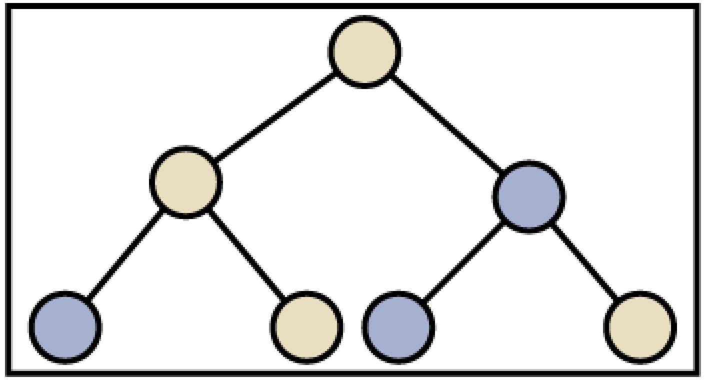
Gentree

Everything you didn't know you needed for decision trees
Now on sale at WUCAI 2024

Decision Trees

- Omnipresent in (tabular) classification tasks
- Enable logic-like reasoning
- Easy and usually fast to learn, almost hyperparameter-free



From bravo to eccellente...

Greedy Trees

Hands-down dominant.

- greedy partitioning
- suboptimal solutions
- fast
- CART, C5, etc.

Optimal Trees

Never heard of them.

- theoretically optimal
- *slow* :(
- Optimal trees, branch-and-bound trees

From bravo to eccellente...

Greedy Trees

Hands-down dominant.

- greedy partitioning
- suboptimal solutions
- *fast*
- CART, C5, etc.

Limitation: greedy

Optimal Trees

- theoretically optimal
- *slow* :(
- Optimal trees, branch-and-bound trees

Limitation: even though optimal, the theory has limitations

... passing through bravissimo

Greedy Trees

Limitation: greedy

Opti-like Trees

- still suboptimal...
- but optimized
- guess what? Sometimes they're genetic!

Optimal Trees

Limitation: even though optimal, the theory has limitations

Genesim: genetic extraction of a single, interpretable model

... passing through bravissimo

Greedy Trees

Limitation: greedy

Marrying the two families

- more accurate than greedy
- faster than optimal

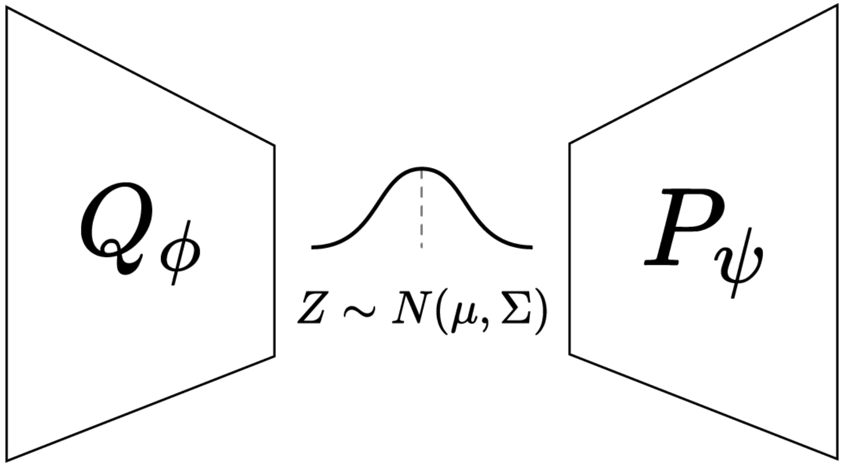
Optimal Trees

Limitation: even though optimal, the theory has limitations

Paradigm shift: candidate search in a continuous space

Ingredient 1: Variational Autoencoders.

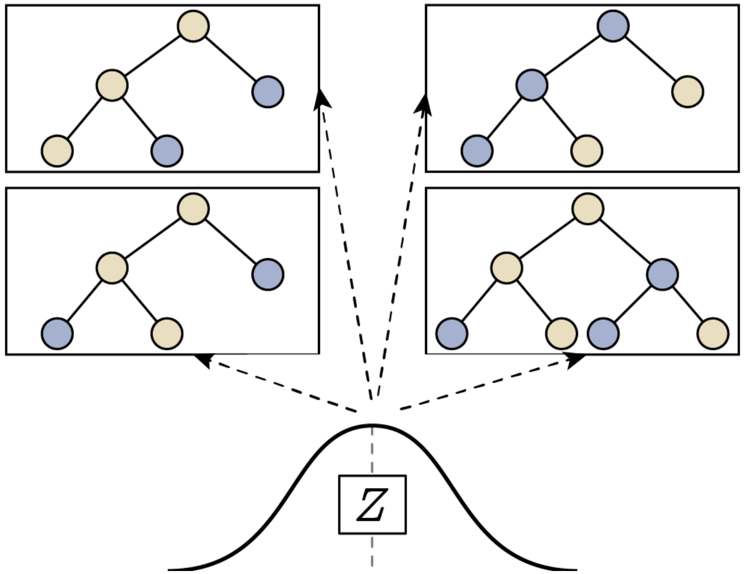
- Encoder: data in...
- Decoder: data out.
- Sampling: where the data (latently) lives.



Paradigm shift: tree space percolation

Ingredient 2: Tree space percolation.

- Searching a latent space is exceedingly difficult (read: intractable)!
- Should condition on a tree, but we don't have it :(
- F**k it, we optimizing



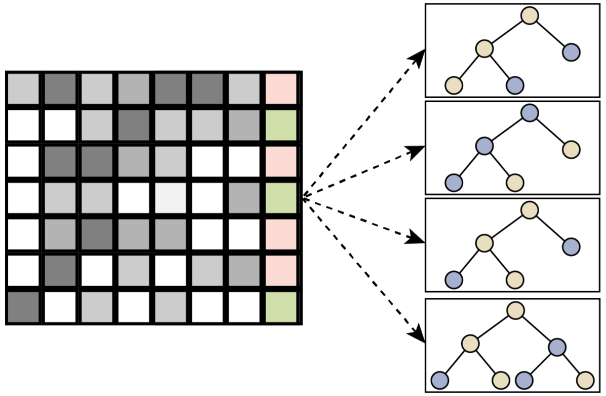
The represent-and-sample pipeline

I love it, I wish I could have worked on it directly.

```
base_models = induce_models(...)
space = induce_space(base_models)
seed_model = induce_model()
model = optimize(seed_model, data)
```

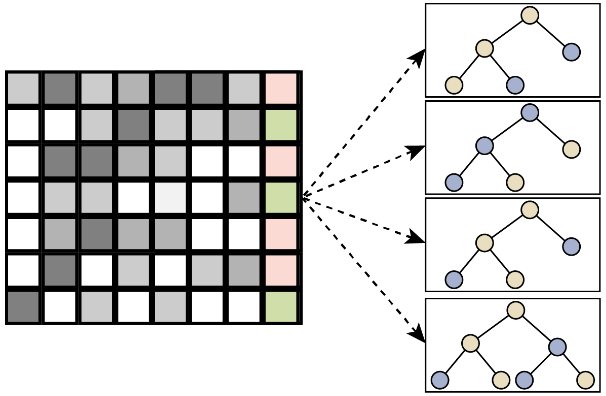
People also do it for logic programs and other discrete models!

Gentree

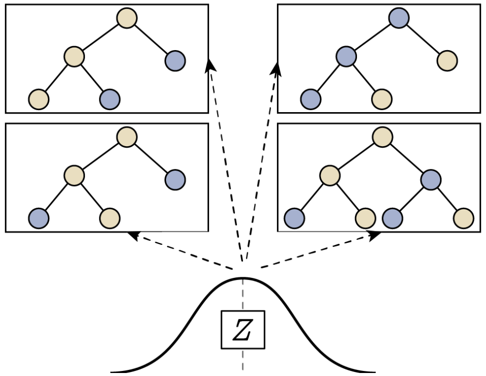


Trees T induced from the dataset.

Gentree

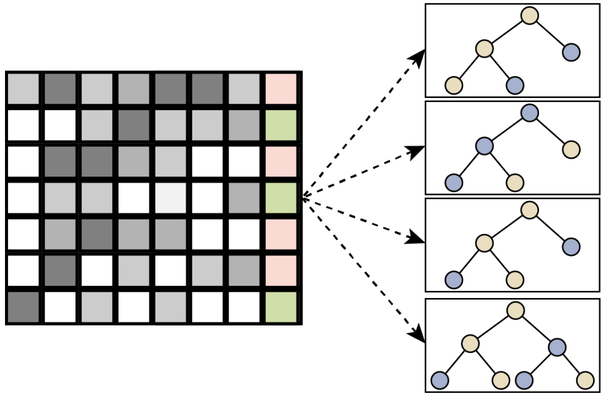


Trees T induced from the dataset.

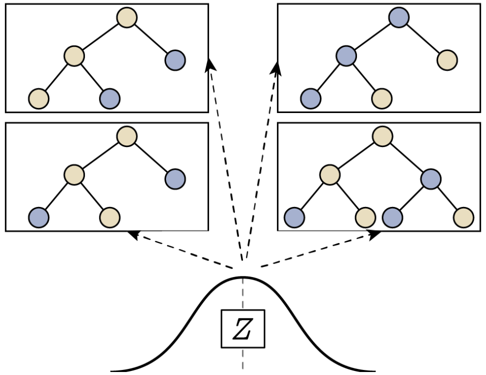


Tree-VAE learns a space Z from T.

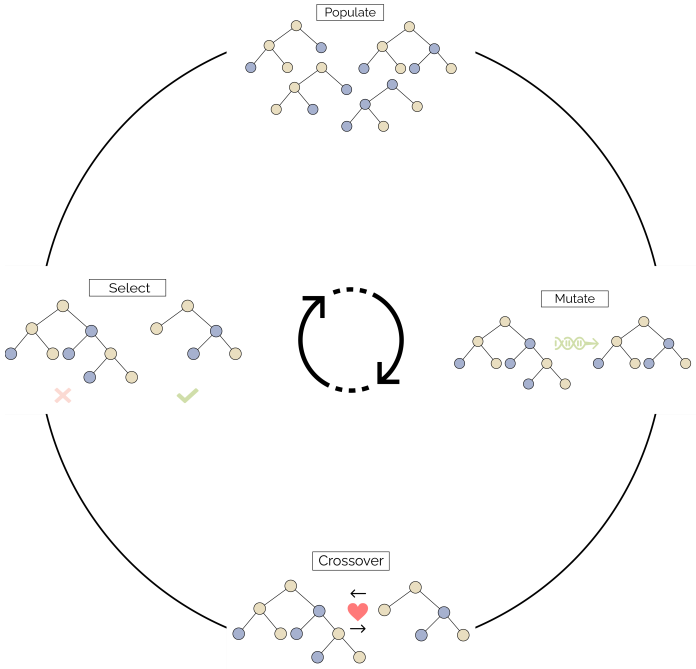
Gentree



Trees T induced from the dataset.



Tree-VAE learns a space Z from T.



Sample from Z, genetic optimization.

Juicy details: Tree-VAE

- A classic Convolutional VAE;
- Trees are encoded the good old fashioned CS way:
  - maximum capacity matrix
  - one row holding features,
  - and one row holding parameters.

Node 1	Node 2	...	Node $2^d - 1$
`age`	`salary`	`...`	`-`
`30`	`10K`	`...`	`0.9`

Experiments

- 18 UCI datasets
- continuous attributes only
- Datasets randomly split as 80% - 10% - 10%
- GenTree (GT) VS greedy trees (CART *DT*, Random Forest *RF*), optimal trees (Optimal Decision Trees *ODT*), opti-like trees (GeneSim *GS*)
- 10 randomized repetitions per run

Dataset	n	m	$\mid \mathcal{Y} \mid$	% majority	% minority	$\bar{d}$
aus	690	38	2	55.51	44.49	4
bnk	45211	48	2	88.30	11.70	4
bnote	1372	4	2	55.54	44.46	6
brst	699	9	2	65.52	34.48	6
cars	1728	6	4	70.02	3.76	8
dbn	13611	16	7	26.05	3.84	7
ecoli	327	6	5	43.73	6.12	4
glass	214	9	6	35.51	4.21	4
heart	270	20	2	55.56	44.44	4
iris	150	4	3	33.33	33.33	4
iso	7797	617	26	3.85	3.82	10
led7	2563	7	8	13.30	10.53	7
lymph	142	47	2	57.04	42.96	4
pima	768	8	2	65.10	34.90	4
sonar	208	60	2	53.37	46.63	7
vlc	846	18	4	25.77	23.52	10
wine	6497	11	7	43.65	0.08	9
yeast	1484	8	10	31.20	0.34	6

Table 1: Datasets size (n), dimensionality (m), class number ($\mid \mathcal{Y} \mid$), and majority and minority class percentages.

Performance

	Accuracy ↑					Complexity ↓				
	DT	GT	GS	ODT	RF	DT	GT	GS	ODT	RF
aus	.829	.855	.855	.855	**.858**	29.0	**3.0**	23.8	**3.0**	26.7
bank	.825	.891	-	**.894**	.881	31.0	11.4	-	**11.0**	30.4
bnk	.916	.915	-	.978	**.996**	43.0	**11.0**	-	19.0	40.8
brst	.864	.914	.950	.929	**.957**	41.0	**3.4**	18.5	7.0	42.1
car	.866	.875	-	.913	**.954**	79.0	**19.6**	-	49.0	147.0
dnb	.817	.665	-	**.902**	.897	183.0	**73.6**	-	75.0	169.7
ecoli	.850	.760	.853	**.940**	.916	25.0	**3.0**	19.1	15.0	26.4
glass	.716	.777	.670	.682	**.855**	23.0	**5.8**	29.7	7.0	23.8
heart	.764	.784	.798	.780	**.920**	29.0	6.2	17.4	**3.0**	26.8
iris	.913	**.967**	.946	.933	.933	13.0	**5.0**	5.9	7.0	12.0
iso	.730	.779	-	.822	**.933**	526.2	815.2	-	**163.0**	597.4
led7	.733	.753	.793	.795	**.804**	217.0	**51.2**	92.0	57.0	195.8
lymph	.797	.800	.787	.800	**.920**	25.0	**5.0**	14.8	**5.0**	21.5
pima	.644	.726	.727	.766	**.797**	31.0	3.6	45.2	**3.0**	28.5
sonar	.711	.695	-	.762	**.900**	39.0	**23.0**	-	31.0	40.0
VCL	.649	.678	.683	**.776**	.764	217.8	**41.8**	83.2	49.0	194.7
wine	.482	.481	**.913**	.503	.595	660.2	39.6	**8.0**	11.0	528.6
yeast	.509	.519	-	.530	**.572**	95.0	**28.4**	-	35.0	86.3
avg	.753	.779	.813	.808	**.858**	128.1	62.7	32.5	**30.5**	124.3
rank	2.10	1.78	-	1.43	**0.99**	2.10	1.10	-	**1.06**	1.86

Complexity

	Accuracy ↑					Complexity ↓				
	DT	GT	GS	ODT	RF	DT	GT	GS	ODT	RF
`aus`	.829	*.855*	*.855*	*.855*	**.858**	29.0	**3.0**	23.8	**3.0**	26.7
`bank`	.825	*.891*	-	**.894**	.881	31.0	*11.4*	-	**11.0**	30.4
`bnk`	.916	.915	-	*.978*	**.996**	43.0	**11.0**	-	*19.0*	40.8
`brst`	.864	.914	*.950*	.929	**.957**	41.0	**3.4**	18.5	*7.0*	42.1
`car`	.866	.875	-	*.913*	**.954**	79.0	**19.6**	-	*49.0*	147.0
`dnb`	.817	.665	-	**.902**	*.897*	183.0	**73.6**	-	*75.0*	169.7
`ecoli`	.850	.760	*.853*	**.940**	*.916*	25.0	**3.0**	*19.1*	15.0	26.4
`glass`	.716	*.777*	.670	.682	**.855**	23.0	**5.8**	29.7	*7.0*	23.8
`heart`	.764	.784	*.798*	.780	**.920**	29.0	*6.2*	17.4	**3.0**	26.8
`iris`	.913	**.967**	*.946*	.933	.933	13.0	**5.0**	*5.9*	7.0	12.0
`iso`	.730	.779	-	*.822*	**.933**	*526.2*	815.2	-	**163.0**	597.4
`led7`	.733	.753	*.793*	.795	**.804**	217.0	**51.2**	92.0	*57.0*	195.8
`lymph`	.797	*.800*	.787	*.800*	**.920**	25.0	**5.0**	*14.8*	**5.0**	21.5
`pima`	.644	.726	*.727*	.766	**.797**	31.0	*3.6*	45.2	**3.0**	28.5
`sonar`	.711	.695	-	*.762*	**.900**	39.0	**23.0**	-	*31.0*	40.0
`VCL`	.649	.678	.683	**.776**	*.764*	217.8	**41.8**	83.2	*49.0*	194.7
`wine`	.482	.481	**.913**	.503	*.595*	660.2	39.6	**8.0**	*11.0*	528.6
`yeast`	.509	.519	-	*.530*	**.572**	95.0	**28.4**	-	*35.0*	86.3
avg	.753	.779	*.813*	.808	**.858**	128.1	62.7	*32.5*	**30.5**	124.3
rank	2.10	1.78	-	*1.43*	**0.99**	2.10	*1.10*	-	**1.06**	1.86

Juicy analysis: the latent space

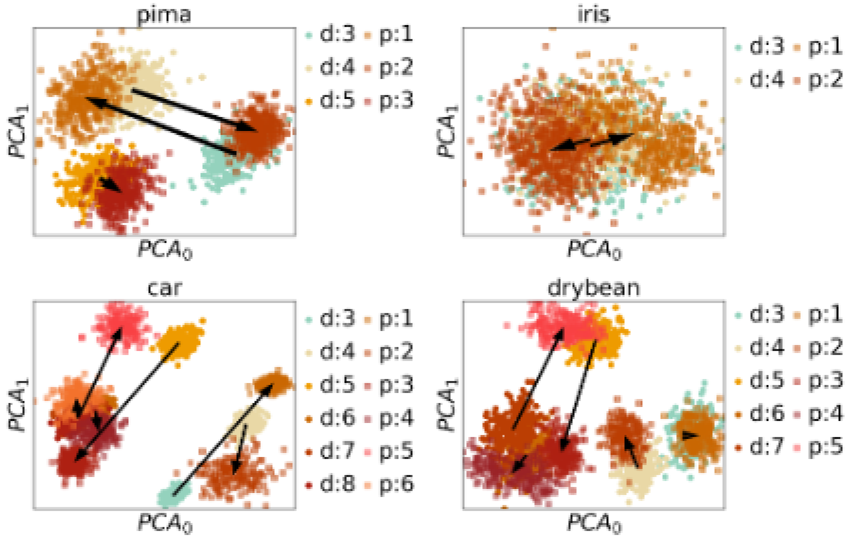


Figure 6: GENTREE latent tree spaces shown with two principal components for 1000 decision trees and their pruned counterpart. Depths (after ":") are colored in different ways.

- Tree-VAE (kinda) learns depth clusters!
- Depth directions exists, other may be interesting to investigate:
  - features
  - structure

Gentree

- Finally another way of inducing decision trees...
- and it works!
- Continuous representations shows some promise in disentangling tree properties
- Pls no question I have this as a poster at WUCAI in a couple of days :)