the hearer belong	"Fam" (family), "Fri" (friends), "Rom" (romantic partners), "Work" (colleagues at work), "Str" (strangers) and "Acq" (acquaintances)
Speaker-related and hearer-related variables		
S_Age, H_Age	The Speaker's age, the Hearer's age	"Child" (younger than 18), "Young" (approximately 18–35), "Middle" (approximately 35–60), "Old" (approximately older than 60)
S_Class, H_Class	The Speaker's social class, the Hearer's social class	"Upper" (top-rank politicians and civil servants, owners of multinational corporations, etc.), "Middle" (white-collar workers, small business owners, military officers, etc.), "Lower" (blue-collar workers, servants, etc.) and "Other" (aliens, animals, as well as gangsters, tramps, prostitutes and other declassed elements)
S_Sex, H_Sex	The Speaker's sex, the Hearer's sex	"M" or "F" (there were no transgenders in the data)

(continued)

Table 25.6 (continued)

Name	Meaning	Values
Variables describing the communicative settings		
Others	The presence of other people who could hear the speaker	"Yes" or "No"
Office	Whether the interaction takes place in an office, a government building, prison, school, etc.	"Yes" or "No"
Before68	Whether the action takes place before 1968	"Yes" or "No"
Place	Where the action of the film takes place	"UK" (in the United Kingdom, "US" in the USA, "Fictive" (in an imaginary world), "Global" (in different parts of the world), "Europe" (somewhere in Europe)
Other variables		
Film	The film	See film titles in Table 25.5.

The dataset and R code (`25_CIT_RF.r`) are provided in the supplementary materials. In order to access the data, the comma-separated file `25_CIT_RF_tv.csv` should be first saved locally in a directory on your computer. Next, you should read it in R as a data frame called `tv`. One of the ways to do so is to choose the file interactively, as shown below.

```
#read the data in R, choosing the file interactively

tv <- read.csv(file = file.choose())
```

25.3.4 Software

At the moment of writing, there are two add-on packages in R, in which conditional inference trees and random forests are implemented. One is `party` and the other one is `partykit`. The latter is a more recent version, which contains a new improved procedure for CITs. There are also some differences in the R syntax. The package `partykit` is still under development, though, and some functionalities available in `party` cannot be used at the moment in `partykit`. For example, one cannot use the Monte-Carlo resampling method of permutation. Only the default asymptotic method can be used. This is why the R code provided in the supplementary materials is based only on the functions from `party`. You will also need two other add-on packages: `Hmisc` and `pdp`. The packages should be first installed, as shown below.

```
install.packages(c("party", "Hmisc", "pdp"))

library(party)

library(Hmisc)

library(pdp)
```

25.3.5 Conditional Inference Tree

In order to fit a CIT, the function `ctree()` should be used:

```
#fit a CIT

tv.cit <- ctree(Form ~  ., data = tv)
```

The code, which uses the default settings, is identical to the following line:

```
#Identical to:

tv.cit <- ctree(Form ~   ., data = tv, controls =

     ctree_control(teststat = "quad",

     testtype = "Bonferroni",

     mincriterion = 0.95,

     minsplit = 20,

     minbucket = 7))
```

The default settings, which are recommended in most cases, can be changed, if necessary, in `ctree_control()`:

- quadratic test statistic `teststat = "quad"`. If necessary, one can use `teststat = "max"`;
- use of *p*-values with the Bonferroni correction `testtype = "Bonferroni"`. Alternatively, one can use `testtype = "MonteCarlo"`, which performs actual reshuffling of the data the number of times specified by `nresample` (9999 by default). It may be useful to try both the default test and run the Monte-

Carlo simulation and compare the results. Note, however, that the permutation may take a while if the dataset is large and the number of replications is high. In principle, it is also possible (but not advisable, unless the user knows well what she or he is doing) to take the *p*-values without the Bonferroni correction (`testtype = "Univariate"`) or to use the test statistics themselves instead of the *p*-values (`testtype = "Teststatistic"`);

- 0.95 as the minimal 1 – *p* value needed to implement a split, defined by `mincriterion = 0.95`. If no *p*-values are computed, then this hyperparameter specifies the minimum score of the test statistic. In some situations, it may be useful to change this setting. For example, fitting a tree with a lower minimal criterion may be useful in a pilot study performed for exploratory purposes;
- minimum 20 observations in a node for a split to be considered, specified by `minsplit = 20`;
- according to the default settings, there should be at least seven observations in one node after a split, `minbucket = 7`.

In order to see the tree, one can use the following simple code:

```
plot(tv.cit)
```

Figure 25.5 shows an individual CIT fitted to the data with the help of `ctree()`.The plot should be interpreted from the top down. The top split (Node 1) is made in the variable *Rel_Circle*. This means that the predictor is the most important one with regard to the use of *ty* and *vy*. The variable selection criteria (i.e. 1 – *p*) can be obtained with the help of the function `node()`. Only the first seven predictors are shown:

```
Check the variable selection criteria for Node 1:

nodes(tv.cit, 1)[[1]]$criterion$criterion

#Film      Rel_Age      Rel_Sex    Rel_Power   Rel_Circle

#0.999800767 0.951730359 0.846906561 0.093920867 0.999999960
#S_Class      H_Class

#0.963888397 0.481392221
```

The data are then split in two subsets. The one on the left includes the situations when the Speaker and the Hearer are friends, family members or romantic partners. These contexts are not split further and constitute together a terminal node (Node 2) with 45 observations. As one can see from the bar plot in that node, the proportion of *ty* is very high in those contexts. The right-hand branch from the top node represents

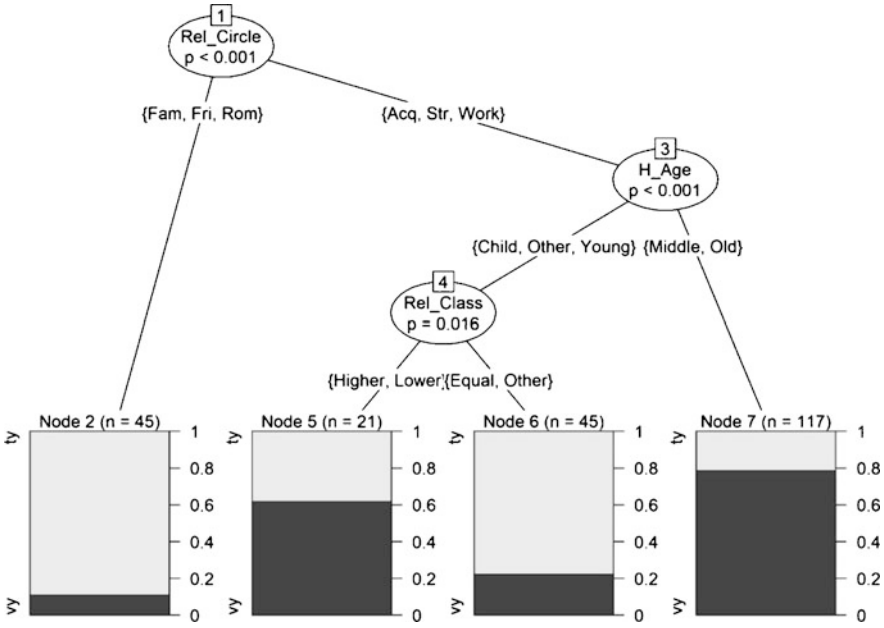

Fig. 25.5 CIT based on the Russian T/V data

the situations when the Speaker and the Hearer are colleagues, acquaintances or strangers. Further splits are made in these contexts. The first split is made in *H_Age* (Node 3). Let us first look at the right-hand branch, which includes middle-aged and old Hearers, and the resulting terminal Node 7. The proportions in the bar plot show that these contexts are characterized by a high chance of *vy*. No more splits are made here. As for the left-hand branch from Node 3, which includes the Hearers who are younger or have no human age at all (like magic creatures), then the relative social class (*Rel_Class*) plays a role, and another split is made (Node 4). If the social classes of the Speaker and Hearer are not equal, then the V form is slightly preferred (see the proportions in Node 5). If the Hearer and the Speaker belong to the same class or the class is not identifiable, the T form is strongly preferred (Node 6).

Since there are two non-final, internal splits in the tree, it may be difficult to interpret them in terms of proportions of T and V. To facilitate the interpretation of those splits, one can create bar plots with proportions in each node, including the inner ones:

```
plot(tv.cit, inner_panel = node_barplot)
```

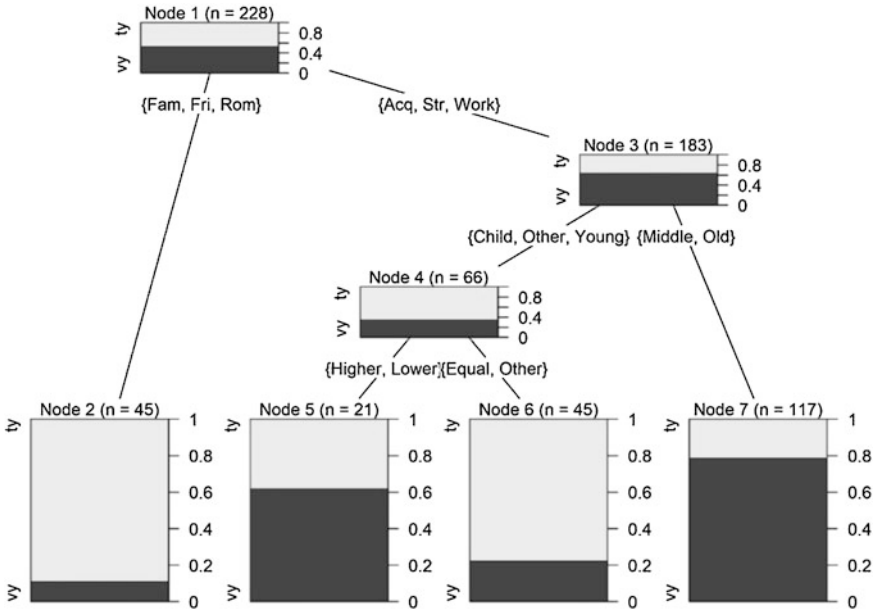

Fig. 25.6 CIT with bar plots in the inner nodes

Such a plot is shown in Fig. 25.6. One can compare, for example, the proportions of T and V after the first split, looking at the terminal Node 2 and the inner Node 3. The bar plots show clearly that the proportion of V is greater in Node 3 than in Node 2.

It is also easy to obtain the proportions of T and V in an inner or terminal node by using the function `nodes()`. For example, for Node 2 these proportions are as follows:

```
#obtain the proportions of T and V in Node 2

nodes(tv.cit, 2)[[1]]$prediction

#[1] 0.8888889 0.1111111 #proportion of T and proportion of V
```

Finally, we need to estimate how well the tree fits the data. A popular measure for classification tasks is accuracy, which is defined as the number of correct predictions divided by the total number of observations.

```
#compute the predicted(fitted) values

pred.cit <- predict(tv.cit)
#cross-tabulate the observed and predicted values

table(pred.cit, tv$Form)

#pred.ctree  ty  vy

#        ty  75  15

#        vy  33 105
#compute the proportion of correct predictions

(75 + 105)/228

#[1] 0.7894737
```

The accuracy is 0.79. Another popular measure, which can be used for binary response variables, is the *C*-index. It shows the proportion of times when the randomly sampled observation with outcome A also has a higher probability of A predicted by the model than a randomly sampled instance of B. It ranges from 0.5 (the model does not discriminate between the outcomes) to 1 (perfect discrimination).

```
#create a vector of predicted probabilities

prob.cit <- unlist(predict(tv.cit, type = "prob"))

               [c(FALSE, TRUE)]
#compute the C-index using a function from the Hmisc package

somers2(prob.cit, as.numeric(tv$Form) - 1)

#          C         Dxy           n     Missing

#  0.8092593   0.6185185 228.0000000   0.0000000
```

The *C*-index of our model is 0.81. One can use the same rule of thumb as the one for logistic regression models. Namely, a model has acceptable discrimination between the response categories if *C* is higher than 0.7, good if it is above 0.8, and excellent if it is above 0.9.

25.3.6 Conditional Random Forest

This subsection demonstrates how one can grow a CRF, compute the conditional variable importance scores, and visualize the partial effects of relevant predictors on the choice between *ty* and *vy*. There are numerous hyperparameters that should be set before a forest is grown:

- `ntree`, which specifies the number of trees in the ensemble (500 by default). According to Strobl et al. (2009), one needs many trees when the number of predictor variables is large, in order to give each variable a chance to occur in enough trees;
- `mtry`, which specifies the number of predictors randomly selected for each individual tree. For some technical reasons, `mtry = 5` by default. One often sees a recommendation to use the square root of the total number of predictors. In the presence of many intercorrelated predictors, is may be useful to try a larger value, so that each variable occurs in a sufficient number of trees, and its importance is not due to some random variation (Strobl et al. 2009);
- sampling with replacement (`replace = TRUE`) or without replacement (`replace = FALSE`). Traditional random forests use bootstrap, i.e. sampling with replacement (see Chap. 24). According to Strobl et al. (2007), only subsampling without replacement can guarantee unbiased variable importance measures. This is particularly important when a) categorical and continuous variables are combined, and b) there are categorical variables with different number of categories;
- various parameters of the individual trees (see Sect. 25.3.5).

These hyperparameters can be specified with the help of different controls. Some default settings are implemented in `cforest_unbiased` and `cforest_classical`, which have the same defaults as `cforest_control`. The default settings in `cforest_unbiased` reproduce the approach in Strobl et al. (2007), who use subsampling without replacement instead of commonly accepted bootstrapping with replacement (`replace = FALSE`). In that case, the algorithm builds a tree based on a random sample 0.632 times the size of the original dataset. The number can be changed with the help of the parameter `fraction`. This approach also uses `teststat = "quad"` and `testtype = "Univariate"`. The behaviour of `cforest_classical`, in contrast, mimics the algorithm in `randomForest` in the eponymous package (Liaw and Wiener 2002) and has the following default settings: `testtype = "max"`, `testtype = "Teststatistic"`, `mincriterion = qnorm(0.9)`, which equals approximately 1.28 (a type of *z*-statistic), and `replace = TRUE`.

In our data, we have categorical variables with the number of values ranging from two to nine. Following the conclusions made by Strobl et al. (2007) about the optimal way of treating such diverse predictors, we use the unbiased approach and will use subsampling without replacement. The default settings of

`cforest_unbiased` are taken, with the exception of `mtry = 4` (the square root of the total number of predictors) and `ntree = 2000`.

```
#set the random seed if you want to reproduce the

#results presented in this chapter

set.seed(61)

tv.crf <- cforest(Form ~  ., data = tv,

          controls=cforest_unbiased(mtry = 4, ntree = 2000))
```

Next, we compute the conditional variable importance scores. By default, the unconditional scores are computed by `varimp()`. To change that, add `conditional = TRUE`.

```
set.seed(23)
#compute the conditional variable importance scores. This may

#take a while, depending on the number of trees

#and the size of the dataset

tv.varimp <- varimp(tv.crf, conditional = TRUE)
#create a dot chart with varimp scores

dotchart(sort(tv.varimp), xlab = "Conditional variable
importance")
#add a vertical line to separate important scores from
#unimportant ones

abline(v = abs(min(tv.varimp)),

          lty = 2, lwd = 2, col = "red")
```

Figure 25.7 displays a dot chart with the sorted values. The horizontal axis represents the conditional variable importance for each predictor, i.e. the average decrease in the OOB prediction accuracy when this predictor is permuted using the conditional approach described in Sect. 25.2.4.2. The red line separates the important scores from the unimportant scores. Note that unimportant predictors fluctuate randomly around zero, with positive or negative values. Negative values appear when the randomly permuted versions of a predictor tend to be better than the original one. The rule of thumb is to take the absolute minimum value as a cut-off point. Note that the importance scores should only be interpreted with regard

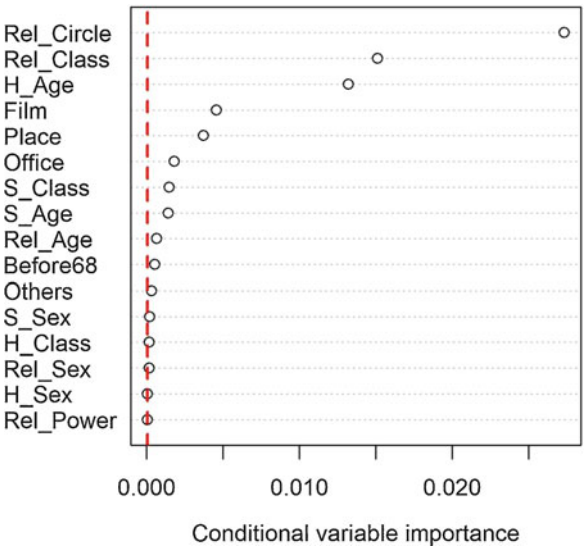

Fig. 25.7 Conditional variable importance scores of the CRF based on Russian T/V data

to their ranking, and not as absolute values. It would be a mistake to compare the scores across different models and data.

The plot in Fig. 25.7 demonstrates that the variable *Rel_Circle* is the most important predictor. It is followed by the difference in the social class (*Rel_Class*) and the Hearer's age (*H_Age*). Recall that these three variables were also the ones that played a role in the CIT. Also, the individual differences between the films (*Film*) play a role. One can attribute that to some pragmatic variables that our course-grained schema does not take into account, or to the translators' individual preferences. The place where the action is developing (*Office* and *Place*) is of some importance, as well, followed by the Speaker's social class and age (*S_Class* and *S_Age*). The age differences (*Rel_Age*), time period (*Before68*) and the presence of others (*Others*) are only marginally important. Interestingly, the sex of the Speaker and Hearer seems to play hardly any role. The individual power relationships do not matter, either.

It is also necessary to evaluate how well the model discriminates between T and V. As was mentioned in Sect. 25.2.4.2, one can use the learning samples and the OOB samples (i.e. those left out during the bootstrap sampling or subsampling). The OOB value is usually more realistic and is closer to the result one can find on new data or during cross-validation. The measures based on the learning samples can be naive and over-optimistic estimates of the error rate (Strobl et al. 2009). By default, the training sample is used. To change that, one should add `OOB = TRUE` in `predict()`.

```
#measures based on the OOB sample

#get predicted (fitted) values

pred.crf.oob <- predict(tv.crf, OOB = TRUE)
#cross-tabulate the predicted and observed values

table(pred.crf.oob, tv$Form)

#pred.rf ty vy

#ty 80 24

#vy 28 96 #compute the proportion of correct predictions
(accuracy)

(80 + 96)/228

#[1] 0.7719298 #compute the predicted probabilities

prob.crf.oob <- unlist(predict(tv.crf,

                    type="prob", OOB=TRUE))[c(FALSE, TRUE)]
#compute C using a function from the Hmisc package

somers2(prob.crf.oob, as.numeric(tv$Form) - 1)

#C         Dxy           n     Missing

#0.8689815   0.7379630 228.0000000   0.0000000
#for comparison, measures based on the learning sample:
pred.crf.train <- predict(tv.crf, OOB = FALSE)
table(pred.crf.train, tv$Form)

#pred.rf  ty  vy

#     ty  92  8

#     vy  16 107 (92 + 107)/228

#[1] 0.872807 #accuracy based on the learning sample
prob.crf.train <- unlist(predict(tv.crf,

                    type = "prob"))[c(FALSE, TRUE)]
somers2(prob.crf.train, as.numeric(tv$Form) - 1)

#C         Dxy           n     Missing

#0.9507716   0.9015432 228.0000000   0.0000000
```

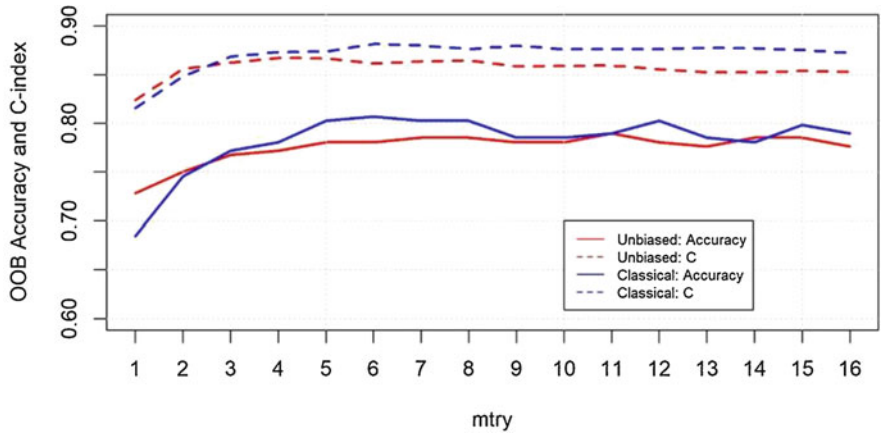

Fig. 25.8 Classification accuracy and *C*-indices of the unbiased and classical methods for different `mtry` based on the TV data

In our case, the learning sample accuracy is 0.87, and the corresponding *C*-index is 0.95. As expected, the OOB measures are lower: the OOB accuracy is 0.77, and the OOB *C*-index is 0.87. The OOB accuracy of the CRF is rather low, even in comparison with the accuracy of the CIT presented in Sect. 25.3.5. This may be due to the difference in the roles of the unbiased approach and the classical method. The unbiased approach with subsampling aims at providing the fairest evaluation of the variable importance, while the classical approach with bootstrapping is particularly good in prediction. Figure 25.8 illustrates the differences in the *C*-values and accuracy scores between the two methods for different values of `mtry`, i.e. the number of randomly sampled predictors. For most values of `mtry`, the classical method outperforms the unbiased approach in terms of predictive power. These results also demonstrate that it is useful to try different values of `mtry` and other settings. It is also recommended to run different models with different random seeds, in order to see whether the results are stable. In our case, as an inspection of several individual models suggests, the top 5 most influential variables remain the same. The differences appear in the order of low-importance predictors. This is a usual result.

25.3.7 Interpretation of the Predictor Effects: Partial Dependence Plots

Unfortunately, the CRF does not return information similar to regression coefficients, which would enable us to interpret the effects of individual variables on the response. Instead, one can use partial dependence plots, which can help to visualize the relationships between different values of the predictors and the response while

accounting for the average effect of the other predictors in the model. The plots shown here are created with the help of the package `pdp` (Greenwell 2017).

```
#Figure 9a: predictor "Rel_Circle"

pdp_circle <- partial(tv.crf, "Rel_Circle", prob = TRUE)

plotPartial(pdp_circle, main = "Rel_Circle")
#Figure 9b: predictor "Office"

pdp_office <- partial(tv.crf, "Office", prob = TRUE)

plotPartial(pdp_office, main = "Office")
```

The left-hand plot in Fig. 25.9 displays the effects of *Rel_Circle* on the probability of *ty*. The vertical axis ('y-hat') corresponds to the average predicted probability of the T form for each value of the predictor. In order to understand better what the y-hat values represent, take the value "Acq" as an example. The algorithm copies the original dataset and replaces all original values of *Rel_Cicle* with "Acq". Next, the predicted values are computed for each of the observations. Finally, the average predicted probabilities are computed. This procedure is repeated for all other values of the predictor (i.e. "Fam", "Fri", etc.). As a result of this procedure, one can obtain the average predicted probabilities that can be directly compared with each other because the remaining predictors in the dataset have the same values. It is also possible to obtain the predicted log-odds of the outcomes (see Chap. 21) instead of probabilities (not shown here).

The plot suggests a cline of (in)formality or intimacy/distance shown in (3):

(3) Family – Friends – Romance – Work – Acquaintances – Strangers

The more to the left, the higher the probability of the T form.

The right-hand plot shows the effect of *Office*, which has not been found on the tree. Being in an office increases the chances of *vy*. Note, however, that the difference is very small.

Other plots, which are not shown here, reveal that the probability of *ty* is higher in *Avatar*, *Black Swan, Frozen* and *Inception* than in the other films. The model also predicts a higher proportion of *ty* when the action takes place in the US or in a fictive world than in the other places. The probability of *ty* is the highest when the Speaker does not belong to any social class, and the lowest when an upper-class person is speaking. As for the Speaker's age, young people, children and magic creatures tend to use *ty* more often than old people, while middle-aged people are the one who are the most likely to use *vy*. The V form is also more probable when in the presence of other people and before 1968.

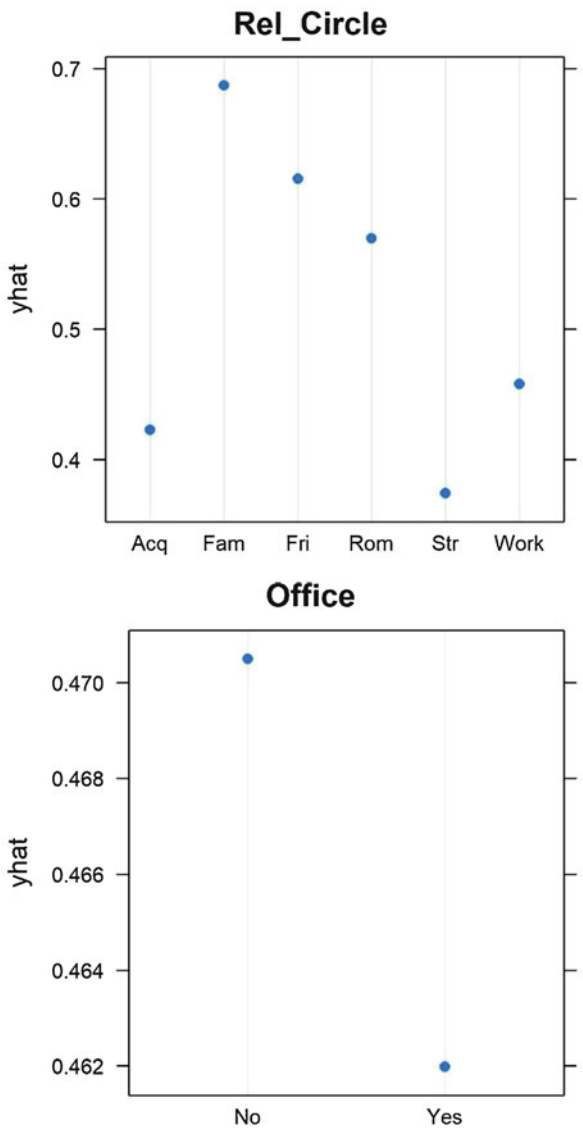

Fig. 25.9 Examples of partial dependence plots. Left: *Rel_Circle*; right: *Office*

25.3.8 Conclusions and Recommendations for Reporting the Results

The case study of T/V forms in Russian has revealed that the solidarity dimension is the strongest one. There is little evidence of the power dimension playing a role. Even the asymmetric variables (e.g. *Rel_Class*) are in fact more related to solidarity than power: if the communicators belong to the same class or do not belong to any

class at all, as magic creatures, the T form is preferred. The translators' perception of different places and time periods as involving more or less formal communication is reflected in the choice of *ty* and *vy*, as well.

When reporting the results, one should include the following information:

- an individual tree (cf. Figure 25.5) and a verbal description of the splits made (see Sect. 25.3.5);
- conditional variable importance scores of the forest (e.g. in the form of a plot, as in Fig. 25.8) and a description (see Sect. 25.3.6);
- measures of predictive accuracy and goodness of fit of the tree and forest (in the latter case, it is preferable to report the statistics based on the OOB samples). For example, one can write the following: "The predictive power of the models is safisfactory. The CIT has the classification accuracy of 0.79 (with the baseline value of 0.51), whereas the concordance index C is 0.81. As for the CRF, its out-of-bag classification accuracy is 0.77, and the out-of-bag C is 0.87."

One also needs to mention in the Methods section the R package and the hyperparameters that were used to grow the trees and forests. For more general info about how to report the results of a quantitative corpus-based study, see also Chap. 26.

To summarize, CITs and CRFs are very flexible and convenient tools that can be used in many situations when traditional parametric methods will fail. They provide easily interpretable results and do not require a tedious check of numerous assumptions. However, these methods may be misleading in some special cases and therefore must be used in a combination with other methods, most importantly, mixed-effects logistic models (see Chap. 23). There remain a few open questions. First of all, we need to have a more generalizable and appropriate way of dealing with the situations when the observations are not independent. Such situations are very common in corpus linguistics. In fact, this is a matter of ongoing research (Torsten Hothorn, p.c.), so hopefully we will see some important innovations in the near future. Second, the software is currently in a state of flux. We still have to wait until all functionalities that are available in the R package `party` are implemented in the optimized R package `partykit`.

Further Reading

Strobl, C., Malley, J., and Tutz, G. 2009. An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods* 14(4):323–348. doi:10.1037/a0016973.

This paper can be useful to those interested in the statistical details. It contains an excellent introduction to recursive partitioning methods. One can find the main principles of recursive partitioning, its advantages and methodological improvements, but also its limitations and pitfalls. The R code is provided, as well.

Kuhn, M. & Johnson, K. 2016. *Applied Predictive Modeling*. New York: Springer.

This book (more specifically, Chap. 8) can be recommended to those who want to get a broader perspective on different popular recursive partitioning methods, including CITs and CRFs. The similarities and differences between these methods are discussed.

Baayen, R.H., Endresen, A., Janda, L.A., Makarova, A., and Nesset, T. 2013. Making Choices in Russian: Pros and Cons of Statistical Methods for Rival Forms. *Russian Linguistics* 37:253–291.

This paper, which deals with functionally similar Russian constructions, compares the advantages and disadvantages of several popular methods (CITs and CRFs, mixed-effects logistic regression models and naïve discriminative learning) along various dimensions, which are important to quantitative linguists, including classification accuracy, cognitive realism and ease of interpretation.

References

Baayen, R. H. (2010). Demythologizing the word frequency effect: A discriminative learning perspective. *The Mental Lexicon, 5*, 436–461. https://doi.org/10.1075/ml.5.3.10baa.

Baayen, R. H., Endresen, A., Janda, L. A., Makarova, A., & Nesset, T. (2013). Making choices in Russian: Pros and cons of statistical methods for rival forms. *Russian Linguistics, 37*, 253–291.

Berk, R. A. (2006). An introduction to ensemble methods for data analysis. *Sociological Methods & Research, 34*(3), 263–295. https://doi.org/10.1177/0049124105283119.

Bernaisch, T., Gries, S. T., & Mukherjee, J. (2014). The dative alternation in south Asian English(es). *English World-Wide, 35*(1), 7–31. https://doi.org/10.1075/eww.35.1.02ber.

Breheny, P., & Burchett, W. (2016). *Visreg: Visualization of regression models*. R package version 2.3–0. https://CRAN.R-project.org/package=visreg. Accessed 22 May 2019.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Brown, R., & Gilman, A. (1960). The pronouns of power and solidarity. In T. A. Sebeok (Ed.), *Style in language* (pp. 253–276). Cambridge, MA: MIT Press.

Greenwell, B. M. (2017). pdp: An R package for constructing partial dependence plots. *The R Journal, 9*(1), 421–436. URL https://journal.r-project.org/archive/2017/RJ-2017-016/index.html. Accessed 22 May 2019.

Harrell, F.E. Jr. 2017. *rms: Regression modeling strategies*. R package version 5.1–1. https://CRAN.R-project.org/package=rms. Accessed 22 May 2019.

Hopper, P., & Thompson, S. A. (1980). Transitivity in grammar and discourse. *Language, 56*, 251–299.

Hothorn, T., & Zeileis, A. (2015). partykit: A modular toolkit for recursive Partytioning in R. *Journal of Machine Learning Research, 16*, 3905–3909. URL http://jmlr.org/papers/v16/hothorn15a.html. Accessed 22 May 2019.

Hothorn, T., Hornik, K., & Zeileis, A. (2006a). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics, 15*(3), 651–674. https://doi.org/10.1198/106186006X133933.

Hothorn, T., Hornik, K., van de Wiel, M. A., & Zeileis, A. (2006b). A Lego system for conditional inference. *The American Statistician, 60*, 257–263. https://doi.org/10.1198/000313006X118430.

Kuhn, M., & Johnson, K. (2016). *Applied predictive modeling*. New York: Springer.

Levshina, N. (2016). Why we need a token-based typology: A case study of analytic and lexical causatives in fifteen European languages. *Folia Linguistica, 50*(2), 507–542. https://doi.org/10.1515/flin-2016-0019.

Levshina, N. (2017). A multivariate study of T/V forms in European languages based on a parallel Corpus of film subtitles. *Research in Language, 15*(2), 153–172. https://doi.org/10.1515/rela-2017-0010.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2*(3), 18–22.

Loh, W.-Y. (2014). Fifty years of classification and regression trees. *International Statistical Review, 82*(3), 329–348. https://doi.org/10.1111/insr.12016.

Lohmann, A. (2013). Is tree hugging the way to go? Classification trees and random forests in linguistic study. *Vienna English Working Papers* 22. https://anglistik.univie.ac.at/research/views/archive/. Accessed 22 May 2019.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random Forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics, 8*, 25. https://doi.org/10.1186/1471-2105-8-25.

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics, 9*, 307. https://doi.org/10.1186/1471-2105-9-307.

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods, 14*(4), 323–348. https://doi.org/10.1037/a0016973.

Szmrecsanyi, B., Grafmiller, J., Heller, B., & Röthlisberger, M. (2016). Around the world in three alternations: Modeling syntactic variation in varieties of English. *English World-Wide, 37*(2), 109–137. https://doi.org/10.1075/eww.37.2.01szm.

Tagliamonte, S., & Baayen, R. H. (2012). Models, forests and trees of York English: Was/were variation as a case study for statistical practice. *Language Variation and Change, 24*(2), 135–178. https://doi.org/10.1017/S0954394512000129.

Wiechmann, D., & Kerz, E. (2012). The positioning of concessive adverbial clauses in English: Assessing the importance of discourse-pragmatic and processing-based constraints. *English Language & Linguistics, 17*(1), 1–23.