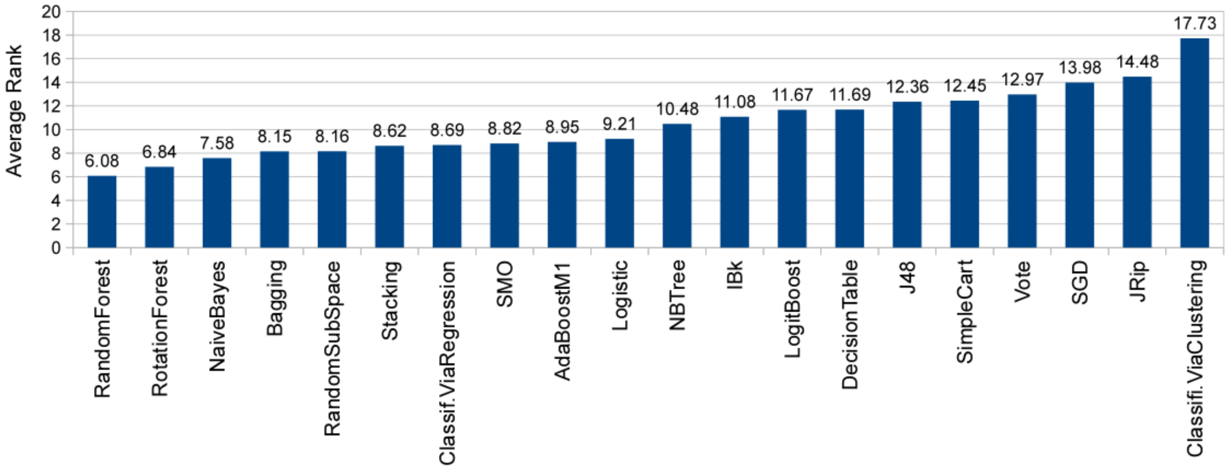
Supplementary Material

Quan Sun and Bernhard Pfahringer. Pairwise Meta-Rules for Better Meta-Learning-Based Algorithm Ranking. Machine Learning, Springer, 2013.

Algorithm Ranking List

(Based on PSO optimized performances (AUC) over 466 datasets)



Pearson's correlation scores between the ranking positions of each pair of algorithms over 466 dataset

	AdaBoos	Bagging	Classifica	Classifica	Decision'	IBk	J48	JRip	Logistic	LogitBoo	NBTree	NaiveBay	RandomI	Random!	Rotation	SGD	SMO	SimpleCa	Stacking	Vote
AdaBoostM1	1.0000	0.4856	-0.1958	-0.0571	-0.0838	-0.3377	-0.0191	-0.0863	-0.4139	0.2142	-0.1746	-0.1316	0.2232	0.0651	0.0043	-0.2832	-0.1906	0.1181	0.0337	-0.2755
Bagging	0.4856	1.0000	-0.2515	0.0232	-0.0686	-0.3820	0.0222	-0.0638	-0.3308	0.4197	-0.1544	-0.1654	0.2290	0.0785	-0.1880	-0.3659	-0.3448	0.2831	0.0049	-0.3061
ClassificationViaClus	-0.1958	-0.2515	1.0000	-0.1887	-0.0501	0.2699	0.0563	0.1364	0.1366	-0.2046	0.0000	-0.0484	-0.3017	-0.2177	-0.2626	0.3026	-0.0233	0.0500	-0.1970	0.2531
ClassificationViaRegı	-0.0571	0.0232	-0.1887	1.0000	0.0334	-0.0777	-0.1899	-0.2390	0.2497	0.1341	-0.0212	-0.0684	-0.0577	-0.1092	-0.0983	-0.1425	-0.1942	-0.0686	0.0958	-0.2161
DecisionTable	-0.0838	-0.0686	-0.0501	0.0334	1.0000	-0.1881	0.0717	0.0789	-0.1207	0.0853	0.2710	-0.2144	-0.0522	0.0672	-0.0736	-0.1668	-0.2628	-0.1609	0.0742	-0.1248
IBk	-0.3377	-0.3820	0.2699	-0.0777	-0.1881	1.0000	-0.1810	-0.0686	0.2980	-0.3333	-0.0745	0.2634	-0.2810	-0.2337	0.0328	0.2273	0.2184	-0.1697	-0.1703	0.0311
J48	-0.0191	0.0222	0.0563	-0.1899	0.0717	-0.1810	1.0000	0.3273	-0.1718	0.0088	-0.1245	-0.2555	-0.0317	0.0160	-0.1556	-0.1003	-0.2330	0.2396	-0.0257	-0.0165
JRip	-0.0863	-0.0638	0.1364	-0.2390	0.0789	-0.0686	0.3273	1.0000	-0.0888	-0.0219	-0.0121	-0.2303	-0.0417	-0.0967	-0.1841	-0.0351	-0.1751	0.1544	-0.2173	0.1685
Logistic	-0.4139	-0.3308	0.1366	0.2497	-0.1207	0.2980	-0.1718	-0.0888	1.0000	-0.2154	-0.0550	0.0504	-0.3841	-0.2941	-0.0388	0.2593	0.1703	-0.0732	-0.2296	0.0954
LogitBoost	0.2142	0.4197	-0.2046	0.1341	0.0853	-0.3333	0.0088	-0.0219	-0.2154	1.0000	-0.0525	-0.3105	0.0248	0.0821	-0.2401	-0.2809	-0.3324	0.1361	0.0954	-0.2476
NBTree	-0.1746	-0.1544	0.0000	-0.0212	0.2710	-0.0745	-0.1245	-0.0121	-0.0550	-0.0525	1.0000	0.0286	0.0006	0.0245	-0.0240	-0.0822	-0.1049	-0.1733	-0.0921	-0.0253
NaiveBayes	-0.1316	-0.1654	-0.0484	-0.0684	-0.2144	0.2634	-0.2555	-0.2303	0.0504	-0.3105	0.0286	1.0000	0.0269	-0.0203	0.0677	0.0608	0.2463	-0.1812	-0.1926	-0.0098
RandomForest	0.2232	0.2290	-0.3017	-0.0577	-0.0522	-0.2810	-0.0317	-0.0417	-0.3841	0.0248	0.0006	0.0269	1.0000	0.2912	0.1102	-0.3904	-0.0915	-0.1327	0.0170	-0.1548
RandomSubSpace	0.0651	0.0785	-0.2177	-0.1092	0.0672	-0.2337	0.0160	-0.0967	-0.2941	0.0821	0.0245	-0.0203	0.2912	1.0000	0.0073	-0.3225	-0.2553	-0.0692	0.1210	-0.0713
RotationForest	0.0043	-0.1880	-0.2626	-0.0983	-0.0736	0.0328	-0.1556	-0.1841	-0.0388	-0.2401	-0.0240	0.0677	0.1102	0.0073	1.0000	-0.0629	0.2611	-0.3276	0.0570	-0.1169
SGD	-0.2832	-0.3659	0.3026	-0.1425	-0.1668	0.2273	-0.1003	-0.0351	0.2593	-0.2809	-0.0822	0.0608	-0.3904	-0.3225	-0.0629	1.0000	0.3069	-0.1040	-0.1827	0.4929
SMO	-0.1906	-0.3448	-0.0233	-0.1942	-0.2628	0.2184	-0.2330	-0.1751	0.1703	-0.3324	-0.1049	0.2463	-0.0915	-0.2553	0.2611	0.3069	1.0000	-0.2455	-0.0884	0.1325
SimpleCart	0.1181	0.2831	0.0500	-0.0686	-0.1609	-0.1697	0.2396	0.1544	-0.0732	0.1361	-0.1733	-0.1812	-0.1327	-0.0692	-0.3276	-0.1040	-0.2455	1.0000	-0.0825	-0.0392
Stacking	0.0337	0.0049	-0.1970	0.0958	0.0742	-0.1703	-0.0257	-0.2173	-0.2296	0.0954	-0.0921	-0.1926	0.0170	0.1210	0.0570	-0.1827	-0.0884	-0.0825	1.0000	-0.1979
Vote	-0.2755	-0.3061	0.2531	-0.2161	-0.1248	0.0311	-0.0165	0.1685	0.0954	-0.2476	-0.0253	-0.0098	-0.1548	-0.0713	-0.1169	0.4929	0.1325	-0.0392	-0.1979	1.0000

The highest positive correlation score (+0.4929) is between Vote and SGD;
The lowest negative correlation score (-0.4139) is between AdaBoostM1 and Logistic.

The correlation scores are all between -0.5 and +0.5 partially implying that different learning algorithms have their own expertise on different problems (datasets), this was a motivation for our meta-learning research.

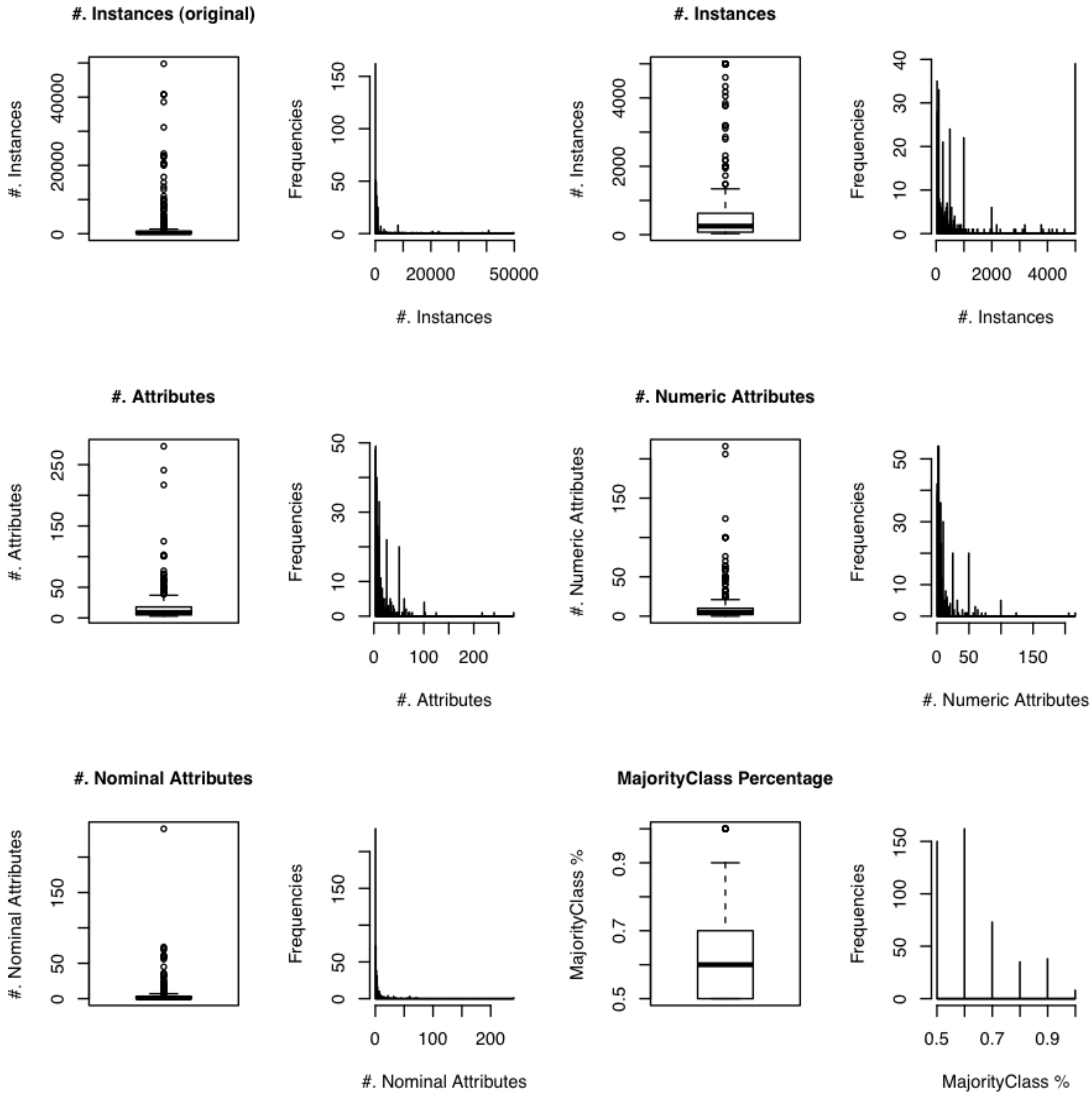
Algorithms/Parameters Table (20 WEKA algorithms)

PSO parameters used for experiments
SwarmSize: 10
numEvolutions: 10
objective function: 1x10-fold CV AUC performance

Algorithm	**Parameters for PSO to tune**
AdaBoostM1	I {10, 30, 100} H {1.0, 0.8, 0.6, 0.4, 0.2} Q {True, False} REPTree depth {-1 to 6}
Bagging	I {10, 30, 100} REPTree depth {-1 to 6} REPTree prune {True, False}
ClassificationViaClustering	numClusters {2 to 30}
ClassificationViaRegression	M5P (with different parameters) REPTree (with different parameters) LinearRegression (with different parameters)
DecisionTable	Evaluation measure {Acc,RMSE,MAE,AUC} numFolds {1 to 5}
IBk	K {1 to numInstances - 1}
J48	BinarySplits {True, False} ConfFactor {0.00001 to 0.5} minInstances {2 to 0.5 * numInstances}
JRip	Folds {3 to 10} minNo {2 to 10} pruning {True, False}
Logistic	R {0.0000001 to 100000.0} M {-1, 300, 500, 1000}
LogitBoost	REPTree depth {-1 to 6} H {0.2 to 1.0} I {10, 30, 100}
NBTree	D {True, False}
NaiveBayes	K {True, False}
RandomForest	I {10, 30, 100} K {0 to numAtts}
RandomSubSpace	I {10, 30, 100} P {0.1 to 0.9} REPTree depth {-1 to 6}
RotationForest	I {10, 30, 100} P {50 t0 90} RandomProjection {Gaussian, sparse1, sparse2}

SGD	F {log loss, hinge loss} N {True, False} M {True, False} R {0.00001 to 100.0} E {100, 500, 1000}
SMO	C {0.00001 to 10000.0} N {0,1,2} kernel {PolyKernel (E 1.0, 2.0), RBF(G 0.0001 to 1000.0)}
SimpleCart	H {True, False} M {2 to 0.3 x numInstances} U {True, False}
Stacking	Meta-classifier: LinearRegression X {2 to 5} 10 base classifiers: NaiveBayes Logistic SMO SGD IBk J48 NBTree SimpleCart DecisionTable JRip
Vote	R {AVG,PROD,MAJ,MIN,MAX,MED} 10 base classifiers: NaiveBayes Logistic SMO SGD IBk J48 NBTree SimpleCart DecisionTable JRip

Properties of the 466 datasets used for experiments



The 80 SIL MetaFeatures used for experiments

1 RandomTreeDepth1AUC_K=0
2 RandomTreeDepth2AUC_K=0
3 RandomTreeDepth3AUC_K=0
4 RandomTreeDepth1AUC_K=0
5 RandomTreeDepth2AUC_K=0
6 RandomTreeDepth3AUC_K=0
7 RandomTreeDepth1ErrRate_K=1
8 RandomTreeDepth1Kappa_K=1
9 RandomTreeDepth2ErrRate_K=1
10 RandomTreeDepth2Kappa_K=1
11 RandomTreeDepth3ErrRate_K=1
12 RandomTreeDepth3Kappa_K=1
13 RandomTreeDepth1ErrRate_K=1
14 RandomTreeDepth1Kappa_K=1
15 RandomTreeDepth2ErrRate_K=1
16 RandomTreeDepth2Kappa_K=1
17 RandomTreeDepth3ErrRate_K=1
18 RandomTreeDepth3Kappa_K=1
19 DecisionStumpErrRate
20 DecisionStumpAUC
21 NBErrRate
22 NBAUC2fCV
23 NBAUC5fCV
24 NBAUC10fCV
25 REPTreeDepth1ErrRate
26 REPTreeDepth1AUC
27 REPTreeDepth2ErrRate
28 REPTreeDepth2AUC
29 REPTreeDepth3ErrRate
30 REPTreeDepth3AUC
31 REPTreeDepth1Kappa
32 REPTreeDepth2Kappa
33 REPTreeDepth3Kappa
34 J48.00001.ErrRate
35 J48.00001.AUC
36 J48.0001.ErrRate
37 J48.0001.AUC
38 J48.001.ErrRate
39 J48.001.AUC
40 J48.00001.kappa
41 J48.0001.kappa
42 J48.001.kappa
43 ObliviousTree.depth.1.ErrRate

44	ObliviousTree.depth.1.AUC
45	ObliviousTree.depth.2.ErrRate
46	ObliviousTree.depth.2.AUC
47	ObliviousTree.depth.3.ErrRate
48	ObliviousTree.depth.3.AUC
49	ObliviousTree.depth.1.kappa
50	ObliviousTree.depth.2.kappa
51	ObliviousTree.depth.3.kappa
52	MeanMeansOfNumericAtts
53	MeanStdDevOfNumericAtts
54	MeanKurtosisOfNumericAtts
55	MeanSkewnessOfNumericAtts
56	NumAttributes
57	Dimensionality
58	NumNominalAtts
59	NumNumericAtts
60	PercentageOfNominalAtts
61	PercentageOfNumericAtts
62	NumBinaryAtts
63	PercentageOfBinaryAtts
64	ClassCount
65	PositivePercentage
66	NegativePercentage
67	DefaultAccuracy
68	IncompleteInstanceCount
69	InstanceCount
70	NumMissingValues
71	PercentageOfMissingValues
72	MaxNominalAttDistinctValues
73	MinNominalAttDistinctValues
74	MeanNominalAttDistinctValues
75	StdvNominalAttDistinctValues
76	ClassEntropy
77	MeanAttributeEntropy
78	MeanMutualInformation
79	EquivalentNumberOfAtts
80	NoiseToSignalRatio

Appendix: A brief derivation of Eq. 11

Here we give the core steps of the derivation. For a more detailed derivation of the intermediate steps, we refer the reader to [2, 30].

Let Q_m be the set of all $m \times m$ permutation matrices, where m is the number of objects (e.g., algorithms) to rank. The sets $\mathscr{S}_m$ (in Eq. 4) and Q_m are in one-to-one correspondence:

$$Q^{(y)} e_m = y, Q^{(y)} \in Q_m, y \in \mathscr{S}_m,$$

where $e'_m = [1, 2, 3, \ldots, m]$. For example, $Q_{ij} = 1$ iff $y_i = j$.

Let d be a Hoeffding distance of the form,

$$d(x,y) = \sum_{i=1}^{m} a(x_i, y_i) = tr(Q^{(x)} \Delta Q^{(y)\prime}).$$

For Spearman distance (which is a form of Hoeffding distance):

$$\Delta = f_m 1'_m + 1_m f'_m - 2 e_m e'_m,$$

where $f'_m = [1, 2^2, 3^2, \ldots, m^2]$ and $1'_m = [1, 1, 1, \ldots, 1]$.

The spread in Eq. 11 is calculated as:

$$\hat{\alpha}_{Spearman} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} d(y^{(i)}, y^{(j)})}{n^2} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} tr(Q_i \Delta Q'_j)}{n^2} = tr(\hat{M} \Delta \hat{M}'),$$

where $\hat{M}$ is the marginal matrix ([30]) and $\hat{M} 1_m = \hat{M}' 1_m = 1_m$. Since it is well know that the sum of the squares of integers $1^2 + 2^2 + 3^2 + \ldots + m^2 = \frac{m(m+1)(2m+1)}{6}$, therefore

$$\hat{\alpha}_{Spearman} = 2(1'_m f_m) - 2(\hat{M} e_m)'(\hat{M} e_m) = \frac{m(m+1)(2m+1)}{3} - 2||\bar{y}||^2.$$

2. Mayer Alvo, Paul Cabilio, and Paul D. Feigin. Asymptotic theory for measures of concordance with special reference to average kendall tau. *The Annals of Statistics*, 10(4):pp. 1269–1276, 1982.

30. John I. Marden. *Analyzing and Modeling Rank Data*. Chapman & Hall, 1995.