innovate achieve lead

BITS Pilani

Machine Learning (IS ZC464) Session 7 :

Evaluation of classification models

Discussion

- Overfitting
- cross validation
- Class-wise accuracy
- true/false positives/negatives
- Precision and recall
- sensitivity analysis
- ROC curves
- Confusion matrix

Classifier's performance

- Classifiers are learned (trained) on a finite training set .
- A learned classifier has to be tested on a different test set experimentally.
- The experimental performance on the test checks the classifier's generalization ability.
- There is a need for a criterion function assessing the classifier performance experimentally, e.g., its error rate, accuracy etc.

Classifier's performance

- Learning the training data too precisely usually leads to poor classification results on new data.

- Classifier has to have the ability to generalize.

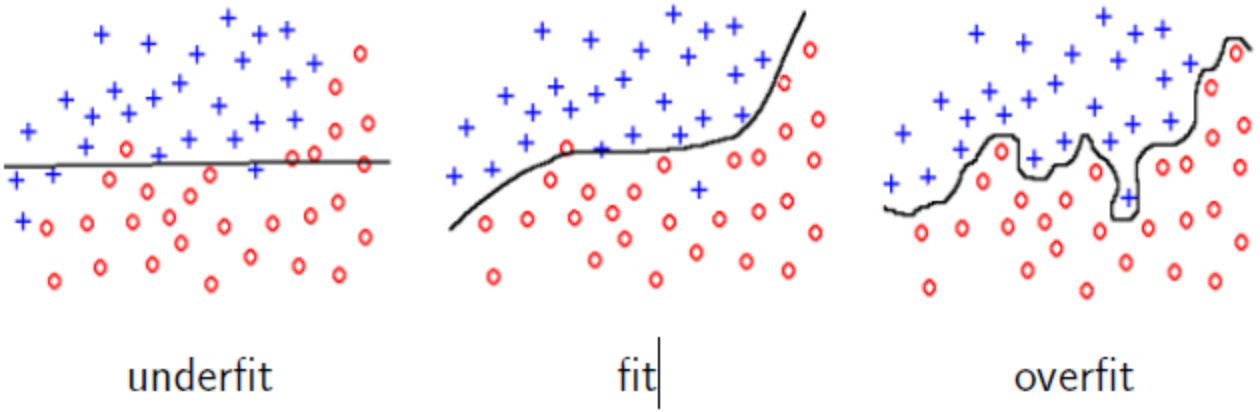
Bias

- It gives us how closeness is our predictive models to training data after averaging predicted value.
- Generally algorithm has high bias which help them to learn fast and easy to understand but are less flexible.
- That looses it ability to predict complex problem, so it fails to explain the algorithm bias.
- This results in underfitting of our model.

Variance

- It define as deviation of predictions, in simple it is the amount which tells us when its point data value change
- Ideally, the predicted value which we predict from model should remain same even changing from one training data-sets to another
- but if the model has high variance then model predict value are affect by value of data-sets.

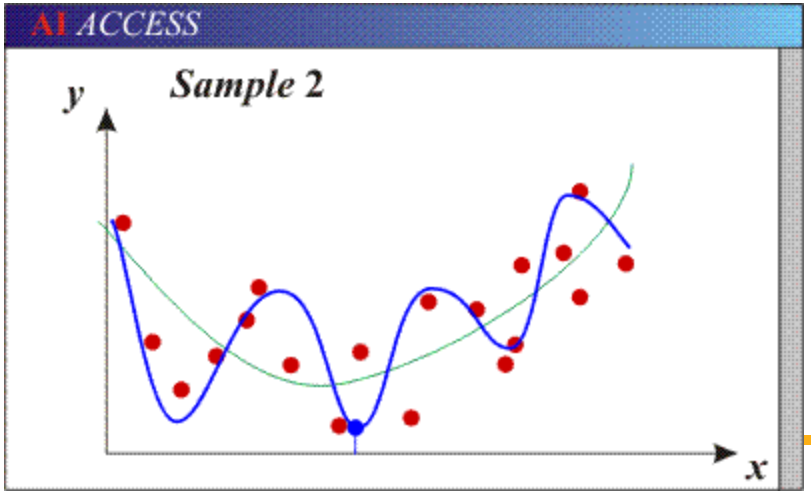
Overfitting

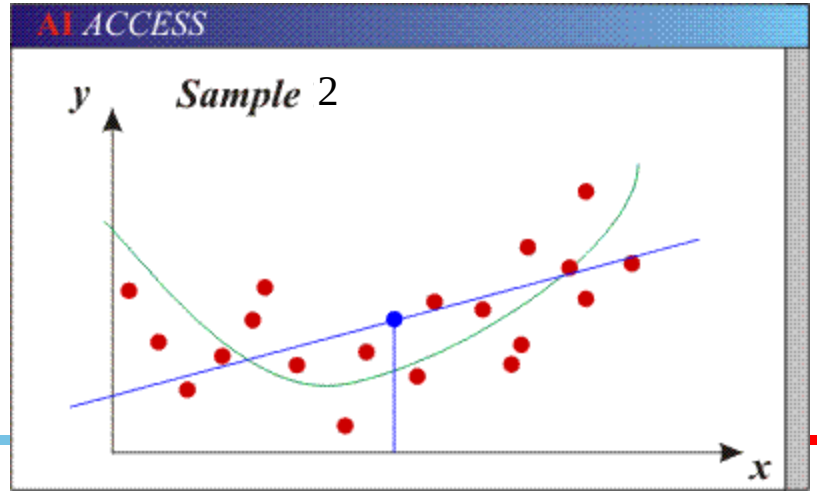


Bias/variance tradeoff

- Models with too many parameters may fit the training data well (**low bias**), but are sensitive to choice of training set (**high variance**)
- Generalization error is due to ***overfitting***

- Models with too few parameters may not fit the data well (**high bias**) but are consistent across different training sets (**low variance**)
- Generalization error is due to ***underfitting***





Reference : http://www.cs.unc.edu/~lazebnik/fall09/lec02_concepts.pptx

Training and testing

- Finite data are available only and have to be used both for training and testing.
- More training data gives better generalization.
- More test data gives better estimate for the classification error probability.
- Never evaluate performance on training data. The conclusion would be optimistically biased.

Partitioning the data

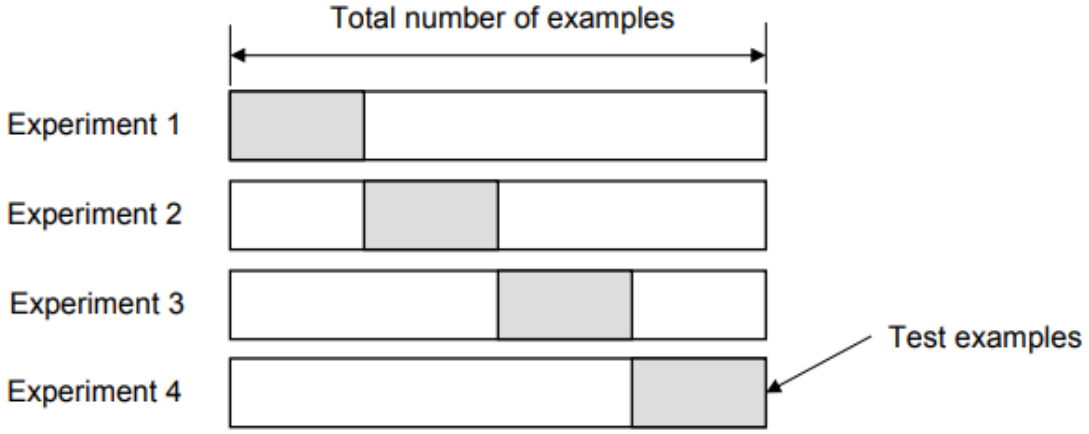
- Partitioning of available finite set of data to training / test sets.
  - **Hold out.**
  - **Cross validation.**
  - **Bootstrap.**
- Once evaluation is finished, all the available data can be used to train the final classifier.

Hold out method

- Given data is randomly partitioned into two independent sets.
- Training multi-set (e.g., 2/3 of data) for the statistical model construction, i.e. learning the classifier.
- Test set (e.g., 1/3 of data) is hold out for the accuracy estimation of the classifier.
- Random sampling is a variation of the hold out method in which the hold out is repeated *k times and the accuracy is estimated as the average of the* accuracies obtained.

K-fold cross validation

- The training set is randomly divided into *K disjoint sets of equal size* where each part has roughly the same class distribution.
- The classifier is trained *K times, each* time with a different set held out as a test set.
- The estimated error is the mean of these *K errors.*



Number of folds

- **With a large number of folds**

  Advantage

  The bias of the true error rate estimator will be small (the estimator will be very accurate)

  Disadvantage

  The variance of the true error rate estimator will be large and the computational time will be very large as well (many experiments)

Reference :https://www.cs.tau.ac.il/~nin/Courses/NC05/pr_l13.pdf

Number of folds

- **With a small number of folds**

  Advantage

  The number of experiments and, therefore, computation time are reduced

  The variance of the estimator will be small

  Disadvantage –

  The bias of the estimator will be large (conservative or smaller than the true error rate)

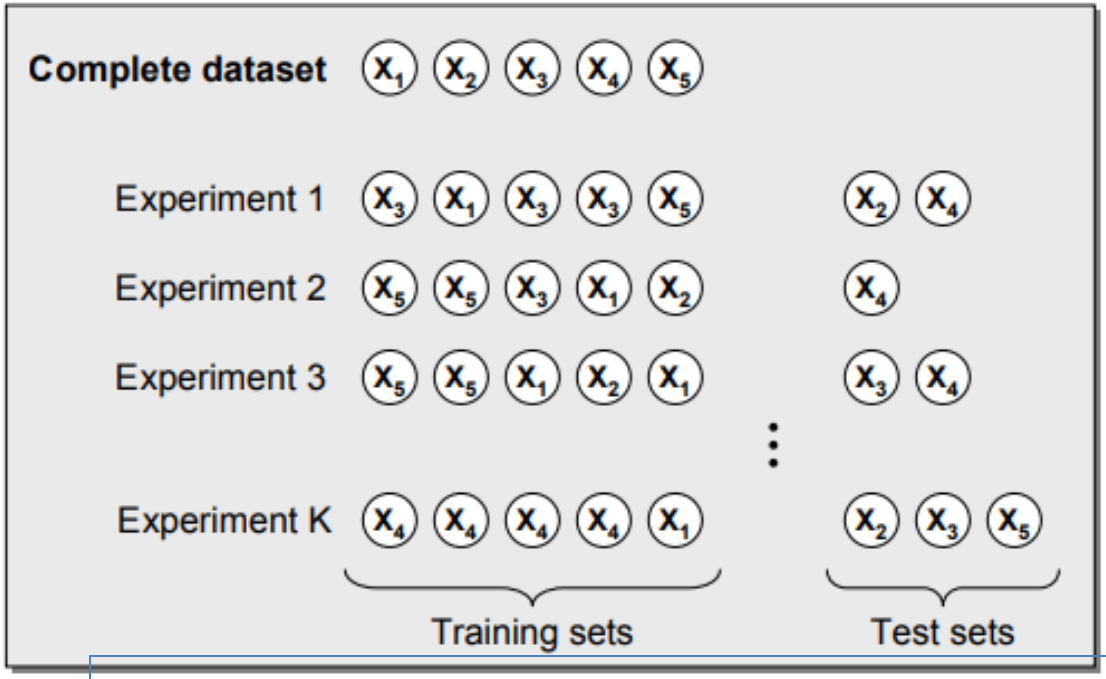
  In practice, the choice of the number of folds depends on the size of the dataset n

  For large datasets, even 3-Fold Cross Validation will be quite accurate n

  For very sparse datasets, we may have to use leave-one-out in order to train on as many examples as possible
- **A common choice for K-Fold Cross Validation is K=10**

Bootstrap Method

- The bootstrap is a resampling technique with replacement
  - From a dataset with N examples, randomly select (with replacement) N examples and use this set for training
  - The remaining examples that were not selected for training are used for testing
  - This value is likely to change from fold to fold



Repeat this process for a specified number of folds (K)
As before, the true error is estimated as the average error rate on test

Bootstrap Method

- Compared to basic cross-validation, the bootstrap **increases the variance** that can occur in each fold and this is a desirable property since it is a more realistic simulation of the real-life experiment from which our dataset was obtained.
- An additional benefit of the bootstrap is its ability to **obtain accurate measures of both the bias and variance of the true error estimate**

Model selection and true error estimates

- If model selection and true error estimates are to be computed simultaneously, the data needs to be divided into three disjoint sets
- **Training set**: a set of examples used for learning: to fit the parameters of the classifier
- **Validation set**: a set of examples used to tune the parameters of a classifier
- **Test set**: a set of examples used only to assess the performance of a fully-trained classifier

Sensitivity and Specificity

- Sensitivity is also called as *Recall* and it measures the *True Positive Rate* (TPR) of the classification
- Specificity is the measure of *True Negative Rate* (TNR).

True positive and False negative

- If a test sample of an authorized class is recognized correctly, it counts to the number of **True Positives (TP)**, while if it is rejected and labeled as impostor due to classifier's limited ability to recognize correctly, the count adds to the number of **False Negatives (FN)**.

True negative and False positive

- If a test data from the impostor database is recognized correctly as impostor, then it counts **True Negatives (TN),** while if an impostor is recognized to be falling in authorized class, it counts as **False Positives (FP)**

Table depicting meaning

Term	Meaning
True Positive (TP)	Correct Identification
True Negative (TN)	Correct Rejection
False Negative (FN)	Incorrect Rejection
False Positive (FP)	Incorrect Identification

Sensitive and specific

- A classifier is said to be efficient when not only does it recognize the authorized test data correctly, but also rejects the impostors (unauthorized persons) in the test samples.
- A perfect classifier is said to be 100% **sensitive**, if all test samples belonging to authorized database are recognized by the classifier as authorized
- A perfect classifier is said to be 100% **specific**, if all impostors are not identified as authorized or are identified as unauthorized.

Sensitivity

- ***Sensitivity*** is defined as the ratio of true positives and the total number of positive samples used in training.

$$Sensitivity\ (Recall) = \frac{TP}{P} = \frac{TP}{(TP + FN)}$$

Precision

- ***Precision*** is computed as Positive Predictive Value (PPV) and is given as

$$Precision = \frac{TP}{(TP + FP)}$$

Specificity

- ***Specificity*** is defined as follows

$$Specificity = \frac{TN}{N} = \frac{TN}{(TN + FP)}$$

Fall out

- ***Fall out*** is a measure of False Positive Rate (FPR) and is defined as 1-Specificity given below

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN + FP)}$$

Accuracy

- ***Accuracy*** is a measure of overall number of correctly recognized face image test samples

$$Accuracy = \frac{(TP + TN)}{(TP + FN + TN + FP)}$$

Percentage accuracy is obtained by multiplying the accuracy computed using above equation by 100.

Confusion Matrix

- A Classifier performance is evaluated more precisely by an error matrix called as *Confusion Matrix*.
- The columns of the matrix depict the instances of the actual classes and the rows of the matrix represent the instances in the predicted classes.
- The confusion matrix is also known as *Contingency Table* or *Error Matrix*.
- The performance of the classifier for two class classification problem is visualized as a 2×2 matrix as show

Two Class Classification based Confusion Matrix

		Actual Classes		
		Authorized (Positive)	Impostor (Negative)	
Predicted Classes	Authorized (Positive)	TP	FP	Positive Predictive Value = TP/(TP+FP)
	Impostor (Negative)	FN	TN	Negative Predictive Value = TN/(TN+FN)
		Sensitivity = TP/(TP+FN)	Specificity = TN/(TN+FP)	

Perfect classifier

- A classifier is said to be perfect if it produces non-zero values in the **diagonal** of the Confusion matrix and has all zeros at the upper and lower triangular matrix entries.
- This means that the classifier is ***not confused*** and knows who is who correctly.
- A perfect classifier must not identify incorrectly an impostor, which means a perfect classifier must have as FP = 0.
- Similarly a perfect classifier must not incorrectly classify an authorized class data as imposter, i.e. FN =0.

Perfect classifier

$$Sensitivity\ (Recall) = \frac{TP}{(TP + FN)} = \frac{TP}{(TP + 0)} = 1$$

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN + FP)} = 0$$

Confusion matrix for many class classification

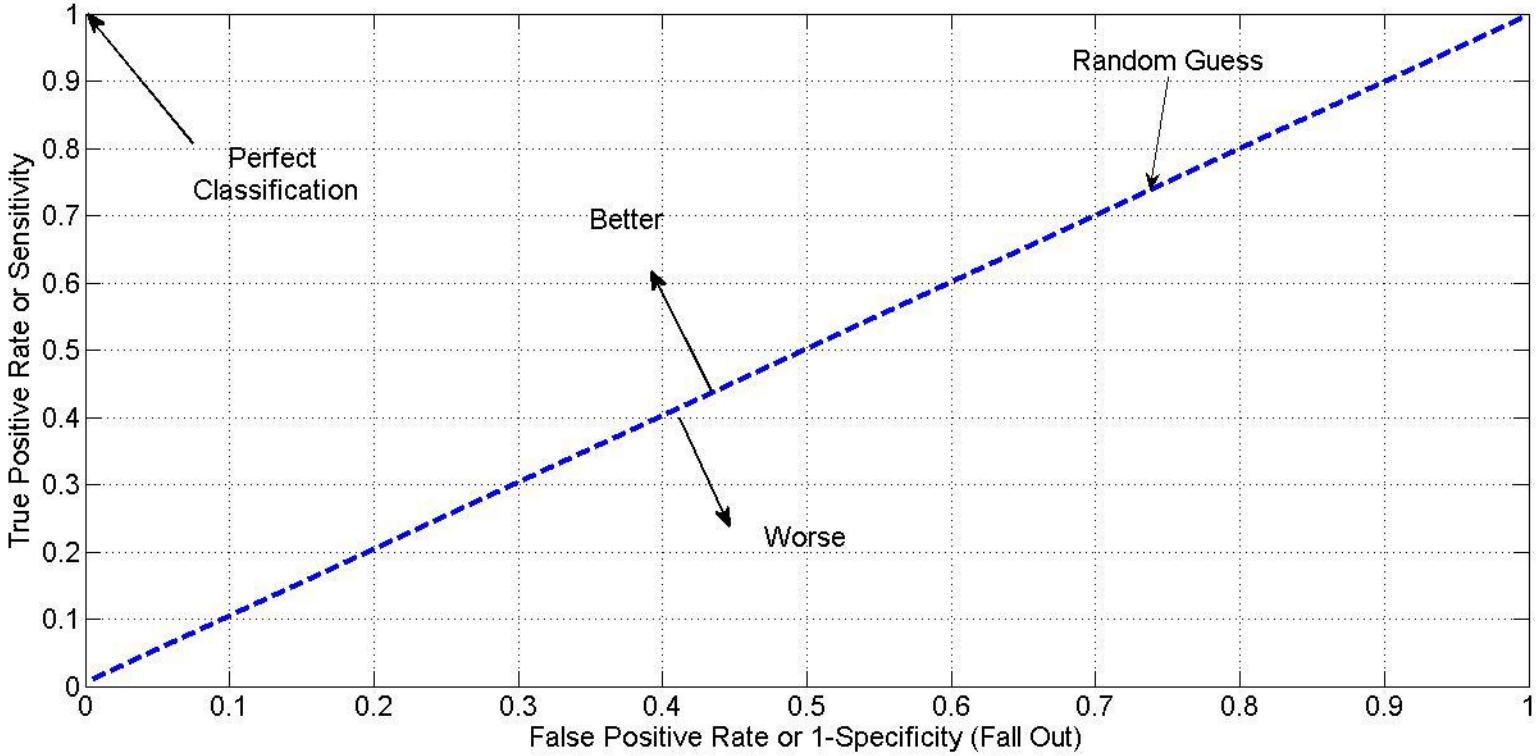
		Actual Classes									
		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Predicted Classes	C1	3	0	0	0	0	0	0	0	0	0
	C2	0	4	0	0	0	0	0	0	0	0
	C3	0	0	5	0	0	0	3	0	0	0
	C4	0	0	0	3	0	0	0	0	0	0
	C5	0	0	0	0	6	0	0	0	0	0
	C6	0	0	0	0	0	3	0	0	0	0
	C7	0	0	0	0	0	0	5	0	0	0
	C8	0	0	0	0	0	0	0	5	0	0
	C9	0	0	0	0	0	0	0	0	6	0
	C10	0	0	0	0	0	0	0	0	0	6

The sum of the entries in the diagonal of Confusion Matrix represents the number of True Positives(TP), while the sum of the non-diagonal entries represents False Negatives (FN).

Receiver Operating Characteristic (ROC) Curve

- ROC curve is a graph between False Positive Rates (FPR) and True Positive Rates (TPR) which are plotted on x-axis and y-axis respectively.
- The FPR is also known as Fall Out and is equal to (1-Specificity) while TPR is measured as Sensitivity.
- The ROC curve is the plot of Sensitivity as a function of Fall out.
- A classifier is said to be perfect if it correctly recognizes all positive test samples (TPR = 1) and rejects all negative samples correctly (FPR = 0).
- The curve with an observation resulting in extreme upper left corner point on the ROC (Sensitivity or TPR = 1 and Fallout or FPR = 0) is said to display the performance of the classifier as perfect .

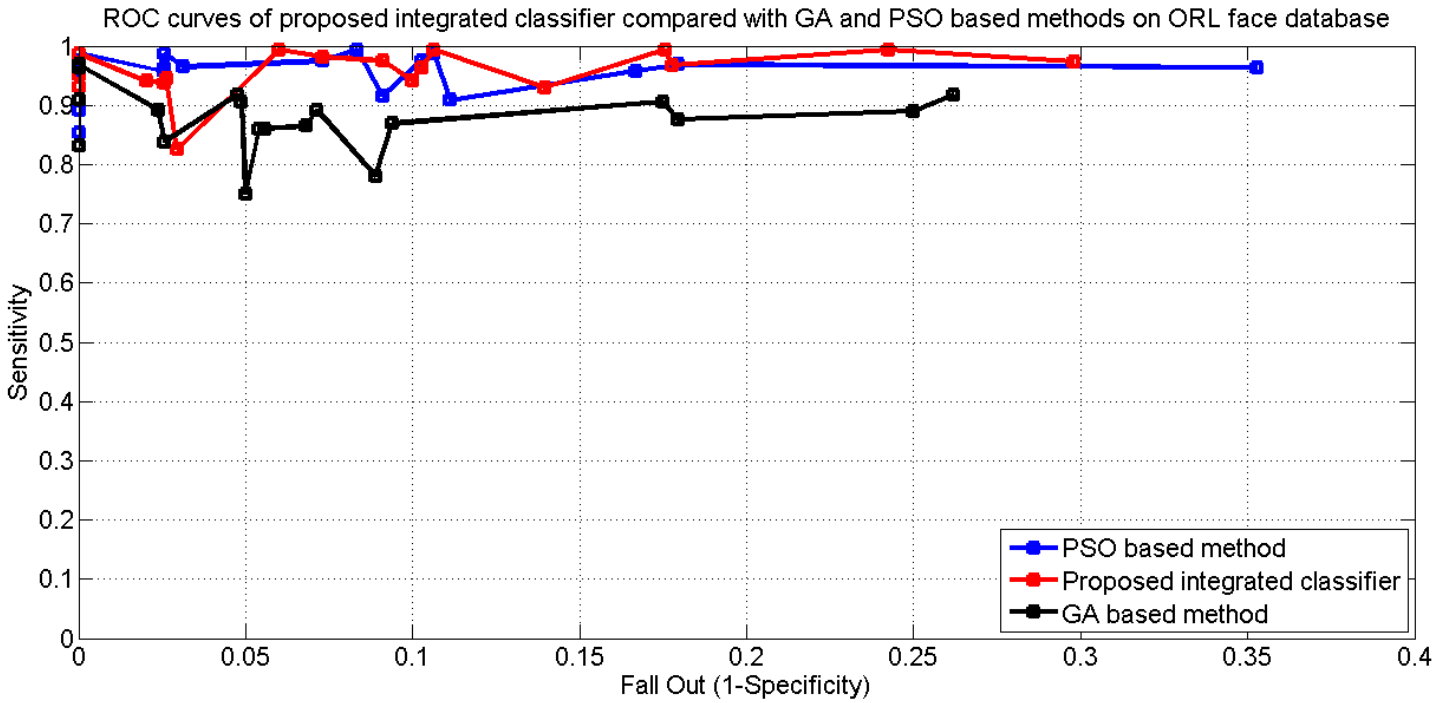
ROC curve


Perfect
Classification
Random Guess
Better
Worse
True Positive Rate or Sensitivity
False Positive Rate or 1-Specificity (Fall Out)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

ROC curve

- A classifier with fall out and sensitivity taken as point P(x,y) is said to be performing worse as P falls below the dotted line.
- If the classifier performs in terms of fall out and sensitivity taken as point Q(x,y), then it is said to be performing better as Q falls above the dotted line.
- If observations about the classifier performance are taken for a varying parameter, the ROC graph must stretch from lower left corner to the upper left corner of the graph and then to upper right corner.
- The area under ROC curve for a perfect classifier ideally is equal to 1, while it is 0.5 or less for an imperfect classifier

ROC curve


ROC curves of proposed integrated classifier compared with GA and PSO based methods on ORL face database
Sensitivity
Fall Out (1-Specificity)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.05
0.15
0.25
0.35
PSO based method
Proposed integrated classifier
GA based method