Introduction to Machine Learning

Evaluation: Simple Measures for Classification

		True Class y	
		$+$	$-$
Pred.	$+$	True Positive (TP)	False Positive (FP)
$\hat{y}$	$-$	False Negative (FN)	True Negative (TN)

Learning goals

- Know the definitions of misclassification error rate (MCE) and accuracy (ACC)
- Understand the entries of a confusion matrix
- Understand the idea of costs
- Know definitions of Brier score and log loss

LABELS VS PROBABILITIES

In classification we predict:

1. Class labels $\rightarrow \hat{h}(\mathbf{x}) = \hat{y}$
2. Class probabilities $\rightarrow \hat{\pi}_k(\mathbf{x})$

$\rightarrow$ We evaluate based on those

LABELS: MCE

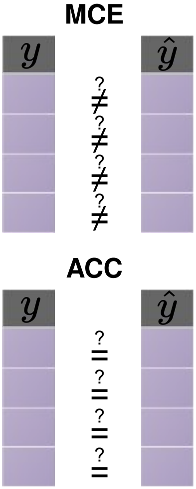
The misclassification error rate (MCE) counts the number of incorrect predictions and presents them as a rate:

$$MCE = \frac{1}{n} \sum_{i=1}^{n} [y^{(i)} \neq \hat{y}^{(i)}] \in [0; 1]$$

Accuracy is defined in a similar fashion for correct classifications:

$$ACC = \frac{1}{n} \sum_{i=1}^{n} [y^{(i)} = \hat{y}^{(i)}] \in [0; 1]$$

- If the data set is small this can be brittle
- The MCE says nothing about how good/skewed predicted probabilities are
- Errors on all classes are weighted equally (often inappropriate)



LABELS: CONFUSION MATRIX

Much better than simply reducing prediction errors to a simple number is tabulating them in a confusion matrix:

- true classes in columns
- predicted classes in rows

We can nicely see class sizes (predicted and true) and where errors occur.

		True classes				
		setosa	**versicolor**	**virginica**	**error**	*n*
Predicted classes	**setosa**	50	0	0	0	50
	versicolor	0	46	4	4	50
	virginica	0	4	46	4	50
	error	0	4	4	8	-
	n	50	50	50	-	150

LABELS: CONFUSION MATRIX

In binary classification

		True Class y	
		$+$	$-$
Pred.	$+$	True Positive (TP)	False Positive (FP)
$\hat{y}$	$-$	False Negative (FN)	True Negative (TN)

e.g.,

- **True Positive** (TP) means that an instance is classified as positive which is also positive (true prediction).
- **False Negative** (FN) means that an instance is classified as negative which is actually positive (false prediction).

LABELS: COSTS

We can also assign different costs to different errors via a cost matrix.

$$Costs = \frac{1}{n}\sum_{i=1}^{n} C[y^{(i)}, \hat{y}^{(i)}]$$

Example: **@BB Elkan Paper! Confusion matrix discussion**

Depending on certain features (age, income, profession, ...) a bank wants to decide, if it grants a 10,000 EUR loan.

Predict if a person is solvent (yes / no).
Should a bank give her/him a loan?

Examplary costs:

Loan cannot be repaid: 10,000 EUR
Interest paid for the loan: 100 EUR

		True classes	
		solvent	**not solvent**
Predicted classes	**solvent**	0	10,000
	not solvent	100	0

LABELS: COSTS

Cost matrix

		True classes	
		solvent	**not solvent**
Predicted classes	**solvent**	0	10,000
	not solvent	100	0

Confusion matrix

		True classes	
		solvent	**not solvent**
Predicted classes	**solvent**	70	3
	not solvent	7	20

- If the bank gives every person a credit, the costs are at:

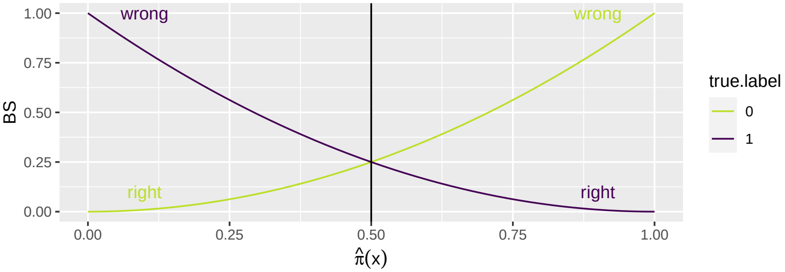
$$
\begin{aligned}
Costs &= \frac{1}{n}\sum_{i=1}^{n} C[y^{(i)}, \hat{y}^{(i)}] \\
&= \frac{1}{100}\left(-37 \cdot 7 + 0 \cdot 0 + 3 \cdot 93 + 0 \cdot 0\right) = 0.2
\end{aligned}
$$

PROBABILITIES: BRIER SCORE

Measures squared distances of probabilities from the true class labels:

$$BS1 = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{\pi}(\mathbf{x}^{(i)}) - y^{(i)} \right)^2$$

- Fancy name for MSE on probabilities
- Usual definition for binary case, $y^{(i)}$ must be coded as 0 and 1.



PROBABILITIES: BRIER SCORE

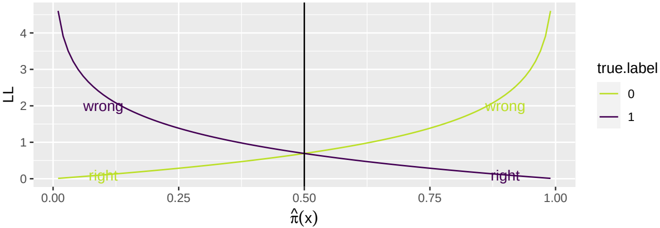
$$BS2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{g}\left(\hat{\pi}_k(\mathbf{x}^{(i)}) - o_k^{(i)}\right)^2$$

- Original by Brier, works also for multiple classes
- $o_k^{(i)} = [y^{(i)} = k]$ is a 0-1-one-hot coding for labels
- For the binary case, BS2 is twice as large as BS1, because in BS2 we sum the squared difference for each observation regarding class 0 **and** class 1, not only the true class.

PROBABILITIES: LOG-LOSS

Logistic regression loss function, a.k.a. Bernoulli or binomial loss, $y^{(i)}$ coded as 0 and 1.

$$LL = \frac{1}{n} \sum_{i=1}^{n} \left(-y^{(i)} \log(\hat{\pi}(\mathbf{x}^{(i)})) - (1 - y^{(i)}) \log(1 - \hat{\pi}(\mathbf{x}^{(i)})) \right)$$



- Optimal value is 0, "confidently wrong" is penalized heavily
- Multiclass version: $LL = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{g} o_k^{(i)} \log(\hat{\pi}_k(\mathbf{x}^{(i)}))$