Diversifying Support Vector Machines for Boosting using Kernel Perturbation: An Application to Class Imbalanced Learning
Supplementary material
(for online publication only)

Shounak Datta, Sayak Nag, Sankha Subhra Mullick, Swagatam Das, and Ivan Zelinka

I. Transformation Factors used by Kernel Perturbation Methods

We list the various transformation factors employed by the kernel perturbation techniques in Table I.

TABLE I
BRIEF DESCRIPTION OF THE TRANSFORMATION FACTORS USED BY DIFFERENT KERNEL PERTURBATION TECHNIQUES

Algorithm	Point-specific transformation factor $D(\mathbf{x})$	Remarks
Amari and Wu [1]	$\sum_{\mathbf{x}_i \in V} \lambda_i \exp\Big\{\Big(-\frac{\lvert\lvert\mathbf{x}-\mathbf{x}_i\rvert\rvert^2}{2\tau^2}\Big)\Big\}$	λ_i determines the relative importance of each SV, while τ controls the decay of D as one moves away from SV.
Wu and Amari [2]	$\sum_{\mathbf{x}_i \in V} \exp\Big\{\Big(-\frac{\lvert\lvert\mathbf{x}-\mathbf{x}_i\rvert\rvert^2}{\tau_i^2}\Big)\Big\}$	τ_i accounts for the density of the other SVs around the SV $\mathbf{x}_i$.
Wu and Chang [3]	$\sum_{\mathbf{x}_i \in V^p} \exp\Big\{\Big(-\frac{\lvert\lvert\mathbf{x}-\mathbf{x}_i\rvert\rvert^2}{\eta_p\tau_i^2}\Big)\Big\} + \sum_{\mathbf{x}_i \in V^n} \exp\Big\{\Big(-\frac{\lvert\lvert\mathbf{x}-\mathbf{x}_i\rvert\rvert^2}{\eta_n\tau_i^2}\Big)\Big\}$	V^p and V^n respectively denotes the sets of SVs for the positive and negative class, while η_p and η_n (where, $\eta_p > \eta_n$) controls the corresponding asymmetric decay of D.
Wu and Change [4], [5]	$\frac{1}{\lvert\chi_b\rvert} \sum_{\mathbf{x}_b \in \chi_b} \exp\left(-\frac{\lvert\lvert\Phi(\mathbf{x})-\Phi(\mathbf{x}_b)\rvert\rvert^2}{\tau_b^2}\right)$	χ_b is the set of interpolated boundary points $\mathbf{x}_b$ which are the linear combinations of SVs. The nature of the parameter τ_b is similar to the τ_i in [2].
Williams *et al.* [6]	$\exp\big(-kf(\mathbf{x})^2\big)$	k controls the decay of D.
Maratea *et al.* [7], [8]	$\exp\big(-k_+f(\mathbf{x})^2\big)$ if $f(\mathbf{x}) \geq 0$ $\exp\big(-k_-f(\mathbf{x})^2\big)$ if $f(\mathbf{x}) < 0$	k_+ controls the decay of D for the positive class. k_- controls the decay of D for the negative class, and $0 < k_+ < k_-$.
Zhang *et al.* [9]	$\exp\big(-k_if(\mathbf{x})^2\big)$	k_i, determined using the χ^2 statistic, controls the decay of D for the i^{th} class.
Xiong *et al.* [10]	$\alpha_0^* + \sum_{\mathbf{x}_i \in P} \alpha_i^* \exp\big(-\gamma_i\lvert\lvert\mathbf{x}-\mathbf{x}_i\rvert\rvert^2\big)$	$\boldsymbol{\alpha}^*$ chosen so as to maximize the Fisher criterion $\frac{\alpha^T\Sigma_b\alpha}{\alpha^T\Sigma_w\alpha}$ in the feature space, while γ_i controls the decay of D.

Abbreviations and notations follow form the original article.

Shounak Datta, is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. Sayak Nag is with the Center for Atmospheric Sciences, Indian Institute of Technology, Delhi, India. Sankha Subhra Mullick and Swagatam Das are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India. Ivan Zelinka is with the Department of Computer Science, VSB - Technical University of Ostrava, Ostrava, Czech Republic.
Corresponding author: Swagatam Das (Email: swagatam.das@isical.ac.in)

II. Description of datasets

A brief description highlighting the key properties of the datasets used in our experiments is presented in Table II (the alias used for each dataset in the subsequent tables is also reported). We have used a total of 52 two-class (42 low dimensional while the rest are high dimensional i.e. having more than 500 dimensions), and 21 multi-class (12 low dimensional and 9 high dimensional) datasets in this study. All the datasets are normalized so that each feature has zero mean and unit standard deviation. The chosen datasets (other than those from ImageNet) retain their original names with the suffixed numerals (if any) denoting either the target class only (in which case all the rest of the classes are combined together to form the non-target class) or the target as well as non-target classes [11]. Apart from this some datasets required special construction and/or processing as listed in the following.

1) mnist_2vs17 is constructed by randomly picking 100 images from the Special Dataset 1 and 500 images from the Special Dataset 3 [12] for each of the three classes 1, 2 and 7. mnist_6vs09 is also constructed in a similar manner.
2) rcv1_1vs36vs245 is generated from Reuters RCV1 by randomly choosing 50 points from each class and using those of C15 as the first class, combining those of E21 and M11 to form the second class, and the rest as the third class.
3) We prepare 4 two-class and 6 multi-class dataset from ImageNet (2011 fall release) natural image classification database [13]. We download images from 8 primary subtrees or classes (plants, geological forms, natural objects, sports, artifacts, fungus, person and animal) and four leaves under the Miscellaneous branch (foods, microbes, collections and documents) ensuring every category to contain instances amounting at least 20 and at most 2% of the number of synsets in the corresponding subtree. Instead of using raw images, a state-of-the-art feature representation is employed to express an image in the form of a 2048-dimensional real valued vector. We use the output of the final global average pooling layer of Inception-v3 [14] deep learning network for the purpose. The two class datasets are created by combining images from two chosen ImageNet classes and named accordingly. imageNet3A (animal, artifacts and foods), imageNet3b (plant, artifacts and documents), imageNet4 (plants, artifacts, foods and documents), and imageNet8 (8 principal subtrees) are constructed by uniting instances from select classes. imageNet9 and imageNet12 are formed by incorporating all the downloaded images. However, in imageNet9, images from Miscellaneous branch are not distinctly labeled by the corresponding leaf name contrary to imageNet12.
4) breastcancer2 contains microRNA profiling of tissue samples collected immediately following surgery and 30 minutes after surgery from 14 patients only half of whom went through a radiotherapy treatment. A detailed description of the collection and processing of this dataset can be found in [15].

III. Notes on cross-validation

We have used a 10-fold stratified cross-validation such that the proportion of the number of representatives from each of the classes in the original dataset (i.e. class imbalance) is also conserved in each of the folds, which consequently preserves the same in the training and set sets. To elaborate we ensure that the difference in the number of points from any class between any pair of folds is ideally zero and can be at most one. If a dataset contains less than 10 points for a minority class, then random sampling with replacement is used to conserve the original class imbalance.

IV. Parameter optimization

The optimization procedure of different parameters associated with the contending methods are detailed in Table III.

V. Detailed results

The detailed performance of the proposed KPBoost-SVM against the rest of the state-of-the-art contenders (namely SVM, SVMBoost, KBSVM, RUSBoost, AKS, AKS-χ^2, AdaBoost-MLP, KBA, ACT, and OCTK) in terms of Gmean and AUROC on the two-class as well as multi-class datasets are detailed in Tables IV-VII.

TABLE II
SUMMARY OF DATASET PROPERTIES.

Dataset	Alias	Number of points	Number of dimensions	Number of classes	Imbalance Ratio
abalone19	TMD1	4177	8	2	129.53
abalone3vs11	TMD2	502	8	2	32.46
abalone9vs18	TMD3	731	8	2	16.40
banana	TMD4	1213	2	2	4.09
car3	TMD5	1728	6	2	24.04
cleveland0vs4	TMD6	177	13	2	12.61
ecoli0137vs26	TMD7	281	7	2	39.14
ecoli0vs1	TMD8	220	7	2	1.85
ecoli1	TMD9	336	7	2	3.36
ecoli2	TMD10	336	7	2	5.46
ecoli3	TMD11	336	7	2	8.60
ecoli4	TMD12	336	7	2	15.80
glass015vs2	TMD13	172	9	2	9.11
glass016vs2	TMD14	192	9	2	10.29
glass04vs5	TMD15	92	9	2	9.22
glass06vs5	TMD16	108	9	2	11.00
glass0	TMD17	213	9	2	2.04
glass123vs456	TMD18	214	9	2	3.19
glass1	TMD19	214	9	2	1.81
glass4	TMD20	214	9	2	15.46
glass5	TMD21	214	9	2	22.77
glass6_aka_glass7	TMD22	214	9	2	6.37
iris12vs3	TMD23	150	4	2	2.00
lymphography_nf	TMD24	148	18	2	23.66
pageblocks13vs4	TMD25	472	10	2	15.85
pima	TMD26	768	8	2	1.86
poker89vs6	TMD27	1485	10	2	58.40
poker8vs6	TMD28	1477	10	2	85.88
shuttle6vs23	TMD29	230	9	2	22.00
shuttleC2vsC4	TMD30	129	9	2	20.50
soybean12	TMD31	683	35	2	14.52
thyroid1	TMD32	215	5	2	5.14
vehicle0	TMD33	846	18	2	3.25
vowel0	TMD34	988	13	2	9.97
winequality_red3vs5	TMD35	691	11	2	68.10
winequality_white3vs7	TMD36	900	11	2	44.00
winequality_white9vs4	TMD37	168	11	2	32.60
yeast0359vs78	TMD38	506	8	2	9.12
yeast0569vs4	TMD39	528	8	2	9.35
yeast1458vs7	TMD40	693	8	2	22.10
yeast2vs4	TMD41	514	8	2	9.07
zoo3	TMD42	101	16	2	19.20
CNAE9_2	THD1	1080	856	2	8.00
CNAE9_35vs6789	THD2	360	856	2	2.00
CNAE9_3vs4567	THD3	480	856	2	3.00
breastcancer2	THD4	28	2227	2	3.00
imageNet_docs_artifacts	THD5	440	2048	2	21.00
imageNet_food_docs	THD6	80	2048	2	3.00
imageNet_plants_artifacts	THD7	600	2048	2	2.33
imageNet_plants_docs	THD8	200	2048	2	9.00
mnist_2vs17	THD9	1800	784	2	2.00
mnist_6vs09	THD10	1800	784	2	2.00
balance	MMD1	625	4	3	5.87
dermatology	MMD2	366	34	6	5.60
ecoli	MMD3	336	7	8	71.50
glass	MMD4	214	10	6	8.44
hayes	MMD5	132	4	3	1.70
lymphography	MMD6	148	18	4	40.50
new-thyroid	MMD7	215	5	3	5.00
pageblocks	MMD8	548	10	5	164.00
shuttle	MMD9	2175	9	5	853.00
thyroid	MMD10	720	21	3	39.17
wine	MMD11	178	13	3	1.47
yeast	MMD12	1484	8	10	92.60
CNAE9_15vs249vs3vs6vs78	MHD1	1080	856	5	3.00
CNAE9_1vs2vs34vs56vs789	MHD2	1080	856	5	3.00
imageNet12	MHD3	1240	2048	12	21.00
imageNet8	MHD4	1120	2048	8	21.00
imageNet9	MHD5	1240	2048	9	21.00
imageNet3A	MHD6	640	2048	3	7.00
imageNet3B	MHD7	620	2048	3	21.00
imageNet4	MHD8	680	2048	4	21.00
rcv1_1vs36vs245	MHD9	300	21513	3	3.00

TABLE III
LIST OF CONTENDING ALGORITHMS ALONG WITH THEIR PARAMETER SETTINGS

Contending algorithm	Base classifier	Detailed parameter settings
Baseline method:		
SVM [16]	-	$C_r \in \mathbb{C}$; $\sigma \in S_1$.
Boosting based techniques:		
SVMBoost [17]	SVM	$C_r \in \mathbb{C}$; $\sigma \in S_3$; $T = 10$.
RUSBoost [18], [19]	C4.5 [20]	In two-class datasets the majority class is under-sampled by 35%, 50%, and 65% as recommended in [19], while in multi-class problems equal number of training representatives are sampled from each class. $T = 10$.
KBSVM [21]	SVM	$\sigma_{step} \in \{1, 2, 3\}$; $T = 40$, both as per [21]
AdaBoost-MLP [22]	MLP	The number of hidden nodes is varied in the set $\{10, 15, 20\}$, $T = 10$.
Kernel perturbation methods:		
KBA [5]	SVM	$C_r \in \mathbb{C}$; $\sigma \in S_3$, $T = 10$.
ACT [3]	SVM	$C_r \in \mathbb{C}$; $\sigma \in S_3$, $T = 10$. All other parameters are set according to the original article.
AKS [7], [8]	SVM	$k_+ \in S_2$; $k_- \in S_2$; $C_r \in \mathbb{C}$; $\sigma \in S_3$, $T = 10$.
AKS-χ^2 [9]	SVM	$C_r \in \mathbb{C}$; $\sigma \in S_3$. $T \in \{2, 3, 5\}$ is set to avoid over-fit.
OCTK [10]	-	$\sigma_0 \in S_3$, and $\gamma_1 \in \{0.01, 0.1, 1, 10\}$ along the lines of [10].
Proposed method:		
KPBoost-SVM	SVM	$\varsigma \in S_2$; $C_r \in \mathbb{C}$; $\sigma \in S_3$; $T = 10$.

[1] $\mathbb{C} = \{10^2, 10^3\}$
[2] $S_1 = \{[0.01 : 0.01 : 0.09] \cup [0.1 : 0.1 : 0.9] \cup [1 : 1 : 9] \cup [10 : 10 : 100] \cup \{200\}\}$.
[3] $S_2 = \{[1 \times 10^{-4} : 1 \times 10^{-4} : 9 \times 10^{-4}] \cup [1 \times 10^{-3} : 1 \times 10^{-3} : 9 \times 10^{-3}] \cup [0.01 : 0.01 : 0.09] \cup [0.1 : 0.1 : 1]\}$
[4] $S_3 = \{\frac{\sigma^g}{2}, \frac{3\sigma^g}{4}, \sigma^g, \frac{5\sigma^g}{4}, \frac{3\sigma^g}{2}\} \cup \{\frac{\sigma^a}{2}, \frac{3\sigma^a}{4}, \sigma^a, \frac{5\sigma^a}{4}, \frac{3\sigma^a}{2}\}$; where σ^g is the best σ in terms of Gmean while σ^a is the same for AUROC observed (over 10-fold cross validation) for the baseline SVM on the dataset in question.
[5] All SVM implementations use an RBF kernel unless otherwise specified. All parameters not exclusively mentioned in the table are tuned as per recommendations from the corresponding original article.

TABLE IV
RESULTS ON TWO-CLASS DATASETS IN TERMS OF GMEAN.

Datasets	KPBoost-SVM	SVM	SVMBoost	KBSVM	AKS	AKS-χ^2	RUSBoost	AdaBoost-MLP	KBA	ACT	OCTK
TMD1	0.424	0.157	0.157	0.080	**0.559**	0.156	0.073	0.000	0.075	0.510	0.026
TMD2	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.000	**1.000**	**1.000**	**1.000**	0.763
TMD3	**0.813**	0.604	0.607	0.400	0.634	0.473	0.522	0.694	0.533	0.381	0.329
TMD4	**0.902**	0.806	0.864	0.000	0.748	0.708	0.870	0.866	0.575	0.113	0.160
TMD5	0.985	0.969	0.857	0.861	**0.998**	0.977	0.950	0.976	0.933	0.901	0.678
TMD6	0.877	0.874	0.833	0.000	**0.905**	0.904	0.855	0.830	0.781	0.863	0.449
TMD7	0.859	0.885	0.883	0.198	0.883	0.881	0.735	0.880	**0.893**	0.883	0.883
TMD8	**0.977**	0.946	0.953	0.960	0.954	0.932	0.976	0.966	0.758	0.946	0.894
TMD9	**0.901**	0.843	0.851	0.343	0.775	0.483	0.862	0.841	0.741	0.000	0.478
TMD10	0.924	0.906	0.907	0.197	**0.932**	0.770	0.879	0.861	0.740	0.000	0.159
TMD11	**0.853**	0.732	0.791	0.013	0.784	0.273	0.728	0.743	0.761	0.400	0.677
TMD12	0.971	0.803	0.914	0.000	**0.979**	0.729	0.886	0.881	0.386	0.718	0.471
TMD13	**0.853**	0.537	0.575	0.000	0.797	0.321	0.228	0.397	0.568	0.230	0.540
TMD14	0.726	0.427	0.415	0.000	**0.779**	0.420	0.000	0.252	0.160	0.096	0.439
TMD15	**1.000**	**1.000**	**1.000**	0.141	**1.000**	0.931	0.723	0.794	0.891	0.341	0.729
TMD16	**1.000**	**1.000**	**1.000**	0.141	1.000	0.985	0.936	0.995	0.755	0.437	0.821
TMD17	0.814	0.771	0.771	0.087	0.801	0.607	0.346	0.725	0.682	**0.842**	0.624
TMD18	**0.928**	0.824	0.817	0.807	0.673	0.720	0.825	0.907	0.739	0.344	0.882
TMD19	**0.722**	0.708	0.700	0.000	0.709	0.712	0.105	0.687	0.638	0.636	0.507
TMD20	0.868	0.693	0.868	0.257	0.868	**0.933**	0.495	0.859	0.596	0.896	0.721
TMD21	**0.995**	0.941	0.956	0.000	**0.995**	0.281	0.528	0.795	0.804	0.409	0.875
TMD22	**0.938**	0.901	0.901	0.930	0.932	0.931	0.913	0.910	0.878	0.929	0.881
TMD23	**0.997**	0.954	0.985	0.980	0.958	0.615	0.766	0.932	0.631	0.760	0.603
TMD24	**1.000**	0.804	0.800	0.741	**1.000**	0.975	0.528	0.608	0.541	0.800	0.920
TMD25	0.998	0.896	0.896	0.127	0.958	0.974	0.797	**0.999**	0.749	0.981	0.624
TMD26	**0.738**	0.727	0.714	0.694	0.603	0.720	0.569	0.736	0.561	0.427	0.618
TMD27	**0.994**	0.863	0.863	0.007	0.931	0.794	0.000	0.894	0.741	0.225	0.523
TMD28	**1.000**	**1.000**	**1.000**	0.200	**1.000**	0.725	0.000	0.899	0.668	0.000	0.232
TMD29	**1.000**	**1.000**	**1.000**	0.483	**1.000**	0.938	0.791	0.941	0.883	**1.000**	0.772
TMD30	**1.000**	0.800	0.800	0.196	**1.000**	0.987	0.000	0.800	0.807	**1.000**	0.396
TMD31	0.990	0.977	0.977	0.893	0.995	0.977	**1.000**	0.954	0.977	0.977	0.661
TMD32	0.985	0.956	0.966	0.789	**0.996**	0.260	0.992	0.961	0.570	0.597	0.606
TMD33	**0.985**	0.982	0.975	0.589	0.984	0.460	0.918	0.983	0.546	0.342	0.934
TMD34	**1.000**	**1.000**	**1.000**	0.865	**1.000**	**1.000**	0.933	**1.000**	**1.000**	**1.000**	0.612
TMD35	**0.710**	0.155	0.141	0.000	0.485	0.140	0.605	0.140	0.087	0.000	0.000
TMD36	0.730	0.646	0.651	0.100	**0.745**	0.511	0.686	0.524	0.557	0.000	0.294
TMD37	**0.797**	**0.797**	**0.797**	0.000	**0.797**	0.591	0.749	0.597	0.400	0.791	**0.797**
TMD38	**0.585**	0.422	0.407	0.407	0.468	0.330	0.135	0.429	0.307	0.000	0.437
TMD39	**0.788**	0.556	0.555	0.458	0.501	0.549	0.395	0.612	0.604	0.000	0.564
TMD40	0.561	0.203	0.197	0.000	**0.660**	0.162	0.000	0.004	0.081	0.000	0.136
TMD41	**0.878**	0.821	0.812	0.152	0.497	0.657	0.871	0.861	0.634	0.629	0.577
TMD42	**0.989**	0.612	0.600	0.400	0.804	0.595	0.400	0.595	0.600	0.595	0.600
THD1	**0.999**	0.961	0.957	0.000	0.985	0.895	0.950	0.970	0.819	0.000	0.533
THD2	**1.000**	0.987	0.987	0.003	0.840	0.925	0.960	0.979	0.987	0.133	0.583
THD3	**1.000**	0.999	**1.000**	0.000	0.973	0.880	0.978	0.987	0.987	0.083	0.702
THD4	**0.618**	0.174	0.173	0.008	0.560	0.502	0.003	0.296	0.000	0.386	0.490
THD5	**1.000**	0.627	0.941	0.632	**1.000**	0.007	0.754	**1.000**	0.141	0.002	0.683
THD6	**1.000**	0.941	**1.000**	0.941	**1.000**	0.000	0.445	0.800	0.151	0.000	0.928
THD7	0.970	0.936	0.931	0.899	**0.979**	0.536	0.888	0.921	0.911	0.920	0.941
THD8	**1.000**	0.741	0.741	0.741	**1.000**	0.000	0.671	**1.000**	0.141	0.000	0.930
THD9	0.970	0.967	0.971	0.001	0.975	0.969	0.135	**0.988**	0.968	0.931	0.919
THD10	0.974	0.973	0.973	0.000	0.990	0.972	0.131	0.992	0.966	**0.995**	0.911

Best results are boldfaced.

TABLE V
RESULTS OF TWO-CLASS DATASETS IN TERMS OF AUROC.

Datasets	KPBoost-SVM	SVM	SVMBoost	KBSVM	AKS	AKS-χ^2	RUSBoost	AdaBoost-MLP	KBA	ACT	OCTK
TMD1	0.600	0.527	0.527	0.512	**0.644**	0.524	0.500	0.500	0.512	0.626	0.501
TMD2	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.500	**1.000**	**1.000**	**1.000**	0.866
TMD3	**0.823**	0.691	0.691	0.601	0.731	0.642	0.613	0.738	0.623	0.623	0.559
TMD4	**0.910**	0.823	0.869	0.500	0.789	0.765	0.871	0.869	0.587	0.543	0.521
TMD5	0.985	0.970	0.868	0.891	**0.998**	0.976	0.961	0.976	0.933	0.906	0.730
TMD6	0.891	0.877	0.875	0.500	**0.921**	0.909	0.875	0.849	0.819	0.877	0.633
TMD7	0.871	**0.915**	0.902	0.598	0.900	0.898	0.844	0.896	0.900	0.900	0.901
TMD8	0.977	0.948	0.954	0.954	0.955	0.938	**0.995**	0.967	0.813	0.948	0.899
TMD9	**0.893**	0.856	0.856	0.617	0.815	0.648	0.866	0.847	0.756	0.500	0.638
TMD10	0.927	0.901	0.901	0.597	**0.935**	0.809	0.885	0.876	0.773	0.500	0.520
TMD11	**0.868**	0.771	0.809	0.517	0.811	0.606	0.764	0.775	0.789	0.639	0.724
TMD12	0.978	0.838	0.924	0.500	**0.979**	0.775	0.899	0.892	0.622	0.823	0.683
TMD13	**0.856**	0.677	0.696	0.500	0.828	0.617	0.528	0.615	0.704	0.582	0.654
TMD14	0.765	0.633	0.633	0.500	**0.798**	0.641	0.500	0.579	0.546	0.519	0.587
TMD15	**1.000**	**1.000**	**1.000**	0.550	**1.000**	0.938	0.821	0.894	0.889	0.651	0.838
TMD16	**1.000**	**1.000**	**1.000**	0.550	**1.000**	0.985	0.945	0.995	0.850	0.642	0.847
TMD17	**0.819**	0.793	0.787	0.525	0.808	0.639	0.503	0.731	0.726	0.845	0.662
TMD18	**0.930**	0.838	0.838	0.848	0.746	0.773	0.844	0.912	0.766	0.644	0.886
TMD19	**0.743**	0.739	0.727	0.500	0.733	0.731	0.518	0.709	0.680	0.686	0.613
TMD20	0.883	0.800	0.883	0.583	0.883	0.935	0.707	0.873	0.752	**0.907**	0.762
TMD21	**0.995**	0.950	0.967	0.500	**0.995**	0.617	0.743	0.895	0.823	0.698	0.893
TMD22	**0.961**	0.910	0.910	0.935	0.938	0.936	0.935	0.923	0.888	0.927	0.891
TMD23	**0.990**	0.955	0.985	0.980	0.973	0.730	0.865	0.935	0.692	0.860	0.725
TMD24	**1.000**	0.900	0.900	0.850	**1.000**	0.991	0.736	0.797	0.750	0.900	0.922
TMD25	0.998	0.916	0.906	0.540	0.960	0.974	0.897	**0.999**	0.788	0.982	0.723
TMD26	**0.741**	0.734	0.734	0.716	0.662	0.723	0.640	**0.741**	0.658	0.637	0.626
TMD27	**0.994**	0.880	0.880	0.500	0.947	0.820	0.500	0.900	0.772	0.564	0.633
TMD28	**1.000**	**1.000**	**1.000**	0.600	**1.000**	0.769	0.500	0.908	0.783	0.500	0.555
TMD29	**1.000**	**1.000**	**1.000**	0.700	**1.000**	0.946	0.891	0.950	0.912	**1.000**	0.805
TMD30	**1.000**	0.900	0.900	0.596	**1.000**	0.988	0.500	0.900	0.900	**1.000**	0.696
TMD31	0.996	0.986	0.978	0.899	0.995	0.978	**0.999**	0.956	0.978	0.978	0.715
TMD32	0.986	0.959	0.972	0.814	0.989	0.583	**0.992**	0.952	0.671	0.797	0.678
TMD33	**0.998**	0.992	0.975	0.678	0.984	0.571	0.919	0.976	0.619	0.550	0.934
TMD34	**1.000**	**1.000**	**1.000**	0.874	**1.000**	**1.000**	0.936	**1.000**	**1.000**	**1.000**	0.683
TMD35	**0.750**	0.557	0.559	0.500	0.654	0.546	0.724	0.546	0.520	0.500	0.500
TMD36	0.776	0.724	0.724	0.532	**0.789**	0.669	0.735	0.674	0.684	0.500	0.530
TMD37	0.912	0.897	**0.916**	0.500	0.897	0.785	0.833	0.794	0.700	0.888	0.897
TMD38	0.621	0.607	0.607	0.607	**0.628**	0.588	0.509	0.613	0.590	0.513	0.561
TMD39	**0.801**	0.661	0.661	0.609	0.679	0.667	0.585	0.691	0.657	0.500	0.651
TMD40	**0.657**	0.546	0.547	0.500	0.691	0.531	0.500	0.500	0.514	0.500	0.530
TMD41	**0.903**	0.831	0.844	0.557	0.658	0.734	0.869	0.869	0.701	0.751	0.663
TMD42	**0.990**	0.802	0.800	0.707	0.908	0.795	0.679	0.795	0.800	0.795	0.795
THD1	**0.987**	0.958	0.958	0.500	0.985	0.902	0.950	0.971	0.837	0.500	0.670
THD2	**1.000**	0.988	0.988	0.504	0.872	0.927	0.971	0.979	0.988	0.525	0.727
THD3	**1.000**	0.988	0.999	0.500	0.974	0.888	0.981	0.988	0.988	0.523	0.726
THD4	**0.725**	0.568	0.557	0.500	0.685	0.595	0.500	0.550	0.500	0.581	0.622
THD5	**1.000**	0.752	0.950	0.750	**1.000**	0.504	0.774	**1.000**	0.550	0.500	0.800
THD6	**1.000**	0.950	**1.000**	0.955	**1.000**	0.500	0.633	0.900	0.556	0.500	0.933
THD7	**0.980**	0.939	0.944	0.906	0.971	0.644	0.882	0.927	0.917	0.930	0.946
THD8	**1.000**	0.850	0.850	0.850	**1.000**	0.500	0.783	**1.000**	0.550	0.500	0.939
THD9	0.970	0.967	0.971	0.500	0.993	0.958	0.509	**0.987**	0.968	0.933	0.913
THD10	0.975	0.973	0.973	0.506	0.984	0.972	0.510	0.992	0.966	**1.000**	0.914

Best results are boldfaced.

TABLE VI
RESULTS ON MULTI-CLASS DATASETS IN TERMS OF GMEAN.

	OVO						OVA					
Datasets	KPBoost-SVM	SVM	AKS-χ^{2*}	RUSBoost*	AdaBoost-MLP*	OCTK*	KPBoost-SVM	SVM	AKS-χ^{2*}	RUSBoost*	AdaBoost-MLP*	OCTK*
MMD1	**1.00**	0.977	0.567	0.474	0.977	0.000	0.853	0.773	0.567	0.474	**0.977**	0.000
MMD2	0.942	0.953	0.926	0.910	**0.992**	0.000	0.960	0.947	0.926	0.910	**0.992**	0.000
MMD3	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
MMD4	**0.939**	0.915	0.871	0.870	0.773	0.718	0.844	0.713	**0.871**	0.870	0.773	0.718
MMD5	**0.889**	0.830	0.600	0.664	0.880	0.431	0.849	0.849	0.600	0.664	**0.880**	0.431
MMD6	**0.923**	**0.923**	0.000	0.000	0.000	0.000	**0.933**	0.771	0.000	0.000	0.000	0.000
MMD7	**0.935**	**0.935**	0.931	0.883	0.865	0.000	**0.969**	0.935	0.931	0.883	0.865	0.000
MMD8	**0.829**	**0.829**	**0.829**	0.783	0.793	0.000	**0.901**	0.829	0.829	0.783	0.793	0.000
MMD9	**1.000**	0.000	0.000	0.000	0.000	0.000	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
MMD10	0.958	0.706	0.687	**0.983**	0.535	0.380	0.822	0.643	0.687	**0.983**	0.535	0.380
MMD11	0.977	0.977	0.965	**0.985**	0.946	0.606	**1.000**	0.985	0.965	0.985	0.946	0.606
MMD12	0.000	0.000	**0.387**	0.000	0.000	0.000	0.360	0.365	**0.387**	0.000	0.000	0.000
MHD1	0.929	0.920	0.912	0.877	**0.948**	0.447	0.928	0.904	0.912	0.877	**0.948**	0.447
MHD2	**0.934**	0.866	0.933	0.905	0.918	0.508	**0.946**	0.920	0.933	0.905	0.911	0.508
MHD3	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.640**	0.000	0.000	0.000	0.000	0.000
MHD4	0.000	0.000	**0.566**	0.000	0.000	0.000	**0.725**	0.000	0.566	0.000	0.000	0.000
MHD5	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.697**	0.601	0.000	0.000	0.000	0.000
MHD6	0.859	0.726	0.452	0.692	0.774	**0.887**	0.844	0.846	0.458	0.692	0.774	**0.887**
MHD7	**0.988**	0.830	0.000	0.525	0.976	0.000	**0.988**	0.973	0.000	0.525	0.976	0.000
MHD8	**0.866**	0.000	0.000	0.000	0.715	0.735	**0.922**	0.878	0.000	0.000	0.715	0.735
MHD9	0.751	0.746	0.455	0.000	**0.783**	0.746	0.756	0.689	0.455	0.000	**0.783**	0.746

* marked algorithms do not have any OVO/OVA variant.
Best results are boldfaced.

TABLE VII
RESULTS ON MULTI-CLASS DATASETS IN TERMS OF AUROC.

	OVO						OVA					
Datasets	KPBoost-SVM	SVM	AKS-χ^{2*}	RUSBoost*	AdaBoost-MLP*	OCTK*	KPBoost-SVM	SVM	AKS-χ^{2*}	RUSBoost*	AdaBoost-MLP*	OCTK*
MMD1	**1.000**	0.983	0.703	0.630	0.983	0.576	0.893	0.864	0.703	0.630	**0.983**	0.576
MMD2	0.967	0.973	0.960	0.950	**0.995**	0.619	0.977	0.970	0.960	0.950	**0.995**	0.619
MMD3	0.824	0.824	**0.891**	0.559	0.804	0.740	**0.895**	0.813	0.891	0.559	0.804	0.740
MMD4	**0.966**	0.950	0.925	0.936	0.884	0.834	0.920	0.869	0.925	**0.936**	0.884	0.834
MMD5	**0.922**	0.878	0.701	0.767	0.913	0.700	0.894	0.894	0.701	0.767	**0.913**	0.700
MMD6	**0.952**	**0.952**	0.792	0.641	0.825	0.710	**0.958**	0.866	0.792	0.641	0.825	0.710
MMD7	**0.954**	**0.954**	0.949	0.910	0.904	0.659	**0.977**	0.954	0.949	0.910	0.904	0.659
MMD8	**0.910**	**0.910**	**0.910**	0.885	0.887	0.625	**0.941**	0.910	0.910	0.885	0.887	0.625
MMD9	**1.000**	0.875	0.872	0.571	0.748	0.503	0.810	0.810	**0.872**	0.571	0.748	0.503
MMD10	0.970	0.810	0.785	**0.987**	0.771	0.621	0.868	0.806	0.785	**0.987**	0.771	0.621
MMD11	0.983	0.983	0.972	**0.990**	0.955	0.750	**1.000**	0.990	0.972	0.990	0.955	0.750
MMD12	0.786	0.786	0.725	0.686	**0.816**	0.548	0.662	0.676	0.725	0.686	**0.816**	0.548
MHD1	0.957	0.951	0.946	0.925	**0.967**	0.697	0.956	0.941	0.946	0.925	**0.967**	0.697
MHD2	**0.959**	0.919	0.959	0.942	0.946	0.712	**0.966**	0.950	0.959	0.942	0.946	0.712
MHD3	**0.804**	0.721	0.672	0.688	0.769	0.718	**0.832**	0.734	0.672	0.688	0.769	0.718
MHD4	**0.847**	0.777	0.780	0.684	0.700	0.774	**0.864**	0.760	0.780	0.684	0.700	0.7742
MHD5	**0.790**	0.741	0.750	0.644	0.694	0.758	**0.875**	0.798	0.750	0.644	0.694	0.758
MHD6	0.905	0.826	0.733	0.776	0.847	**0.918**	0.892	0.888	0.733	0.776	0.847	**0.918**
MHD7	**0.991**	0.881	0.741	0.683	0.982	0.728	**0.991**	0.973	0.741	0.683	0.982	0.728
MHD8	**0.917**	0.754	0.735	0.680	0.853	0.868	**0.951**	0.937	0.735	0.680	0.853	0.868
MHD9	0.836	0.825	0.622	0.561	**0.850**	0.825	0.833	0.800	0.622	0.561	**0.850**	0.825

* marked algorithms do not have any OVO/OVA variant.
Best results are boldfaced.

References

[1] S.-I. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.

[2] S. Wu and S.-I. Amari, "Conformal transformation of kernel functions: A data-dependent way to improve support vector machine classifiers," *Neural Processing Letters*, vol. 15, no. 1, pp. 59–67, 2002.

[3] G. Wu and E. Y. Chang, "Adaptive feature-space conformal transformation for imbalanced-data learning," in *ICML*, 2003, pp. 816–823.

[4] ——, "Aligning boundary in kernel space for learning imbalanced dataset," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE, 2004, pp. 265–272.

[5] ——, "Kba: Kernel boundary alignment considering imbalanced data distribution," *IEEE Transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 786–795, 2005.

[6] P. Williams, S. Li, J. Feng, and S. Wu, "Scaling the kernel function to improve performance of the support vector machine," in *International Symposium on Neural Networks*. Springer, 2005, pp. 831–836.

[7] A. Maratea and A. Petrosino, "Asymmetric kernel scaling for imbalanced data classification," in *International Workshop on Fuzzy Logic and Applications*. Springer, 2011, pp. 196–203.

[8] A. Maratea, A. Petrosino, and M. Manzo, "Adjusted f-measure and kernel scaling for imbalanced data learning," *Information Sciences*, vol. 257, pp. 331–341, 2014.

[9] Y. Zhang, P. Fu, W. Liu, and G. Chen, "Imbalanced data classification based on scaling kernel-based support vector machine," *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 927–935, 2014.

[10] H. Xiong, W. Yu, X. Yang, M. N. S. Swamy, and Q. Yu, "Learning the conformal transformation kernel for image recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 1, pp. 149–163, 2017.

[11] I. Triguero, S. González, J. M. Moyano, S. García *et al.*, "Keel 3.0: An open source software for multi-stage analysis in data mining," *International Journal of Computational Intelligence Systems*, vol. 10, pp. 1238–1249, 2017.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009, pp. 248–255.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818—-2826.

[15] L. Fabris, S. Berton, F. Citron *et al.*, "Radiotherapy-induced mir-223 prevents relapse of breast cancer by targeting the egf pathway," *Oncogene*, vol. 35, no. 37, pp. 4914–4926, 2016.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] B. X. Wang and N. Japkowicz, "Boosting support vector machines for imbalanced data sets," *Knowledge and information systems*, vol. 25, no. 1, pp. 1–20, 2010.

[18] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2012.

[19] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Rusboost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 1, pp. 185–197, 2010.

[20] J. R. Quinlan, *C4.5: programs for machine learning*. Elsevier, 2014.

[21] X. Li, L. Wang, and E. Sung, "Adaboost with SVM-based component classifiers," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 5, pp. 785–795, 2008.

[22] H. Schwenk and Y. Bengio, "Boosting neural networks," *Neural Computation*, vol. 12, no. 8, pp. 1869–1887, 2000.