Metrics in tensorflow

Neural Networks for Health Technology Applications
Spring 2020
Sakari Lukkarinen
Helsinki Metropolia University of Applied Sciences

Contents

- Review of metrics
  - Confusion matrix, sensitivity and specificity, ROC curve
- Metrics in tensorflow
  - Examples: accuracy, false negative, false positives, sensitivity at specificity
- Example – phonocardiographic screening
  - Sensitivity and specificity
  - False negatives and false positives
  - How to select the threshold
  - ROC curves

Confusion (error) matrix

		Actual class		
		Cat	Dog	Rabbit
Predicted class	Cat	5	2	0
	Dog	3	3	2
	Rabbit	0	1	11

Source: [Confusion matrix (Wikipedia)]

Accuracy

Total	True (+)	True (-)
Test (+)	TP	FP
Test (-)	FN	TN

Accuracy = (TP + TN) / Total

Source: [Accuracy in binary classification (Wikipedia)]

Sensitivity and specificity

True condition (diagnosis)

Test says: **"Disease" (+)** **"Healthy" (-)**		**Disease (+)**	**Healthy (-)**
	Total	**True (+)**	**True (-)**
	Test (+)	**TP**	FP
	Test (-)	FN	**TN**

$$\text{Sensitivity} = \frac{\textbf{TP}}{\text{True (+)}} \qquad \frac{\textbf{TN}}{\text{True (-)}} = \text{Specificity}$$

Sensitivity (also called the **true positive rate**, the **recall**, or **probability of detection**) measures *the percentage (%) of sick people who are correctly identified* by the test having the condition.

Specificity (also called the **true negative rate**) measures the *percentage (%) of healthy people who are correctly identified* by the test as not having the condition

Source: Sensitivity and specificity (Wikipedia)

Example

	True (+)	True(-)	Sum
Test (+)	50	10	60
Test (-)	10	30	40
SUM	60	40	100

What are the sensitivity, specificity and accuracy?

Example

	True (+)	True(-)	Sum
Test (+)	**50 (TP)**	10	60
Test (-)	10	**30 (TN)**	40
SUM	**60 (All disease)**	**40 (All healthy)**	**100 (All patients)**

Sensitivity = True positives / All disease = 50/60 ~ 0.83

Specificity = True negatives / All healthy = 30/40 = 0.75

Accuracy = (True positives + True negatives) / All = (50 + 30)/100 = 0.80

Confusion matrix [edit]

Let us consider a group with **P** positive instances and **N** negative instances of some condition. The four outcomes can be formulated in a 2×2 *contingency table* or *confusion matrix*, as follows:

		True condition				
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

Source: Sensitivity and specificity (Wikipedia)

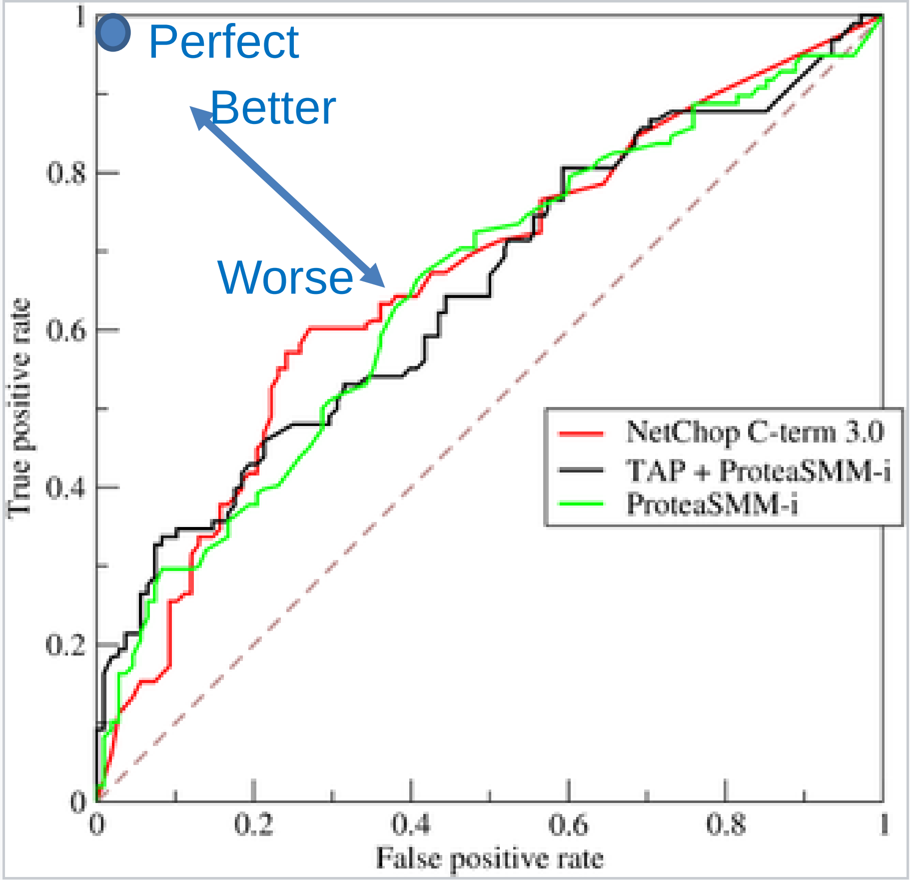
ROC curve

In statistics, a **receiver operating characteristic curve**, i.e. **ROC curve**, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

True positive rate (TPR) = sensitivity

False positive rate (FPR) = 1 - specificity

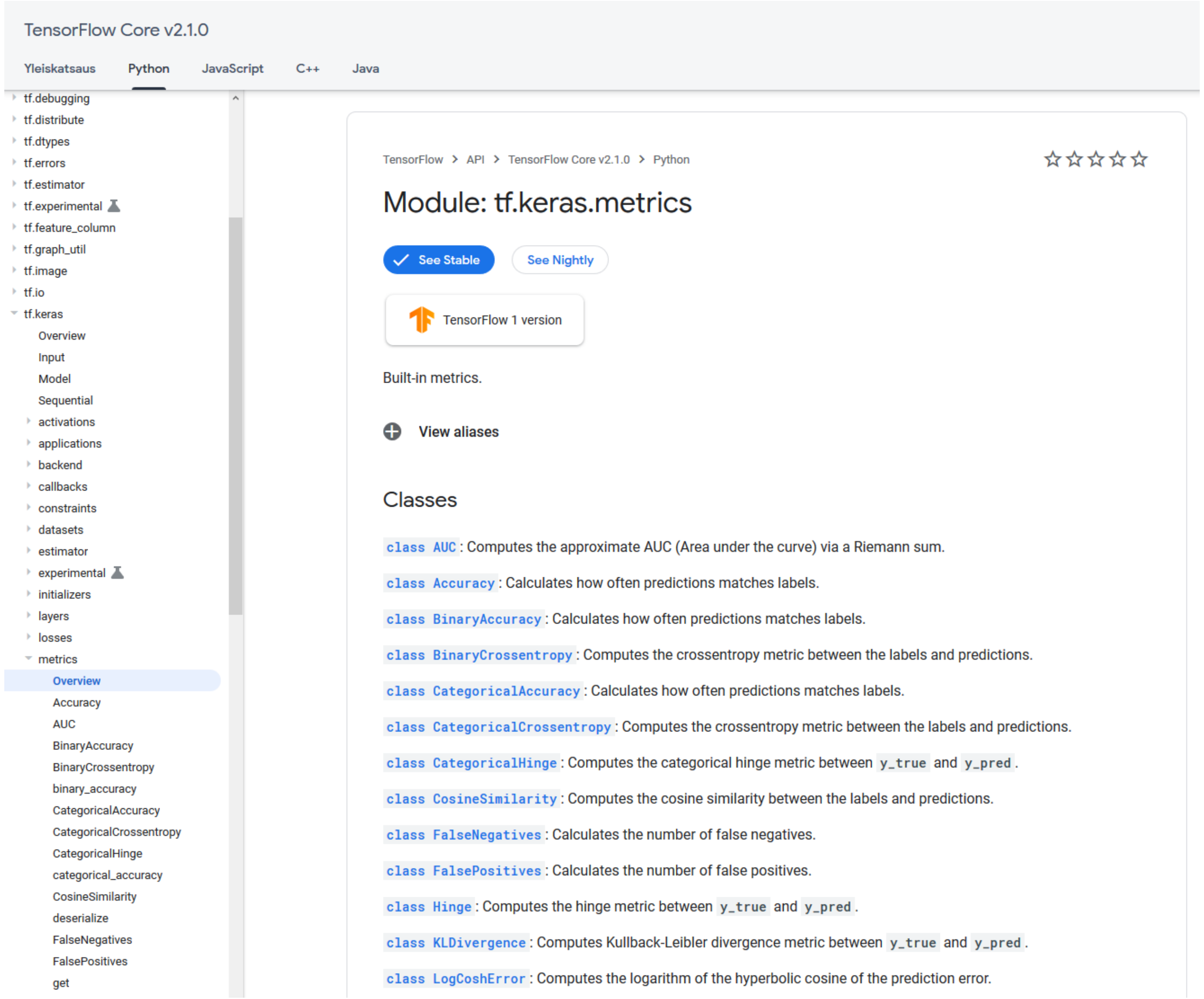
Source: Receiver operating characteristics (Wikipedia)



ROC curve of three predictors of peptide cleaving in the proteasome.

tf.keras.metrics

METRICS IN TENSORFLOW



TensorFlow Core v2.1.0

Yleiskatsaus | Python | JavaScript | C++ | Java

- tf.debugging
- tf.distribute
- tf.dtypes
- tf.errors
- tf.estimator
- tf.experimental
- tf.feature_column
- tf.graph_util
- tf.image
- tf.io
- tf.keras
  - Overview
  - Input
  - Model
  - Sequential
  - activations
  - applications
  - backend
  - callbacks
  - constraints
  - datasets
  - estimator
  - experimental
  - initializers
  - layers
  - losses
  - metrics
    - Overview
    - Accuracy
    - AUC
    - BinaryAccuracy
    - BinaryCrossentropy
    - binary_accuracy
    - CategoricalAccuracy
    - CategoricalCrossentropy
    - CategoricalHinge
    - categorical_accuracy
    - CosineSimilarity
    - deserialize
    - FalseNegatives
    - FalsePositives
    - get

TensorFlow > API > TensorFlow Core v2.1.0 > Python

Module: tf.keras.metrics

See Stable | See Nightly

TensorFlow 1 version

Built-in metrics.

View aliases

Classes

`class AUC` : Computes the approximate AUC (Area under the curve) via a Riemann sum.

`class Accuracy` : Calculates how often predictions matches labels.

`class BinaryAccuracy` : Calculates how often predictions matches labels.

`class BinaryCrossentropy` : Computes the crossentropy metric between the labels and predictions.

`class CategoricalAccuracy` : Calculates how often predictions matches labels.

`class CategoricalCrossentropy` : Computes the crossentropy metric between the labels and predictions.

`class CategoricalHinge` : Computes the categorical hinge metric between `y_true` and `y_pred` .

`class CosineSimilarity` : Computes the cosine similarity between the labels and predictions.

`class FalseNegatives` : Calculates the number of false negatives.

`class FalsePositives` : Calculates the number of false positives.

`class Hinge` : Computes the hinge metric between `y_true` and `y_pred` .

`class KLDivergence` : Computes Kullback-Leibler divergence metric between `y_true` and `y_pred` .

`class LogCoshError` : Computes the logarithm of the hyperbolic cosine of the prediction error.

https://www.tensorflow.org/api_docs/python/tf/keras/metrics

Example - Accuracy

import tensorflow as tf

Usage:

```
>>> m = tf.keras.metrics.Accuracy()
>>> _ = m.update_state([1, 2, 3, 4], [0, 2, 3, 4])
>>> m.result().numpy()
0.75
>>> m.reset_states()
>>> _ = m.update_state([1, 2, 3, 4], [0, 2, 3, 4], sample_weight=[1, 1, 0, 0])
>>> m.result().numpy()
0.5
```

Usage with tf.keras API:

```
model = tf.keras.Model(inputs, outputs)
model.compile('sgd', loss='mse', metrics=[tf.keras.metrics.Accuracy()])
```

https://www.tensorflow.org/api_docs/python/tf/keras/metrics/Accuracy

Example - False negatives

import tensorflow as tf

Usage:

```
m = tf.keras.metrics.FalseNegatives()
m.update_state([0, 1, 1, 1], [0, 1, 0, 0])
print('Final result: ', m.result().numpy())  # Final result: 2
```

Usage with tf.keras API:

```
model = tf.keras.Model(inputs, outputs)
model.compile('sgd', loss='mse', metrics=[tf.keras.metrics.FalseNegatives()])
```

https://www.tensorflow.org/api_docs/python/tf/keras/metrics/FalseNegatives

Example - SensitivityAtSpecificity

import tensorflow as tf

Usage:

```
m = tf.keras.metrics.SensitivityAtSpecificity(0.4, num_thresholds=1)
m.update_state([0, 0, 1, 1], [0, 0.5, 0.3, 0.9])
print('Final result: ', m.result().numpy())  # Final result: 0.5
```

Usage with tf.keras API:

```
model = tf.keras.Model(inputs, outputs)
model.compile(
    'sgd',
    loss='mse',
    metrics=[tf.keras.metrics.SensitivityAtSpecificity()])
```

https://www.tensorflow.org/api_docs/python/tf/keras/metrics/SensitivityAtSpecificity

Remember!

TPR + FNR = 1

Sensitivity = 1- FNR

* TPR = True positive rate = Sensitivity

* FNR = False negative rate

FPR + TNR = 1

Specificity = 1 – FPR

* TNR = True negative rate = Specificity

* FPR = False positive rate

	True condition	
Total population	Condition positive	Condition negative
Predicted condition positive	**True positive**, Power	**False positive**, Type I error
Predicted condition negative	**False negative**, Type II error	**True negative**
	True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$
	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$

Example – Phonocardiographic screening

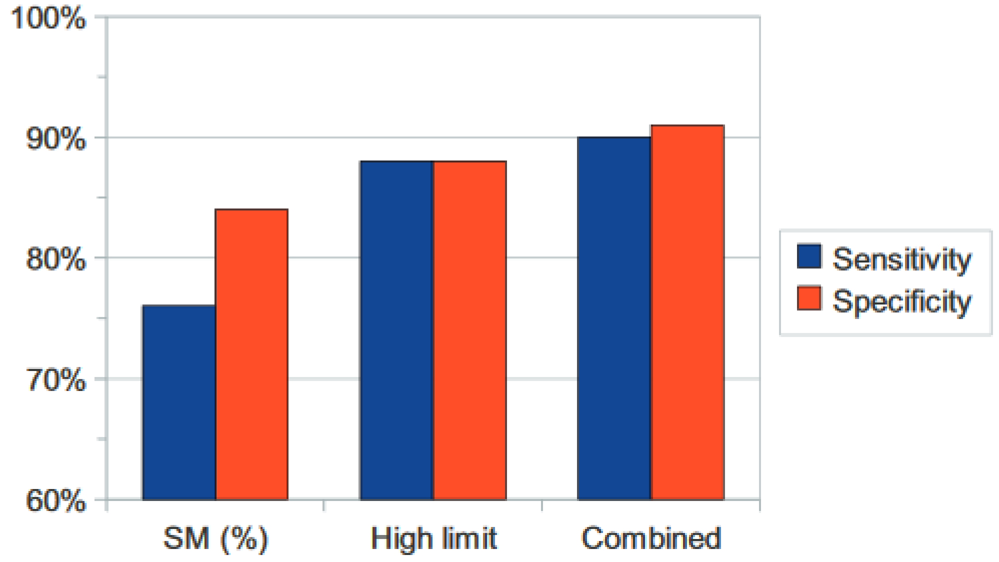


Figure 6.6. *Increase in sensitivity and specificity when changing the decision criterion from relative duration of systolic murmur (SM (%)) to high frequency limit and combined criteria (SM(%) >= 80 % OR High_limit >= 200 Hz).*

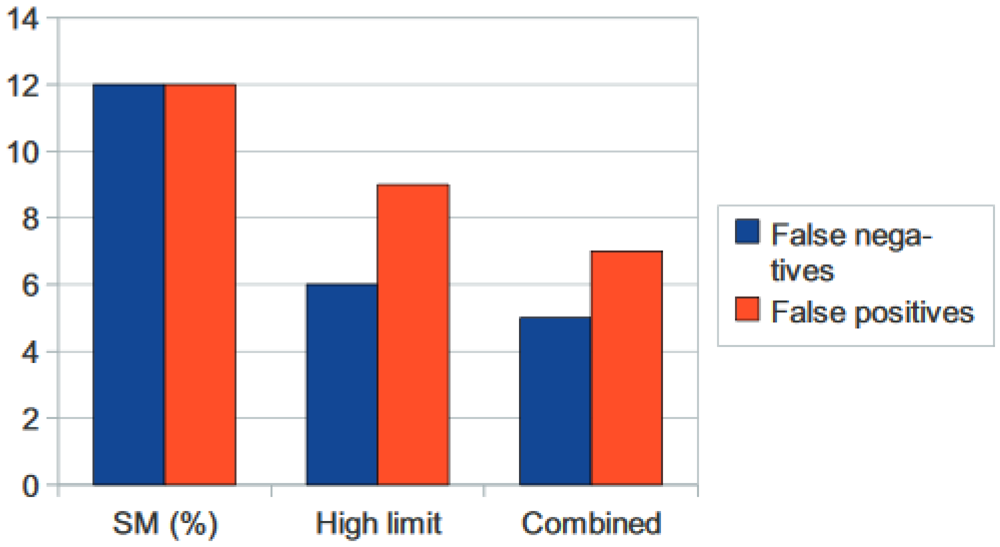


Figure 6.7. *Decrease in number of false negative and false positive cases using relative duration of systolic murmur, high frequency limit or combined criteria for selection criteria. Total number of pathological cases was 50 and total number of physiological cases was 75.*

https://aaltodoc.aalto.fi/handle/123456789/4480

How to select the threshold decision parameter

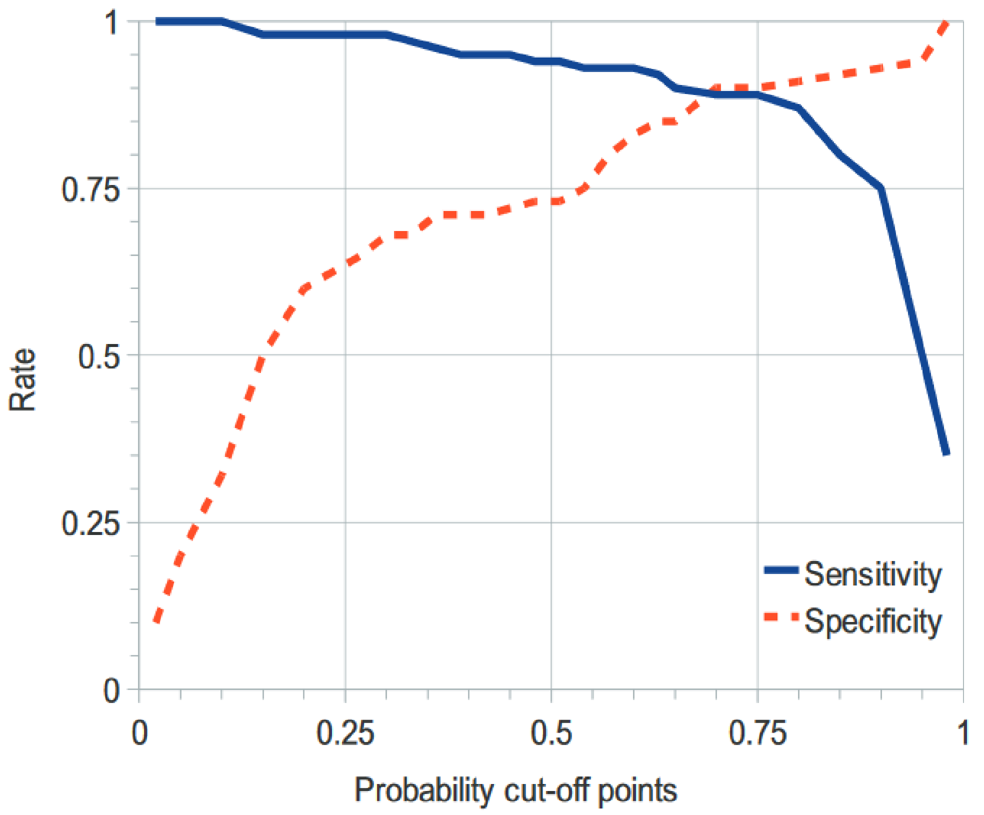


Figure 6.2. *The sensitivity and specificity of the stepwise logistic regression model at different cut-off points in detection cardiac disease in children (Publication VII).*

https://aaltodoc.aalto.fi/handle/123456789/4480

ROC curves

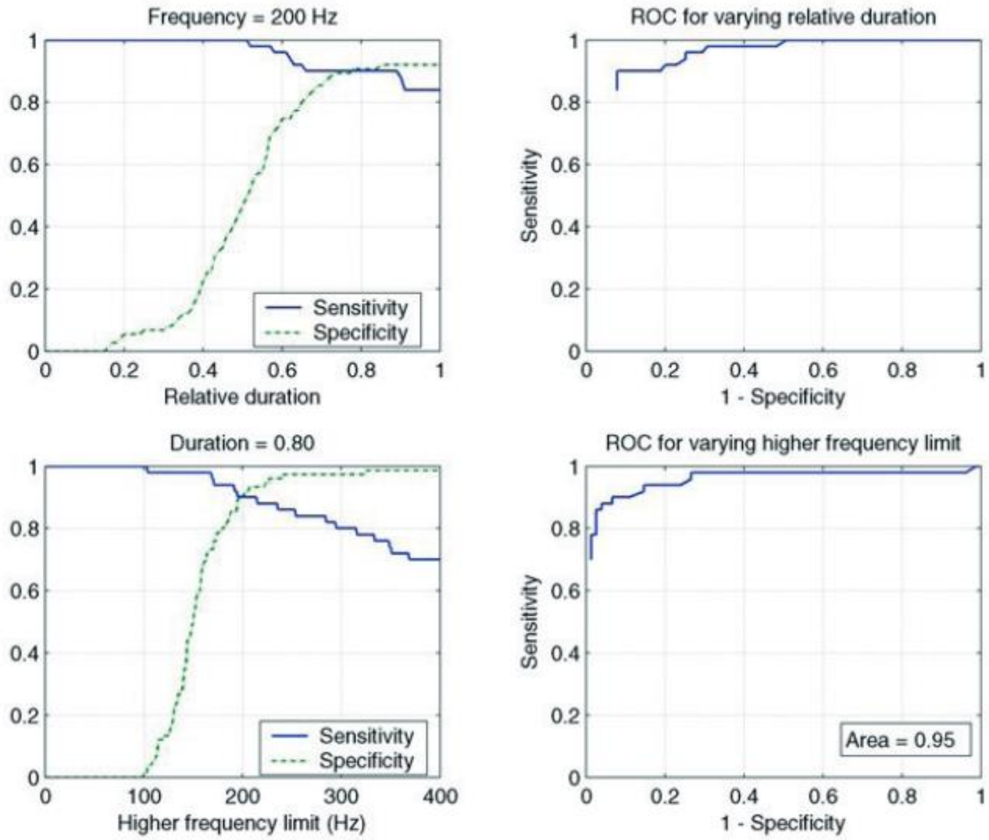


Figure 6.8. *Sensitivity and specificity for the combined criteria around the optimal point (closest to 100 % sensitivity and specificity) to differentiate between the pathological and physiological murmurs (top left). The highest frequency limit is fixed to 200 Hz and the relative duration is varied (top right). The ROC vs. the relative duration around the optimal point. The relative duration is fixed to 0.80 and the high frequency limit is varied (bottom left). ROC vs. the highest frequency limit. Area under the curve is 0.95 (bottom right).*