Supervised and Experiential Learning
Practical work 2: decision and random forests

Pol Roca Llaberia

May 10, 2021

Master in Artificial Intelligence

Universitat Politècnica de Catalunya

1 Introduction

In this document we will see a possible implementation of the CART algorithm used to induce classifier trees for decision and random forests [1], along with a brief comparison of their performance by applying it to three different datasets and analyzing the obtained results.

2 Implementation

Algorithm 1 shows the pseudo-code of this implementation of CART, which is the core logic of what was then programmed in Python.

Algorithm 1 CART

Input: A supervised dataset $\mathcal{D}$
Output: The root node R of the induced classifier tree

$R \leftarrow$ New node ▷ Tree root
$\mathcal{T} \leftarrow \{\langle R, \mathcal{D}\rangle\}$ ▷ Set of nodes to be treated
while $\mathcal{T} \neq \varnothing$ **do**
  $N, D \leftarrow \text{pop}(\mathcal{T})$
  $A \leftarrow$ get_attribute_subset($D.attributes$)
  if $A = \varnothing$ **then**
    $N.is\_leaf \leftarrow$ True
    $N.label \leftarrow$ class_majority(D)
    continue
  end if
  $a, v \leftarrow$ find_best_split(D)
  (Update feature importance of attribute a)
  Set a, v as the selector used to split data with N
  Split D into $\langle L, R\rangle$ using the splitting condition of N
  $N.left \leftarrow$ New node
  if all instances in L have the same class label **then**
    $N.left.is\_leaf \leftarrow$ True
    $N.left.label \leftarrow$ the only label in L
  else
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{\langle N.left, L\rangle\}$
  end if
  $N.right \leftarrow$ New node
  if all instances in R have the same class label **then**
    $N.right.is\_leaf \leftarrow$ True
    $N.right.label \leftarrow$ the only label in R
  else
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{\langle N.right, R\rangle\}$
  end if
end while

Algorithm 2 find_best_split

Input: A supervised dataset $\mathcal{D}$
Output: An attribute-value pair $\langle a_{best}, v_{best} \rangle$ that splits $\mathcal{D}$ and minimizes the Gini impurity
$S \leftarrow \varnothing$ ▷ List of splits attempted and their scores
for each attribute a of $\mathcal{D}$ **do**
  if a is numeric **then**
    $V \leftarrow$ All midpoints between contiguous sorted values of a in $\mathcal{D}$
  else
    $V \leftarrow$ All possible binary partitions of the values of a in $\mathcal{D}$
  end if
  for v in V **do**
    $L, R \leftarrow$ Split $\mathcal{D}$ using v
    $L_{score} \leftarrow$ gini_impurity(L)
    $R_{score} \leftarrow$ gini_impurity(R)
    $s \leftarrow \frac{|L|}{|\mathcal{D}|} * L_{score} + \frac{|R|}{|\mathcal{D}|} * R_{score}$
    $S \leftarrow S \cup \{\langle a, v, s \rangle\}$
  end for
end for
$a_{best}, v_{best} \leftarrow$ Attribute-value pair in S having the minimum score

Note that this algorithm can be used to build both decision and random trees. In a decision tree, the function `get_attribute_subset()` returns all the attributes/features; in a random tree, this function returns a random subset of attributes, of length specified by parameter.

From this point, the process of building a decision forest or a random forest is simple. Decision forests can be built by inducing NT trees using only F random features initially for each tree. As for random forests, they can be built by bootstrapping the dataset for each tree and using it entirely to induce random trees since the subsets of F features will be chosen randomly on the go, as mentioned above.

3 Evaluation

In this section we will analyze the results of applying the algorithm on three datasets from the UCI repository, namely: *hepatitis*. *cmc* (Contraceptive Method Choice) and *nursery* [2]. It is worth mentioning that since some of these datasets contain missing values, some preprocessing steps were performed previously to inducing the forests where these values were filled with modes and means respectively to categorical and numerical attributes. By the way, data was randomly split and stratified into train and test sets with a proportion of 80/20.

The methodology employed was the following. Different forests were trained having these dimensions $NT_{all} = \{1, 10, 25, 50, 75, 100\}$. For decision forests, the number of features tried out were $F_{dec} = \{int(\frac{M}{4}), int(\frac{M}{2}), int(\frac{3*M}{4}), Runif(1, M)\}$ with M being the number of features/attributes in a specific dataset and $Runif$ a function that returns a random number between a given range. As for random forests, these where the lengths of the subsets of features tried: $F_{rand} = \{1, 3, int(\log_2 M + 1), \sqrt{M}\}$.

In appendix A we can see the whole list of experimental results separated in tables. Note

that each classifier was trained and tested 5 times with different random seeds to combat poor initializations, and the best result achieved was the one that was preserved (also known as *best of 5*). However, in this section we will analyze only some of these results, i.e. the *best* decision and random forest in terms of accuracy for each possible size NT among all the possible numbers of features F. Additionally, we will be able to see the importance of features computed from the frequency with which each feature was chosen when building a forest. For that matter, feature importance was computed by training a random forest of 1000 trees while using all the features of the dataset. This was thought to be a better way to obtain a representative result instead of using a decision forest.

3.1 Small dataset

The first dataset to apply the algorithm is the *hepatitis* dataset. This small dataset has only 155 instances, 6 numeric attributes, 13 categorical attributes and a class attribute with 2 different labels.

# trees	classifier	# features	# nodes	train time (s)	infer time (s)	accuracy
1	decision	4	8	0.01	0.01	0.8387
	random	3	19	0.03	0.01	**0.9355**
10	decision	4	336	0.61	0.02	**0.9677**
	random	3	196	0.29	0.03	**0.9677**
25	decision	4	679	1.20	0.07	0.9355
	random	5	431	0.92	0.08	**0.9677**
50	decision	9	1247	4.20	0.14	**0.9355**
	random	1	1406	1.64	0.19	0.9032
75	decision	unif	1517	6.32	0.20	**0.9355**
	random	1	2070	2.45	0.28	**0.9355**
100	decision	unif	2109	7.78	0.28	0.9355
	random	1	2722	3.09	0.36	**0.9677**

Table 1: Classification results for the *hepatitis* dataset.

Table 1 shows a comparison of both classifiers. The way to interpret this table is that for instance in the first row, the best decision forest of 1 tree was the one that used 4 features, induced 8 nodes in total (in this case of the same tree) and achieved 0.9355 score in accuracy. Random forest has achieved in 3 cases the maximum accuracy of 0.9677 in the results, while decision forest only once. Overall, random forest has obtained better results, with only 1 exception with forests of 50 trees. We can also see how the best results from random forests of sizes 50, 75 and 100 only employed a single feature per choice. This means that these classifiers were built totally randomly, since 1 feature was chosen each time, no comparison had to be made to decide which feature to use to split.

feature	frequency
age	0.2681
bilirubin	0.1309
alk_phosphate	0.1166
albumin	0.0685
spiders	0.0614
protime	0.0494
ascites	0.0442
sgot	0.0428
varices	0.032
liver_firm	0.0268
sex	0.0252
liver_big	0.0213
malaise	0.0212
histology	0.0202
steroid	0.0187
anorexia	0.0172
fatigue	0.0167
spleen_palpable	0.0117
antivirals	0.0069

Table 2: Feature importance for the *hepatitis* dataset.

Feature importance of this dataset is listed in table 2. It is interesting to see that actually the top 4 features are numeric. The *age* feature is more than twice as frequently used as the second most important feature. The same happens with the 3rd and the 4th. After that, importance decreases less drastically.

3.2 Medium dataset

The *cmc* dataset is a medium-sized database with 1473 instances. It has 2 numeric attributes, 7 categorical attributes and 3 possible class labels.

# trees	classifier	# features	# nodes	train time (s)	infer time (s)	accuracy
1	decision	4	285	0.53	0.04	0.4915
	random	3	445	0.54	0.05	**0.5153**
10	decision	4	1995	3.61	0.45	**0.5525**
	random	3	4282	5.18	0.51	0.5458
25	decision	6	10820	18.38	1.28	0.5458
	random	3	10679	13.12	1.31	**0.5729**
50	decision	6	22625	39.10	2.49	0.5424
	random	4	20449	27.58	2.47	**0.5661**
75	decision	6	32013	53.30	3.59	0.5458
	random	4	30809	41.19	3.65	**0.5932**
100	decision	6	42804	72.84	5.19	0.5424
	random	4	40650	53.74	4.93	**0.5831**

Table 3: Classification results for the *cmc* dataset.

In this case, table 3 shows how both kinds of forests struggled to classify this dataset correctly. Random forest achieved the best accuracy among all, with a score of 0.5932, and also better results in general. In comparison to the small dataset, forests here sum in total approximately 20x more nodes, even though the dataset is only about 10x larger. It could be that in this case features are just less correlated or informative in relation to the class attribute.

feature	frequency
Wifes_age	0.398
Number_of_children_ever_born	0.1605
Standard-of-living_index	0.0915
Husbands_occupation	0.0879
Wifes_education	0.0758
Husbands_education	0.0728
Wifes_now_working	0.0549
Wifes_religion	0.0477
Media_exposure	0.0109

Table 4: Feature importance for the *cmc* dataset.

Table 4 presents feature importance for the *cmc* dataset. The feature *Wifes_age* was by a huge margin the most important feature, being chosen about 40% of the time. The second one was less frequently used, but it significantly more than the third one. After that, the remaining features are pretty much on par.

3.3 Large dataset

In the last experiment a larger dataset was employed, i.e. the *nursery* dataset. This database is formed by 12960 instances with 8 categorical attributes and 5 possible class labels.

# trees	classifier	# features	# nodes	train time (s)	infer time (s)	accuracy
1	decision	4	72	0.32	0.43	0.7218
	random	4	408	0.67	0.50	**0.9726**
10	decision	6	7717	18.50	4.93	0.9371
	random	3	5139	6.97	4.63	**0.9973**
25	decision	unif	9894	25.61	10.46	0.9464
	random	4	8765	15.36	10.45	**0.9992**
50	decision	unif	18408	47.58	20.16	0.9630
	random	3	26063	35.85	22.12	**0.9985**
75	decision	unif	19277	56.12	29.70	0.9730
	random	4	26742	47.68	32.30	**0.9992**
100	decision	unif	25393	71.06	37.75	0.9691
	random	4	35441	64.32	43.53	**0.9988**

Table 5: Classification results for the *nursery* dataset.

The results on the *nursery* dataset can be seen in table 5. There are a few aspects to notice in these results. First of all, again, random forest provided the best results. Surprisingly, it was able to achieve up to 0.9992 in accuracy, which is practically a perfect classification. Decision forests also provided good results, however, they were always behind random forests and not just slightly. Notice how decision forests benefited from using the technique of deciding the number of features dynamically per tree (random **unif**orm), as most of their best results came from forests that employed it.

It is also interesting to see that the total number of nodes from these forests is similar to that of the medium dataset, and even though both datasets have an equivalent amount of features (9 and 8), this one is about 10x larger in terms of instances.

feature	frequency
form	0.2619
children	0.245
housing	0.1927
finance	0.1012
has_nurs	0.0626
parents	0.0514
social	0.0483
health	0.037

Table 6: Feature importance for the *nursery* dataset.

Finally, table 6 shows the importance of features derived from their frequency of choice with this dataset. We can see that the top 3 features are quite on par, accumulating around 60% of probability of chosing one of them. The rest of features are more or less equally less significant.

4 Conclusions

In conclusion we could say that random forests have better capabilities and provide better results compared to decision forests. One of the reasons could be because these forests apply bagging (bootstrap aggregating) separately for each tree, which allows each tree to be more specialized in some instances different from the others, providing more heterogeneous criteria when classifying new instances. Another reason could be that, even though decision forests also involve some randomness, random forests are *more random* by nature, as the act of choosing which features will be considered is repeated at each node. From what I know, the more randomness is involved in training these models, the less probable it will be that they overfit to the noise in the dataset, and with this they will offer a better generalization.

From what we have seen in the experiments and also from my own intuition, when choosing the hyperparameters (number of trees or features) I would probably set to use half of the amount of features in the dataset to enforce more variety in the choices since not always the best feature will be available. Regarding the number of trees, I would definitely train forests of at least 20 trees, ranging from that number up to 100. This way, the poor performance or initialization of some trees would not impact the overall performance by that much, making the forest more robust.

5 Reproducibility

A simple CLI was built so as to be able to reproduce the experiments described in this document. Following are some examples of commands needed to run the algorithm on the different datasets:

Listing 1: Command examples to execute the experiments

```
$ python3 main.py --help
$ python3 main.py --dataset hepatitis --classifier-type random \
                  --n-trees 10 --n-features 3
$ python3 main.py -d hepatitis -c random -t 10 -f 3
$ python3 main.py -d nursery -c decision -t 100   # no -f, so it will use all features
$ python3 main.py -d cmc -c decision -t 5 -f 0    # -f 0 means random uniform
$ python3 main.py -d cmc -c random --seed 123
```

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and regression trees," *Cytometry*, vol. 8, no. 5, pp. 534–535, 1987. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.990080516

[2] Christopher L. DuBois, "UCI network data repository," http://networkdata.ics.uci.edu, 2008.

A Experiment results

A.1 Decision forest results

A.1.1 Small dataset

# trees	# features	compute time	infer time	# nodes	accuracy	feature importance
1	4	0.0169	0.0054	8	0.8387	D, I, N, F
1	9	0.0736	0.0048	18	0.7742	C, M, A, I, P, D, L, H
1	14	0.1083	0.0044	24	0.8387	B, G, M, L, A, S, F, K, D, H, E
1	unif	0.1314	0.0046	20	0.8065	A, G, R, M, B, C, S, F, D, L, N
10	4	0.6148	0.0296	336	0.9677	O, B, A, G, C, D, R, H, E, Q, J, F, S, I, P, L
10	9	0.8161	0.0347	273	0.9355	B, O, C, G, A, M, D, L, Q, F, H, N, K, R, P, I, S, E, J
10	14	0.9927	0.0294	185	0.7742	A, C, G, B, M, O, K, F, S, P, L, R, H, N, I, D, J, Q
10	unif	0.7905	0.0310	218	0.8710	C, O, B, M, G, A, Q, K, I, D, P, J, S, N, L, E
25	4	1.2066	0.0716	679	0.9355	B, O, G, M, C, R, A, K, D, I, E, N, J, H, L, S, P, F, Q
25	9	2.3542	0.0793	638	0.9032	C, B, O, G, A, M, L, Q, K, D, P, F, J, R, S, H, N, I, E
25	14	3.2330	0.0743	526	0.8065	B, C, A, G, M, O, P, L, F, D, I, K, Q, E, J, S, R, N, H
25	unif	1.9898	0.0731	482	0.9032	G, C, O, M, A, B, D, P, I, K, N, Q, J, E, F, L, H, S, R
50	4	2.4124	0.1565	1337	0.9032	B, O, C, A, G, M, K, I, R, Q, N, J, D, E, H, F, L, P, S
50	9	4.2080	0.1487	1247	0.9355	B, A, C, O, G, M, J, K, L, N, S, P, R, I, F, H, Q, D, E
50	14	6.1214	0.1532	1063	0.8710	A, C, G, B, O, M, L, P, F, I, R, K, Q, E, N, D, J, S, H
50	unif	4.2315	0.1461	1136	0.8387	A, C, O, G, B, M, D, R, K, L, I, N, Q, P, J, S, E, F, H
75	4	3.7502	0.2240	2071	0.8710	A, O, G, C, B, M, K, J, D, N, R, E, I, S, Q, L, F, H, P
75	9	5.6771	0.2156	1841	0.8387	G, O, C, A, B, M, I, K, L, Q, S, P, J, H, R, F, D, N, E
75	14	9.2574	0.2029	1488	0.8387	C, G, O, A, B, M, P, R, I, S, L, F, K, H, N, E, J, Q, D
75	unif	6.3287	0.2045	1517	0.9355	C, O, A, B, M, G, P, J, I, D, S, N, L, R, Q, K, F, H, E
100	4	4.8328	0.3067	2670	0.9032	A, C, O, B, G, M, R, J, I, D, K, S, Q, H, L, N, E, F, P
100	9	8.3041	0.2950	2460	0.9032	O, C, G, B, A, M, R, L, I, J, K, H, Q, S, P, D, N, F, E
100	14	11.4385	0.3151	2183	0.8387	A, G, O, C, B, M, P, L, K, F, S, Q, I, D, N, J, R, H, E
100	unif	7.7843	0.2809	2109	0.9355	G, C, B, A, O, M, P, K, Q, D, F, L, I, R, J, H, S, N, E

letter	feature
A	age
B	albumin
C	alk_phosphate
D	anorexia
E	antivirals
F	ascites
G	bilirubin
H	fatigue
I	histology
J	liver_big
K	liver_firm
L	malaise
M	protime
N	sex
O	sgot
P	spiders
Q	spleen_palpable
R	steroid
S	varices

Table 7: Decision forest results for the *hepatitis* dataset.

A.1.2 Medium dataset

# trees	# features	compute time	infer time	# nodes	accuracy	feature importance
1	2	0.4074	0.0293	207	0.4644	F, D
1	4	0.5383	0.0408	285	0.4915	F, D, I, C
1	6	0.8485	0.0578	481	0.4373	F, E, B, I, H, C
1	unif	0.4637	0.0566	246	0.4814	D, E, H, I, G
10	2	0.7451	0.3425	308	0.4576	D, F, B, E, A, G, H, C
10	4	3.6152	0.4540	1995	0.5525	F, D, A, E, B, H, G, I, C
10	6	7.0069	0.5036	4140	0.5085	F, D, B, E, A, H, I, G, C
10	unif	6.1469	0.4510	3518	0.5153	F, D, B, E, A, G, H, I, C
25	2	2.3161	0.8015	1037	0.4576	F, D, B, A, G, H, C, E, I
25	4	9.5296	1.0805	5360	0.5322	F, D, B, E, H, G, I, A, C
25	6	18.3853	1.2898	10820	0.5458	F, D, E, B, A, I, H, G, C
25	unif	13.9705	1.0931	7453	0.5356	F, D, E, A, H, B, G, I, C
50	2	4.3222	1.6245	1864	0.4678	F, D, G, B, A, E, H, C, I
50	4	18.9853	2.2109	10755	0.5220	F, D, A, E, B, I, H, G, C
50	6	39.1090	2.4995	22625	0.5424	F, D, E, B, A, H, G, I, C
50	unif	31.4540	2.2387	17151	0.5424	F, D, E, A, B, H, I, G, C
75	2	7.0543	2.3794	3128	0.4847	F, D, A, B, E, G, I, H, C
75	4	27.8361	3.3875	15296	0.5254	F, D, E, A, B, H, I, C, G
75	6	53.3043	3.5976	32013	0.5458	F, D, E, B, A, H, G, I, C
75	unif	47.3045	3.2668	26531	0.5153	F, D, E, B, A, H, I, G, C
100	2	8.2604	3.2010	3609	0.4644	F, D, B, E, G, A, H, C, I
100	4	37.3747	4.4378	21371	0.5322	F, D, E, B, A, H, G, I, C
100	6	72.8464	5.1984	42804	0.5424	F, D, E, B, A, H, I, G, C
100	unif	60.5878	4.3013	33142	0.5322	F, D, E, B, H, A, G, I, C

letter	feature
A	Husbands_education
B	Husbands_occupation
C	Media_exposure
D	Number_of_children_ever_born
E	Standard-of-living_index
F	Wifes_age
G	Wifes_education
H	Wifes_now_working%3F
I	Wifes_religion

Table 8: Decision forest results for the *cmc* dataset.

A.1.3 Large dataset

# trees	# features	compute time	infer time	# nodes	accuracy	feature importance
1	2	0.0987	0.3551	8	0.5610	children, health
1	4	0.3223	0.4302	72	0.7218	social, form, housing, health
1	6	0.8593	0.4320	282	0.6154	social, children, has_nurs, finance, parents, health
1	unif	0.7912	0.5764	239	0.4703	form, children, social, has_nurs
10	2	0.9710	3.1276	97	0.7280	form, children, housing, health, has_nurs, parents, finance
10	4	4.4178	4.3281	1099	0.8584	form, children, social, has_nurs, housing, finance, parents, health
10	6	18.5085	4.9300	7717	0.9371	form, social, children, finance, has_nurs, housing, parents, health
10	unif	7.1117	3.8252	2390	0.6617	form, finance, social, children, housing, has_nurs, parents, health
25	2	2.3556	7.9187	237	0.5324	children, has_nurs, social, form, parents, finance, health, housing
25	4	11.0047	11.0518	2711	0.8661	form, children, social, has_nurs, finance, housing, parents, health
25	6	43.1553	12.0836	17570	0.6562	form, children, social, finance, has_nurs, housing, parents, health
25	unif	25.6175	10.4698	9894	0.9464	form, children, social, finance, has_nurs, housing, parents, health
50	2	4.5032	15.1217	438	0.6470	children, housing, has_nurs, social, form, finance, health, parents
50	4	20.0680	21.0472	4937	0.9313	form, children, social, has_nurs, housing, finance, parents, health
50	6	92.1679	24.7942	37878	0.9421	form, social, children, has_nurs, finance, parents, housing, health
50	unif	47.5842	20.1677	18408	0.9630	form, children, social, finance, has_nurs, parents, housing, health
75	2	7.4230	23.5115	746	0.6763	children, form, has_nurs, housing, social, health, finance, parents
75	4	30.7268	32.6377	7485	0.8252	form, children, social, has_nurs, finance, housing, parents, health
75	6	129.7754	35.8653	52192	0.9653	form, children, social, has_nurs, finance, housing, parents, health
75	unif	56.1224	29.7032	19277	0.9730	form, children, social, finance, has_nurs, housing, parents, health
100	2	9.5250	31.6499	950	0.5818	form, has_nurs, children, social, housing, finance, parents, health
100	4	37.0088	41.4136	8710	0.8900	form, children, social, has_nurs, finance, housing, parents, health
100	6	160.8445	46.8087	65289	0.9599	form, children, social, finance, has_nurs, housing, parents, health
100	unif	71.0635	37.7563	25393	0.9691	form, children, social, finance, housing, has_nurs, parents, health

Table 9: Decision forest results for the *nursery* dataset.

A.2 Random forest results

A.2.1 Small dataset

# trees	# features	compute time	infer time	# nodes	accuracy	feature importance
1	1	0.0306	0.0055	28	0.8387	M, B, I, F, E, P, G, A, S, C, Q, J, D, L, O, R, N
1	3	0.0364	0.0054	19	0.9355	C, G, F, M, B, O, P, K, J, D, H, R, N, A
1	4	0.0268	0.0048	14	0.8387	B, G, I, M, O, C, S, P, D, L, H, N
1	5	0.0318	0.0044	12	0.8710	G, O, M, C, F, P, D, R, A
10	1	0.2980	0.0397	272	0.9032	B, C, M, K, G, A, D, O, P, R, Q, H, J, F, L, N, I, S, E
10	3	0.2924	0.0383	196	0.9677	G, B, O, M, A, C, R, I, F, L, P, Q, K, N, J, D, S, H, E
10	4	0.3135	0.0329	170	0.9032	G, B, C, A, O, M, K, R, P, J, I, Q, F, L, S, N, H, D
10	5	0.3910	0.0377	183	0.9032	O, C, G, B, M, A, P, F, K, R, S, N, D, Q, L, H, I, E, J
25	1	0.7693	0.0931	627	0.8710	O, A, B, C, M, G, R, I, F, Q, D, L, P, J, K, S, H, N, E
25	3	0.7861	0.0834	515	0.9355	O, G, B, C, A, M, R, L, P, S, Q, K, F, H, I, D, E, J, N
25	4	0.8954	0.0821	454	0.9032	G, O, B, A, C, M, P, L, I, Q, S, F, K, D, J, H, R, N, E
25	5	0.9235	0.0809	431	0.9677	G, B, A, C, O, M, P, I, F, S, R, Q, L, H, K, J, D, N, E
50	1	1.6493	0.1919	1406	0.9032	C, O, G, B, A, M, K, L, R, P, I, Q, F, J, S, N, D, H, E
50	3	1.4913	0.1552	957	0.9032	G, B, C, M, A, O, L, I, P, F, R, Q, J, S, D, N, K, H, E
50	4	1.7273	0.1540	958	0.9032	B, A, G, C, O, M, P, F, I, K, R, Q, S, L, D, J, H, N, E
50	5	1.7765	0.1488	793	0.9032	B, G, O, A, C, M, P, S, R, L, F, Q, I, D, J, K, H, N, E
75	1	2.4564	0.2835	2070	0.9355	O, A, C, G, B, M, P, K, L, R, D, I, S, Q, H, F, J, E, N
75	3	2.3183	0.2428	1434	0.9032	G, A, O, C, B, M, P, I, F, R, K, S, L, Q, J, H, D, N, E
75	4	2.3490	0.2335	1329	0.9032	B, G, A, C, O, M, P, Q, S, I, K, L, R, F, D, J, H, E, N
75	5	2.6487	0.2171	1221	0.8710	G, A, O, C, B, M, P, L, I, F, K, D, S, Q, R, H, J, N, E
100	1	3.0983	0.3658	2722	0.9677	B, C, A, O, G, M, R, Q, P, K, I, L, J, D, S, E, N, F, H
100	3	3.0402	0.3190	2096	0.9677	B, O, G, A, C, M, P, L, I, R, K, J, D, Q, F, S, H, N, E
100	4	3.2881	0.3063	1819	0.9355	G, A, B, O, M, C, P, L, K, I, R, D, F, Q, H, J, S, N, E
100	5	3.3099	0.2800	1535	0.8387	G, B, C, A, O, M, P, L, I, F, R, Q, J, K, D, H, S, N, E

letter	feature
A	age
B	albumin
C	alk_phosphate
D	anorexia
E	antivirals
F	ascites
G	bilirubin
H	fatigue
I	histology
J	liver_big
K	liver_firm
L	malaise
M	protime
N	sex
O	sgot
P	spiders
Q	spleen_palpable
R	steroid
S	varices

Table 10: Random forest results for the *hepatitis* dataset.

A.2.2 Medium dataset

# trees	# features	compute time	infer time	# nodes	accuracy	feature importance
1	1	0.6476	0.0600	470	0.4983	F, D, B, G, E, A, I, H, C
1	3	0.5499	0.0583	445	0.5153	F, D, E, B, G, H, A, I, C
1	4	0.5082	0.0525	364	0.5017	F, D, B, E, A, H, I, G, C
10	1	6.2027	0.5948	4704	0.5390	F, D, E, B, G, A, H, I, C
10	3	5.1812	0.5125	4282	0.5458	F, D, E, B, A, G, H, I, C
10	4	5.3618	0.5031	3983	0.5220	F, D, B, E, G, H, A, I, C
25	1	15.5553	1.5248	11875	0.5119	F, D, E, B, G, A, H, I, C
25	3	13.1289	1.3109	10679	0.5729	F, D, E, B, G, A, H, I, C
25	4	13.7444	1.2589	10103	0.5559	F, D, E, B, G, A, H, I, C
50	1	31.5248	3.0825	23884	0.5220	F, D, G, E, B, A, I, H, C
50	3	25.9186	2.5991	21449	0.5525	F, D, E, B, G, A, H, I, C
50	4	27.5851	2.4774	20449	0.5661	F, D, E, B, A, G, H, I, C
75	1	46.3260	4.8864	35600	0.5424	F, D, E, G, B, A, H, I, C
75	3	38.4704	3.8565	31956	0.5322	F, D, B, E, A, G, H, I, C
75	4	41.1944	3.6551	30809	0.5932	F, D, E, B, A, G, H, I, C
100	1	61.4524	5.7145	47315	0.5288	F, D, E, G, B, A, H, I, C
100	3	51.9370	5.1323	43410	0.5695	F, D, E, B, A, G, H, I, C
100	4	53.7473	4.9332	40650	0.5831	F, D, E, B, A, G, H, I, C

letter	feature
A	Husbands_education
B	Husbands_occupation
C	Media_exposure
D	Number_of_children_ever_born
E	Standard-of-living_index
F	Wifes_age
G	Wifes_education
H	Wifes_now_working%3F
I	Wifes_religion

Table 11: Random forest results for the *cmc* dataset.

A.2.3 Large dataset

# trees	# features	compute time	infer time	# nodes	accuracy	feature importance
1	1	1.9837	0.6070	1651	0.8557	has_nurs, health, children, parents, housing, form, finance, social
1	2	0.7802	0.5014	693	0.9433	form, housing, social, children, has_nurs, health, parents, finance
1	3	0.7347	0.5378	540	0.9633	children, housing, form, has_nurs, social, health, finance, parents
1	4	0.6797	0.5044	408	0.9726	children, form, social, health, parents, has_nurs, housing, finance
10	1	18.8263	5.8132	15671	0.9676	health, has_nurs, form, parents, children, social, housing, finance
10	2	9.7031	5.2168	8498	0.9938	health, form, has_nurs, children, housing, social, parents, finance
10	3	6.9725	4.6308	5139	0.9973	form, children, has_nurs, social, housing, finance, health, parents
10	4	6.2805	4.2915	3627	0.9973	form, children, housing, social, finance, has_nurs, parents, health
25	1	43.2992	13.6615	37669	0.9796	health, has_nurs, children, form, parents, social, housing, finance
25	2	24.9960	12.8604	21762	0.9954	health, has_nurs, form, children, housing, social, parents, finance
25	3	17.4400	11.1120	13044	0.9961	form, children, housing, has_nurs, social, finance, parents, health
25	4	15.3671	10.4599	8765	0.9992	form, children, housing, social, has_nurs, finance, parents, health
50	1	101.8841	30.5551	81614	0.9811	health, has_nurs, form, children, parents, social, housing, finance
50	2	49.7844	25.1892	43469	0.9969	health, form, has_nurs, children, housing, social, parents, finance
50	3	35.8508	22.1206	26063	0.9985	form, children, housing, health, social, has_nurs, finance, parents
50	4	31.0644	21.6034	17480	0.9981	form, children, housing, has_nurs, social, finance, parents, health
75	1	141.5689	41.7957	119745	0.9888	health, has_nurs, children, form, parents, social, housing, finance
75	2	72.8997	37.6181	63521	0.9973	health, form, has_nurs, children, housing, social, parents, finance
75	3	54.2853	35.1383	40574	0.9981	form, children, has_nurs, health, social, housing, finance, parents
75	4	47.6890	32.3083	26742	0.9992	form, children, housing, finance, social, has_nurs, parents, health
100	1	183.2779	56.9543	156060	0.9896	health, has_nurs, parents, children, form, housing, social, finance
100	2	100.5444	51.0121	88331	0.9938	health, has_nurs, form, children, social, housing, parents, finance
100	3	69.6816	44.0344	52992	0.9985	form, children, has_nurs, housing, social, health, finance, parents
100	4	64.3259	43.5373	35441	0.9988	form, children, housing, social, finance, has_nurs, parents, health

Table 12: Random forest results for the *nursery* dataset.