Neural Network Theory and Applications

Homework Assignment 3

1 Problem one

Implement traditional one-versus-one and one-versus-rest task decomposition methods. In this task, I use the java version of *libsvm* library. My codes are under the *workspace/* directory. Typically, for a one-versus-one decomposition method who has 12 classes, there are totally 66 svms. In order to save the training time of svms, I use **multi-threaded** programming in coding. Also, to select the best parameters for each method, **cross validation (CV)** is used. What's more, the CV has been conducted in two stages – loose grid-search and fine grid-search. In the loose grid-search stage, I use a *bigger* step to find the optimal parameters. Then, in the fine grid-search stage, in order to find more **precise** optimal parameters**,** I search in the region around the optimal parameters found in the previous stage with a *smaller* step.

1.1 One-Versus-One

Use the max-win voting strategy

1.1.1 Linear Kernel

I use linear kernel $K(x_i,x_j) = x_i^T x_j$ as the kernel method. For the linear kernel does not introduce any new parameters, we only need to select the best *C* value.

i. Loose grid-search

C	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}	2^{5}	2^{7}	2^{9}	2^{11}	2^{13}	2^{15}
accuracy	0.2547	0.3066	0.4662	0.5079	0.5688	0.5871	0.5904	0.5910	0.5902	0.5904	0.5894

ii. Fine grid-search

C	$2^{7.75}$	$2^{8.00}$	$2^{8.25}$	$2^{8.50}$	$2^{8.75}$	$2^{9.00}$	$2^{9.25}$	$2^{9.50}$	$2^{9.75}$	$2^{10.00}$	$2^{10.25}$
accuracy	0.5901	0.5909	0.5902	0.5904	0.5906	0.5910	0.5924	0.5922	0.5919	0.5917	0.5901

With C = $2^{9.25}$, the accuracy for the test data is: 63.540%

1.1.2 RBF Kernel

I use RBF kernel $K(x_i,x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$ as the kernel method. For the RBF kernel introduces a new parameter γ, we need to select the best (C, γ) pairs.

i. Loose grid-search

C \ gamma	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}	2^{5}	2^{7}	2^{9}	2^{11}	2^{13}
2^{-5}	0.2547	0.2547	0.2547	0.2850	0.4737	0.4870	0.4892	0.3048	0.2547	0.2547
2^{-3}	0.2547	0.2547	0.3027	0.4669	0.5084	0.5699	0.5629	0.4125	0.2547	0.2547
2^{-1}	0.2547	0.3058	0.4664	0.5093	0.5757	0.6087	0.6334	0.5175	0.2659	0.2547
2^{1}	0.3066	0.4664	0.5086	0.5723	0.5996	0.6351	0.6713	0.6168	0.3544	0.2567
2^{3}	0.4662	0.5083	0.5694	0.5927	0.6161	0.6532	0.6786	0.6138	0.3546	0.2567
2^{5}	0.5079	0.5688	0.5892	0.6009	0.6288	0.6537	0.6560	0.6126	0.3546	0.2567
2^{7}	0.5686	0.5884	0.5953	0.6143	0.6410	0.6408	0.6428	0.6126	0.3546	0.2567
2^{9}	0.5876	0.5927	0.5981	0.6253	0.6417	0.6263	0.6405	0.6126	0.3546	0.2567
2^{11}	0.5899	0.5934	0.6105	0.6384	0.6260	0.6130	0.6405	0.6126	0.3546	0.2567
2^{13}	0.5904	0.5973	0.6239	0.6389	0.6230	0.6056	0.6405	0.6126	0.3546	0.2567
2^{15}	0.5930	0.6084	0.6382	0.6258	0.6075	0.6044	0.6405	0.6126	0.3546	0.2567

ii. Fine grid-search

C \ gamma	$2^{5.75}$	$2^{6.00}$	$2^{6.25}$	$2^{6.50}$	$2^{6.75}$	$2^{7.00}$	$2^{7.25}$	$2^{7.50}$	$2^{7.75}$	$2^{8.00}$	$2^{8.25}$
$2^{1.75}$	0.6539	0.6601	0.6631	0.6672	0.6730	0.6766	0.6783	0.6807	0.6827	0.6811	0.6694
$2^{2.00}$	0.6578	0.6616	0.6661	0.6705	0.6728	0.6778	0.6801	0.6827	0.6832	0.6793	0.6671
$2^{2.25}$	0.6590	0.6643	0.6667	0.6699	0.6766	0.6774	0.6816	0.6840	0.6824	0.6766	0.6652
$2^{2.50}$	0.6613	0.6647	0.6667	0.6700	0.6769	0.6791	0.6802	0.6827	0.6822	0.6741	0.6652
$2^{2.75}$	0.6621	0.6659	0.6664	0.6704	0.6768	0.6773	0.6817	0.6832	0.6799	0.6722	0.6643
$2^{3.00}$	0.6621	0.6649	0.6669	0.6712	0.6760	0.6786	0.6811	0.6806	0.6771	0.6697	0.6631
$2^{3.25}$	0.6624	0.6633	0.6689	0.6737	0.6753	0.6773	0.6786	0.6781	0.6750	0.6699	0.6628
$2^{3.50}$	0.6628	0.6628	0.6684	0.6728	0.6760	0.6768	0.6765	0.6730	0.6708	0.6682	0.6628
$2^{3.75}$	0.6590	0.6649	0.6685	0.6717	0.6740	0.6743	0.6728	0.6717	0.6702	0.6680	0.6636
$2^{4.00}$	0.6593	0.6669	0.6664	0.6700	0.6722	0.6713	0.6699	0.6677	0.6689	0.6682	0.6624
$2^{4.25}$	0.6626	0.6651	0.6674	0.6659	0.6669	0.6682	0.6664	0.6669	0.6684	0.6672	0.6624

With $(C,\gamma) = (2^{2.25}, 2^{7.5})$, the accuracy for the test data is: 75.495%.

1.2 One-Versus-Rest

Use the winner-takes-all strategy

1.2.1 Linear Kernel

i. Loose grid-search

C	2^{1}	2^{3}	2^{5}	2^{7}	2^{9}	2^{11}
accuracy	0.2605	0.3129	0.3314	0.3366	0.3376	0.3376

ii. Fine grid-search

C	$2^{7.75}$	$2^{8.00}$	$2^{8.25}$	$2^{8.50}$	$2^{8.75}$	$2^{9.00}$	$2^{9.25}$	$2^{9.50}$	$2^{9.75}$	$2^{10.00}$	$2^{10.25}$
accuracy	0.3375	0.3371	0.3378	0.3376	0.3378	0.3376	0.3378	0.3376	0.3376	0.3375	0.3376

With $C = 2^{8.25}$, the accuracy for the test data is: 34.610%

1.2.2 RBF Kernel

i. Loose grid-search

C \ gamma	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}	2^{5}	2^{7}	2^{9}	2^{11}
2^{-5}	0.0	0.0	0.0	0.0	0.0314	0.2097	0.1859	0.0128	0.0
2^{-3}	0.0	0.0	0.0	0.0479	0.2583	0.3129	0.3007	0.1051	0.0
2^{-1}	0.0	0.0	0.0497	0.2615	0.3162	0.3653	0.3993	0.2544	0.0016
2^{1}	0.0	0.0516	0.2615	0.3155	0.3535	0.4240	0.5073	0.4146	0.0319
2^{3}	0.0516	0.2610	0.3132	0.3365	0.3891	0.4732	0.5620	0.4313	0.0319
2^{5}	0.2605	0.3129	0.3317	0.3676	0.4285	0.5210	0.5711	0.4308	0.03198
2^{7}	0.3127	0.3312	0.3505	0.3924	0.4679	0.5507	0.5577	0.4308	0.0319
2^{9}	0.3312	0.3391	0.3718	0.4260	0.5099	0.5515	0.5546	0.4308	0.0319
2^{11}	0.3368	0.3543	0.3927	0.4639	0.5358	0.5485	0.5549	0.4308	0.0319

ii. Fine grid-search

C \ gamma	$2^{5.75}$	$2^{6.00}$	$2^{6.25}$	$2^{6.50}$	$2^{6.75}$	$2^{7.00}$	$2^{7.25}$	$2^{7.50}$	$2^{7.75}$	$2^{8.00}$	$2^{8.25}$
$2^{3.75}$	0.5302	0.5413	0.5500	0.5597	0.5686	0.5698	0.5691	0.5648	0.5531	0.5413	0.5226
$2^{4.00}$	0.5356	0.5444	0.5543	0.5629	0.5714	0.5726	0.5699	0.5637	0.5531	0.5416	0.5228
$2^{4.25}$	0.5391	0.5485	0.5591	0.5685	0.5731	0.5714	0.5716	0.5637	0.5533	0.5408	0.5218
$2^{4.50}$	0.5444	0.5523	0.5622	0.5683	0.5741	0.5723	0.5701	0.5624	0.5518	0.5426	0.5236
$2^{4.75}$	0.5470	0.5551	0.5643	0.5726	0.5713	0.5727	0.5685	0.5624	0.5505	0.5434	0.5256
$2^{5.00}$	0.5507	0.5581	0.5675	0.5739	0.5711	0.5711	0.5661	0.5610	0.5495	0.5429	0.5248
$2^{5.25}$	0.5543	0.5607	0.5683	0.5701	0.5714	0.5686	0.5661	0.5589	0.5502	0.5429	0.5239
$2^{5.50}$	0.5539	0.5648	0.5683	0.5694	0.5686	0.5681	0.5635	0.5568	0.5505	0.5421	0.5228
$2^{5.75}$	0.5572	0.5681	0.5675	0.5678	0.5668	0.5660	0.5614	0.5569	0.5521	0.5427	0.5233
$2^{6.00}$	0.5614	0.5647	0.5660	0.5653	0.5661	0.5640	0.5599	0.5574	0.5515	0.5417	0.5241
$2^{6.25}$	0.5652	0.5638	0.5652	0.5652	0.5642	0.5612	0.5600	0.5572	0.5510	0.5417	0.5241

With $(C,\gamma) = (2^{4.50}, 2^{6.75})$, the accuracy for the test data is: 63.738%

2 Problem two

Implement part-versus-part task decomposition method.

In this task, first, I've divided the multi-class classification problem into binary classification using **one-versus-one** decomposition method. Then, in each one-versus-one problem, part-versus-part method is used to deal with some of the imbalanced data set. Also, Min-Max modular is applied to compose results from each "part" of the whole data set.

2.1 Linear Kernel

i. Loose grid-search

C	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}	2^{5}	2^{7}	2^{9}	2^{11}	2^{13}	2^{15}
accuracy	0.2552	0.3065	0.4600	0.5459	0.5599	0.5706	0.5838	0.5922	0.5963	0.5927	0.5935

ii. Fine grid-search

C	$2^{9.75}$	$2^{10.00}$	$2^{10.25}$	$2^{10.50}$	$2^{10.75}$	$2^{11.00}$	$2^{11.25}$	$2^{11.50}$	$2^{11.75}$	$2^{12.00}$	$2^{12.25}$
accuracy	0.5940	0.5947	0.5940	0.5932	0.5943	0.5963	0.5953	0.5943	0.5940	0.5942	0.5934

With $C = 2^{11}$, the accuracy for the test data is: 62.615%

2.2 RBF Kernel

i. Loose grid-search

C \ gamma	2^{-5}	2^{-3}	2^{-1}	2^{1}	2^{3}	2^{5}	2^{7}	2^{9}	2^{11}	2^{13}
2^{-5}	0.2552	0.2552	0.2552	0.2847	0.4647	0.5272	0.5025	0.3055	0.2547	0.2547
2^{-3}	0.2552	0.2552	0.3027	0.4600	0.5508	0.5874	0.5782	0.4070	0.2547	0.2547
2^{-1}	0.2552	0.3058	0.4598	0.5487	0.5690	0.5927	0.6230	0.5264	0.2662	0.2547
2^{1}	0.3065	0.4596	0.5469	0.5620	0.5792	0.6181	0.6582	0.6212	0.3820	0.2573
2^{3}	0.4598	0.5455	0.5604	0.5732	0.6062	0.6403	0.6593	0.6199	0.3820	0.2573
2^{5}	0.5459	0.5594	0.5721	0.5938	0.6247	0.6504	0.6455	0.6199	0.3820	0.2573
2^{7}	0.5599	0.5719	0.5877	0.6130	0.6379	0.6390	0.6336	0.6199	0.3820	0.2573
2^{9}	0.5708	0.5869	0.6009	0.6293	0.6422	0.6212	0.6308	0.6199	0.3820	0.2573
2^{11}	0.5851	0.5920	0.6141	0.6356	0.6298	0.6082	0.6308	0.6199	0.3820	0.2573
2^{13}	0.5919	0.6057	0.6281	0.6408	0.6153	0.6023	0.6308	0.6199	0.3820	0.2573
2^{15}	0.5996	0.6140	0.6342	0.6283	0.6047	0.6008	0.6308	0.6199	0.3820	0.2573

ii. Fine grid-search

C \ gamma	$2^{5.75}$	$2^{6.00}$	$2^{6.25}$	$2^{6.50}$	$2^{6.75}$	$2^{7.00}$	$2^{7.25}$	$2^{7.50}$	$2^{7.75}$	$2^{8.00}$	$2^{8.25}$
$2^{1.75}$	0.6369	0.6460	0.6521	0.6570	0.6651	0.6679	0.6671	0.6695	0.6654	0.6601	0.6547
$2^{2.00}$	0.6410	0.6501	0.6558	0.6605	0.6687	0.6661	0.6684	0.6669	0.6611	0.6591	0.6553
$2^{2.25}$	0.6460	0.6532	0.6555	0.6652	0.6685	0.6654	0.6638	0.6636	0.6598	0.6595	0.6527
$2^{2.50}$	0.6506	0.6537	0.6585	0.6659	0.6666	0.6628	0.6618	0.6593	0.6615	0.6582	0.6527
$2^{2.75}$	0.6501	0.6557	0.6623	0.6667	0.6629	0.6600	0.6611	0.6601	0.6619	0.6563	0.6511
$2^{3.00}$	0.6517	0.6580	0.6644	0.6647	0.6603	0.6593	0.6591	0.6595	0.6578	0.6540	0.6497
$2^{3.25}$	0.6558	0.6598	0.6629	0.6619	0.6591	0.6572	0.6590	0.6580	0.6563	0.6527	0.6481
$2^{3.50}$	0.6558	0.6593	0.6600	0.6590	0.6583	0.6563	0.6582	0.6540	0.6552	0.6507	0.6473
$2^{3.75}$	0.6565	0.6583	0.6596	0.6585	0.6555	0.6572	0.6553	0.6532	0.6516	0.6502	0.6468
$2^{4.00}$	0.6568	0.6577	0.6555	0.6558	0.6544	0.6544	0.6537	0.6511	0.6512	0.6499	0.6468
$2^{4.25}$	0.6572	0.6563	0.6552	0.6516	0.6525	0.6525	0.6502	0.6504	0.6497	0.6474	0.6474

With $(C,\gamma) = (2^{1.75}, 2^{7.50})$, the accuracy for the test data is: 73.844%

3 Problem three

Use two different kernel functions, namely linear and RBF.

In the previous two tasks, I've used both linear and RBF kernels for each decomposition methods.

4 Problem four

Compare the advantages and disadvantages of these three task decomposition methods.

4.1 Comparison between different decomposition methods

decomposition method	*One-Versus-One*	*One-Versus-Rest*	*Part-Versus-Part*
number of SVMs	(classes-1)*classes/2	classes	*much more than* (classes-1)*classes/2
training time	computationally intensive	the least	same with *One-Versus-One*, even more
accuracy	the best	the worst	competitive with *One-Versus-One*

From the table above, we can see that One-Versus-Rest has the worst performance. The main reason may be the unbalanced training datasets. On the other hand, even though the One-Versus-One and Part-Versus-Part decomposition methods may be computationally intensive, since the training of each SVM can be easily paralleled, we can use parallel computing to reduce the overall training time.

4.2 Comparison between different kernels

kernels	*One-Versus-One*	*One-Versus-Rest*	*Part-Versus-Part*
Linear	63.540%	34.610%	62.615%
RBF	75.495%	63.738%	73.844%

Generally, we can draw the conclusion that the RBF kernel is superior to the linear kernel. However, RBF kernel will need more training time, for the simple reason that RBF kernel has two parameters while linear kernel has only one.