Estimation of parameter importance and interaction with RF

Leslie Pérez Cáceres

Below there is an example using ACOTSP:

- 20 secs cut off time
- 11 parameters
- 200 instances of size 2000
- 5000 experiments for configuration

1. Configuration ranking as dependent variable

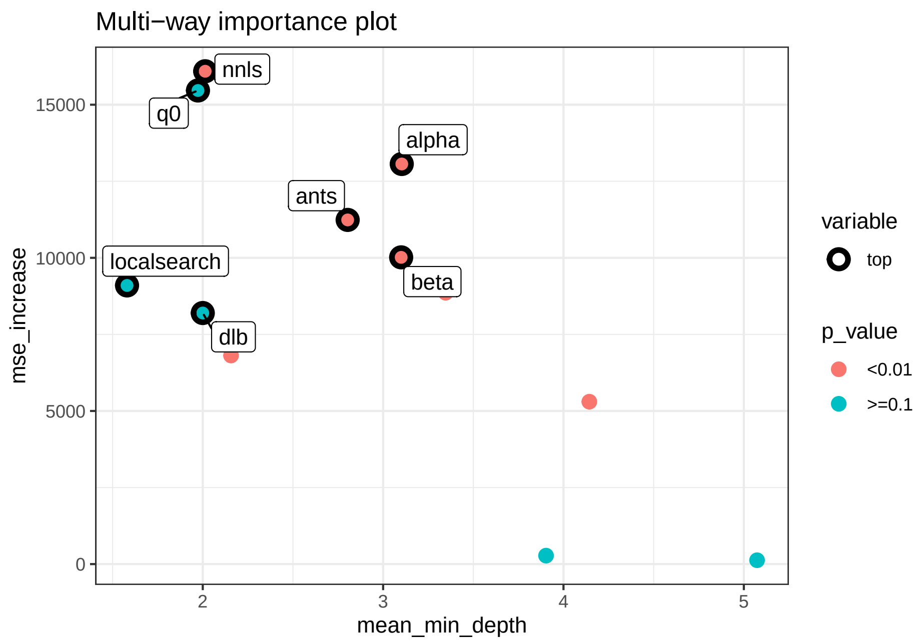
```
load("model_data/model-acotsp2000-ranking.Rdata")
importance_frame = model$importance_frame
important_parameters = model$important_parameters
full_interactions_frame = model$full_interactions_frame
interactions_frame = model$interactions_frame
kable(importance_frame[order(importance_frame[,"mean_min_depth"]),])  %>%
      kable_styling(latex_options="scale_down")
```

	variable	mean_min_depth	no_of_nodes	mse_increase	node_purity_increase	no_of_trees	times_a_root	p_value
9	localsearch	1.580000	11712	9099.0143	22939928	300	48	1
11	q0	1.973333	45043	15463.9853	59530677	300	88	1
5	dlb	2.000000	8965	8198.4909	18601123	300	38	1
10	nnls	2.013333	67056	16092.5830	50521182	300	18	0
1	algorithm	2.093333	17393	7626.4907	27977278	300	78	1
8	instance	2.156667	227931	6810.7911	85093917	300	0	0
3	ants	2.803333	84446	11238.2352	37397603	300	14	0
4	beta	3.100000	89000	10016.2334	40649159	300	0	0
2	alpha	3.103333	88859	13065.0803	45253707	300	5	0
13	rho	3.346667	84217	8860.2857	31832405	300	6	0
7	elitistants	3.903333	11005	276.6651	1364817	300	5	1
6	dummy	4.143333	66966	5302.7057	20237786	300	0	0
12	rasrank	5.073333	15798	126.5911	1929760	300	0	1

We can plot some measures using the randomForestExplainer package. In this case, since we are not predicting performance, analizing the importance of parameters not including the instance would be an error.

```
plot_multi_way_importance(importance_frame, size_measure = "p_value",
                          y_measure="mse_increase", x_measure="mean_min_depth",
                          no_of_labels=7)
```

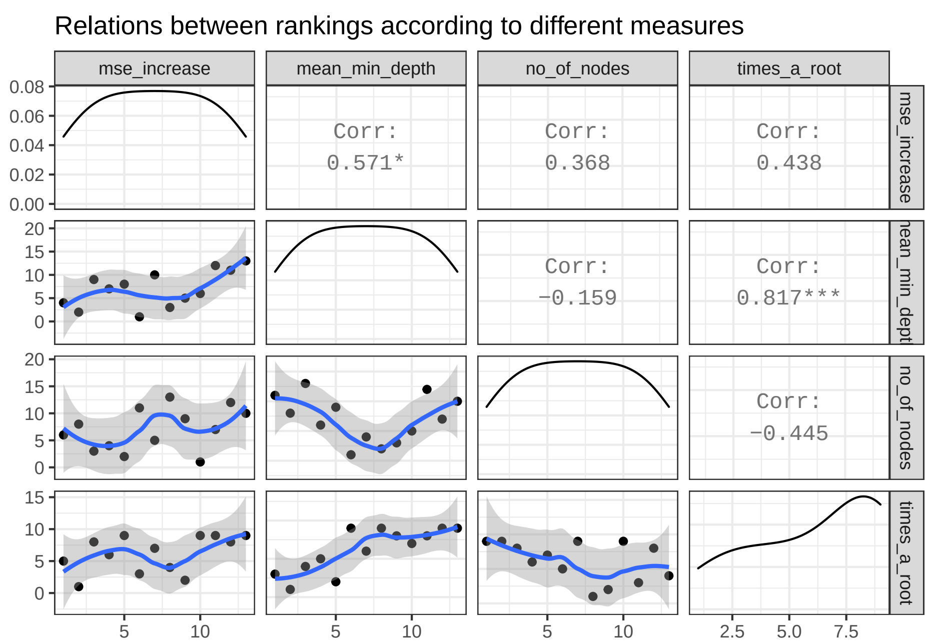
```
## Warning: Using alpha for a discrete variable is not advised.
```



We can also visualize the relationship between importance measures

```r
#plot_importance_ggpairs(importance_frame)
plot_importance_rankings(importance_frame, measures=c("mse_increase", "mean_min_depth", "no_of_nodes",
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



After the analysis of conditional parameters the most important 5 parameters are (plus reference added for interaction analysis):

```
print(important_parameters)
```

```
## [1] "localsearch" "nnls"        "algorithm"   "dummy"
```

We run the interaction analysis and find the importance of interactions:

```
kable(full_interactions_frame) %>% kable_styling(latex_options="scale_down")
```

variable	root_variable	mean_min_depth	occurrences	interaction	uncond_mean_min_depth
nnls	algorithm	0.9100719	278	algorithm:nnls	1.366667
algorithm	algorithm	1.3782254	274	algorithm:algorithm	1.310000
dummy	algorithm	0.9681535	274	algorithm:dummy	1.433333
nnls	dummy	0.8765468	270	dummy:nnls	1.366667
dummy	dummy	1.1423022	267	dummy:dummy	1.433333
nnls	localsearch	1.1505755	265	localsearch:nnls	1.366667
nnls	nnls	1.1925180	265	nnls:nnls	1.366667
localsearch	localsearch	1.7280576	258	localsearch:localsearch	1.403333
dummy	nnls	1.4168345	256	nnls:dummy	1.433333
dummy	localsearch	1.5509353	250	localsearch:dummy	1.433333
localsearch	algorithm	1.7513909	247	algorithm:localsearch	1.403333
algorithm	nnls	2.1723741	230	nnls:algorithm	1.310000
localsearch	nnls	2.4574101	224	nnls:localsearch	1.403333
algorithm	dummy	2.3841727	223	dummy:algorithm	1.310000
localsearch	dummy	2.4846043	221	dummy:localsearch	1.403333
algorithm	localsearch	2.5634532	216	localsearch:algorithm	1.310000

Once we filter dummy interactions and aggregate bidirectional interactions we get a matrix where the importance of interactions are summarized:

```
print(interactions_frame)
```

```
##   variable root_variable mean_min_depth occurrences    interaction
## 1     nnls     algorithm      0.9100719         508 algorithm:nnls
##   uncond_mean_min_depth
## 1              1.366667
```

2. Configuration ranking quartile as dependent variable

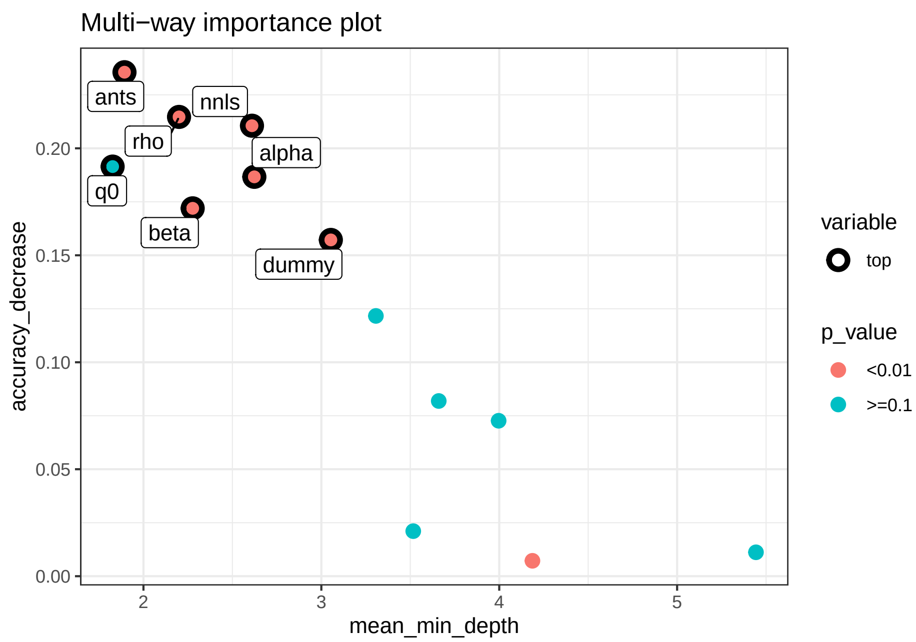
```
load("model_data/model-acotsp2000-qranking.Rdata")
importance_frame = model$importance_frame
important_parameters = model$important_parameters
full_interactions_frame = model$full_interactions_frame
interactions_frame = model$interactions_frame
kable(importance_frame[order(importance_frame[,"mean_min_depth"]),])  %>%
      kable_styling(latex_options="scale_down")
```

	variable	mean_min_depth	no_of_nodes	accuracy_decrease	gini_decrease	no_of_trees	times_a_root	p_value
11	q0	1.826667	20875	0.1914654	744.2653	300	70	1
3	ants	1.893333	32935	0.2355764	1038.1176	300	50	0
13	rho	2.200000	32707	0.2146773	1035.0556	300	28	0
4	beta	2.276667	34873	0.1719534	1125.2795	300	13	0
10	nnls	2.610000	27377	0.2105706	854.9377	300	25	0
2	alpha	2.623333	34671	0.1866669	1104.9939	300	7	0
6	dummy	3.053333	25785	0.1571873	797.8111	300	4	0
1	algorithm	3.306667	7826	0.1216607	220.9197	300	55	1
12	rasrank	3.516667	6258	0.0210585	175.7423	300	5	1
5	dlb	3.660000	5230	0.0819014	148.2808	300	30	1
9	localsearch	3.996667	8358	0.0726623	235.1934	300	11	1
8	instance	4.186667	69848	0.0071969	1164.0060	300	0	0
7	elitistants	5.443333	4427	0.0111779	112.1926	300	2	1

We can plot some measures using the randomForestExplainer package. In this case, since we are not predicting performance, analizing the importance of parameters not including the instance would be an error.

```
plot_multi_way_importance(importance_frame, size_measure = "p_value",
                          y_measure="accuracy_decrease", x_measure="mean_min_depth",
                          no_of_labels=7)
```

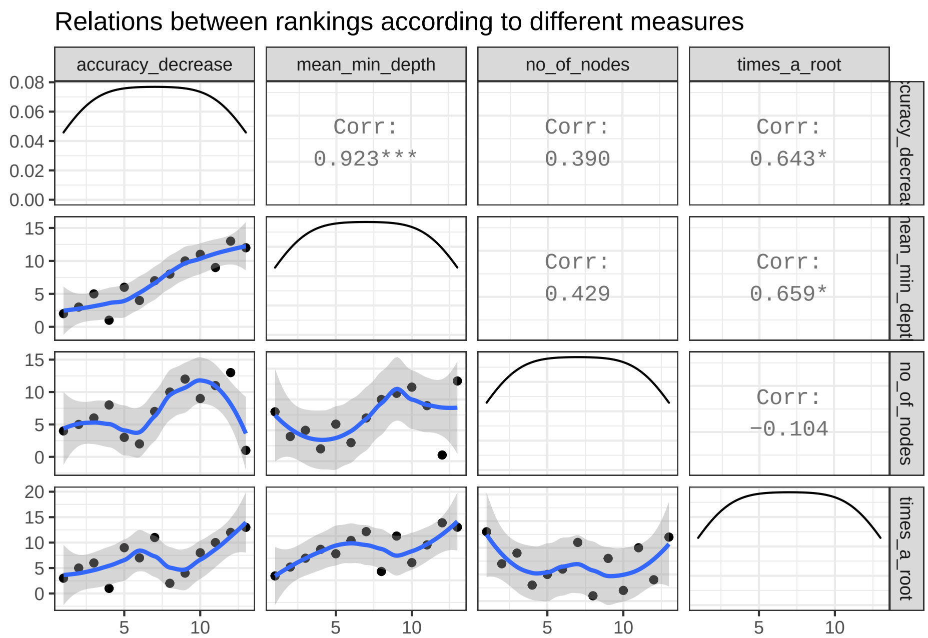
```
## Warning: Using alpha for a discrete variable is not advised.
```



We can also visualize the relationship between importance measures

```
#plot_importance_ggpairs(importance_frame)
plot_importance_rankings(importance_frame, measures=c("accuracy_decrease", "mean_min_depth", "no_of_node
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



After the analysis of conditional parameters the most important 5 parameters are (plus reference added for interaction analysis):

```r
print(important_parameters)
```

```
## [1] "q0"    "ants"  "rho"   "beta"  "dummy"
```

We run the interaction analysis and find the importance of interactions:

```r
kable(full_interactions_frame)  %>%  kable_styling(latex_options="scale_down")
```

variable	root_variable	mean_min_depth	occurrences	interaction	uncond_mean_min_depth
ants	q0	0.9932660	297	q0:ants	1.2200000
q0	q0	0.8989899	297	q0:q0	0.8933333
rho	q0	1.1212121	297	q0:rho	1.6233333
beta	q0	0.8831650	296	q0:beta	1.6133333
dummy	q0	1.7764310	295	q0:dummy	2.6100000
beta	beta	2.3166554	280	beta:beta	1.6133333
beta	rho	2.0961279	280	rho:beta	1.6133333
beta	ants	2.2905724	279	ants:beta	1.6133333
rho	ants	2.3309764	279	ants:rho	1.6233333
ants	beta	2.8853199	276	beta:ants	1.2200000
q0	rho	2.4550505	276	rho:q0	0.8933333
rho	rho	2.8085859	276	rho:rho	1.6233333
ants	ants	2.8598204	275	ants:ants	1.2200000
rho	beta	2.7522334	275	beta:rho	1.6233333
q0	ants	2.8363636	273	ants:q0	0.8933333
dummy	ants	3.3700898	272	ants:dummy	2.6100000
q0	beta	3.1761167	271	beta:q0	0.8933333
ants	rho	3.0405724	270	rho:ants	1.2200000
dummy	beta	3.6625589	270	beta:dummy	2.6100000
dummy	rho	3.4119529	268	rho:dummy	2.6100000
beta	dummy	3.5158249	237	dummy:beta	1.6133333
rho	dummy	4.0573176	230	dummy:rho	1.6233333
dummy	dummy	5.3166891	220	dummy:dummy	2.6100000
q0	dummy	4.7113356	217	dummy:q0	0.8933333
ants	dummy	4.6722896	216	dummy:ants	1.2200000

Once we filter dummy interactions and aggregate bidirectional interactions we get a matrix where the importance of interactions are summarized:

```r
print(interactions_frame)
```

```
##   variable root_variable mean_min_depth occurrences interaction
## 1     ants            q0       0.993266         570     q0:ants
## 2      rho            q0       1.121212         573      q0:rho
## 3     beta            q0       0.883165         567     q0:beta
##   uncond_mean_min_depth
## 1              1.220000
## 2              1.623333
## 3              1.613333
```