Bridging performance gap between minimal and maximal SVM models

Anonymous authors
Paper under double-blind review

Abstract

Support vector machine models are typically built using all possible pairs of SVM in one-against-one fashion. This requires too much computation for datasets with hundreds or thousands of classes, which motivates the search for multi-class models with a smaller number of edges in the underlying model graph. We conduct experiments to uncover metricàl and topological properties that impact the accuracy of a multi-class SVM model. Based on these results we propose two ways to build smaller multi-class SVM models. The key insight is that for model graphs of diameter two, we can estimate missing pairwise probabilities from known ones thus transforming the computation of posteriors to the usual complete (maximal) case. The first approach is incremental starting from a star graph. The second approach uses complete bipartite graphs. These approaches allow one to reduce computational effort by 50-90% while keeping accuracy near, or even above that of a complete model. Finally, we study how the choice of the coupling method impacts classification errors. We find that the currently used method of Wu-Lin-Weng relies mostly on the decision of the so-called critical SVM, similarly to a newly proposed stratified coupling method. This provides a partial explanation for its success in practice and suggests that it is a suitable method for element-wise ensemble improvement. All experiments are done using convolutional data sets, which have multiple advantages for benchmarking methodology to build multiclass SVM models.

1 Introduction

For almost 20 years support vector machines (SVMs) have been a popular tool for many diverse tasks in applied machine learning (Cortes & Vapnik (1995); Nayak et al. (2015); Tian et al. (2012)). The underlying geometric concept, division of space by a hyperplane into two halfspaces, is a natural fit for two-class classification problems. To apply SVM to multi-class classification problems one has to construct an ensemble of SVM for multiple two-class problems and arrive at a decision by combining the predictions of the ensemble's elements.

The optimization problem underlying SVM avoids modeling conditional probabilities $p(c|x)$, where c is a class and x is the feature vector. Rather it provides a *hard decision* - a binary classification decision is made according to which halfspace of the feature space an object belongs. This suggests that an ensemble decision should be made by counting the votes of its members (see e.g. Hsu & Lin (2002)). However, voting ensembles have drawbacks e.g. how to resolve tied votes, and have proven suboptimal in practical applications (Wu & Weng (2004)).

An improved approach to forming a multiclass model is to convert each ensemble member to a probabilistic *soft classifier* which predicts a likelihood distribution over the set of all classes. Building a multiclass model then requires three methodological choices.

(S1) deciding on a set of binary problems that will form the multiclass ensemble's elements, which we call the *model graph*,

(S2) deciding on a way to convert each SVM in the ensemble to a soft classifier, (S3) deciding on a way to combine predictions of the ensemble's members.

One of the most popular libraries for SVM is LIBSVM (Chang & Lin (2011)). The library has made the following choices. It forms a multi-class classifier by training SVM for each pair of classes, thus adopting for (S1) so-called one-vs-one classification paradigm. For (S2) it adopts modeling of conditional probabilities via logistic function drawing on the work of J. Platt (Platt et al. (1999); Lin et al. (2007)). For (S3) it has chosen a combining (also called coupling) method proposed by Wu, Lin and Weng Wu & Weng (2004).

The primary aim of this work is to study the performance implications of the choice of model's graph. One-vs-one classification adopts complete model graphs. This is acceptable only for datasets with a modest number of classes. The widely used Imagenet dataset is an example of when the complete model graph requiring $\binom{1000}{2}$ SVM models is utterly impractical. We thus study smaller model graphs and their properties that affect the resulting model's accuracy.

The secondary aim of our work is to better understand the behavior of the Wu-Lin-Weng coupling method and compare it with newer alternatives. The alternatives are outlined in Table 1.

Methodological choice	**LIBSVM choice**	**this paper**
(S1) Set of binary SVM problems	maximal	minimal, maximal, and intermediate
(S2) Conversion to soft classifier	Platt scaling	Platt scaling
(S3) Combining ensemble's models predictions	coupling method of Wu-Lin-Weng	• method of Wu-Lin-Weng • stratified coupling • normal coupling
Datasets used for benchmarking	various ML datasets (Dua & Graff (2017))	features provided by a convolutional neural network

Table 1: A comparison of models investigated in the present paper with models formed in the LIBSVM library.

Our work also differs in the choice of datasets. In this work, we concentrate on the problem of classifying images based on input features to the final softmax layer of a convolutional neural network. This class of problems has several compelling attributes, namely

- the set of classes is well defined (e.g. in comparison with much less clear phoneme classes in speech processing),
- benchmark image datasets have been thoroughly examined for the correctness of annotation; moreover, correctness of annotation can be easily assessed by a layperson,
- convolutional neural networks are trained with the softmax layer to high classification accuracy, which suggests that class separation boundaries are close to linear. This simplifies training of SVM models since linear kernel models do not require additional hyperparameters used by other kernels such as scale parameter for radial basis functions,
- digital image classification has widespread applications, and any potential improvements would be quite valuable,
- this class of problems has not been considered during methodological preparation for LIBSVM, nor in the initial evaluation of the alternative coupling methods.

2 Experimental methodology

2.1 Datasets

In sections 2-4 we have used three datasets each having ten classes: CIFAR-10 (Krizhevsky et al. (2009)), Imagenette, and Imagewoof (Howard), the latter two being subsets of the well-known Imagenet 2012 dataset (Russakovsky et al. (2015)).

0	n01558993	robin	25	n02790996	barbell
1	n01582220	magpie	26	n02807133	bathing_cap
2	n01622779	great_grey_owl	27	n03017168	chime
3	n01698640	American_alligator	28	n03089624	confectionery
4	n01980166	fiddler_crab	29	n03180011	desktop_computer
5	n02018795	bustard	30	n03196217	digital_clock
6	n02025239	ruddy_turnstone	31	n03201208	dining_table
7	n02028035	redshank	32	n03417042	garbage_truck
8	n02091134	whippet	33	n03658185	letter_opener
9	n02091244	Ibizan_hound	34	n03721384	marimba
10	n02094114	Norfolk_terrier	35	n03776460	mobile_home
11	n02096177	cairn	36	n03929855	pickelhaube
12	n02096585	Boston_bull	37	n03950228	pitcher
13	n02101388	Brittany_spaniel	38	n04026417	purse
14	n02101556	clumber	39	n04033995	quilt
15	n02114548	white_wolf	40	n04118776	rule
16	n02119789	kit_fox	41	n04254120	soap_dispenser
17	n02123045	tabby	42	n04277352	spindle
18	n02177972	weevil	43	n04525038	velvet
19	n02264363	lacewing	44	n04589890	window_screen
20	n02423022	gazelle	45	n06596364	comic_book
21	n02641379	gar	46	n06874185	traffic_light
22	n02655020	puffer	47	n07730033	cardoon
23	n02676566	acoustic_guitar	48	n09256479	coral_reef
24	n02777292	balance_beam	49	n12144580	corn

Table 2: The list of 50 classes of Imagenet-50 dataset used in Sections 4 and 5.

In sections 4 and 5 we have used a subset of Imagenet created by choosing 50 classes at random among the thousand classes in ImageNet 2012. We refer to the set as Imagenet-50 and we list the classes in Table 2.

2.2 Neural networks

For CIFAR-10 we have used a custom network designed by David C. Page, which can be quickly trained to 94 % accuracy (Page).

For Imagenette and Imagewoof datasets, we used Resnet-18 and Resnet-34 architectures (He et al. (2016)). These architectures were designed to solve the Imagenet classification problem, which has 1000 classes. We used multiple ways to adapt these architectures to solve classification problems with 10 classes:

- (fresh) to replace 1000 output neurons with 10 output neurons and train from scratch,
- (adapt) pretrain the network on the whole Imagenet 2012 problem, then replace 1000 output neurons with 10 neurons and repeat the training process with only the ten classes in the dataset,
- (whole) train a network on the whole Imagenet, and use only probabilities provided by 10 output neurons corresponding the classes in the smaller dataset.

We have trained 20 networks of each of the three kinds (whole, fresh & adapt) for both Imagenette and Imagewoof problems, as well as 20 network instances for CIFAR-10 dataset.

For Imagenet-50 we used Resnet-18 architecture and fine-tuned a network trained on all of Imagenet to this dataset.

2.3 Pairwise models for convolutional data sets

Given a convolutional neural network trained on a chosen data set, we proceed by extracting the activations of the penultimate layer. All network architectures in this paper use 512 neurons in this layer. These 512 activations constitute features of our convolutional data sets. Each training matrix had 10000 entries per class, which included all training samples and also augmented data samples.

The next step is a computation of pairwise likelihoods for each pair of classes. One could do this by simply training an SVM model and applying Platt's method to estimate likelihoods on the testing test. However, we adopt a variant of folding used inthe work Wu & Weng (2004), where we divide the training data into four folds and train four SVM models. We apply Platt's method to derive pairwise likelihoods for each of the four then models and then average resulting probabilities using geometric mean. An immediate advantage is the reduction of training time, since the training time of SVM scales superlinearly with training size (Abdiansah & Wardoyo (2015)). Moreover, by averaging, one may expect to arrive at more precise likelihoods.

2.4 Coupling algorithms

Consider a general multi-class classification problem of assigning to a given sample a probabilistic distribution $(p_1, \ldots, p_C)$ among C classes. For simplicity, we will index the C classes by integers from one to C.

Suppose pairwise SVM models are trained for some subset E of edges of the complete graph on C vertices $1, 2, \ldots, C$. We assume that for any edge (i, j) in E the corresponding model $M_{i,j}$ gives a probabilistic prediction i.e. predicts that with probability r_{ij} the sample belongs to the class i, and with the probability $1 - r_{ij}$ it belongs to the class j. We note that throughout this paper r_{ij} will always be positive which allows us to avoid degenerate cases.

If the model $M_{i,j}$ were the Bayesian classifier then the so-called Bradley-Terry equation would hold (Hastie & Tibshirani (1998)):

$$r_{ij} = \frac{p_i}{p_i + p_j}, \tag{1}$$

which we can rewrite as a linear equation in unknowns p_i and p_j

$$p_i(r_{ij} - 1) + r_{ij} p_j = 0, \tag{2}$$

or equivalently, as

$$p_j = \frac{1 - r_{ij}}{r_{ij}} p_i \tag{3}$$

The last equation shows that if any p_i is known, then we can deduce the value of any other p_j for j's that j are connected by a path in E.

If E forms a spanning tree then, then all probabilities p_i are uniquely determined by virtue of the total probability requirement

$$\sum_{i=1}^{C} p_i = 1.$$

The case when E forms a spanning tree is the *minimal* subset of SVM models required to deduce multi-class probability distribution (p_i).

It is more common to consider the *maximal* case when E corresponds to all edges of the complete graph on C vertices. Since pairwise models $M_{i,j}$ will be approximations to Bayesian classifiers, one may expect that Bradley-Terry equations will not be consistent and thus the complete set of Bradley-Terry equations will be over-determined.

In our work, we consider three different methods to solve Bradley-Terry equations in the maximal case. The first is the well-known method of Wu-Lin-Weng, which is commonly used via LIBSVM implementation. We will abbreviate it as WLW2. The other two, stratified coupling and normal coupling are newer and have not yet been thoroughly studied.

The method of Wu-Lin-Weng amounts to optimizing a quadratic functional

$$\min_{\mathbf{p}} \frac{1}{2} \sum_{i=1}^{C} \sum_{j:j\neq i} (r_{ji}p_i - r_{ij}p_j)^2$$

This method naturally leads to a system of linear equations. They note that there is a quickly converging iterative method (similar to Jacobi's) that allows one to solve the system quickly.

The second coupling method, the *stratified coupling*, aims to overcome the objection of G. Hinton, namely that the pairwise predictors in the maximal case are expected to make meaningful predictions on objects from classes they have not seen in training. Thus the method aims to restrict itself to using classifiers only for the true class, although this information is not known. This aim leads to a system of C linear equations. The matrix of the system is Markovian, which again allows for implementing a quickly converging iterative method.

The third coupling method, *normal coupling*, was proposed in 2016. It satisfies *Bayes covariance* requirement, which stipulates that coupling methods should commute with change of priors according to Bayes theorem. Computationally the method amounts to solving a set of normal equations of a special form which can be done very quickly, and thus this method is the fastest of all three.

2.5 Software used for experiments and analysis

For computations of SVM we have used R package e1071 (R Core Team (2021); Meyer et al. (2020)). For data processing we used tidyverse (Wickham et al. (2019)) and for visualizations packages ggplot2, tidyverse and ggrepel (Wickham (2016); Slowikowski (2020)).

3 Comparison of minimal and maximal SVM models

We study the performance gap between minimal and maximal SVM models, showing that there is a measurable gap. We will demonstrate various aspects of three, five and ten class classification problems derived from 10 class datasets: CIFAR-10, Imagenette and ImageWoof.

3.1 Comparison of models for three class subsets of ten class datasets

We shall consider six different probabilistic SVM models, three of them minimal and another three maximal.

A minimal probabilistic model for three classes consists of the choice of two pairwise models. We expect that their performance will vary depending on which pairwise SVM was omitted. To that end, we will order the three classes so that d' distances among them satisfies $d'_{12} < d'_{13} < d'_{23}$. We will then consider the following minimal models (see also Figure 1):

- MIN1 with edges corresponding to $(1, 2), (1, 3)$,
- MIN2 with edges corresponding to $(1, 2), (2, 3)$,
- MIN3 with edges corresponding to $(1, 3), (2, 3)$.

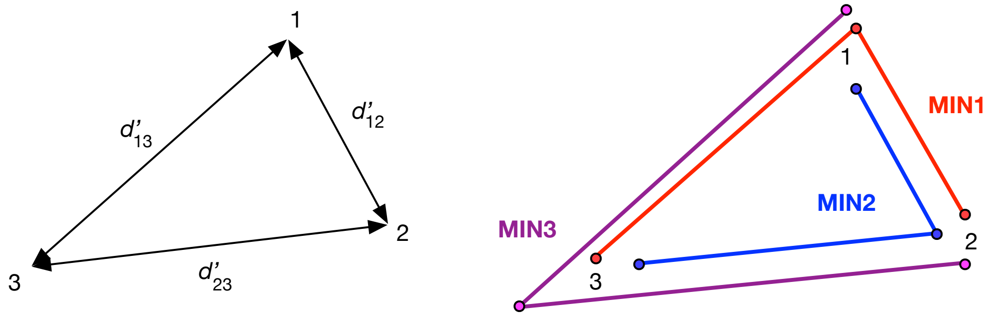


Figure 1: Left: Ordering classes so that $d'_{12} < d'_{13} < d'_{23}$. Right: Graphical indication of the three trees on three vertices, where each vertex corresponds to a classification category.

The three maximal models differ only by the applied coupling method.

We evaluate all $\binom{10}{3}$ three class problems and report resulting averaged errors in Table 3.

Model			Minimal case			Maximal case		
dataset	model	method	MIN3	MIN2	MIN1	WLW2	stratified	normal
CIFAR-10			2.3	2.1	2.0	1.9	1.9	1.9
imagenette	resnet18	whole	3.8	3.2	2.8	2.6	2.6	2.6
imagenette	resnet18	fresh	4.8	4.7	4.5	4.4	4.4	4.3
imagenette	resnet18	adapt	1.1	1.1	1.0	0.9	0.9	0.9
imagenette	resnet34	whole	3.3	2.8	2.5	2.4	2.3	2.3
imagenette	resnet34	fresh	5.5	5.4	5.2	4.9	4.9	5.0
imagenette	resnet34	adapt	1.1	1.1	0.9	0.8	0.8	0.8
imagewoof	resnet18	whole	11.5	10.7	10.2	9.8	9.8	9.7
imagewoof	resnet18	fresh	8.0	7.6	7.5	7.3	7.2	7.2
imagewoof	resnet18	adapt	3.5	3.5	3.4	3.2	3.2	3.2
imagewoof	resnet34	whole	11.0	10.5	10.0	9.7	9.7	9.5
imagewoof	resnet34	fresh	10.0	9.4	9.1	8.9	8.9	8.9
imagewoof	resnet34	adapt	3.3	3.4	3.4	3.1	3.1	3.1

Table 3: Accuracies of minimal and maximal models trained on triplets of classes.

We can see that the maximal models are consistently more accurate than the minimal models. Among minimal models, MIN1 is the most accurate and MIN3 the least accurate model. The differences among maximal models are negligible.

3.2 Comparison of models for five class subsets of ten class datasets

We shall again consider six different probabilistic SVM models, three of them minimal and another three maximal. The maximal models differ only by the choice of a coupling method.

The differences among the three minimal models are topological. There are three nonisomorphic trees on 5 vertices with diameters 2, 3 and 4. We shall name the corresponding models G2, G3, G4. These are drawn in Figure 2.

In Table 4 we report on average errors made by the various models for 500 randomly selected 5-class subsets of the ten classes.

We can again see that maximal models are consistently more accurate. Among the minimal models, the one corresponding to a star on five vertices (G2) is the most accurate. Among the maximal models, the performance is approximately equal, with only two exceptions, when the normal coupling yields noticeably more accurate models on the imagewoof dataset.

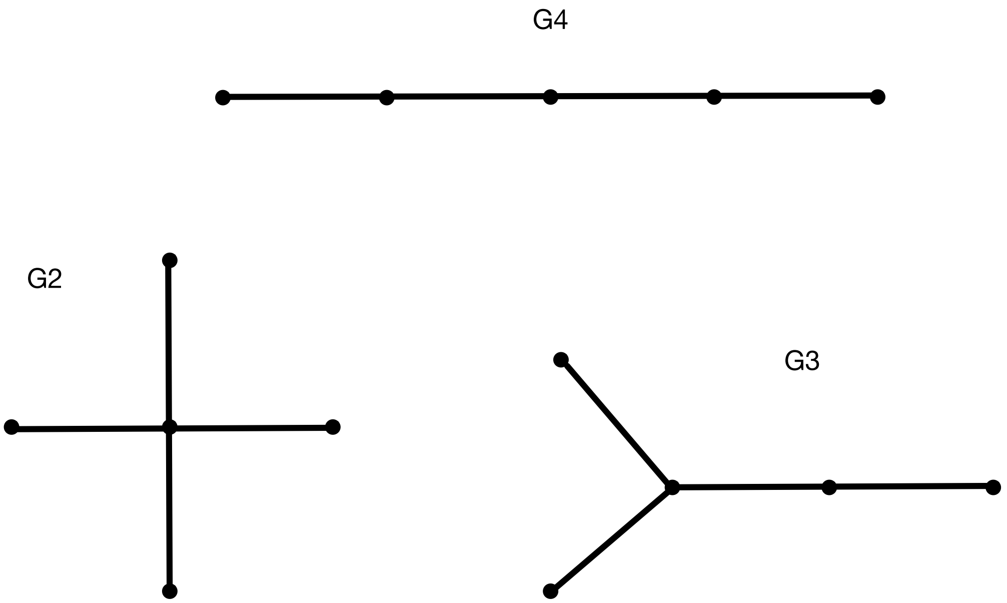


Figure 2: Trees on five vertices

Model			Minimal case			Maximal case		
dataset	architecture	method	G4	G3	G2	WLW2	stratified	normal
CIFAR-10			4.6	4.4	4.0	3.4	3.4	3.4
imagenette	resnet18	whole	7.1	6.7	6.2	4.3	4.2	4.3
imagenette	resnet18	fresh	7.8	7.6	7.2	6.4	6.4	6.3
imagenette	resnet18	adapt	2.6	2.2	1.9	1.4	1.4	1.4
imagenette	resnet34	whole	6.4	6.0	5.5	3.9	3.9	3.8
imagenette	resnet34	fresh	9.2	8.9	8.5	7.4	7.4	7.4
imagenette	resnet34	adapt	3.0	2.4	1.9	1.2	1.2	1.2
imagewoof	resnet18	whole	19.7	19.3	18.6	15.8	15.8	15.4
imagewoof	resnet18	fresh	13.8	13.5	13.1	11.9	11.9	11.8
imagewoof	resnet18	adapt	6.3	6.1	5.8	5.3	5.3	5.2
imagewoof	resnet34	whole	18.8	18.3	17.7	15.3	15.3	14.8
imagewoof	resnet34	fresh	16.8	16.4	16.0	14.1	14.1	14.2
imagewoof	resnet34	adapt	6.4	6.1	5.7	5.1	5.1	5.1

Table 4: Comparison of average error rates of minimal and maximal models on five class subsets.

3.3 Models for ten class problems

In this section we compare models for the complete ten class datasets: CIFAR-10, ImageWoof and Imagenette. We compare two minimal models, and three maximal models with the neural network models.

Based on the results of the previous two sections we compare two minimal models. The first one, named MST, is motivated by results of the section 3.1, in which spanning trees with minimal length yielded better accuracy. Thus MST model is constructed by computing the minimal spanning tree for edges weighted by d' metric. The second one, named star, corresponds to the star, a spanning tree graph of diameter which was shown in 3.2 to be the most accurate topology. The center the star is chosen so that the total sum of d' corresponding to the edges of the tree is minimal.

The averaged results for 20 different neural networks are shown in Table 5.

Model			Minimal case		Maximal case			Network
dataset	model	method	star	MST	WLW2	stratified	normal	
CIFAR-10			6.6	6.5	6.1	6.1	6.2	5.9
imagenette	resnet18	whole	9.5	9.8	7.1	7.0	7.5	9.5
imagenette	resnet18	fresh	10.7	11.1	9.4	9.3	9.2	8.6
imagenette	resnet18	adapt	2.5	4.8	2.1	2.0	2.1	1.9
imagenette	resnet34	whole	8.4	9.1	6.6	6.5	6.5	8.7
imagenette	resnet34	fresh	12.3	13.0	10.9	10.8	11.2	10.4
imagenette	resnet34	adapt	2.6	5.6	1.8	1.8	1.8	1.7
imagewoof	resnet18	whole	28.4	28.4	25.3	25.2	24.6	27.2
imagewoof	resnet18	fresh	20.9	21.6	19.8	19.8	19.8	19.4
imagewoof	resnet18	adapt	9.8	12.9	9.2	9.2	9.0	9.3
imagewoof	resnet34	whole	26.9	27.0	23.8	23.8	22.9	26.7
imagewoof	resnet34	fresh	25.6	25.4	22.8	22.8	23.2	22.6
imagewoof	resnet34	adapt	10.3	15.4	8.9	8.9	8.8	8.9

Table 5: Comparison of average error rates of minimal and maximal models on the ten-class dataset.

We can see that maximal models are consistently more accurate than either of the two minimal models. In most cases

- the neural network is more accurate than the maximal models,
- all coupling methods yield similar results.

The two exceptions are imagewoof models trained using the 'whole' methodology, where maximal models are more accurate and the normal coupling yields noticeably better accuracy compared to other coupling methods.

The performance of the minimal models is similar for CIFAR-10, but the star model yields better results overall.

4 Intermediate SVM models

Training costs of maximal SVM models becomes prohibitive for datasets with thousands of classes. The reason is that the number of pairwise SVM problems grows quadratically with the number of classes. In the previous sections, we have seen that there is a measurable performance gap between the accuracy achieved by minimal models compared to that of maximal models. It is therefore natural to ask whether an intermediate model could be found, when a model requiring significantly less training time would provide most of the accuracy benefits of maximal models.

The key problem is that the proposed coupling methods all require results for all pairwise odds. However, one can notice that a missing odds ratio p_i/p_j can be estimated provided estimates o_{ik}, o_{kj} of odd ratios p_i/p_k and p_k/p_j are known for some k due to the following identity

$$\frac{p_i}{p_j} = \frac{p_i/p_k}{p_k/p_j} \approx \frac{o_{ik}}{o_{kj}}. \tag{4}$$

In practice there may be multiple such k yielding differing estimates for p_i/p_j. Therefore we will use the estimate

$$\frac{p_i}{p_j} \approx \left(\prod_{k \in K_{ij}} \frac{o_{ik}}{o_{kj}} \right)^{1/|O_{ij}|}, \tag{5}$$

where O_{ij} is the set of classes k for which estimates of both p_i/p_k and p_k/p_j are known from a pairwise model.

4.1 The choice of the initial graph

Results of sections 3.1 and 3.3 indicate that to choose a minimal star graph, one should aim to minimize an aggregate of distances of the graph edges. To verify this hypothesis we evaluated all 50 possible graphs. To visualize them, we projected the 50 classes to two dimensions using multidimensional scaling (Gower (2015)) with distance being d' distance computed on pairwise logits. The results are shown in Figure 3. The visualization shows that it is partly true that choosing an off-center class (e.g. redshank, cardoon, or desktop computer) as the center of the star graph may yield lower initial accuracy). However, there are many off-center classes providing high accuracy (e.g. cairn or marimba). Another example is the 'letter opener' class which lies very close to the best class (corn), yet yields relatively poor accuracy. We conclude that the benefit of choosing the minimal aggregate distance for the initial graph outweighs the costs of finding one, and one may choose it simply at random.

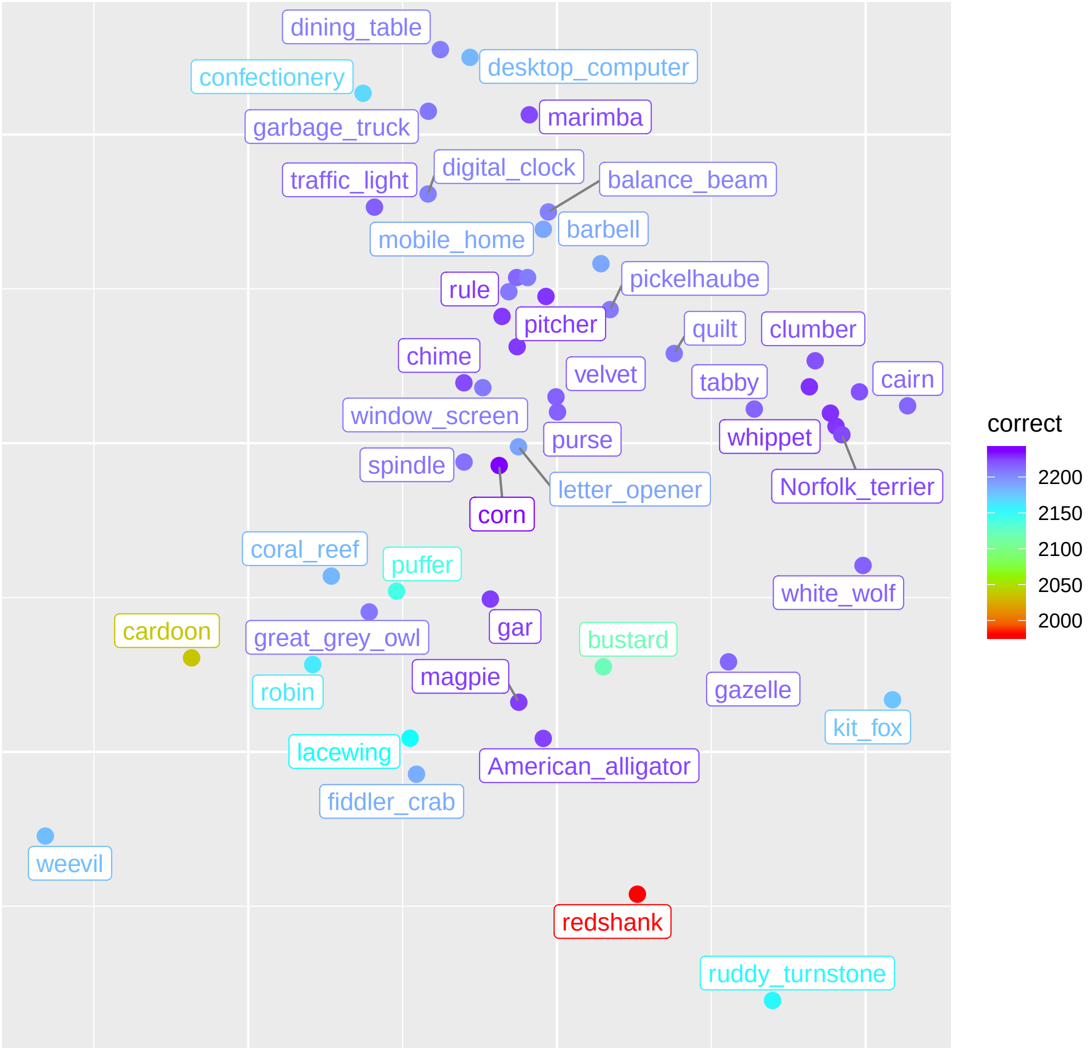


Figure 3: Projection of classes based on d' distance. The colors indicate the number of correctly classified instances in the test set (out of 2500).

4.2 Algorithm for incremental SVM models

This key insight together with the results of previous sections allows us to formulate an incremental multiclass SVM algorithm. It starts from a minimal model and at each step trains one more pairwise SVM which is used to update pairwise probabilities, which are then converted by a coupling method to the final probabilistic classification.

The key variable is F, the set of edges representing potential pairwise SVMs that could be added to an intermediate model. It is initialized on line 17 to the complement of the initial star graph and updated whenever a new pairwise SVM is added to the model.

The cost function is included to accommodate the findings of Section 3.1, which showed that it is preferable to include edges with smaller d' distances. This information may not be available, in which case one can simply use constant function d, which amounts to choosing the edge on line 22 at random. However, one may have an approximation available, for instance from a confusion matrix of another model. When that is the case, it may prove advantageous to add lower-cost edges first.

The algorithm uses logit representation of pairwise probabilities stored in a tensor of shape $N \times N \times R$, where N is the number of classes and R is the number of samples in the test dataset.

Tensor L is filled in lines 9–15 for the initial minimal model. Whenever a new SVM is added to the model, it is updated on lines 28–29 and 31–32 according to equation (5).

4.3 Experimental results

We have evaluated the incremental SVM algorithm with two different cost functions on pairs of classes. First was the uniform (non-informative) cost function which is just a constant and causes one to add a random edge to the model graph at each step. The second was confusion which for classes i and j is defined as the average of the corresponding two entries in confusion matrix as computed using the softmax layer of a neural network. Graphs below summarize mean results after 200 repetitions.

For the first experiment, we used the 10 class datasets used in the previous section. The results are shown in the first three rows in figure 4. We can see that the expected accuracy increases rapidly with the increasing number of added SVM models. Except for CIFAR-10, the normal coupling yields noticeably better results, but even for CIFAR-10, in the case of uniform cost function the normal coupling provides better accuracy until about half of extra SVMs are added to the initial star graph.

The second experiment used a single network trained on the dataset of 50 Imagenet classes. The results are also shown in the last row in Figure 4. We can see that the accuracy quickly increases in both settings when using non-informative cost as well as the confusion cost. An intriguing behavior is decreasing expected accuracy for WLW2 (and also stratified) coupling method with increasing size of the models in the case of the uniform cost. To be sure, the decrease is not large, nevertheless, it shows that intermediate size models may have better accuracy than the maximal models.

Using the confusion cost function proved beneficial for CIFAR-10 dataset, since accuracy increased faster compared to the uniform cost function.

In all cases we notice that adding about half of the remaining SVM models yields almost all possible accuracy gains, bridging the performance gap between the minimal and maximal models. The relative performance of stratified and WLW2 coupling is very similar. Very often, normal coupling provides an extra performance boost, except for CIFAR-10 experiment.

4.4 Bipartite model graphs

The choice of the center of the star graph can impact the accuracy of intermediate models for a long initial period. This is illustrated in Figure 5, comparing performance when choosing the worst (redshank) versus the best (corn) class as the center of the initial graph.

Algorithm 1 Incremental multiclass SVM model creation

```
procedure GROW SVM($S$, $T$, $d$, $\Xi$)
    # Argument $S$ is a training dataset with $N$ classes
    # Argument $T$ is a test dataset with $R$ samples
    # Argument $d$ is a cost function on pairs of classes of $S$
    # Argument $\Xi$ is a (vectorized) coupling function

    Allocate $L$ to be a tensor of shape $N \times N \times R$
    Choose a spanning tree with star topology. Denote by $t$ its center, and by $E$ the set of its edges.
    for $e := (i, j) \in E$ with $i < j$ do
        Train the pairwise SVM model on $S$ distinguishing classes $i$ and $j$
        Set $L[i, j, :]$, $L[j, i, :]$ according to logits of posterior probabilities modelled using Platt's method
    for $1 \leq i \leq N$; $i \neq t$ do
        for $1 \leq j \leq N$; $j \notin \{t, i\}$ do
            $L[i, j, :] \leftarrow L[i, t, :] - L[t, j, :]$
            $L[j, i, :] \leftarrow -L[i, j, :]$
    Set $step \leftarrow 1$
    Set $F$ to be the complement of $E$ in the set of edges of the complete graph on $N$ vertices.
    function $\mathrm{O}(a, b, F)$
        return $\{m$ in $\{1, \ldots, N\} - \{a, b\}$ such that $(a, m) \notin F$ and $(m, b) \notin F\}$
    while $F$ is nonempty do
        yield($list$(step $= i$, prediction $= \Xi(L)$))
        Let $f = (i, j)$ with $i < j$ be any edge in $F$ with minimal length according to the cost function $d$.
        Set $F \leftarrow F - \{f\}$
        Set $step \leftarrow step + 1$
        Train pairwise SVM model on $S$ distinguishing classes $i$ and $j$
        Set $L[i, j, :]$, $L[j, i, :]$ according to logits of posterior probabilities modelled using Platt's method
        for $k$ such that $(k, j) \in F$ do
            Set $L[k, j, :]$ to be the mean of $L[k, m, :] - L[m, j, :]$ where $m$ runs over $\mathrm{O}(k, j, F)$
            Set $L[j, k, :] = -L[k, j, :]$
        for $k$ such that $(i, k) \in F$ do
            Set $L[i, k, :]$ to be the mean of $L[i, m, :] - L[m, k, :]$ where $m$ runs over $\mathrm{O}(i, k, F)$
            Set $L[k, i, :] = -L[i, k, :]$
```

Therefore we investigate an alternative approach, when the model graph is a complete bipartite graph $K_{d,C-d}$, where $1 \leq d < C$. The key property of these graphs is that their diameter is 2, which allows for estimation of the pairwise probabilities via (5).

Note that the graph $K_{1,C-1}$ i.e. the case when $d = 1$ is just the star graph considered previously. We expect better performance for larger values of d. We have conducted an experiment on Imagenet-50 dataset comparing the performance of complete bipartite graphs with incremental graphs with the same number of edges (which equals to the number of SVM that have to be trained). Figure 6 summarizes the results of the experiment. We can see in that bipartite graphs consistently outperform incremental models and satisfactory performance is achieved with d values as low as five. Furthermore, bipartite graphs outperformed the maximal model for both normal and WLW2 coupling. We attribute the excellent performance of bipartite graphs to the increased likelihood of using SVM providing more information, and decreased likelihood of starting with all non-informative SVM.

5 Analysis of differences between coupling methods

In this section, we compare differences in predictions between coupling methods in an experiment on the 50 class subset of Imagenet. We concentrate on the cases when one coupling method errs, yet other yields a correct prediction. The number of such cases is summarized in Table 6.

	WLW2	stratified	normal
WLW2	-	5	20
stratified	0	-	17
normal	18	20	-

Table 6: Number of incorrectly classified samples by the method on the left, but correctly classified by the method on top

Suppose a maximal ensemble of SVM is constructed. For an erroneous prediction by a coupling method, we call the *critical model* the SVM model comparing the class incorrectly predicted by the ensemble with the correct class.

In many cases, the critical SVM model incorrectly predicts pairwise odds > 1. But we will see that some coupling methods can correct this misprediction.

In the subsequent subsections, we analyze the differences in more detail.

5.1 Visualization methodology

For comparison of two coupling methods we look at errors made by one, which have not been made by the other. We randomly select 4 samples. For each of the samples, we visualize from top to bottom the following information:

- We draw the center crop of the corresponding Imagenet picture used to provide input for the neural network.
- We draw the pairwise likelihoods for the critical pair. Likelihoods > 1 mean incorrect decision since we compare probabilities of the incorrect class to the correct one. In all cases, the ensemble prediction, shown in red color above the axis, is > 1, since we visualize only incorrect predictions. We also present the pairwise likelihoods for the critical SVM model, shown in grey color below the axis. The axis is logarithmic.
- We visualize the probabilities for the top four predicted classes by drawing horizontal bars. If the probability is $> 5\%$ then we also attach the textual description of the corresponding class.
- The label of the cell consists of two parts. The first is the name of the Imagenet class (directory), and the other is the variable part of the file in the directory.

5.2 Errors made by the WLW2 method but corrected by the normal method

We can see that in each case shown in Figure 7 the ensemble with WLW2 coupling tried to correct the over-confidence of the critical SVM model. However, it was not enough, and errors were thus made. However, the normal method was able to arrive at the correct decision. It shows that the normal method gives more weight to other models besides the critical one.

5.3 Errors made by the normal method but correctly predicted by the WLW2 method

In Figure 8 we can see cases where the normal method erred. In all of them, the critical SVM made the correct decision, meaning the pairwise odds were < 1. However, as the result of the coupling of the maximal set of SVM models, the ensemble output a decision contradicting the critical SVM. In the cases presented in the Figure, the correct decision was made by the WLW2 model, which indicates it more closely follows the critical SVM model.

5.4 Errors made by the WLW2 method but correctly predicted by the stratified method

The number of differences between WLW2 method and the stratified coupling is considerably less, as seen in Table 6. In fact, there are no errors that the stratified coupling method made, that were avoided by the WLW2 method. The other way around, there were only 5 instances where WLW2 method erred, and the stratified didn't (compared to 20 where WLW2 erred, and the normal method didn't). Four of them are shown in Figure 9. We can see that in all cases, both the critical SVM and the ensemble predictions were near the inter-class boundary where the odds are equal to 1.

5.5 Discussion

The first key finding of this section is that WLW2 method, similarly to the stratified method strongly relies on the prediction of the critical SVM. A valuable practical implication is that WLW2 and stratified methods are suitable for element-wise ensemble improvement. This is a procedure where one replaces pairwise estimates with a better model, perhaps trained using a different cost or other hyperparameters, while keeping the rest of SVMs in the ensemble unchanged. A method that mostly relies on the critical SVM can convert improvements in the ensemble element to better performance of the whole ensemble.

Results of this section confirmed previous works showing that stratified and WLW2 are quite similar (Šuch et al. (2015)), whereas the classification predictions of the normal method differ (Šuch & Barreda (2016)). We remark that the stratified method was designed deductively. Very similar performance of WLW2 method thus provides a theoretical basis why the WLW2 coupling method has been very successful in practice.

We should note that the performance variation among the methods is not very large. In hindsight, this can be explained by considering the number of classification boundaries that are close to a given sample. If there is none, then one may expect the sample to be correctly identified by any coupling method. If the sample lies close to only one boundary, then disagreement among coupling methods may occur if the sample is very near the boundary meaning the pairwise odds are very near 1. We can see this happening for instance in Figure 7 (c), or Figure 9 (b). Most disagreements occur when a sample lies near two or more pairwise boundaries as seen in Figures 7, 8 and 9. The last case is however relatively infrequent which explains the similar performance of coupling methods. The relative frequencies of such events are illustrated in Figure 10.

6 Conclusion

It has long been known that probabilistic SVM models perform better than voting ensembles. In this work, we looked more closely at the performance of various probabilistic multiclass classification models built from SVM.

In section 3 we showed that there may be a significant performance gap between the smallest and the largest models. Two properties of the minimal models were demonstrated to influence the gap: metrical and topological. The metrical property can be described by saying that it is beneficial to include SVM for the pairs of classes that are hard to separate. The topological property can be summarized that model graphs of smaller diameter exhibit better performance.

In section 4 we proposed building intermediate-size SVM models. The key insight is equation (5), which provides a way to estimate pairwise likelihoods even when an SVM is missing in a model graph. This can be applied straightforwardly for graphs of diameter 2. We formulated an algorithm that starts with a star tree and adds models according to a predefined cost function. One example of a cost function is non-informative cost function, with all edges' costs being equal. Another cost function can be based on the confusion matrix yielded by a neural network model. In our experiments incremental models reached almost peak accuracy using just half of the available SVM models. The performance of incremental models may be affected by the choice of the initial star graph. Another approach, less prone to this problem is to use complete bipartite graphs. In our experiments, they yielded accuracy close or even above that of maximal models while using only about 20% of pairwise SVMs.

The differences among coupling methods were generally small, but they became more pronounced on the challenging imagewoof dataset. In general, the coupling methods prove most useful for samples lying near the boundary of three classes. In the most common case - when there is no doubt about classification, coupling methods agree. Most disagreements occur when a sample lies near one or two pairwise boundaries. For good classifiers or easily categorized datasets, this is an infrequent occurrence, which explains the similar performance of coupling methods.

Our evaluations provide more insight into the relative performance of the coupling methods. First, our experiments confirmed on computer vision datasets the previous findings from audio datasets that stratified coupling follows closely the established method of Wu-Lin-Weng. Stratified coupling seems to have a slight edge, which was most apparent in the challeging imagewoof dataset.

In section 5 we closely examined differing predictions on the 50 class subset of Imagenet. They showed that differences may be accounted for by the tendency of Wu-Lin-Weng method and stratified coupling to rely on the accuracy of a single SVM − the critical SVM classifier. When the critical SVM classifier errs, then the whole ensemble errs. This is not the case for normal coupling, which balances predictions of many SVM models, and is thus able to correct even an erroneous critical SVM.

Normal coupling shows different performance from both WLW2 and stratified coupling. Notably, we found it to be more accurate on smaller intermediate models. Our findings also suggest a possible explanation for the success of Wu-Lin-Weng method for SVM, since stratified coupling was devised in a deductive manner.

The results in the paper relied on Platt's method for estimating pairwise probabilities. This method may not be optimal and other density modeling methods could yield better results, which provides an avenue for future research.

References

Abdiansah Abdiansah and Retantyo Wardoyo. Time complexity analysis of support vector machines (svm) in libsvm. *International journal computer and application*, 128(3):28–34, 2015.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995. doi: 10.1007/BF00994018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

John C. Gower. *Principal Coordinates Analysis*, pp. 1–7. John Wiley & Sons, Ltd, 2015. ISBN 9781118445112. doi: https://doi.org/10.1002/9781118445112.stat05670.pub2. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat05670.pub2`.

Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, pp. 451–471, 1998.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jeremy Howard. Imagenette. URL `https://github.com/fastai/imagenette`.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C Weng. A note on platt's probabilistic outputs for support vector machines. *Machine learning*, 68(3):267–276, 2007.

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2020. URL `https://CRAN.R-project.org/package=e1071`. R package version 1.7-4.

Janmenjoy Nayak, Bighnaraj Naik, and HS Behera. A comprehensive survey on support vector machine in data mining tasks: applications & challenges. *International Journal of Database Theory and Application*, 8(1):169–186, 2015.

David Page. cifar10-fast. URL `https://github.com/davidcpage/cifar10-fast`.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2021. URL `https://www.R-project.org/`.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

Kamil Slowikowski. *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*, 2020. URL `https://CRAN.R-project.org/package=ggrepel`. R package version 0.8.2.

Ondrej Šuch, Štefan Benuš, and Andrea Tinajová. A new method to combine probability estimates from pairwise binary classifiers. *rmj*, 1:12, 2015.

Yingjie Tian, Yong Shi, and Xiaohui Liu. Recent advances on support vector machines research. *Technological and Economic Development of Economy*, 18(1):5–33, 2012. doi: 10.3846/20294913.2012.661205. URL `https://doi.org/10.3846/20294913.2012.661205`.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

Hadley Wickham, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, Alex Hayes, Lionel Henry, Jim Hester, Max Kuhn, Thomas Lin Pedersen, Evan Miller, Stephan Milton Bache, Kirill Müller, Jeroen Ooms, David Robinson, Dana Paige Seidel, Vitalie Spinu, Kohske Takahashi, Davis Vaughan, Claus Wilke, Kara Woo, and Hiroaki Yutani. Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, 2019. doi: 10.21105/joss.01686.

Ting-Fan Wu and Ruby C Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

Ondrej Šuch and Santiago Barreda. Bayes covariant multi-class classification. *Pattern Recognition Letters*, 84:99–106, 2016. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2016.08.014. URL `https://www.sciencedirect.com/science/article/pii/S0167865516302161`.

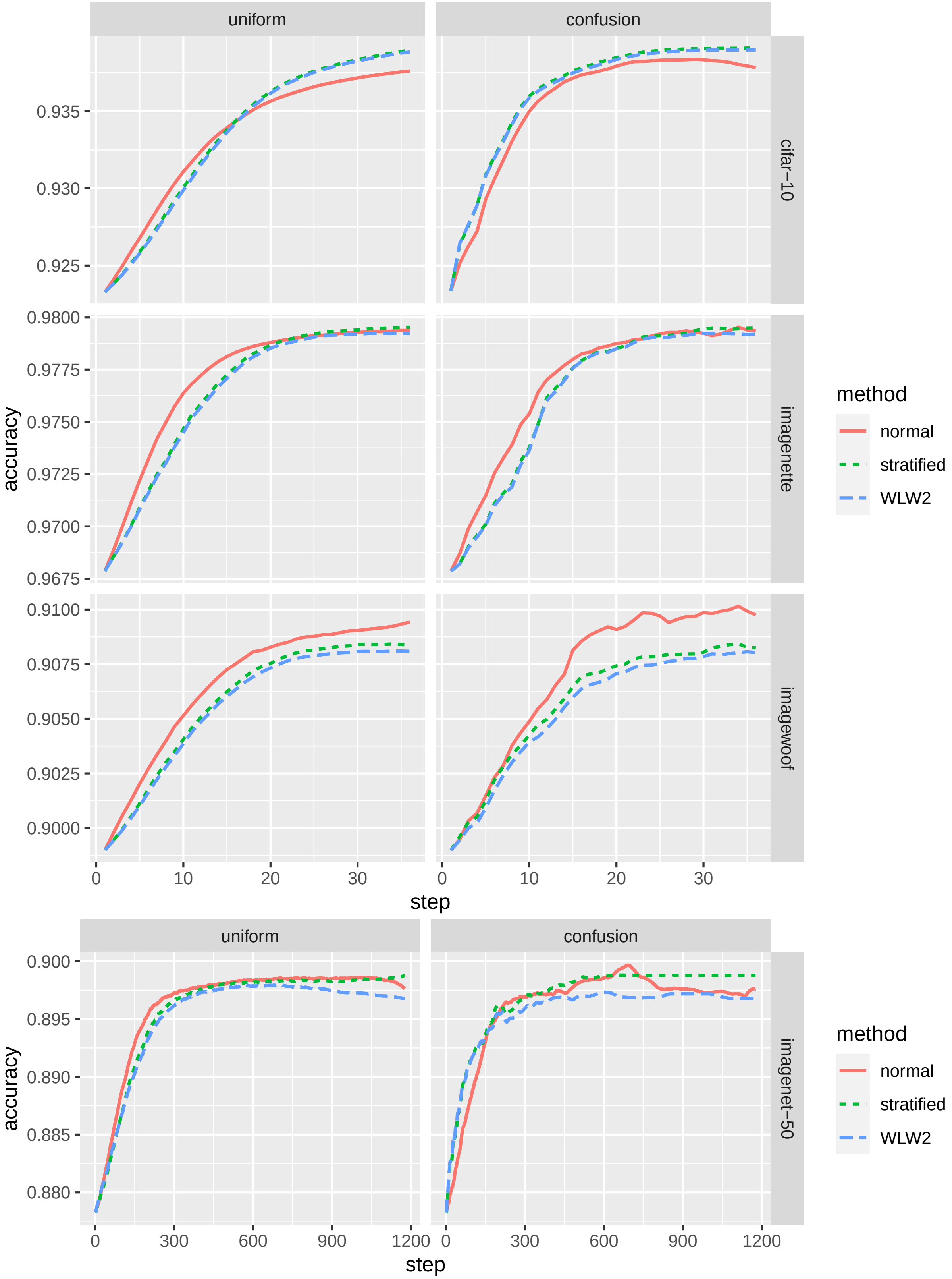


Figure 4: Expected accuracy of incremental models based on the number of additional SVM models added to an initial star model. Left, the cost matrix is uniform, right the cost matrix is defined by the confusion matrix of a neural network. Mean is taken over 200 repetitions.

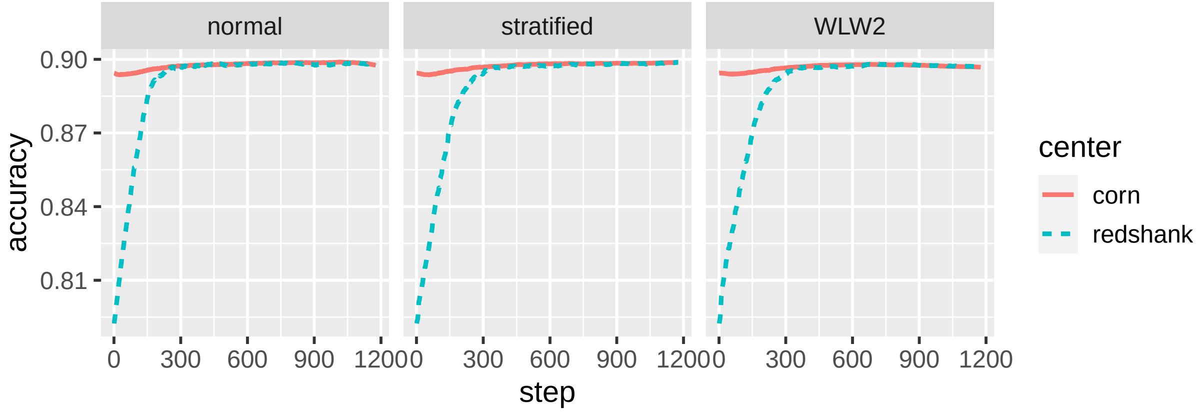


Figure 5: Comparison of accuracy of incremental models depending on the the center of the initial star graph (corn or redshank). Averaged over 100 runs with non-informative cost function.

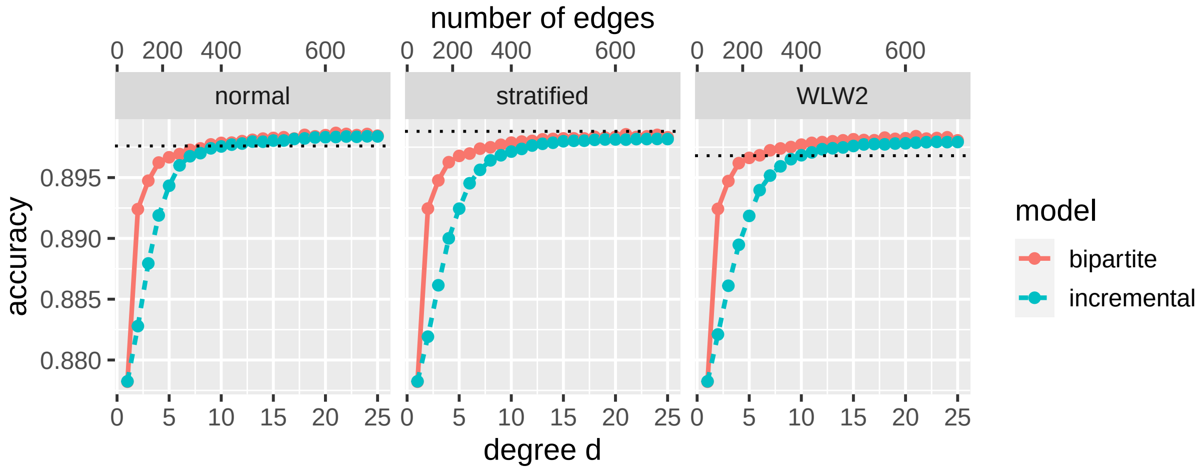
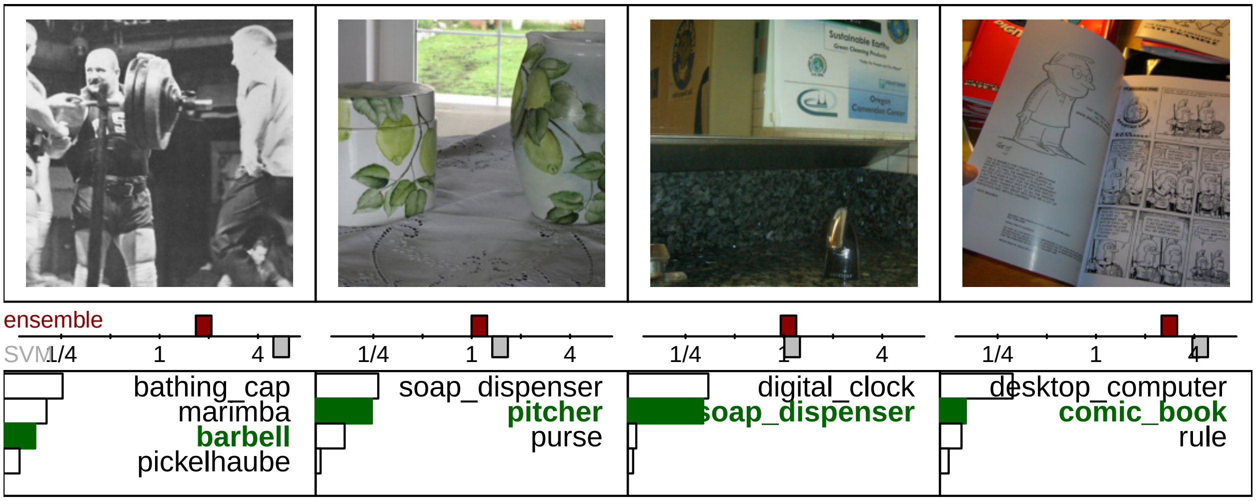
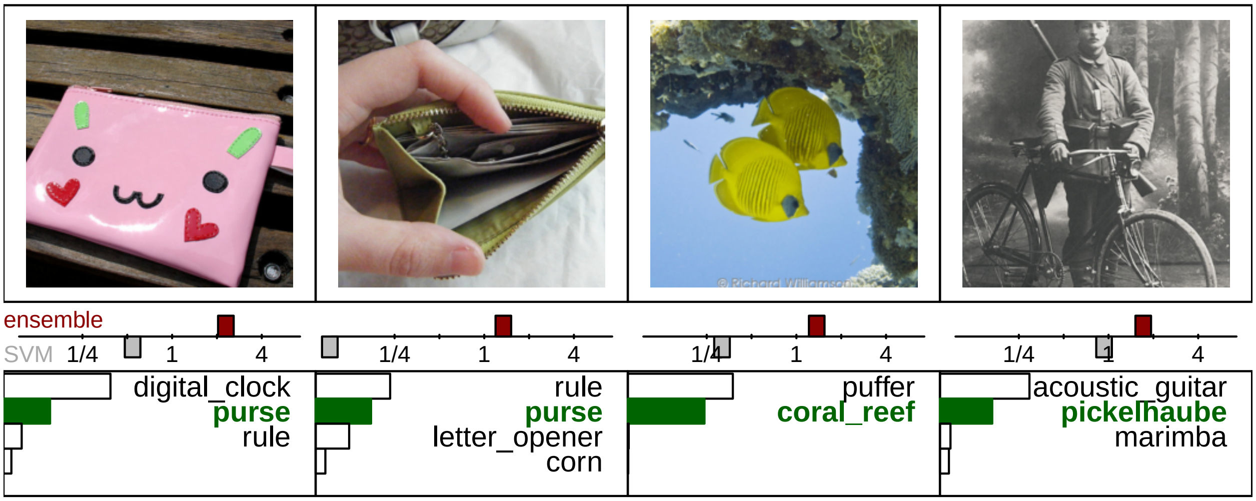


Figure 6: Comparison of accuracy of incremental models with bipartite models. Average is taken over 200 repetitions of the experiment. Incremental models used non-informative cost function and randomly chosen center of the initial star graph. Horizontal lines indicate the performance of complete maximal pairwise SVM model with a given coupling function.



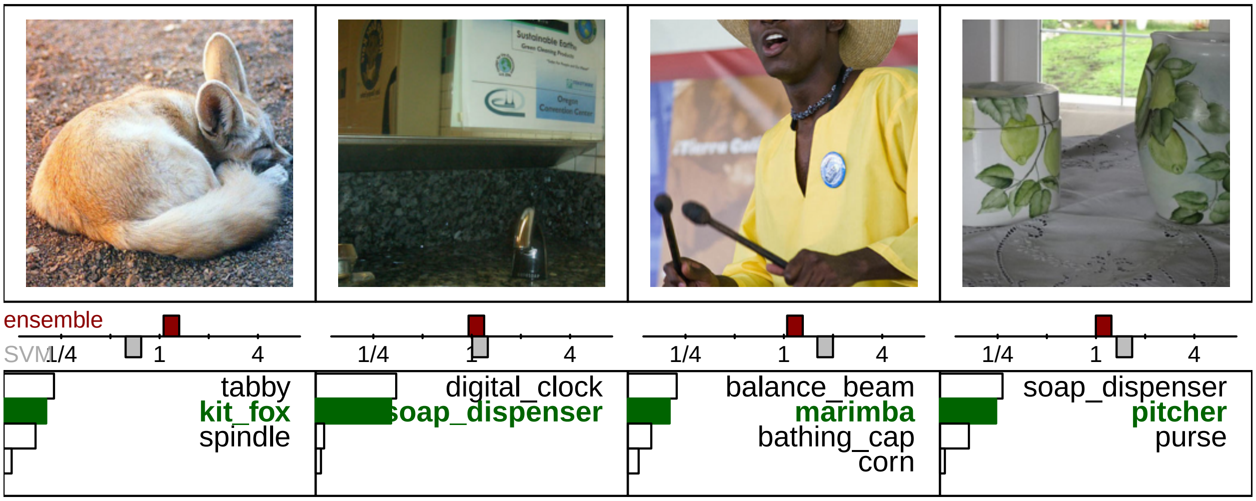
(a) n02790996:00011974 (b) n03950228:00010264 (c) n04254120:00046421 (d) n06596364:00034905

Figure 7: Sample of 4 images that are incorrectly classified by the WLW2 method, but correctly by the normal method. See section 5.1 for detailed description of the figure.



(a) n04026417:00035568 (b) n04026417:00039356 (c) n09256479:00008164 (d) n03929855:00003281

Figure 8: Sample of 4 images that are incorrectly classified by the normal method, but correctly by WLW2 method. See section 5.1 for methodology description.



(a) n02119789:00024529 (b) n04254120:00046421 (c) n03721384:00019692 (d) n03950228:00010264

Figure 9: Four of the images that are incorrectly classified by WLW2 method, but correctly by the stratified method. See section 5.1 for detailed description of the figure.

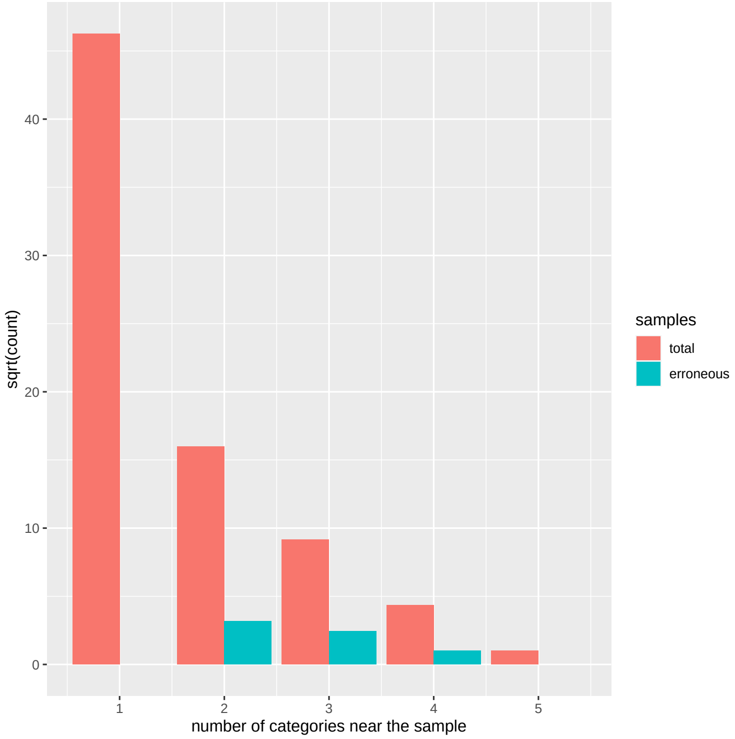


Figure 10: Distribution of samples in the test set across the number of categories a given sample lies near to. This is quantified as predicted probability of a given class is bigger than 0.1 by the stratified method. Erroneous samples are those incorrectly predicted by the stratified method, but correctly by the normal coupling method. Note that y-scale represents square roots of actual counts.