Classification - Logistic regression

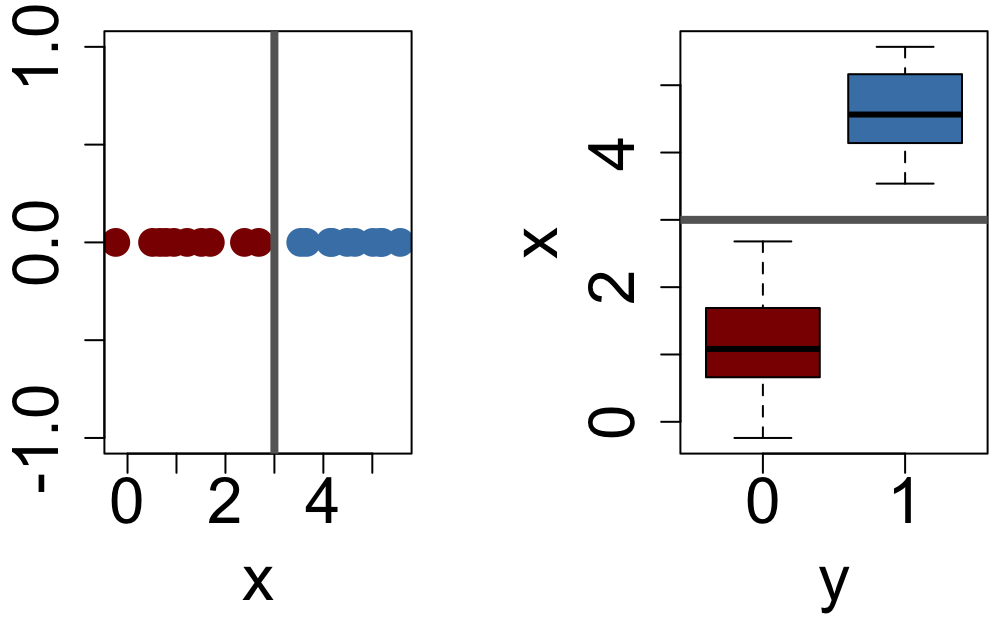
07/12/2022

Classification setup

- Assume a binary outcome Y (multiclass problems are also important and there are suitable methods for them like linear discriminant analysis and multinomial logistic regression):
  - $Y = 0$ for class 1 and $Y = 1$ for class 2
- Training data: $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)$ with $y_i = 0, 1$
- Goal is to come up with a rule based on training data to classify a new instance with feature $\mathbf{x_0}$ as $y_0 = 0$ or $y_0 = 1$.
- Classification rule is a function $\widehat{f} : \mathbb{R}^p \to \{0, 1\}$. We denote $\widehat{y}_i = \widehat{f}(\mathbf{x}_i)$.

Classification with a single feature

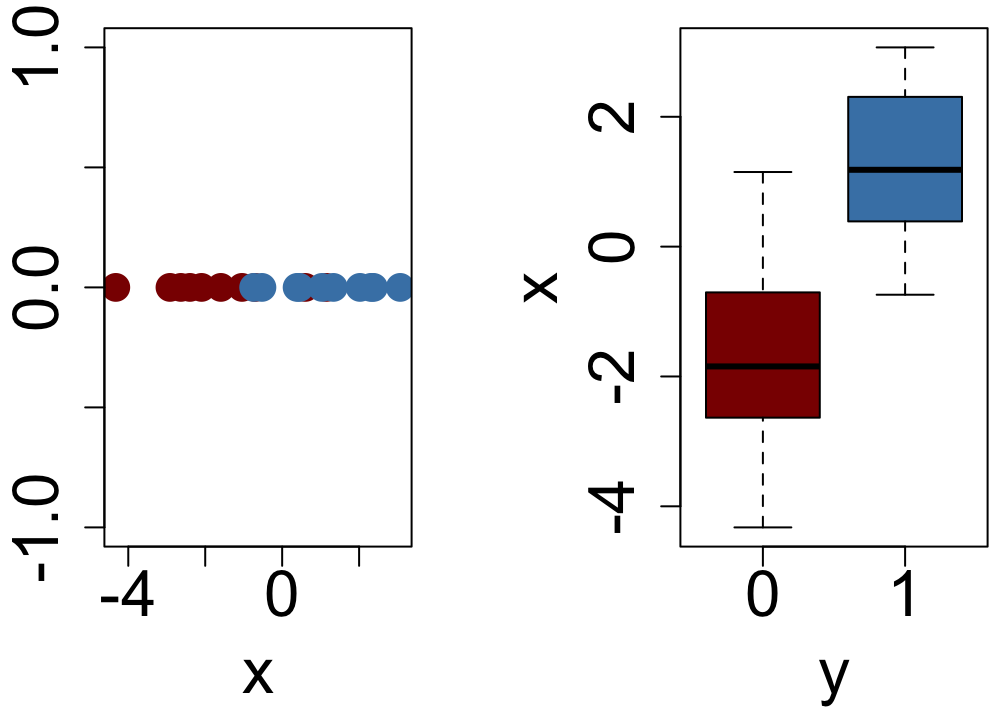
- Training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ with $y_i = 0, 1$



- Because the classes don't overlap classification is easy in this example: e.g. classify a new point as $Y = 0$ (red) if $x \leq 3$ and $Y = 1$ (blue) if $x > 3$.

Classification with a single feature

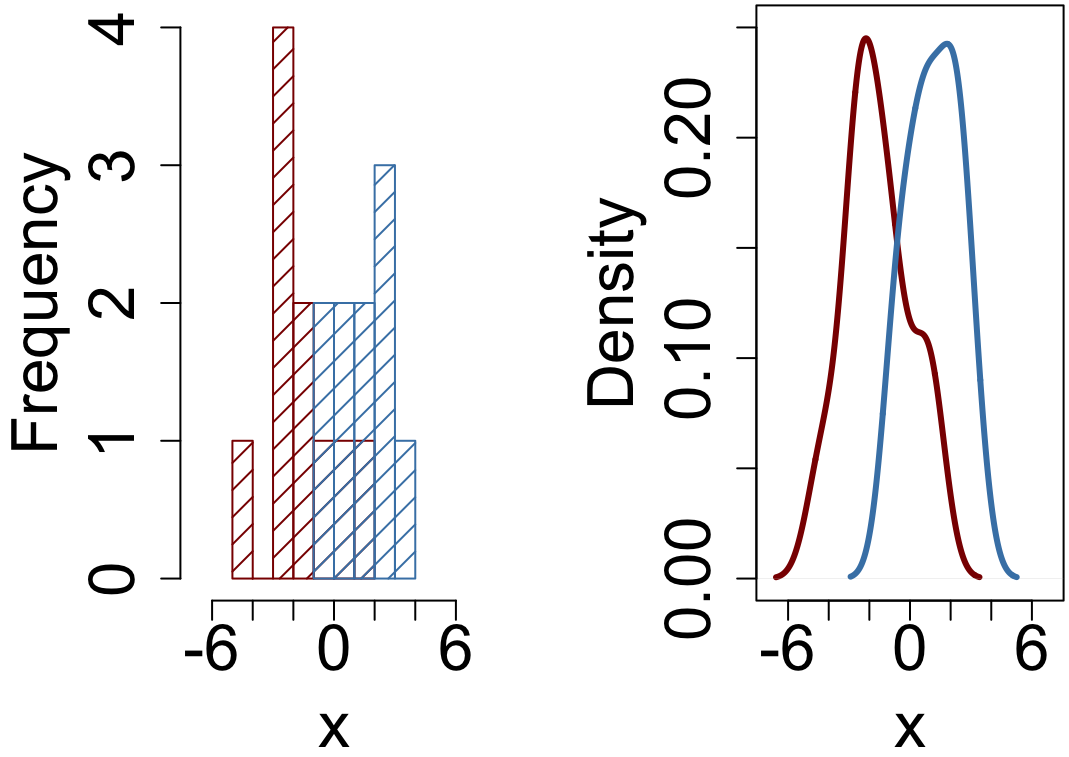
- But classes more typically overlap



- Coming up with a classification rule is less obvious

Classification with a single feature

- Same example with histograms and density (smooth histograms) estimates:



So, how do we choose a classification rule that will perform well in test samples?

Classification setup

- Training data: $(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_n}, y_n)$ with $y_i = 0, 1$
- Training error rate = proportion of missclassified instances in training set:

$$\frac{1}{n} \sum_{i=1}^{n} I\left(\widehat{f}(\mathbf{x_i}) \neq y_i\right) = \frac{1}{n} \sum_{i=1}^{n} I\left(\widehat{y}_i \neq y_i\right)$$

- Test error rate in new observation $(\mathbf{x_0}, y_0)$:

$$P(\widehat{f}(\mathbf{x_0}) \neq y_0) = Ave\left(I(\widehat{f}(\mathbf{x_0}) \neq y_0)\right)$$

- **Want a classification rule $\widehat{f}$ with low test error.**

We can use logistic Regression for classification

With multiple predictors $\mathbf{X} = \left(X_1, \ldots, X_p\right)$:

$$\log\left(\frac{p(\mathbf{X})}{1 - p(\mathbf{X})}\right) = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

or equivalently:

$$p(\mathbf{X}) = \frac{e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p}}$$

- Coefficient β_i is the log-odds ratio for $X_i = x + 1$ vs. $X_i = x$ when all other predictors are fixed

Decision rule

$$\hat{p}(\mathbf{x}) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\ldots+\hat{\beta}_p x_p}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_1+\ldots+\hat{\beta}_p x_p}}$$

- Classify to $y = 1$ if $\hat{p}(\mathbf{x}) > c$
- Classify to $y = 0$ if $\hat{p}(\mathbf{x}) \leq c$

Equivalent to:

- Classify to $y = 1$ if $\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p > t$
- Classify to $y = 0$ if $\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p \leq t$

**Decision boundary is linear

- Often $c = 0.5$ ($t = 0$) but can shift balance between sensitivity and specificity (defined later) by choosing larger or smaller value of c (or t)

Breast Cancer data

286 women diagnosed with breast cancer, underwent surgery and were followed up.

Features:

- Age: age (in years at last birthday) of the patient at the time of diagnosis
- Menopause: whether the patient is pre- or postmenopausal at time of diagnosis
- Tumor size: the greatest diameter (in mm) of the excised tumor
- Invasive nodes: the number (range 0 - 39) of lymph nodes that contain metastasis
- Node capsule: Cancer remain “contained” by the capsule?
- Degree of malignancy: the histological grade (range 1-3) of the tumor.
- Breast: cancer in left or right breast breast
- Breast quadrant: the breast is divided into four quadrants
- Irradiation: treatment with high-energy x-rays to destroy cancer cells

Outcome

- **Recurrence: cancer back within follow-up period?**

Logistic regression in R

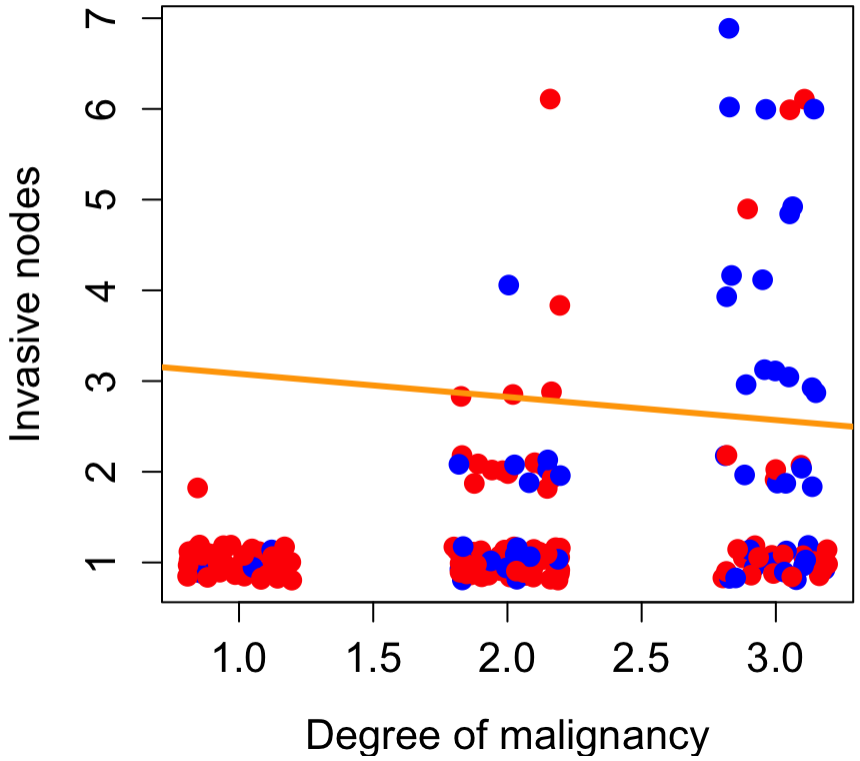
```
breast_glm = glm(recurrence ~  inv_nodes_quant + deg_malig, family='binomial', data=breast[train,])
summary(breast_glm)
```

```
##
## Call:
## glm(formula = recurrence ~ inv_nodes_quant + deg_malig, family = "binomial",
##     data = breast[train, ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7915  -0.7103  -0.4158   0.9749   2.2322
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.8558     0.6310  -6.111 9.92e-10 ***
## inv_nodes_quant   0.2946     0.1473   2.000   0.0455 *
## deg_malig         1.1564     0.2731   4.234 2.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 235.96  on 192  degrees of freedom
## Residual deviance: 198.94  on 190  degrees of freedom
## AIC: 204.94
##
## Number of Fisher Scoring iterations: 4
```

Logistic regression in R

	Estimate	Std..Error	z.value	Pr...z..	OR
(Intercept)	-3.8558	0.6310	-6.1107	0.0000	0.0212
inv_nodes_quant	0.2946	0.1473	1.9998	0.0455	1.3425
deg_malig	1.1564	0.2731	4.2337	0.0000	3.1783

Breast Cancer - classification boundary



- The decision boundary in logistic regression is linear

Breast Cancer - prediction

```
pred_glm_train = factor(predict(breast_glm, newdata=breast[train,], type='response') > 0.5)

levels(pred_glm_train) = c("no-recurrence", "recurrence")

confMatrix_train = table(true=breast[train,]$recurrence, predicted=pred_glm_train)

confMatrix_train
```

```
##                predicted
## true            no-recurrence recurrence
##   no-recurrence           127          8
##   recurrence               37         21
```

```
error_train = (confMatrix_train[1,2] + confMatrix_train[2,1])/ntrain; round(error_train, 2)
```

```
## [1] 0.23
```

Sensitivity and specificity

- In binary classification the two classes are typically not on an equal footing
- We often 'care more' about one of the classes; e.g. recurrence in the Breast cancer problem, email is spam in the spam recognition problem
- This is called the 'positive class'. The other class is called the 'negative class' (e.g. no recurrence, no stroke, no spam)
- **Sensitivity** = $P($**classify subject to the positive class | subject is in the positive class** $)$
- **Specificity** = $P($**classify to the negative class | subject is in the negative class** $)$
- In a hypothesis testing context **power = sensitivity** and **type I error = 1 - specificity**

Breast Cancer - prediction

```
pred_glm_test = factor(predict(breast_glm, newdata=breast[-train,], type='response') > 0.5)

levels(pred_glm_test) = c("no-recurrence", "recurrence")

confMatrix_test = table(true=breast[-train,]$recurrence, predicted=pred_glm_test)

confMatrix_test
```

```
##                predicted
## true            no-recurrence recurrence
##   no-recurrence            60          1
##   recurrence               17          6
```

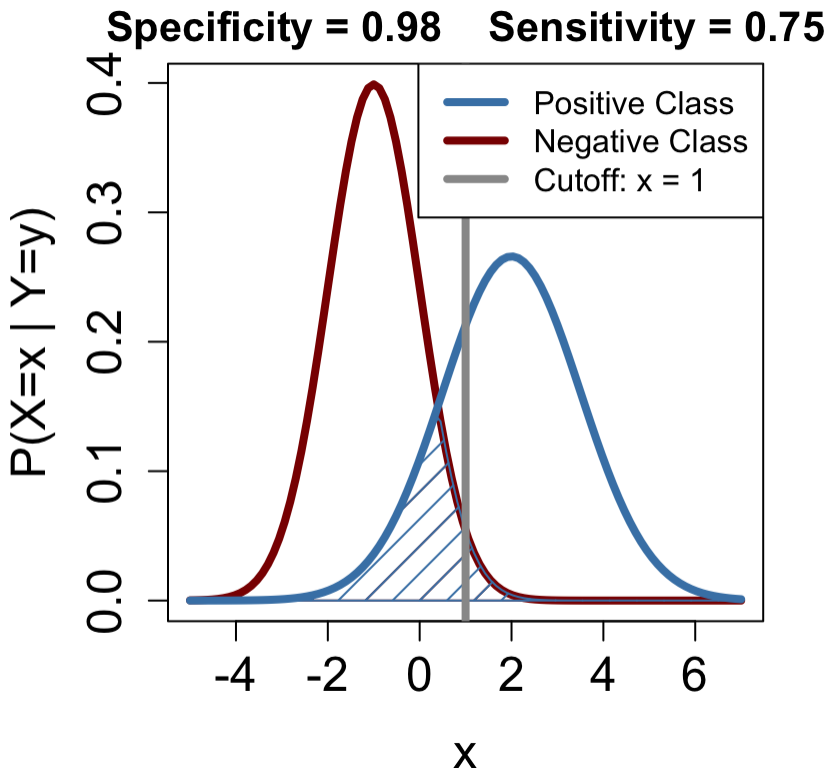
```
error_test = (confMatrix_test[1,2] + confMatrix_test[2,1])/ntest; round(error_test, 2)
```

```
## [1] 0.21
```

Estimate of sensitivity = proportion of true recurrences identified `$= 6/23 = 0.26$ **Estimate of specificity** = proportion of true non-recurrences identified $= 60/61 = 0.98$

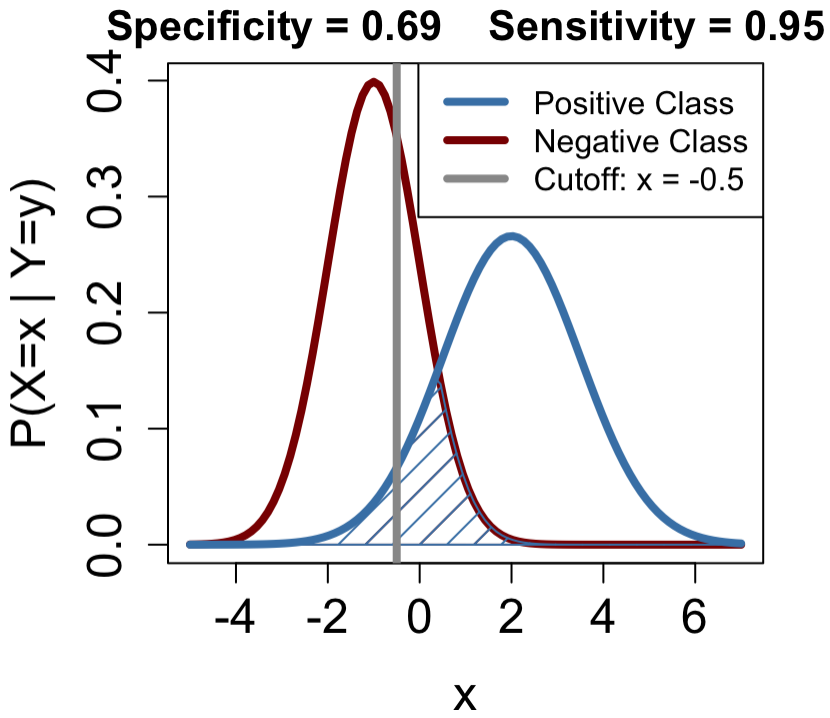
Very poor Sensitivity – not good as a screening test

Trade-off between sensitivity and specificity



- Because classes overlap, we cannot get a classifier with perfect sensitivity and specificity
- True positive rate = Sensitivity
- False positive rate = 1 - Specificity

Trade-off between sensitivity and specificity



- Changing the decision boundary changes the sensitivity and specificity:
  - if one improves the other gets worse
  - Can't make both better at the same time