Using machine learning for communication classification

Stefan P. Penczynski[*]

May 28, 2018

The present study explores the value of machine learning techniques in the classification of communication content in experiments. Previously human-coded datasets are used to both train and test algorithm-generated models that relate word counts to categories. For various games, the computer models of the classification are able to match out-of-sample the human classification to a considerable extent. The analysis raises hope that the substantial effort going into such studies can be reduced by using computer algorithms for classification. This would enable a quick and replicable analysis of large-scale datasets at reasonable costs and widen the applicability of such approaches. The paper gives an easily accessible technical introduction into the computational method.

Keywords: Communication, classification, machine learning.

JEL Classification: C63, D83, C91

[*] Special thanks go to David Hugh-Jones who introduced me to the method of machine learning. I further thank the editor, two anonymous referees, Ayala Arad, Christian Koch and the audiences at the Mannheim Experimental Workshop and the Thurgau Experimental Meeting 2016 for useful comments. School of Economics, University of East Anglia, Norwich, United Kingdom, S.Penczynski@uea.ac.uk, Tel. +41 1603 59 1796.

1. Introduction

This study investigates the possible contribution of machine learning techniques to the coding of natural language transcripts from experiments. The aim is to evaluate whether simple tools from Natural Language Processing (NLP) and machine learning (ML) provide valid and economically viable assistance to the manual approach of coding even when complex concepts are coded.

In recent years, the analysis of communication has been an increasingly important element of many studies in economics. Communication transcripts are being consulted to understand behavior beyond what can be inferred from choice data and to obtain insights into team deliberation processes (e. g. Cooper and Kagel, 2005; Burchardi and Penczynski, 2014; Goeree and Yariv, 2011; Penczynski, 2016a). Computerized experiments make the collection of communication data very easy. And communication data are potentially very informative about reasoning processes. This strength, however, comes with the natural disadvantage that the coding of text – which is usually done manually – is time-intensive and based entirely on human judgment.[1]

Enabling the assistance of computers in the processing of natural language is the aim of the many different research fields of NLP, such as machine translation, question answering and speech recognition.[2] A basic judgment of texts can be made with the help of simple statistics, such as message counts, word counts and word ranks. Moellers, Normann, and Snyder (2017) fruitfully use those concepts when they experimentally investigate communication in vertical markets. More automated approaches like the Linguistic Inquiry and Word Count program (LIWC) group words in semantic classes such as positive or negative emotions, money, past tense etc. Abatayo, Lynham, and Sherstyuk (2017) analyse communication in cooperation experiments with the help of such software. This automation comes at the cost that "the semantic classes may or may not fit the theory being investigated" (Crowston, Allen, and Heckman, 2012, p. 526). A closer fit with a specific economic theory and a higher level of automation can be achieved when statistical techniques such as ML use manually coded examples to build models of linguistic phenomena, an approach that I follow here.[3]

Machine learning – or statistical learning – is a way of obtaining statistical models for pre-

[1] See Krippendorff (2013) for a general introduction into the methodology of content analysis.

[2] General introductory textbooks of NLP are, for example, Manning and Schütze (1999) and Jurafsky and Martin (2014).

[3] Alternatively, linguists extract meaning from texts by establishing human-developed rules that link text and meaning (Crowston, Allen, and Heckman, 2012).

diction in large datasets. Due to the increasing importance of Big Data and variable selection, ML is making its way into the toolbox of econometricians and applied economists (Varian, 2014). For example, its strong out-of-sample prediction capabilities support causality studies by estimating policy implementation and counterfactuals (Mullainathan and Spiess, 2017). The computational handling of text data leads to datasets with many variables and makes these techniques appropriate.

Across the sciences, text analysis with the help of ML has gotten more popular in recent years. Physicians classify suicide notes and observe that the trained computer model outperforms experienced specialists in suicide predictions (Pestian, Nasrallah, Matykiewicz, Bennett, and Leenaars, 2010). Linguists use ML to sift Twitter for useful information during mass emergencies (Verma, Vieweg, Corvey, Palen, Martin, Palmer, Schram, and Anderson, 2011). Based on large volumes of text such as party programs and speeches, political scientists use ML to locate politicians and parties in the political space, for example in the left-right spectrum (Benoit, Laver, and Mikhaylov, 2009). Similarly, economists have used it to quantify the slant of media (Gentzkow and Shapiro, 2010) or the consequences of transparency rules for central banks (Hansen, McMahon, and Prat, 2014). To my knowledge, this is the first study to investigate this technique's usefulness for experimental text data. A great advantage of experimental data is the fact that the experimenter knows the topic of the chat conversation by designing the decision problem at hand.[4]

The communication transcripts studied here are obtained from implementations of Burchardi and Penczynski's (2014) intra-team communication design in beauty contest, hide and seek, social learning and asymmetric-payoff coordination games. Among the applications in experimental work, the classification of reasoning in terms of the level-k model is certainly one of the more ambitious tasks.

Still, the results are clearly positive and show that the out-of-sample computer classification is able to replicate many results of the human classification. They suggest that in similar or easier classification tasks, computer classification can be a valid option to reduce the additional effort that comes with communication analyses, especially large ones. The following sections will introduce the data and the machine learning techniques that are used. Afterwards, results will be presented for three different applications. The technical appendix introduces the computational method based on an example code.

[4] To the extent that this is not the case and the topic needs to be inferred, the analysis becomes more complex and more similar to Hansen, McMahon, and Prat (2014).

2. Data

All communication transcripts in this study are generated by the intra-team communication protocol that was introduced in Burchardi and Penczynski (2014). Teams of two subjects play as one entity and exchange arguments as follows. Both subjects individually make a suggested decision and write up a justifying message. Upon completion, this information is exchanged simultaneously and both subjects can enter individually a final decision. The computer draws randomly one final decision to be the team's action in the game. The protocol has the advantage of recording the arguments of the individual player at the time of the decision making. Furthermore, the subject has incentives to convince his team partner of his reasoning as the partner determines the team action with 50% chance.

The original communication analyses have two research assistants (RA) – usually PhD or Master students – classify the messages according to a standard procedure of content analysis. From the authors of the study, they are provided written instructions as to which concepts to look for in the text. Initially, they code the messages individually in order not to be influenced by the opinion of the other. Afterwards, they meet or are informed about disagreements and have the chance to revise their classification. Finally, only the coding that the two RAs agree upon is entering the messages' data analysis.

In all analyses of this study, the RAs looked for similar concepts described in the level-k model of strategic reasoning (Nagel, 1995; Stahl and Wilson, 1995). RAs were asked to indicate the lower and upper bound of level of reasoning and in some cases the characteristics of the level-0 belief. Due to a possible ambiguity of messages with respect to the level of reasoning, lower and upper bounds are given that determine the interval within which the level of reasoning is likely to lie.

Here, three datasets will be used to investigate the usefulness of machine learning for the classification. Note that the studies were not chosen based on the particular characteristics of the games, but rather on the kinds of results to be replicated and the content extracted from the text, namely levels of reasoning and level-0 belief characteristics.

First, to see the general features of the computerized level classification, I unite observations from the beauty contest game in Burchardi and Penczynski (2014) with observations from the hide and seek game (Penczynski, 2016b). This dataset is referred to as BCHS.

The second, larger dataset is from a study of social learning (SL, Penczynski, 2017) and allows me to investigate whether one of the main results of the paper, namely that the mode behavior is level-2 (or “naïve inference” as in Eyster and Rabin, 2010), can be found via the computer classification. It features scenarios from the standard social learning framework as introduced by Anderson and Holt (1997).

Finally, the third and largest dataset is from a study of asymmetric-payoff coordination games (APC) as investigated in van Elten and Penczynski (2015) based on games introduced by Crawford, Gneezy, and Rottenstreich (2008, CGR). Beyond the out-of-sample replication of the result that the incidence of level-k reasoning is low in symmetric, pure coordination games and high in asymmetric, “battle of the sexes”-type coordination games, this dataset allows me to go one step further and investigate the classification of level-0 beliefs. Specifically, it can be tested whether the computer classification replicates differences in the relevance of label and payoff salience between symmetric and asymmetric games.

3. Technique

The classification method studied here combines techniques of Natural Language Processing (NLP, section 3.1) and machine learning (ML, section 3.2). Appendix A provides further technical details and annotated example code in the software language R.

3.1. Natural Language Processing

In order to transform a set of natural language messages – a text corpus – into a computer-friendly dataset, the text of each message is represented by a bag-of-words model as a multiset of its words, abstracting from grammar and word order. Specifically, in a process of tokenization, the messages of a corpus are broken down into single strings of letters, numbers, or marks that are divided by a space. Each of the M messages can then be represented by a vector of the frequencies of the T unique tokens.[5] This way, the set of messages is converted into a highly sparse $T \times M$-dimensional, so-called document-feature matrix. Denote the frequency of token t in message m as x_t^m and the vector generated by message m as $\mathbf{x}^m$.

Some measures can be taken to usefully reduce the number of features T. Here this is done by a) removing so-called stopwords, common words that are not indicative of the text content[6], b) reducing inflected words to their stem so that, for example, “team”, “teams” and “teamed”

[5] An alternative to single tokens (unigrams) can be the use of bigrams of two consecutive tokens (or more in n-grams) in order to keep some information on word order and syntax. The use of bigrams has commonly been found of little use, while it increases the number of variables considerably (Verma, Vieweg, Corvey, Palen, Martin, Palmer, Schram, and Anderson, 2011; Pang, Lee, and Vaithyanathan, 2002).

[6] In English, for example, stopwords are “the”, “to”, “and”, “that”, “as”, “about”, “from”, etc.

all appear under "team", and c) dropping tokens that appear rarely in the whole document ($\sum_m x_t^m < 5$). For simplicity and objectivity, I did not remove typos from obviously mistyped words although this could further strengthen the results.[7]

3.2. Machine learning

Due to the large number of independent variables T and the possibly nonlinear relationship between word frequencies and level of reasoning, standard linear regression approaches cannot be used. The statistic method of choice should feature a selection of variables and the ability to represent highly nonlinear relationships. The field of machine learning has available a large variety of algorithms for various purposes. Precedent cases of text analysis with random forests (Agrawal, Gupta, Prabhu, and Varma, 2013), the ease of their implementation and their general usefulness (Varian, 2014) let me choose the random forest technique (Breiman, 2001; Hastie, Tibshirani, and Friedman, 2008, henceforth HTF).[8] It does not require prior calibration and has featured good accuracy and little overfitting across applications.

Machine learning is generally used for out-of-sample prediction, in our case for the prediction of reasoning characteristics based on word counts in messages. The out-of-sample performance can easily and precisely measured and is therefore the deciding measure of the usefulness of a model and guides many if not all of the choices of algorithms and parameters. It is thus indispensable to split the data into two separate sets for training and testing of the model.

For initial analyses and for a very simple linear model that relates the count of a particular token x_t^m to the level of reasoning y^m in message m, $f(x_t^m) = \beta \cdot x_t^m$, I chose to have 70% of the observations to formulate the model in-sample ("train") and the remaining 30% of observations to test the model out-of-sample.

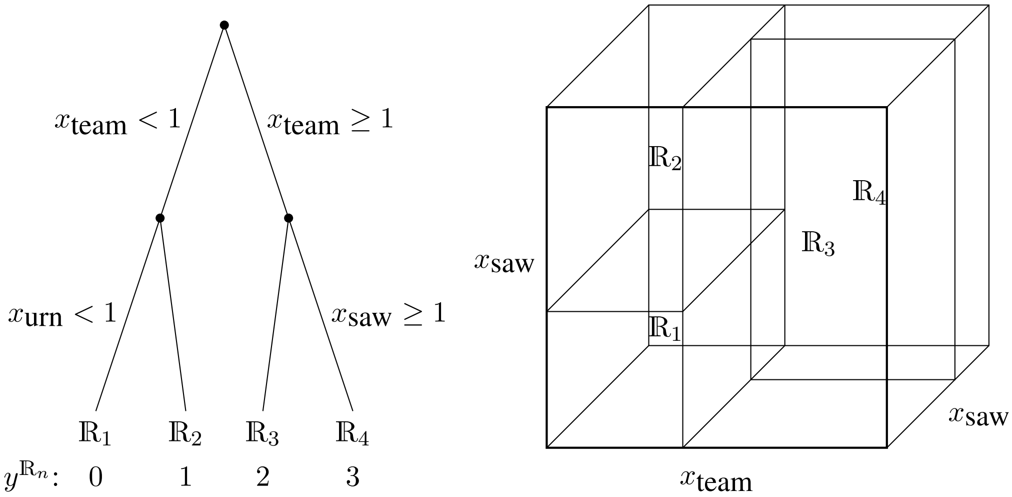
The in-depth evaluation of the random forest results will make use of cross-validation. For 10 consecutive times, a specific 10% subset of the dataset is taken out for testing and the remaining 90% are used for training. The advantage of this more involved process is that eventually all observations will have been predicted based on a model that was trained exclusively on other observations. In all analyses, the in-sample vs. out-of-sample split is balanced

[7] Although increasing the matrix size and not pursued here, it might be useful in some cases to follow linguists' practices and further engage in disambiguation, part-of-speech tagging, adding readability scores, adding number of misspellings, and others (for an example see Pestian, Nasrallah, Matykiewicz, Bennett, and Leenaars, 2010).

[8] The exposition on trees follows section 9.2 of HTF. The introduction to random forests is following section 15 of the same book. An excellent introductory online lecture on machine learning is by Abu-Mostafa (2012). Varian (2014) gives an economist-friendly introduction to machine learning and specifically random forests.

across treatments/games to avoid that results vary due to differences in the number of training observations from particular treatments/games.

As in nature, the concept of a forest is conceptually based on the idea of "trees". Trees partition the space spanned by the independent variables into subspaces. The splits are performed sequentially, dividing a dimension t along a split point s_t into two subspaces, as shown in the illustrative tree and variable space in figure 1. For example, one could divide messages into those with less than one token "team", $x_{\text{team}} < 1$, and messages with more instances of "team", $x_{\text{team}} \geq 1$. The first subspace could be split again by $x_{\text{urn}} < 1$ and $x_{\text{urn}} \geq 1$, the second by $x_{\text{saw}} < 1$ and $x_{\text{saw}} \geq 1$. The online appendix A.4 gives details on how the trees are grown in random forests. To each subsample, one can now associate a level of reasoning $\hat{y}^{\mathbb{R}_n}$, as is done illustratively in figure 1a.



(a) Decision tree.

(b) Partition of messages.

Figure 1: Exemplary decision tree.

Models in machine learning are fundamentally different depending on the nature of the dependent variable. With numerical dependent variables for which differences and means are defined like levels of reasoning, one speaks of a "regression model". When the dependent variable takes a limited number of non-ordered values – discrete variables in economics – one speaks of a "classification model".

A simple regression model reflects the response as a constant c_n in each of the subspaces $\mathbb{R}_n$. The dependent variable y is predicted by

$$f(\mathbf{x}^m) = \sum_n c_n \mathbb{1}(\mathbf{x}^m \in \mathbb{R}_n), \tag{1}$$

and the model error criterion is the mean squared error $Q_{mse} = \frac{1}{M}\sum_m (y^m - f(\mathbf{x}^m))^2$. In our case, the random forest algorithm grows 500 trees. In regression, the prediction for a message m of the collection of 500 trees is the average over all trees' predictions, $f(\mathbf{x}^m) = \frac{1}{500}\sum_{b=1}^{500} f^b(\mathbf{x}^m)$.

In classification models, the mean cannot be used for aggregation of outcomes in the subspaces. The mode outcome can and therefore the aggregation works like a ballot, each of the randomly generated trees casts one vote for its predicted category. The winner of the ballot turns into the model prediction for the message. In each subspace $\mathbb{R}_n$, the proportion of class d messages is $\hat{p}_{nd} = \frac{1}{N_n}\sum_{m:\,\mathbf{x}^m \in \mathbb{R}_n} \mathbb{1}(y^m = d)$. The majority class $d(n)$ in $\mathbb{R}_n$ determines the response that the tree model attributes to a message, that is,

$$f(\mathbf{x}^m) = d(n : \mathbf{x}^m \in \mathbb{R}_n) = \arg\max_d(\hat{p}_{nd} : \mathbf{x}^m \in \mathbb{R}_n). \tag{2}$$

With 500 trees, the majority class d over all 500 trees is the prediction for $\mathbf{x}^m$.

In classification, various error criteria can be conceived. The misclassification error counts the number of misclassified messages and is thus intuitive but not differentiable. I will report the Gini impurity, which gives the error rate not for majority classification, but for a mixture model of classifying a randomly chosen observation in $\mathbb{R}_n$ of category d into category d' with a probability that corresponds to the proportion $\hat{p}_{nd'}$: $Q_{Gini} = \sum_{d \neq d'} \hat{p}_{nd}\hat{p}_{nd'}$. This criterion measures dispersion in the categorization and is 0 if all messages in $\mathbb{R}_n$ fall into one category.

In random forests, many uncorrelated trees are grown and then aggregated. "They can capture complex interactions structures in the data, and if grown sufficiently deep, have relatively low bias. Since trees are notoriously noisy, they benefit greatly from the averaging." (HTF, p. 587f.).

While a single tree as in figure 1a is quite transparent about the modelled relationships, a forest clearly is not. Still, the structure of the model is representable by the so-called variable importance, which tracks over all trees the improvement in the model error thanks to each variable. The higher the reduction in the model error, the more important is the variable for the prediction of the model.

While the level-0 characteristics are discrete variables and hence treated in classification models, the level of reasoning can be treated in either regression or classification models. Given my understanding of levels of reasoning, I would probably see them as categories rather than typical numerical variables. However, in order to also treat and show regression models and results in this paper, I will report both regression and classification results for the levels of reasoning.

4. Results

4.1. Beauty contest and hide and seek games

The beauty contest game (Nagel, 1995) requires players to indicate an integer between 0 and 100, the winner is the player that is closest to 2/3 of the average indicated number. In the hide and seek game, hiders hide a treasure at one of four positions, labelled ABAA (Rubinstein and Tversky, 1993). Seekers can search for the treasure at one position. Whoever holds the treasure at the end wins a prize. The BCHS dataset contains 78 BC and 98 HS messages. I use the rounded average of the agreed-upon lower and upper bounds in the hide and seek game and – for robustness – the rounded average of more than 40 level classifications of the BC dataset obtained on Amazon Mechanical Turk (Eich and Penczynski, 2016).[9]

English stopwords, numbers between 0 and 100, and, due to the game frames, the tokens “a”, “b”, “a’s”, “b’s”, “two”, “third”, “two-third”, “thirds”, “two-thirds”, “half” are excluded from the analysis. Word clouds illustrate the quantified tokens nicely as they indicate more frequent tokens in larger font size. The tokens in the dataset are represented in figure 2.

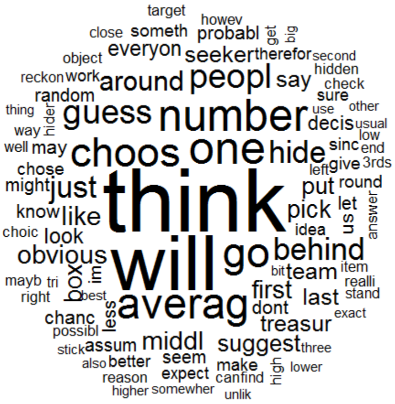
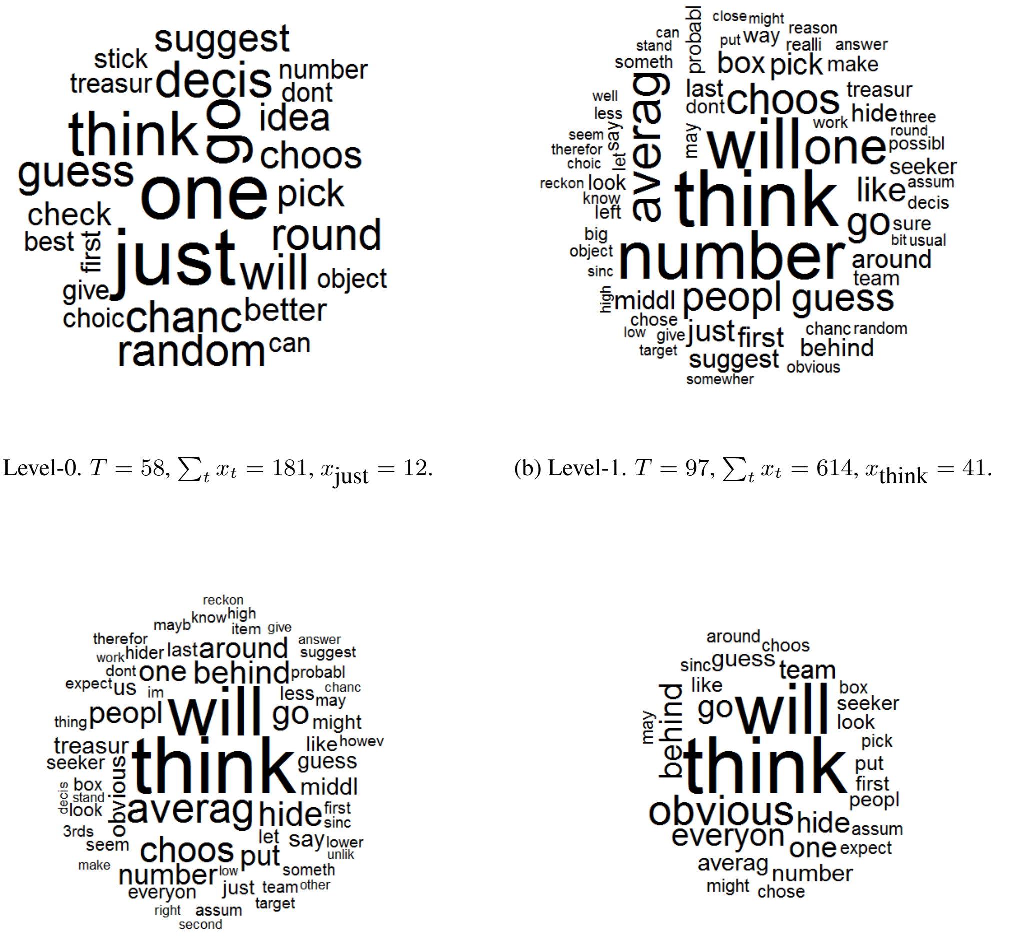


Figure 2: Message tokens in the BCHS dataset. $M = 176$, $T = 98$, $\sum_t x_t = 1605$, $x_{\text{think}} = 127$.

In figure 3, splitting the dataset by the level of reasoning as classified by the RAs gives a first idea whether the content in terms of tokens is different and potentially predictive of the

[9] The results do not change when using lower or upper bounds of the RAs classifications in the beauty contest dataset. Levels are rounded to the integer.

level. Indeed, figure 3a shows for level-0 the words “just” and “one” to be most frequent and others such as “random”, “chance”, or “guess” to come up often. In contrast, higher levels feature words such as “think” and “will” more and more prominently and show fewer instances of “guess” or “random”.



(a) Level-0. $T = 58$, $\sum_t x_t = 181$, $x_{\text{just}} = 12$.

(b) Level-1. $T = 97$, $\sum_t x_t = 614$, $x_{\text{think}} = 41$.

(c) Level-2. $T = 92$, $\sum_t x_t = 568$, $x_{\text{think}} = 50$.

(d) Level-3. $T = 65$, $\sum_t x_t = 228$, $x_{\text{think}} = 25$.

Figure 3: Message tokens in the BCHS dataset by level.

In the BCHS dataset, the frequency of one single token is significantly correlated with the level of reasoning both in- and out-of-sample: “think”. Table 1 reports the correlation coefficients as well as the parameters of the linear model. The R^2 indicates that the word alone accounts for around 48% of the variation in levels.

In a random forest model all tokens are considered. For the two kinds of random forest models, regression and classification, table 2 tabulates the human classification against the

	In-sample		Out-of-sample		Full sample		
t	Corr. coeff.	p-value	Corr. coeff.	p-value	$\hat{\beta}$	s.e.	R^2
“think”	0.404	0.000	0.546	0.002	0.849	0.066	0.48

Notes: p-values are Bonferroni corrected for $T = 98$ simultaneous hypotheses.

Table 1: Bivariate correlations and linear regression between token count and level of reasoning in BCHS.

computer model’s out-of-sample prediction from cross-validation.

$\rho = 0.66$		Human					
$R^2 = 0.80$		0	1	2	3	4	Σ
	0	19	7	1	0	0	27
Comp.	1	19	59	18	3	0	99
	2	1	10	27	11	1	50
	Σ	39	76	46	14	1	176

(a) Random forest regression.

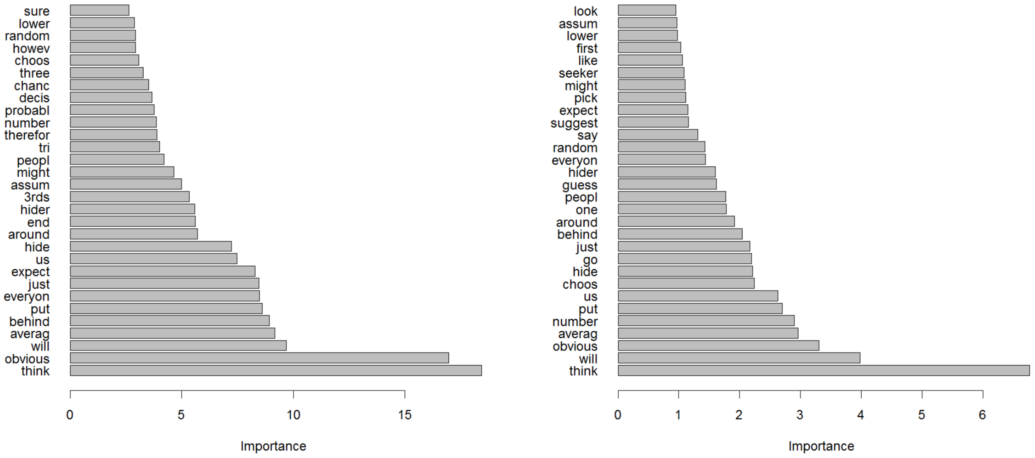
$\rho = 0.53$		Human					
$R^2 = 0.71$		0	1	2	3	4	Σ
	0	11	13	3	0	0	38
Comp.	1	15	51	25	4	0	95
	2	2	12	18	10	1	43
	Σ	39	76	46	14	1	176

(b) Random forest classification.

Table 2: Human classification versus computer prediction from cross-validation in BCHS. ρ gives the correlation coefficient.

In both cases, the computer prediction correlates significantly with the human classification and explains around 71% and 80% of the variation, respectively. The numbers of correctly classified messages, 105 (60%) and 91 (52%), are also significant. In order to test whether the numbers of correctly classified messages could have possibly been obtained by chance, I randomly permute the training levels and observe the number of correctly classified messages 2000 times (Random permutation test, Golland, Liang, Mukherjee, and Panchenko, 2005). For both regression and classification, the numbers 105 and 91, respectively, are above the 99.9th percentile in the resulting distribution. Hence, chance success is rejected with $p < 0.001$.

The structure of the random forest model is illustrated by the importance of the explanatory variables. Figure 4 illustrates the 30 most important tokens in the dataset. Between the two models, the ranking of the most important words is fairly correlated, with the tokens “think”, “will”, “obvious”, and “averag” appearing in the top 4 tokens of both models. Looking back at figure 3, the latter are indeed quite discriminatory, since “obvious” is mainly appearing in level-3 and “averag” is strong in level-1.



(a) Regression model with MSE criterion.

(b) Classification model with Gini criterion.

Figure 4: Variable importance in the BCHS dataset.

Overall, this first analysis on a small and diverse dataset shows that the method can work. The computer classification is not perfect, but it shows promise for larger datasets. In the machine learning literature, the BCHS dataset would be deemed as quite small and in the range where more training datapoints have a positive impact on the prediction performance (HTF).

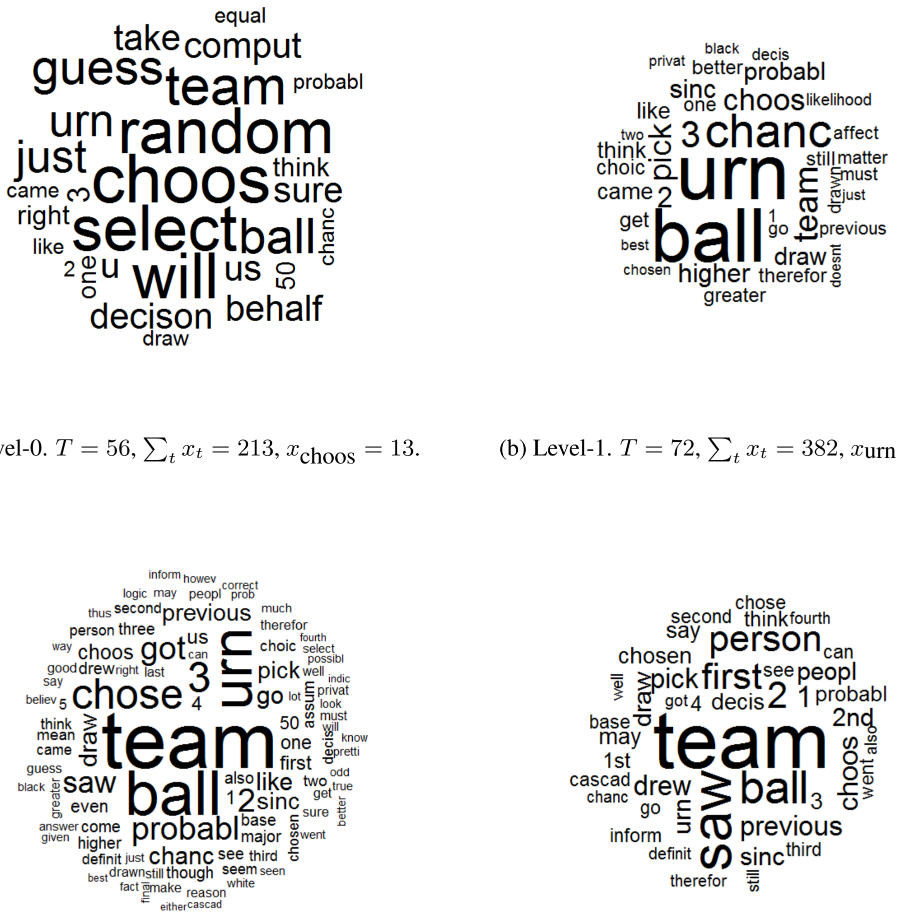
4.2. Social learning

The social learning dataset is taken from Penczynski (2017) and studies the framework introduced by Anderson and Holt (1997). Subjects subsequently receive binary signals (“white”, “black”) about the binary state of the world, A or B, and can observe the decisions of their predecessors in the sequence. Their aim is to match the state of the world with the decision. The private signals are correct with probability 2/3. The dataset contains $M = 348$ messages and their agreed level of reasoning classification from 2 RAs. The messages feature $T = 115$ unique tokens after stemming and disregarding common and rare words.[10]

Figure 5 illustrates the token clouds by level of reasoning. As before, a transition can be noticed, from words such as “choose”, “random”, and “select” in level-0, over “urn” and “ball” in level-1, to a predominant occurrence of considerations including the token “team” in levels 2 and 3. Figure 5 inspired the exemplary decision tree in figure 1a.

In this dataset, there is no single token whose frequency in a message correlates with the level

[10] English stopwords and the tokens “a”, “b”, “a’s”, “b’s”, “as”, “bs”, “A”, “B”, “black”, “white” are excluded from the analysis.



(a) Level-0. $T = 56$, $\sum_t x_t = 213$, $x_{\text{choos}} = 13$.

(b) Level-1. $T = 72$, $\sum_t x_t = 382$, $x_{\text{urn}} = 41$.

(c) Level-2. $T = 115$, $\sum_t x_t = 1986$, $x_{\text{team}} = 178$.

(d) Level-3. $T = 76$, $\sum_t x_t = 340$, $x_{\text{team}} = 31$.

Figure 5: Message tokens in the SL dataset by level.

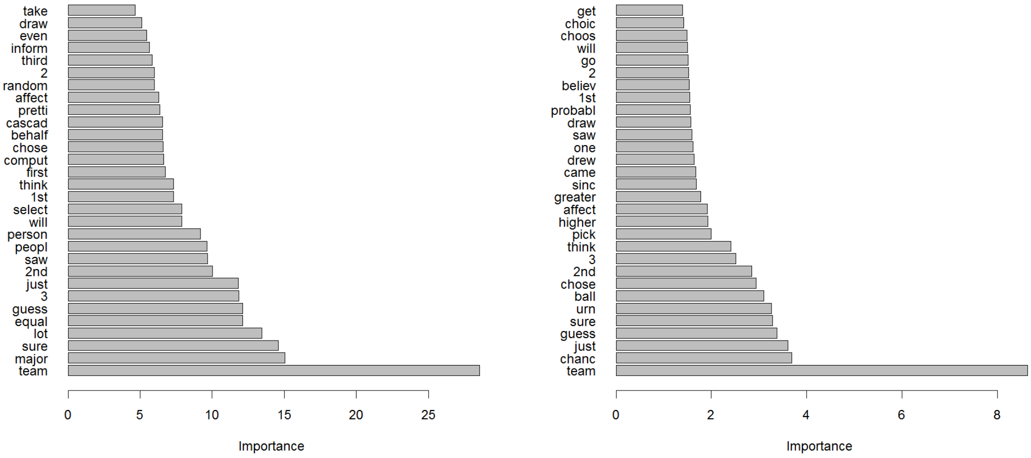
of reasoning both in- and out-of-sample. The strongest correlation and R^2 can be observed with the token "team". Close to the previous dataset, this token accounts for 37% of the outcome variation.

	In-sample		Out-of-sample		Full sample		
t	Corr. coeff.	p-value	Corr. coeff.	p-value	$\hat{\beta}$	s.e.	R^2
"team"	0.367	0.000	0.206	> 0.500	0.950	0.067	0.37

Notes: p-values are Bonferroni corrected for $T = 115$ simultaneous hypotheses.

Table 3: Bivariate correlations and linear regression between token counts and level of reasoning in SL.

In the random forest analyses, the token "team" is turning out to be the most important one in both regression and classification, as figure 6 shows. Further, the tokens “just”, “chance”, and “chose” appear in both models’ top 10 important tokens.



(a) Regression model with MSE criterion.

(b) Classification model with Gini criterion.

Figure 6: Variable importance in the SL dataset.

One of the major results of the original study is the observation that the level of reasoning of the large majority of subjects is 2. In the prediction of the random forest model from cross-validation as shown in table 4, the same conclusion would be drawn from the computer classification. In both regression and classification model, the mode level of reasoning is 2, far ahead of level-1 and level-0.

Here, both models again lead to significant correlation ρ and explain 85% and 88% of the variation. The number of correctly classified messages, 219 (63%) and 239 (69%), is higher than in the BCHS dataset. The random permutation test rejects chance success of that magnitude with $p < 0.001$ in the regression and $p = 0.004$ in the classification.[11] Compared to the BCHS dataset, these numbers show that a larger dataset can improve the percentage of correctly classified messages.

[11] The slightly higher p-value in classification is due to the frequent occurrence of level-2 in the dataset. This increases the probability of chance success of 239 agreeing categorizations.

$\rho = 0.55$		Human				
$R^2 = 0.88$		0	1	2	3	Σ
	0	16	0	4	0	20
Comp.	1	16	35	57	2	110
	2	8	28	167	14	217
	3	0	0	0	1	1
	Σ	40	63	228	17	348

(a) Random forest regression.

$\rho = 0.46$		Human				
$R^2 = 0.85$		0	1	2	3	Σ
	0	18	1	6	0	25
Comp.	1	5	14	15	0	34
	2	17	48	206	16	287
	3	0	0	1	1	2
	Σ	40	63	228	17	348

(b) Random forest classification.

Table 4: Human classification versus computer prediction from the cross-validation in SL. ρ gives the correlation coefficient.

4.3. Asymmetric-payoff Coordination Games

The final dataset in this study results from asymmetric-payoff coordination games (APC) as investigated by Crawford, Gneezy, and Rottenstreich (2008) and van Elten and Penczynski (2015). The challenge here is not only the replication of the result that, roughly speaking, symmetric coordination games lead to significantly lower levels of reasoning than asymmetric ones, but also the test whether characteristics such as level-0 features can be classified. In particular, the analysis of van Elten and Penczynski (2015) showed that asymmetric, "battle of the sexes"-type games predominantly led to payoff salience in the level-0 belief while symmetric, pure coordination games were mostly approached with reference to the salience of the labels.

The dataset consists of $M = 851$ messages and $T = 311$ unique tokens. The analysis uses the agreed upon classification for lower bounds of level of reasoning. Similar results are obtained for the upper bounds or averaged bounds. Table 5 describes the 4 X-Y games and 4 Pie games. In contrast to payoff-symmetric games (in bold), payoff-asymmetric games feature a higher coordination payoff π for one of the two players, depending on the action on which they coordinate. The miscoordination payoff is 0 for both players. The choice is between letters X and Y in the X-Y games and between 3 pie slices (L, R, B) which are identified by ($, #, §) and of which B is uniquely white.[12]

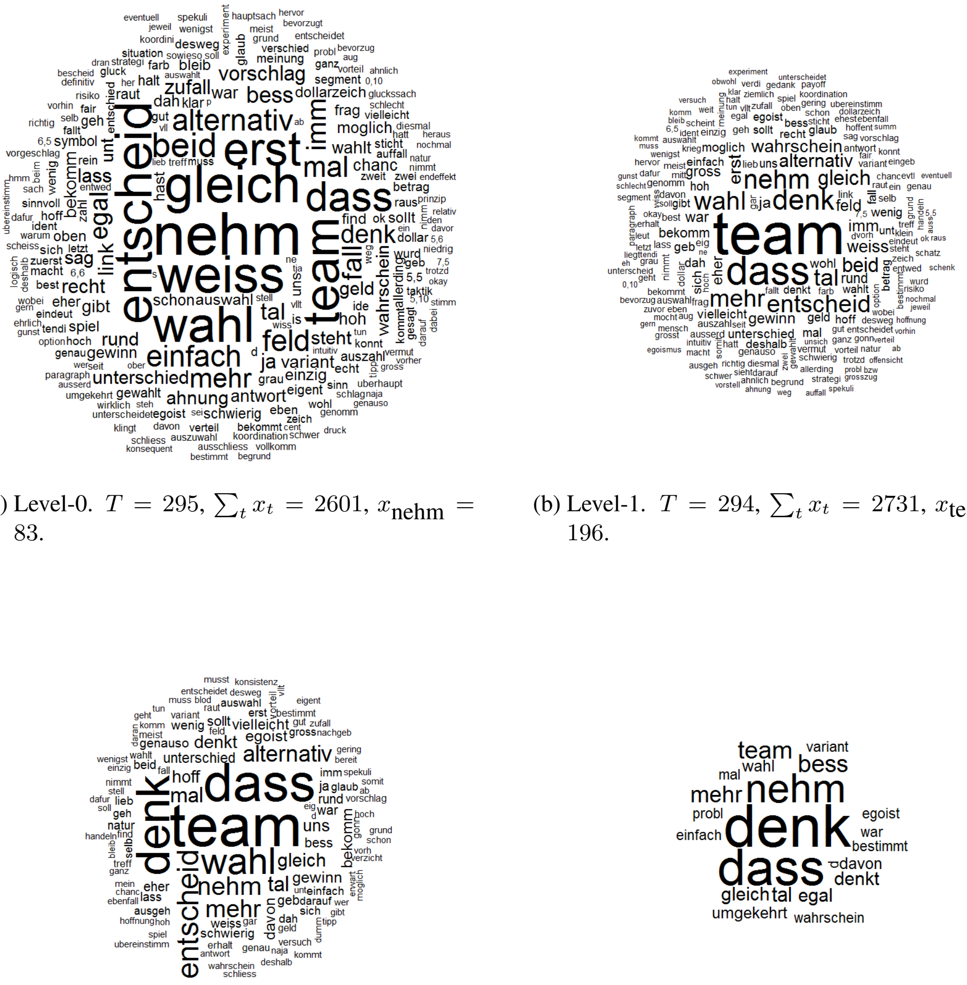
Table 5: Payoff structure of coordination games.

X-Y games (CGR notation)	a	π_1, π_2	Pie games (CGR notation)	a	π_1, π_2
Symmetric Payoffs (**SL**)	X	5, 5	Symmetric Payoffs (**S1**)	L ($)	5, 5
	Y	5, 5		R (#)	5, 5
				B (§)	5, 5
Slight Asymmetry (ASL)	X	5, 5.1	Symmetric Payoffs (**S2**)	L ($)	6, 6
	Y	5.1, 5		R (#)	6, 6
				B (§)	5, 5
Moderate Asymmetry (AML)	X	5, 6	Moderate Asymmetry (AM2)	L ($)	5, 6
	Y	6, 5		R (#)	6, 5
				B (§)	6, 5
Large Asymmetry (ALL)	X	5, 10	Moderate Asymmetry (AM4)	L ($)	6, 7
	Y	10, 5		R (#)	7, 6
				B (§)	7, 5

[12] German stopwords, numbers between 1 and 100, and the tokens "x", "y", "#", "$", "§" are excluded from the analysis.

4.3.1. Levels of reasoning

As before, figure 7 shows the most common tokens by the level of reasoning of the containing message. The experiment communication is in German.[13] As before, one can see a characteristic transition from level-0 to level-3. While take (“nehm”), white (“weiss”), same (“gleich”), first (“erst”) are some of the most common tokens in level-0, the levels 1 and 2 feature most prominently “team” and that (“dass”). The incidence of think (“denk”) is steadily rising in levels 1 and 2, becoming the most common token in level-3.



(a) Level-0. $T = 295$, $\sum_t x_t = 2601$, $x_{\text{nehm}} = 83$.

(b) Level-1. $T = 294$, $\sum_t x_t = 2731$, $x_{\text{team}} = 196$.

(c) Level-2. $T = 239$, $\sum_t x_t = 1363$, $x_{\text{team}} = 97$.

(d) Level-3. $T = 95$, $\sum_t x_t = 234$, $x_{\text{denk}} = 24$.

Figure 7: Message tokens in the APC dataset by level.

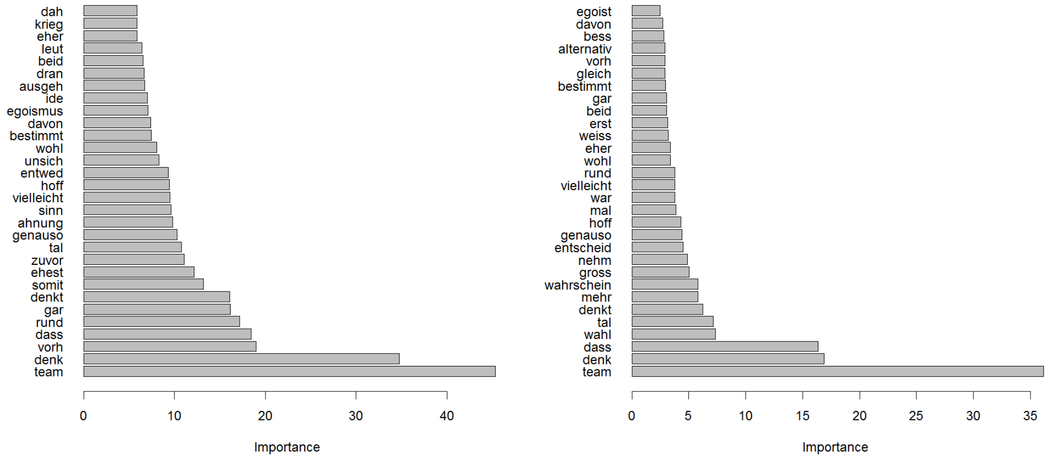
[13] For the computer algorithm, the language of the messages is irrelevant. The training data are generated by RAs that understand the language. Only the compilation of the dataset in R, such as the dropping of stopwords or the stemming of words is easier for common languages. Packages for them are readily available.

Table 6 shows the 5 of the 100 most frequent tokens whose frequencies in messages correlate significantly with the level of reasoning in- and out-of-sample. Among those are two related ones, “denk” and “denkt”, which surprisingly are not pooled during stemming. Again, for objectivity, I do not correct for this manually. The correlations and R^2 reach similar levels as in earlier data and suggest that the token count can again help predict the level of reasoning. Figure 8 shows that these tokens are among the most important variables for the random forest models.

		In-sample		Out-of-sample		Full sample		
	t	Corr. coeff.	p-value	Corr. coeff.	p-value	$\hat{\beta}$	s.e.	R^2
team	“team”	0.198	0.000	0.215	0.050	0.809	0.073	0.12
that	“dass”	0.429	0.000	0.273	0.001	0.777	0.040	0.31
think	“denk”	0.458	0.000	0.510	0.000	0.811	0.039	0.33
us	“uns”	0.303	0.000	0.225	0.026	0.733	0.048	0.22
think	“denkt”	0.282	0.000	0.231	0.018	1.447	0.144	0.10

Notes: p-values are Bonferroni corrected for 100 simultaneous hypotheses.

Table 6: Bivariate correlations and linear regressions between word counts and level of reasoning.



(a) Regression model with MSE criterion.

(b) Classification model with Gini criterion.

Figure 8: Variable importance in the APC dataset.

Table 7 shows the predicted levels for the random forest models. While the correlation between human and computer classification is high and above 0.5, the R^2 is lower than in the previous analyses. The reason is that the computer has difficulties identifying level-2 or higher players, recognizing only 41 and 54, respectively, out of 122. Both models feature an amount of correctly identified messages, 568 (67%) and 536 (63%), similar as in the SL dataset.[14] The question why the performance is not better than in the smaller SL dataset cannot be answered with the data at hand. It could be the different games, the heterogeneity among the 8 coordination games, the German language, a natural limit of a bag-of-words model etc.

Probably due to the numerical nature of the dependent variable and the role of averaging, the regression model identifies many more level-1 players than the classification model or the human classification. A similar but smaller effect can be seen in the SL dataset. I choose the classification model for the following analysis.

$\rho = 0.61$		Human					
$R^2 = 0.64$		0	1	2	3	4	Σ
	0	298	53	9	0	0	360
Comp.	1	127	219	94	9	1	450
	2	4	12	19	5	1	41
	Σ	429	284	122	14	2	851

(a) Random forest regression.

$\rho = 0.55$		Human					
$R^2 = 0.56$		0	1	2	3	4	Σ
	0	344	68	19	2	0	433
Comp.	1	79	198	77	9	1	364
	2	6	18	26	3	1	54
	Σ	429	284	122	14	2	851

(b) Random forest classification.

Table 7: Human classification versus computer prediction from cross-validation. ρ gives the correlation coefficient.

To conclude the analysis of the level of reasoning, let us take a look at the level predictions by game. Table 8 shows the average level of reasoning in the human and computer classifications and the difference Δ between the two. The reduced ability of the computer to identify level-2 players shows most strongly in the asymmetric games. There, the difference Δ is on average -0.19. Importantly, however, the ranking of games in terms of level averages is very similar between human and computer classification. Both feature lower absolute levels in symmetric games SL and S1 on the one side and higher levels in asymmetric games on the other. Despite the reduced identification of higher level players, the computer classification indicates qualitatively similar level differences between games.

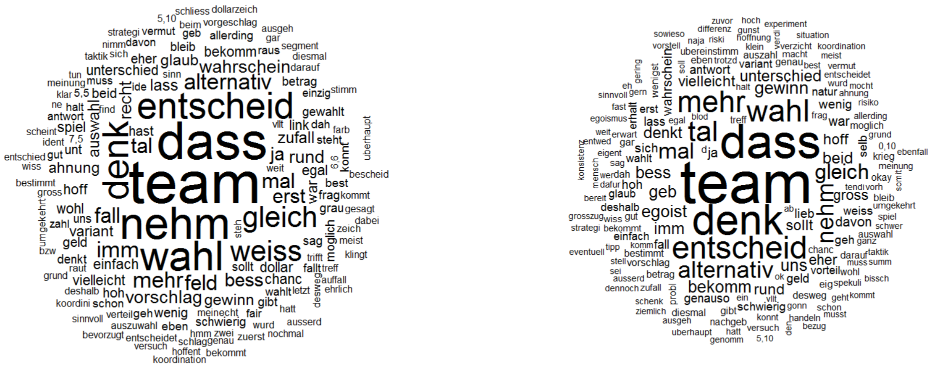
[14] Again, the random permutation test rejects chance success with $p < 0.001$ in both regression and classification.

		Human	Computer	Δ
X-Y	**SL**	0.27	0.26	-0.01
	ASL	1.03	0.78	-0.25
	AML	1.03	0.74	-0.29
	ALL	1.00	0.81	-0.19
Pie	**S1**	0.28	0.18	-0.10
	S2	0.49	0.55	0.07
	AM2	0.64	0.45	-0.19
	AM4	0.69	0.66	-0.03

Table 8: Level averages of human and computer classifications by APC game.

4.3.2. Level-0 salience

The level-0 salience in the APC games can be divided into payoff and label salience. For both, I use the classification model of the random forest method since the attitudes towards salience are non-numerical categories. Payoff salience implies that subjects mention a belief as to how their opponent reacts to the asymmetric payoffs. Figure 9 shows the most frequently used tokens by the two most important categories, "no salience" and "high payoff". There are no striking differences across categories, in both the token "team" is most frequent, although it appears more often in "high payoff".



(a) No payoff salience.
$T = 273$, $\sum_t x_t = 1477$, $x_{\text{team}} = 70$.

(b) High payoff salience.
$T = 261$, $\sum_t x_t = 2355$, $x_{\text{team}} = 156$.

Figure 9: Message tokens in the APC dataset by payoff salience.

Table 9 illustrates the prediction of the classification model based on the 5 payoff-asymmetric games. Out of 534 observations, 353 are classified correctly (66%), a substantial amount.[15]

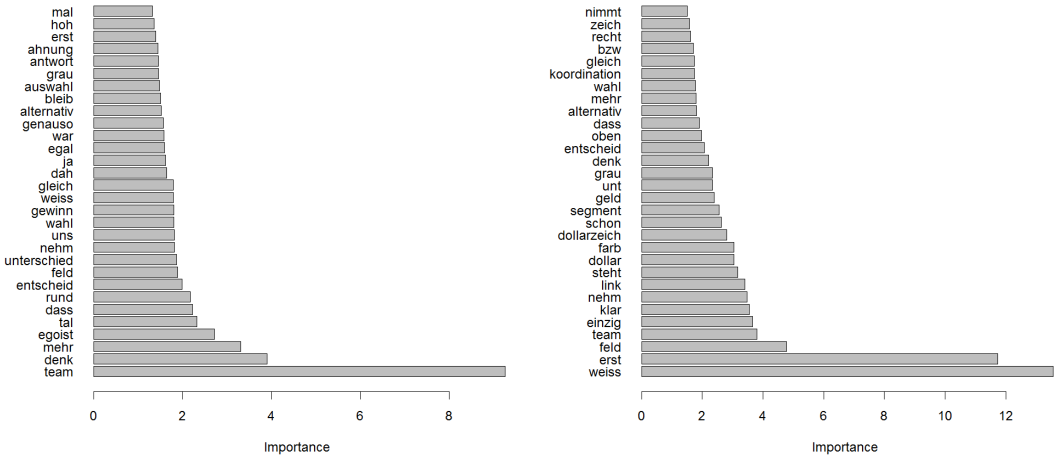
The important tokens for the classification model are illustrated in figure 10a. Compared to

[15] The random permutation test rejects chance success with $p < 0.001$.

		Human				
		no salience	indifference	high payoffs	low payoffs	Σ
	no salience	80	6	49	0	135
Comp.	indifference	0	0	0	0	0
	high payoffs	121	4	273	1	399
	low payoffs	0	0	0	0	0
		201	10	322	1	534

Table 9: Human payoff salience classification versus computer prediction from cross-validation.

the important tokens in the model for the level of reasoning, the notable difference lies in the importance of more (“mehr”), egoistic (“egoist”) and taler (“tal”), which is plausible for the payoff salience. The token “team” stays relevant since payoff salience is correlated with higher level messages that feature this token more often than lower level messages.

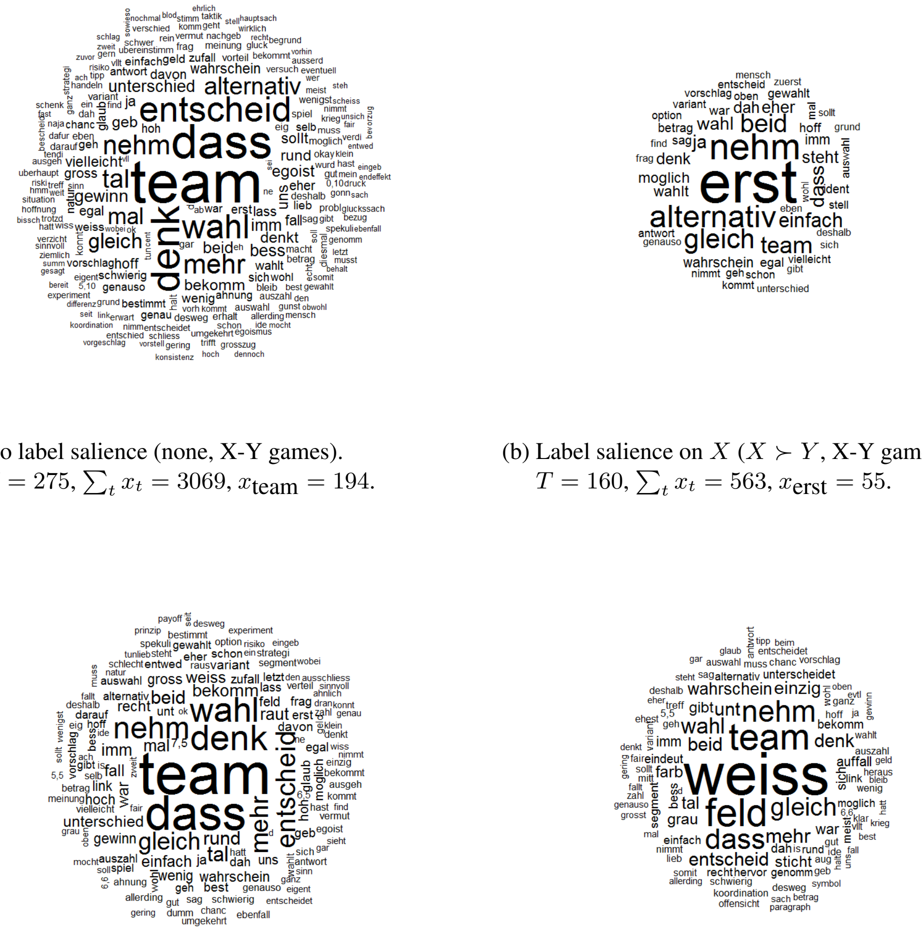


(a) Payoff salience (payoff-asymmetric games).

(b) Label salience (all games).

Figure 10: Variable importance in the APC dataset. Classification model with Gini criterion.

Label salience implies that participants are attracted or averse to actions due to a salient label in the game, which improves the coordination probability. Figures 11a and 11b illustrate the most frequently used tokens for X-Y games by label salience category. It is telling that the "label salience on X" category ($X \succ Y$) features the token first ("erst") most frequently, a term that alludes to the first position of the X in the displayed action space (11b). Similarly for the Pie games in figures 11c and 11d, the latter features white ("weiss") most prominently.



(a) No label salience (none, X-Y games).
$T = 275$, $\sum_t x_t = 3069$, $x_{\text{team}} = 194$.

(b) Label salience on X ($X \succ Y$, X-Y games).
$T = 160$, $\sum_t x_t = 563$, $x_{\text{erst}} = 55$.

(c) No label salience (Pie).
$T = 271$, $\sum_t x_t = 1428$, $x_{\text{team}} = 87$.

(d) Label salience on White (Pie).
$T = 219$, $\sum_t x_t = 1062$, $x_{\text{weiss}} = 82$.

Figure 11: Message tokens in the APC dataset by label salience.

In terms of the prediction of the label salience, with the example of games SL and ALL, table 10 shows that differences between games can be detected in the computer classification. While in the symmetric game SL 37 subjects are classified to hold a belief of preference for

X (table 10a), only 3 are classified to hold such a belief in the asymmetric game ALL (table 10b). In both games, the computer classification is close to the human classification with 74 out of 105 (70%) on the diagonal in SL and 99 out of 104 (95%) in ALL. Recall that the model is not trained in a game-specific way, but trained with a balanced number of observations from all games.[16]

In figure 10b, the important tokens for a joint model in X-Y and Pie games clearly relate to the level-0 label salience: white, first, and field. I conclude that the computer classification is indeed able to indicate differences in level-0 belief characteristics.

		Human			
		none	$X \succ Y$	$Y \succ X$	Σ
Comp.	none	37	26	1	64
	$X \succ Y$	4	37	0	41
	$Y \succ X$	0	0	0	0
	Σ	41	63	1	105

(a) Payoff-symmetric game SL.

		Human			
		none	$X \succ Y$	$Y \succ X$	Σ
Comp.	none	96	2	0	98
	$X \succ Y$	3	3	0	6
	$Y \succ X$	0	0	0	0
	Σ	99	5	0	104

(b) Payoff-asymmetric game ALL.

Table 10: Human classification versus computer prediction from cross-validation.

5. Economic viability and discussion

An important aspect of the presented coding exercise is its economic viability for a research project. What would be the costs and benefits of implementing machine learning?

Regarding the costs, the requirement of a training dataset implies that the manual coding effort cannot fully be substituted. For small projects of the size of the ones treated here, it is unlikely that the effort of manual coding can be reduced by a lot. However, for larger projects, the time and money spent on manual coding can be capped at, say conservatively, the effort of coding 1000 messages. In the context of involved applications such as the ones treated here, this should provide sufficient training data.

[16] For the entire APC dataset, a random permutation test rejects chance success with $p < 0.001$. The same is true for the individual ALL game. In the individual SL game, chance success cannot be rejected due to the specific realisation of a near 50-50 split in the frequencies of categories "none" and "$X \succ Y$". The rejection of chance success in the general APC dataset and in other APC games so far is taken as strong evidence that chance success would be limited here if other distributions had realized in this particular instance.

For example, the extrapolated cost of coding 1000 messages were at the time about 180 Euros and 12 RA student hours. With experimental datasets becoming larger as scientific standards improve and costs of experiments decrease – due to platforms such as Amazon MTurk – the mentioned cap can be valuable. Coding 10000 messages would have resulted in a cost of 1800 Euros and 120 RA student hours, a significant dent in the project's money and time budget.

Beyond the availability of a training dataset, the costs of implementing machine learning as I present it here are relatively low. The software environment R as well as the required packages are freely available. Machine learning methods are quickly absorbed by quantitatively trained economists. Based on the exposition and references here as well as the example in appendix A, I estimate that 3-5 researcher hours are enough to generate a first computer coding output. The statistical training of the model implies that the expertise of a linguist or NLP-trained analyst is not needed (Crowston, Allen, and Heckman, 2012).

For large projects and for researchers that work frequently with text, these numbers suggest that the investment in machine learning expertise is highly economical. Some future developments might shift these numbers further in favor of the investment.

Economists work with a finite set of concepts to be looked for in text. Linguists have developed off-the-shelf tools like sentiment analysis which do not need further training data and thus work without human coding. It is thus conceivable to eventually have enough training data and validated models for off-the-shelf tools that code strategic sophistication, lying aversion, social preferences, etc. Already now, the body of coded text and messages is considerable and could be used as manually-coded training data.[17]

Certainly, more research is required to understand the scope of applications and research questions that can be investigated on this way. Since the present study investigates a rather complex phenomenon of strategic sophistication and aspires to code the degree of this sophistication, I view it as a relatively strong test of the feasibility of machine coding. The estimates given in the context of economic viability should be applicable in other coding tasks and possibly understate the benefits. Other concepts that have been studied with communication such as strategicness in Cooper and Kagel (2005) or the extent of social conversation in Abatayo, Lynham, and Sherstyuk (2017) are probably more easily coded in general, both manually and by machine coding.

An important facilitator in the current study is the researcher's knowledge of the topic of

[17] Linguists are working on tools generally useful for social scientists, including machine learning supported manual coding (Yan, McCracken, and Crowston, 2014).

discussion. In studies with field data, the topic of a text needs to be found out first (Hansen, McMahon, and Prat, 2014), which imposes further costs of analysis. The control in laboratory studies makes the topic of conversations in a given text to be generally set by the game and thus known by the experimenter. This control makes it particularly simple for experimentalists to use the method shown here for coding.

More work is also required to understand possible differences between human and machine classification. Certainly, the conversion of text data into, here, a document-feature matrix risks losing information that is relevant for the theory at hand. Further, since the machine learning coding cannot easily be reconstructed and intuitively understood, it will require more studies and the input of linguists to clearly see the possibilities and limitations of machine classification.

Finally, machine classification might not substitute but rather complement human classification. Existing or to-be-established off-the-shelf models for the coding of specific economic concepts can add evidence or a new perspective beyond the manual classification, as is done, for example in Moellers, Normann, and Snyder (2017); Abatayo, Lynham, and Sherstyuk (2017). Other than reduced labor costs and reduced time of analysis, the computer approach further has the potential to improve consistency where extended coding or the use of multiple coders jeopardizes consistency. Further, the establishment of standard methods would have the potential to improve the acceptance of rigorous qualitative analyses by sceptical quantitatively-minded economists.

References

ABATAYO, A. L., J. LYNHAM, AND K. SHERSTYUK (2017): "Facebook-to-Facebook: online communication and economic cooperation," Applied Economics Letters, pp. 1–6.

ABU-MOSTAFA, Y. (2012): "Learning From Data," `https://www.youtube.com/watch?v=mbyG85GZ0PI`.

AGRAWAL, R., A. GUPTA, Y. PRABHU, AND M. VARMA (2013): "Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages," in Proceedings of the 22nd international conference on World Wide Web, pp. 13–24. ACM.

ANDERSON, L. R., AND C. A. HOLT (1997): "Information Cascades in the Laboratory," American Economic Review, 87(5), 847–62.

BENOIT, K., M. LAVER, AND S. MIKHAYLOV (2009): "Treating words as data with error: Uncertainty in text statements of policy positions," American Journal of Political Science, 53(2), 495–513.

BREIMAN, L. (2001): "Random forests," Machine learning, 45(1), 5–32.

BURCHARDI, K. B., AND S. P. PENCZYNSKI (2014): "Out of your mind: Eliciting individual reasoning in one shot games," Games and Economic Behavior, 84(1), 39 – 57.

COOPER, D. J., AND J. H. KAGEL (2005): "Are Two Heads Better than One? Team versus Individual Play in Signaling Games," American Economic Review, 95(3), 477–509.

CRAWFORD, V. P., U. GNEEZY, AND Y. ROTTENSTREICH (2008): "The Power of Focal Points Is Limited: Even Minute Payoff Asymmetry May Yields Large Coordination Failures," American Economic Review, 98(4), 1443–1458.

CROWSTON, K., E. E. ALLEN, AND R. HECKMAN (2012): "Using natural language processing technology for qualitative data analysis," International Journal of Social Research Methodology, 15(6), 523–543.

EICH, T., AND S. P. PENCZYNSKI (2016): "On the replicability of intra-team communication classification," Discussion paper, University of Mannheim.

EYSTER, E., AND M. RABIN (2010): "Naïve Herding in Rich-Information Settings," American Economic Journal: Microeconomics, 2(4), 221–43.

GENTZKOW, M., AND J. M. SHAPIRO (2010): "What drives media slant? Evidence from US daily newspapers," Econometrica, 78(1), 35–71.

GOEREE, J. K., AND L. YARIV (2011): "An experimental study of collective deliberation," Econometrica, 79(3), 893–921.

GOLLAND, P., F. LIANG, S. MUKHERJEE, AND D. PANCHENKO (2005): "Permutation Tests for Classification," Learning Theory, pp. 36–39.

HANSEN, S., M. MCMAHON, AND A. PRAT (2014): "Transparency and deliberation within the FOMC: a computational linguistics approach," Discussion paper, Centre for Economic Performance, London School of Economics and Political Science.

HASTIE, T., R. TIBSHIRANI, AND J. FRIEDMAN (2008): The elements of statistical learning, vol. 2. Springer series in statistics Springer, Berlin.

JURAFSKY, D., AND J. H. MARTIN (2014): Speech and language processing, vol. 3. Pearson London:.

KRIPPENDORFF, K. (2013): Content analysis: An introduction to its methodology. Sage.

MANNING, C. D., AND H. SCHÜTZE (1999): Foundations of statistical natural language processing. MIT press.

MOELLERS, C., H.-T. NORMANN, AND C. M. SNYDER (2017): "Communication in vertical markets: Experimental evidence," International Journal of Industrial Organization, 50, 214–258.

MULLAINATHAN, S., AND J. SPIESS (2017): "Machine learning: an applied econometric approach," Journal of Economic Perspectives, 31(2), 87–106.

NAGEL, R. (1995): "Unraveling in Guessing Games: An Experimental Study," American Economic Review, 85(5), 1313–1326.

PANG, B., L. LEE, AND S. VAITHYANATHAN (2002): "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. 79–86. Association for Computational Linguistics.

PENCZYNSKI, S. P. (2016a): "Persuasion: An experimental study of team decision making," Journal of Economic Psychology, 56, 244–261.

——— (2016b): "Strategic Thinking: The Influence of the Game," Journal of Economic Behavior and Organization, 128, 72–84.

PENCZYNSKI, S. P. (2017): "The nature of social learning: Experimental evidence," European Economic Review, 94, 148–165.

PESTIAN, J., H. NASRALLAH, P. MATYKIEWICZ, A. BENNETT, AND A. LEENAARS (2010): "Suicide note classification using natural language processing: A content analysis," Biomedical informatics insights, 3, BII–S4706.

RUBINSTEIN, A., AND A. TVERSKY (1993): "Naive Strategies in Zero-Sum Games," Working Paper 17-93, The Sackler Institute of Economic Studies.

STAHL, D. O., AND P. W. WILSON (1995): "On Players' Models of Other Players: Theory and Experimental Evidence," Games and Economic Behavior, 10(1), 218–254.

VAN ELTEN, J., AND S. P. PENCZYNSKI (2015): "Coordination games with asymmetric payoffs: An experimental study with intra-group communication," Discussion paper, University of Mannheim.

VARIAN, H. R. (2014): "Big data: New tricks for econometrics," Journal of Economic Perspectives, 28(2), 3–28.

VERMA, S., S. VIEWEG, W. J. CORVEY, L. PALEN, J. H. MARTIN, M. PALMER, A. SCHRAM, AND K. M. ANDERSON (2011): "Natural Language Processing to the Rescue? Extracting" Situational Awareness" Tweets During Mass Emergency.," in

ICWSM, pp. 385–392. Barcelona.

YAN, J. L. S., N. MCCRACKEN, AND K. CROWSTON (2014): “Semi-automatic content analysis of qualitative data,” iConference 2014 Proceedings.

A. Technical Appendix

This appendix gives a brief primer on using the tools of NLP and ML for your next project. Conveniently, many computational concepts of both NPL and ML have been implemented in the language and software R. They are therefore freely available and quickly implementable for any researcher.

Since the implementation of any analysis presented here consists basically of a number of lines of R code, it is instructive to walk along the lines of the code and comment on procedures in detail if necessary.[18] The electronic version of the code can be accessed in the online Appendix.

A.1. Starting and importing text

The concepts of NLP and ML programmed in R are accessed with the help of packages that have to be installed prior to running the code. Packages that have been installed using `install.packages(x)` can then be called upon using `library(x)`.

```
### Example Code -----------------------------------------------------------
library(quanteda)
library(SnowballC)
library(gtools)
library(refset)
library(randomForest)

setwd("C:/Work/Papers/ML")

set.seed(324789632)

# Read Data -----------------------------------------------------------------

d <- read.delim("SL.txt", stringsAsFactors = FALSE, na.strings = ".")
d <- d[! is.na(d$level),]
```

After the setting of the working directory, a seed for quasi-randomization is set that allows the researcher to replicate results and to grow the same random forest more than once. The example text-file `SL.txt` contains messages and manually coded levels of reasoning.

Having imported the messages, they can now be transformed into a document-feature matrix that has a column for each unique token t and indicates the token's frequency x_t^m in the message m. This functionality is provided by the R package `quanteda` that is maintained by Kenneth Benoit.[19]

```
# Create corpus (cps) and document-feature-matrix -------------------------

cps <- corpus(d$message)
mystop <- c(".", ",", "!", "/", "(", ")", "-", ":", "’", "?", "%", "a", "b",
            "a’s", "b’s", "as", "bs", "A", "B",  "black", "white",
```

[18] I am indebted to David Hugh-Jones for providing me with an R code at the start of this project. Many lines of code are his or adapted from his.

[19] The example is provided based on the `quanteda` package version 1.1.1 from March 2018, documentation url: `https://cran.r-project.org/web/packages/quanteda/quanteda.pdf`.

```
            stopwords("english"))
dfmat <- dfm(cps, remove = mystop, stem = TRUE)
dfmat <- dfmat[,colSums(dfmat) >= 5]
```

The main command of the package `quanteda` is `dfm()`, which tokenizes the text corpus `cps` and establishes the document-feature matrix `dfmat`. The argument `remove` takes away the previously defined set of string tokens `mystop`, which here contain general and game-specific symbols and words. For example, actions of the game are removed so that any association of actions with levels of reasoning is not picked up in the text analysis. `mystop` also contains `stopwords(‘english’)` a pre-established set in R of “stopwords” in English, words that are very frequent in any text but too general to contribute any context-specific meaning like “the”, “to”, “and”, “that”, “as”, “about”, “from”, etc. The argument `stem = TRUE` enables word stemming so that words conveying similar meaning like “team”, “teams”, and “teamed” all appear under the token “team”.[20] Finally, any token that appears less than 5 times in the corpus is removed.

```
# Create Word Cloud plot

topfeatures(dfmat[d$level<0.5], 100)
png(file = "example//cloudSL.png", width = 600, height = 600, pointsize=24)
textplot_wordcloud(dfmat[d$level<0.5], random.order = FALSE)
dev.off()
```

In order to get an overview of the remaining set of tokens or words, `topfeatures()` displays the most frequent tokens. A graphical version of this information is a word cloud such as in figure 5, which is established through the command `textplot_wordcloud`.

A.2. Machine learning

In order to start a first linear regression exercise, the whole sample is split into a *training sample* which informs the model and a *test sample* which tests the performance of the obtained model. We choose the quite common 0.7/0.3 split, but other splits are also used. For larger datasets, a smaller testing set might be sufficient. Note that the 10-fold cross-validation (see A.3) is a better approach to evaluate an algorithm, but might be involved for a first analysis.

```
# Create test and training data -----------------------------------------

train.prop <- .7
train.rows <- test.rows <- numeric(0)
for (tm in unique(d$treatment)) {
  rows <- which(d$treatment == tm)
  nr <- length(rows)
  train.rows <- c(train.rows, sample(rows, floor(nr * train.prop)))
  test.rows <- c(test.rows, setdiff(rows, train.rows))
}
train.rows <- train.rows[ ! is.na(d$level[train.rows]) ]
test.rows <- test.rows[ ! is.na(d$level[test.rows]) ]
dtr %r% d[train.rows,]
dtest %r% d[test.rows,]
dfmtr <- dfmat[train.rows,]
dfmtest <- dfmat[test.rows,]
```

[20] The package `SnowballC` provides access to a library of stemmed words that results from a word stemming algorithm.

Here, the original data d as well as the document-feature matrix is split into training and testing sets.

Then, there is only a line of code to program a complex computational routine. Here, the random forest functionality is provided by the package randomForest that is maintained by Andy Liaw.[21]

```
### random forest ---------------------------------------------------------
## training
# Regression
rf1 <- randomForest(x = as.matrix(dfmtr), y = dtr$level, keep.forest = TRUE,
                    importance = TRUE)
# Classification
rf2 <- randomForest(x = as.matrix(dfmtr), y = factor(dtr$level),
                    keep.forest = TRUE, importance = TRUE)
```

The code shows that both regression and classification require the input matrix dfmtr as independent variables $\mathbf{x}$ and the level classification as dependent variable y. While a numerical vector enters the regression in form of the levels, a vector of a categorical variable – the factorized levels – enters the random forest algorithm in classification. The importance of predictors is set to be assessed in importance = TRUE, which allows for a judgment of the contribution of each token to the accuracy of the model.

```
## testing ----------------------------------------------------------------
# Regression
predict(rf1, as.matrix(dfmtest)) -> rf1p
round(rf1p, 0) -> rf1pround

print(tab <- table(rf1pround, dtest$level))
cor.test(rf1p, dtest$level)
print(summary(lm(dtest$level ~ 0 + rf1p)))
```

For testing, the command predict applies the trained algorithm rf1 to the messages from the test sample dfmtest. The integer-rounded predictions can be compared to the human-coded levels, here by tabulation, correlation test and simple linear regression.

```
# Classification
predict(rf2, as.matrix(dfmtest)) -> rf2p
rf2pchar <- as.character(rf2p)
rf2pnum <- as.numeric(rf2pchar)

print(tab <- table(rf2p, dtest$level))
cor.test(rf2pnum, dtest$level)
print(summary(lm(dtest$level ~ 0 + rf2pnum)))
```

The testing of the classification results works analogously, the only difference is the conversion of the factorized levels into a character variable and then numerical variable before its use in the correlation test and linear regression.

The file in the online appendix further includes the code for the calculation and graphical illustration of the variable importance, as shown in figure 4.

[21] The example is provided based on the randomForest package version 4.6-12 from October 2015, documentation url: https://cran.r-project.org/web/packages/randomForest/randomForest.pdf.

A.3. Cross-validation

Cross-validation is a very common procedure in machine learning to judge the out-of-sample performance of a model based on as many out-of-sample observations as possible. In k-fold cross-validation, the dataset is divided in k equally large subsets. For each subset, the variable of interest can now be predicted based on a model that was trained on the union of the remaining $k - 1$ subsets. This way, the entire dataset can be predicted out-of-sample. A common choice for k is 10, but other values like 5 are used. For relatively small datasets, one can use a k equal to the sample size minus 1. There is little reason to not use this method more frequently in economics (Varian, 2014).

The file in the online appendix includes the code for the cross-validation. If further includes the code for the random permutation test, which tests whether the numbers of correctly classified messages could have possibly been obtained by chance. It randomly permutes the training levels and observes the number of correctly classified messages 2000 times (Random permutation test, Golland, Liang, Mukherjee, and Panchenko, 2005). In the APC game SL, for example, it shows that an almost exact 50-50 split of predictions could also be obtained by chance.

A.4. Growing and tuning forests

The details of how trees are being grown determine the complexity of the model used for prediction. The growing of a tree works as follows. For each terminal node of the tree, a split is implemented by randomly selecting k of the T variables and picking the best variable (and split-point s) of them as long as at least l observations fall into the created subspaces. The criterion for 'best' variable is the minimization of the model error, mse or Gini impurity, respectively.

In the regression here, out of a third of the variables, $k = T/3$, the best variables and split-points are chosen as long as at least $l = 5$ observations populate each subspace. For classification, out of $k = \sqrt{T}$ variables the best are chosen until at least $l = 1$ observation falls in a subspace. With the size of the dataset, these settings imply a certain depth of the trees. Alternatively, one could specify this depth directly.

The parameters that determine the model complexity can be treated as problem-specific tuning parameters to improve the model performance. Judging the out-of-sample performance with the help of cross-validation, one can "tune" the model to highest performance by choosing the details of the model appropriately. When looking for a few percent better performance, one can further combine models from various algorithms that complement each other.