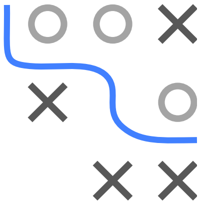
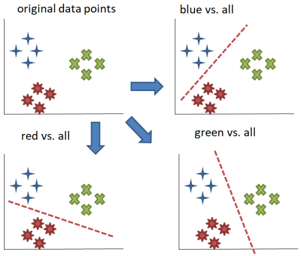
Introduction to Machine Learning



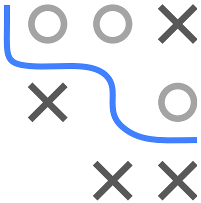
Multiclass Classification
One-vs-Rest and One-vs-One



Learning goals

- Reduce a multiclass problem to multiple binary problems in a model-agnostic way
- Know one-vs-rest reduction
- Know one-vs-one reduction

MULTICLASS TO BINARY REDUCTION



- Assume we have a way to train binary classifiers, either outputting class labels $h(\mathbf{x})$, scores $f(\mathbf{x})$ or probabilities $\pi(\mathbf{x})$.
- We are now looking for a model-agnostic reduction principle to reduce a multiclass problem to the problem of solving **multiple binary problems**.
- Two common approaches are **one-vs-rest** and **one-vs-one** reductions.

CODEBOOKS

How binary problems are generated can be defined by a codebook.

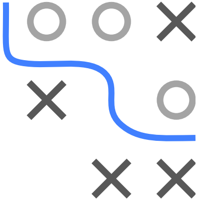
Example:

Class	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$
1	1	-1	-1
2	-1	1	1
3	0	1	-1

- The k-th column defines how classes of all observations are encoded in the binary subproblem / for binary classifier $f_k(\mathbf{x})$.
- Entry (m, i) takes values $\in \{-1, 0, +1\}$
  - if 0, observations of class $y^{(i)} = m$ are ignored.
  - if 1, observations of class $y^{(i)} = m$ are encoded as 1.
  - if -1, observations of class $y^{(i)} = m$ are encoded as -1.

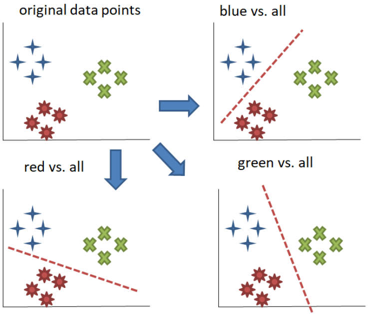
One-vs-Rest

ONE-VS-REST

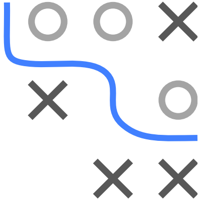


Create g binary subproblems, where in each the k-th original class is encoded as $+1$, and all other classes (the **rest**) as -1.

Class	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$
1	1	-1	-1
2	-1	1	-1
3	-1	-1	1



ONE-VS-REST / 2



- Making decisions means applying all classifiers to a sample $\mathbf{x} \in \mathcal{X}$ and predicting the label k for which the corresponding classifier reports the highest confidence:

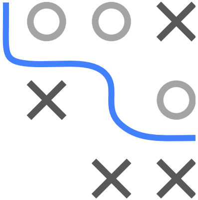
$$\hat{y} = \text{arg max}_{k \in \{1,2,\ldots,g\}} \hat{f}_k(\mathbf{x}).$$

- Obtaining calibrated posterior probabilities is not completely trivial, we could
  fit a second-stage, multinomial logistic regression model on our output scores, so with inputs $\left(\hat{f}_1(\mathbf{x}^{(i)}), \ldots, \hat{f}_g(\mathbf{x}^{(i)})\right)$ and outputs $y^{(i)}$ as training data.

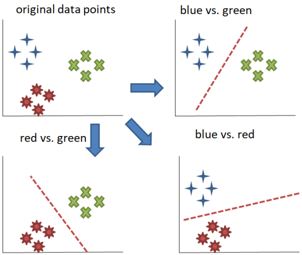
One-vs-One

ONE-VS-ONE

We create $\frac{g(g-1)}{2}$ binary sub-problems, where each $\mathcal{D}_{k,\tilde{k}} \subset \mathcal{D}$ only considers observations from a class-pair $y^{(i)} \in \{k, \tilde{k}\}$, other observations are omitted.

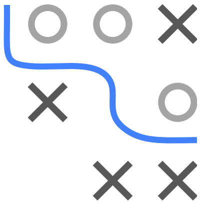


Class	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$
1	1	-1	0
2	-1	0	1
3	0	1	-1

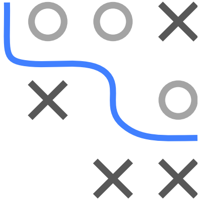


ONE-VS-ONE / 2

- Label prediction is done via **majority voting**. We predict the label of a new **x** with all classifiers and select the class that occurred most often.
- **Pairwise coupling** (see *Hastie, T. and Tibshirani, R. (1998). Classification by Pairwise Coupling*) is a heuristic to transform scores obtained by a one-vs-one reduction to probabilities.



COMPARISON ONE-VS-ONE AND ONE-VS-REST



- Note that each binary problem has now much less than n observations!
- For classifiers that scale (at least) quadratically with the number of observations, this means that one-vs-one usually does not create quadratic extra effort in g, but often only approximately linear extra effort in g.
- We experimentally investigate the train times of the one-vs-rest and one-vs-one approaches for an increasing number of classes g.
- We train a support vector machine classifier (SVMs will be covered later in the lecture) on an artificial dataset with $n = 1000$.

COMPARISON ONE-VS-ONE AND ONE-VS-REST / 2

We see that the computational effort for one-vs-one is much higher than for one-vs-rest, but it does not scale proportionally to the (quadratic) number of trained classifiers.

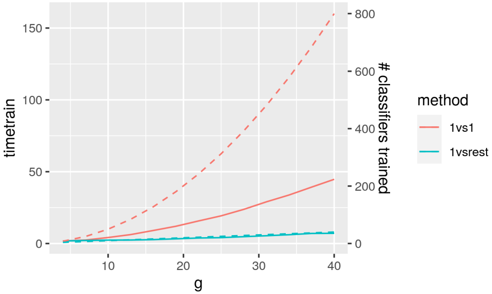


Figure: The number of classes vs. the training time (solid lines, left axis) and number of learners (dashed lines, right axis) for each of the two approaches.