Evaluation of classifiers

Lecture 8

Outline

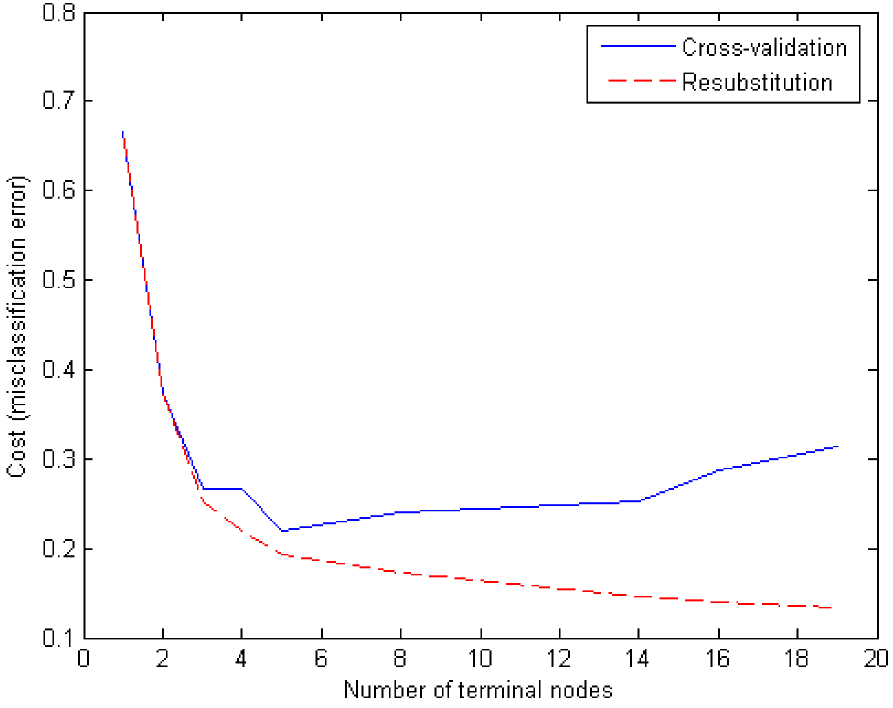
- Performance measure: error rate
- Generating test set
- Predicting performance interval
- Comparing two classifiers
- Cost-based evaluation

Error rate

- Natural performance measure for classification problems: *error rate*
  - *Success*: instance's class is predicted correctly
  - *Error*: instance's class is predicted incorrectly
  - *Error rate*: proportion of errors made over the whole set of instances

Resubstitution (training) error

- Training error - error rate obtained from training data.



Resubstitution error is (hopelessly) optimistic!

Error rate on test set

- *Test set*: independent instances that have played no part in formation of classifier
  - Assumption: both training data and test data are representative samples of the underlying problem
- Generally, the larger the training data the better the classifier
- The larger the test data the more accurate the error estimate

Outline

- Performance measure: error rate
- Generating test set
- Predicting performance interval
- Comparing two classifiers
- Cost-based evaluation

Test set?

- Simple solution that can be used if lots of (labeled) data is available:
  - Split data into training and test set

- However: (labeled) data is usually limited
  - More sophisticated techniques need to be used

Making the most of the data

- *Holdout procedure*: method of splitting original data into training and test set
  - Dilemma: ideally both training set *and* test set should be large!

- The *holdout* method reserves a certain amount for testing and uses the remainder for training
  - Usually: one third for testing, the rest for training

- Problem: the samples might not be representative
  - Example: class might be missing in the test data

- Advanced version uses *stratification*
  - Ensures that each class is represented with approximately equal proportions in both subsets

Repeated holdout method

- Holdout estimate can be made more reliable by repeating the process with different subsamples
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratificiation)
  - The error rates on the different iterations are averaged to yield an overall error rate

- This is called the *repeated holdout* method
- Still not optimum: the different test sets overlap
  - Can we prevent overlapping?

Cross-validation

- *Cross-validation* avoids overlapping test sets
  - First step: split data into *k* subsets of equal size
  - Second step: use each subset in turn for testing, the remainder for training

- Called *k-fold cross-validation*
- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

- Standard method for evaluation: stratified 10-fold cross-validation

Leave-One-Out cross-validation

- Leave-One-Out:
  a particular form of cross-validation:
  - Set number of folds to number of training instances
  - I.e., for *n* training instances, build classifier *n* times
- Makes best use of the data
- Involves no random subsampling
- But, computationally expensive

Leave-One-Out-CV and stratification

- Disadvantage of Leave-One-Out-CV: stratification is not possible
  - It *guarantees* a non-stratified sample because there is only one instance in the test set!
- Extreme example: completely random dataset split equally into two classes
  - Best inducer predicts majority class
  - 50% accuracy on fresh data
  - Leave-One-Out-CV estimate is 100% error!

The bootstrap

- Cross Validation uses sampling *without replacement*
  - The same instance, once selected, can not be selected again for a particular training/test set
- The *bootstrap* uses sampling *with replacement* to form the training set
  - Sample a dataset of *n* instances *n* times *with replacement* to form a new dataset of *n* instances
  - Use this data as the training set
  - Use the instances from the original dataset that don't occur in the new training set for testing
- Also called the *0.632 bootstrap (Why?)*

The 0.632 bootstrap

- A particular instance has a probability of 1–1/*n* of *not* being picked
- Thus its probability of ending up in the test data is:

$$\left(1-\frac{1}{n}\right)^{n} \approx e^{-1} = 0.368$$

- This means the training data will contain approximately 63.2% of the instances

Estimating error with the bootstrap

- The error estimate on the test data will be very pessimistic: trained on just ~63% of the instances
- Therefore, combine it with the training error:

$$err = 0.632 \cdot e_{\text{test instances}} + 0.368 \cdot e_{\text{training instances}}$$

The training error gets less weight than the error on the test data

- Repeat process several times with different replacement samples; average the results
- Probably the best way of estimating performance for very small datasets

Outline

- Performance measure: error rate
- Generating test set
- Predicting performance interval
- Comparing two classifiers
- Cost-based evaluation

Predicting true performance

- Assume the estimated error rate is 25%. How close is this to the true error rate?
  - Depends on the amount of test data
- Prediction is just like tossing a (biased!) coin
  - “Head” is a “success”, “tail” is an “error”
- In statistics, a succession of independent events like this is called a *Bernoulli process*
  - Statistical theory provides us with confidence intervals for the true underlying proportion

Predicting performance *interval*

- We can say: *p – probability of success* of a classifier – lies within a certain specified interval with a certain specified confidence

- Example: S=750 successes in N=1000 trials
  - Estimated success rate: 75%
  - How close is this to the true success rate p?
    - Answer: with 80% confidence $p \in [73.2,76.7]$
- Another example: S=75 and N=100
  - Estimated success rate: 75%
  - With 80% confidence $p \in [69.1,80.1]$
    - I.e. the probability that $p \in [69.1,80.1]$ is 0.8.

- Bigger the N more precise we are in our evaluation, i.e. the surrounding interval is smaller.
  - Above, for N=100 we were less confident than for N=1000.

Predicting performance *interval*

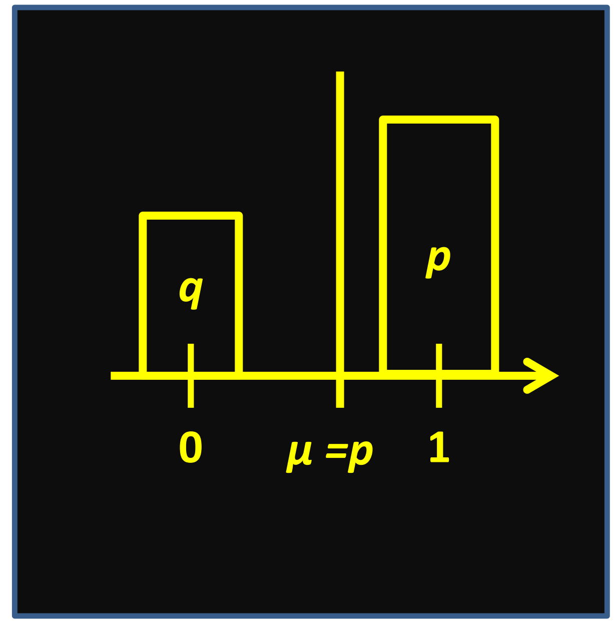
- How do we compute the predicted interval of classifier's success for a certain level of confidence?

- There is a large number of samples to be classified in the future. Out of this population we tested classifier only on N instances (N-the size of our test set).

Success as a random variable

- Let Y be the random variable with possible values
  1 for success and
  0 for error.
- Let probability of success be p.
- Then probability of error is $q=1-p$.

- What's the mean of the Y distribution?
  $\mu=1*p + 0*q = p$
- What's the variance of Y distribution?
  $\sigma^2=(1-p)^2*p + (0-p)^2*q$
  $= q^2*p+p^2*q$
  $= pq(p+q)$
  $= pq(p+1-p)$
  $=pq$

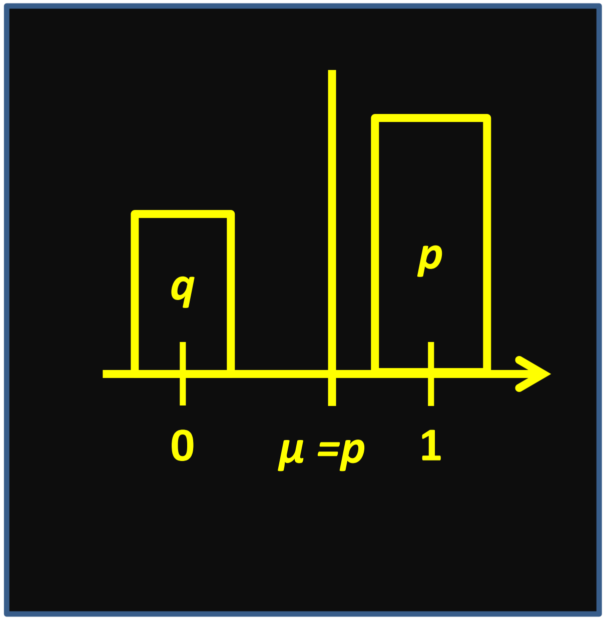


True distribution of classification success

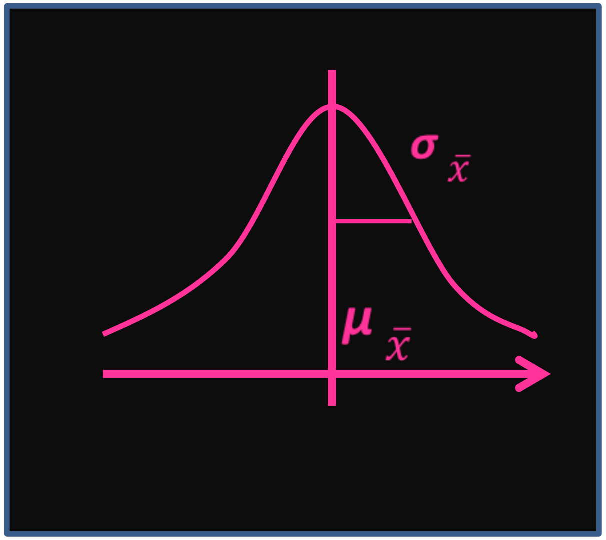
We do not know μ=p at this point!

Distribution of sampling means

We can take a random sample of size N from the entire population of Y values. The average of this one sample, $\bar{x}$, might be close to the real mean μ, and might be not.
However, if we perform many random samplings, and plot the average of each sampling, the sampling averages would have normal distribution

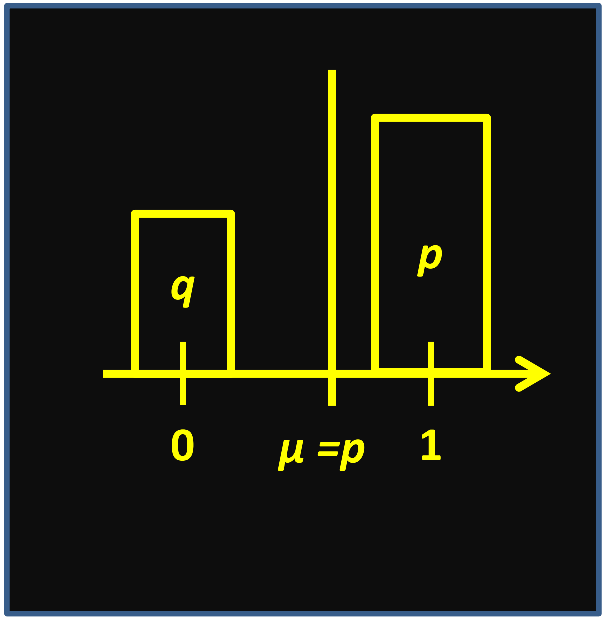


True probability distribution of *Y* in the entire population

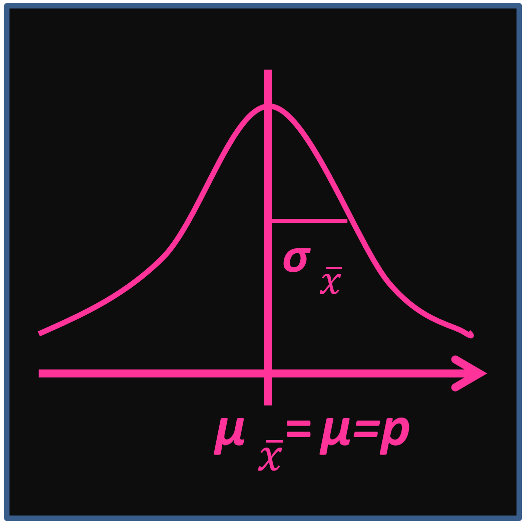


Distribution of sampling averages $\bar{x}$ for N=10

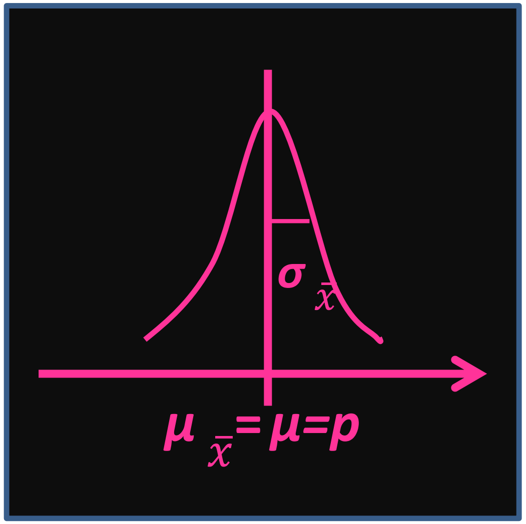
Distribution of sampling means



True distribution of classification success



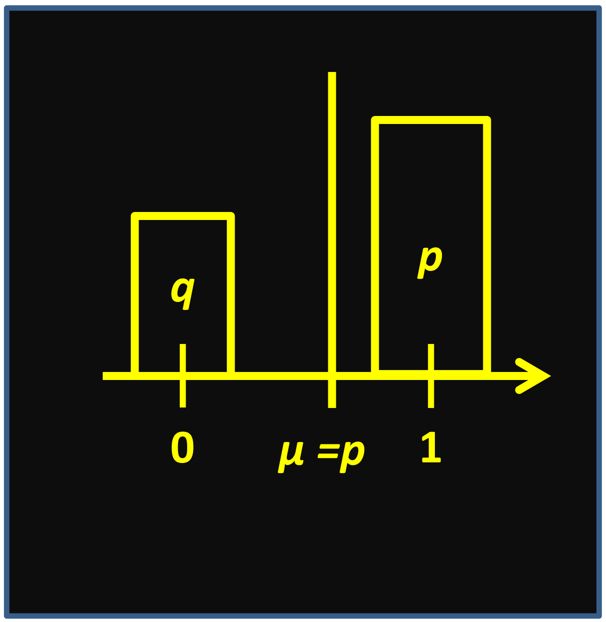
Distribution of sampling averages $\bar{x}$ for N=10



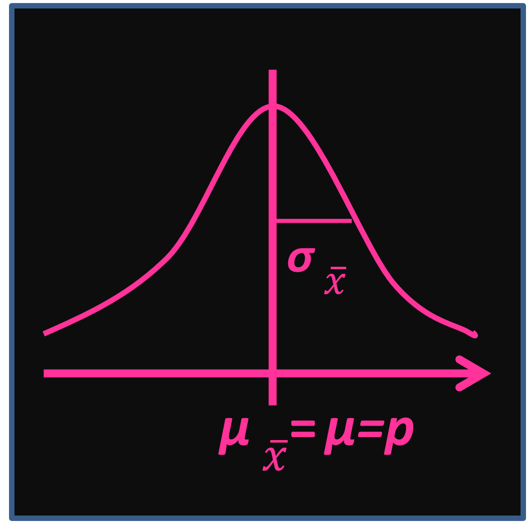
Distribution of sampling averages $\bar{x}$ for N=100

Given large enough number of samplings, the mean of sampling averages will approach the real mean of the entire population

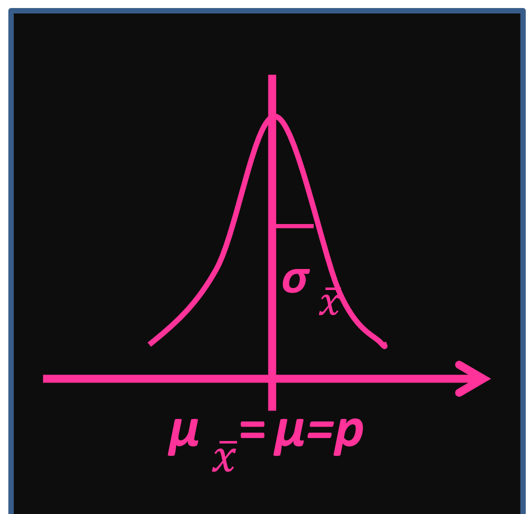
Standard deviation of sampling means



True distribution of classification success



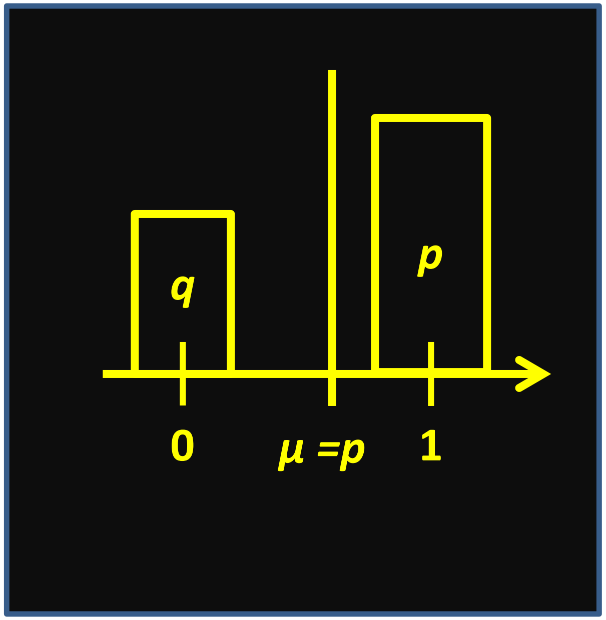
Distribution of sampling averages $\bar{x}$ for N=10



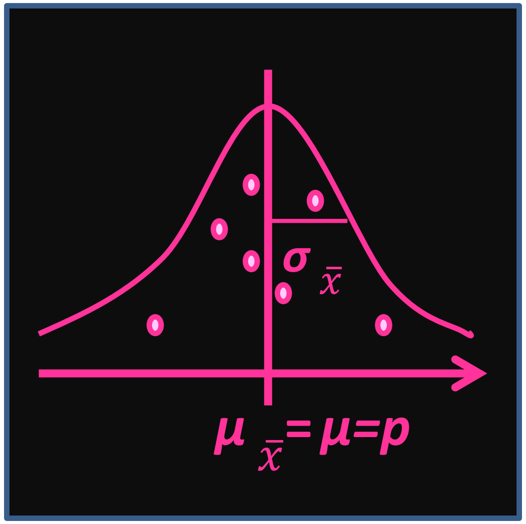
Distribution of sampling averages $\bar{x}$ for N=100

The standard deviation will be smaller if the size of each sample is larger – the larger is each sample, the less is the error of estimating the real mean from this sample

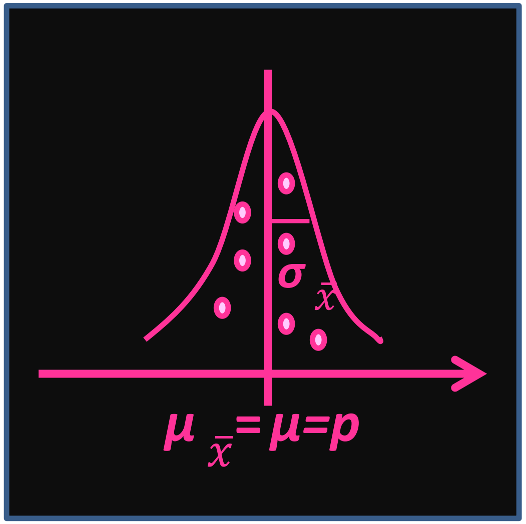
Standard deviation of sampling means



True distribution of classification success



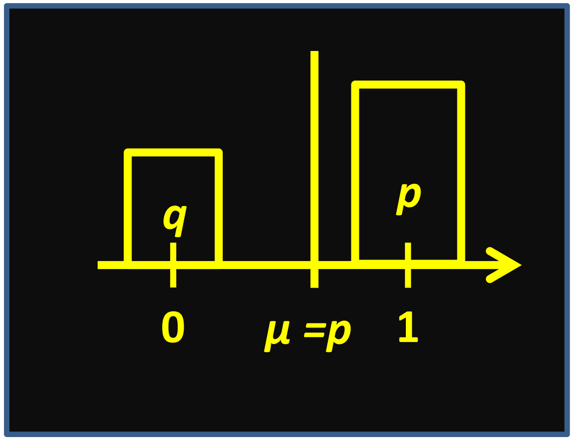
Distribution of sampling averages $\bar{x}$ for N=10



Distribution of sampling averages $\bar{x}$ for N=100

The dots, where each dot represents a mean of a particular sample, will fall closer to the real mean, if the size of each sample is large

Formula for standard deviation of the distribution of sampling means

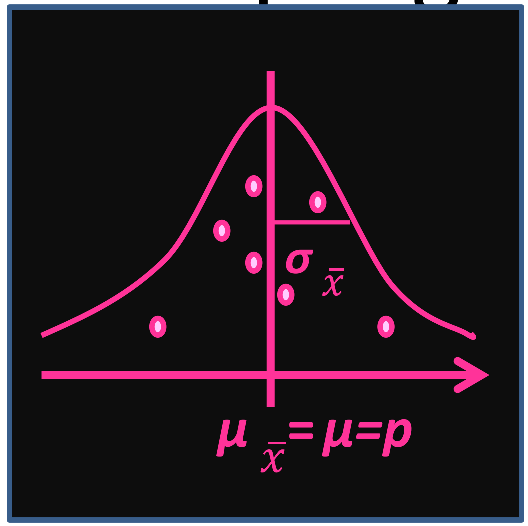


True distribution of classification success

If you take N=100 samples, you are much closer to the real mean than if you take N=2.

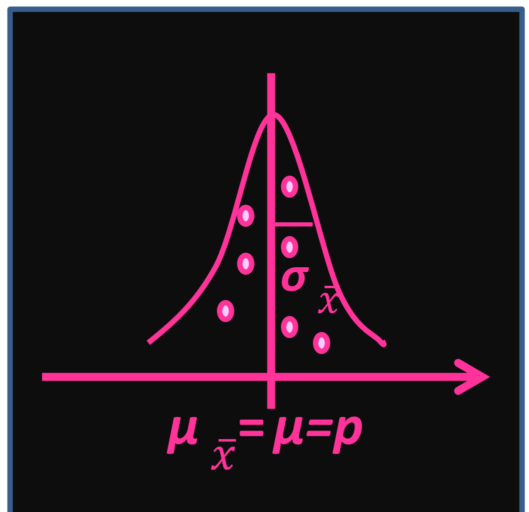
Turns out that: $\sigma^2_{\bar{x}} = \sigma^2/N$

Variance of the sampling mean distribution is inversely proportional to the size of the sample N



Distribution of sampling averages $\bar{x}$ for N=10

$\sigma_{\bar{x}} = \sigma/\sqrt{10}$



Distribution of sampling averages $\bar{x}$ for N=100

$\sigma_{\bar{x}} = \sigma/10$

Computing performance interval. Example

- How do we compute the predicted interval of classifier's success for a certain level of confidence?

- We sampled 100 instances: 75 correctly classified.
- Sample mean:

$\bar{x}$=(1*75+0*25)/100=0.75

- Sample variance:

S^2=[75*(1-0.75)^2+25*(0-0.75)^2]/ N-1=0.19

Adjustor – so we do not underestimate sample variance

Computing performance interval. Example

- How do we compute the predicted interval of classifier's success for a certain level of confidence?

- We sampled 100 instances: 75 correctly classified.
- Sample mean:

$\bar{x}$=(1*75+0*25)/100=0.75

- Sample variance:

s^2=[75*(1-0.75)^2+25*(0-0.75)^2]/ N-1=0.19

- Sample standard deviation:

s=sqrt(0.19)=0.435

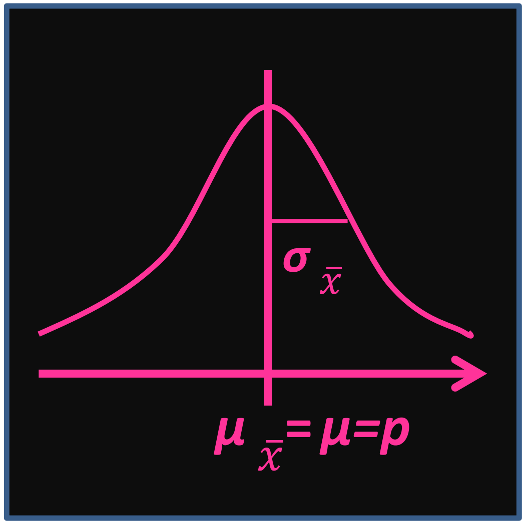
Computing performance interval. Example

- N=100 instances: 75 correctly classified.
- Sample standard deviation: ***s***=0.435

- We estimate the true standard deviation **σ** by sample standard deviation ***s***

- Now we can estimate one standard deviation of the distribution of sampling means:

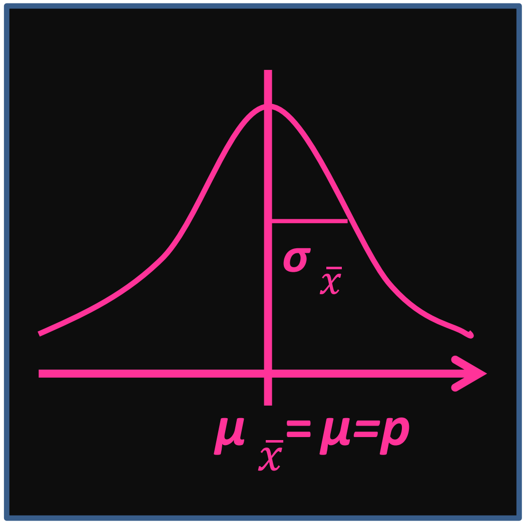
$\sigma_{\bar{x}}$= s/sqrt(N)=0.435/10=0.0435



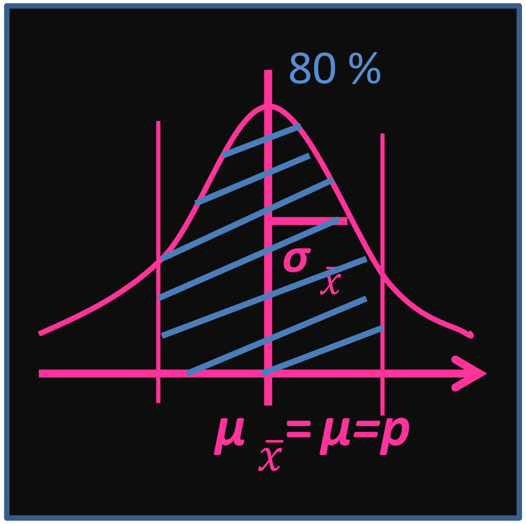
Computing performance interval. Example

$\sigma_{\bar{x}}$ = 0.0435

How many such standard deviations away from the samplings mean we need to be to have 80% confidence that any random sample mean is within this interval?



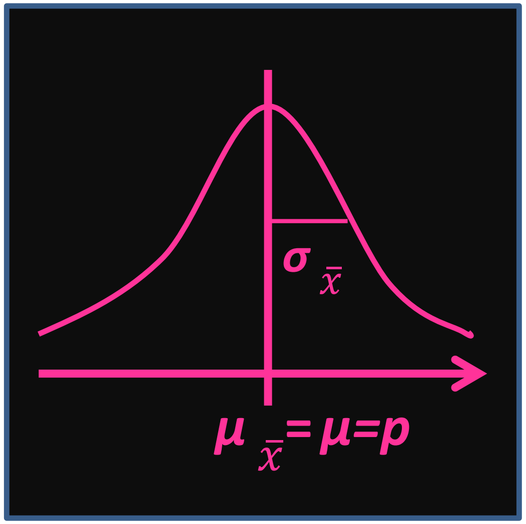
Because the mean of the distribution of the sampling means is equal to the real mean μ, answering the previous question will answer: how big an interval should we allocate around μ, such that any random sampling of size N will have its mean within this interval



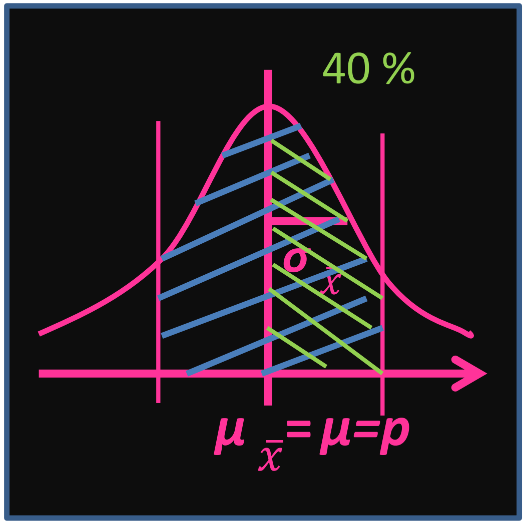
Computing performance interval. Example

$\sigma_{\bar{x}} = 0.0435$

How many such standard deviations away from the samplings mean we need to be to have 80% confidence that any random sample mean is within this interval?



Because the mean of the distribution of the sampling means is equal to the real mean μ, answering the previous question will answer: how big an interval should we allocate around μ, such that any random sampling of size N will have its mean within this interval



We want the upper part (above mean) to be 40%, since normal distribution is symmetric.

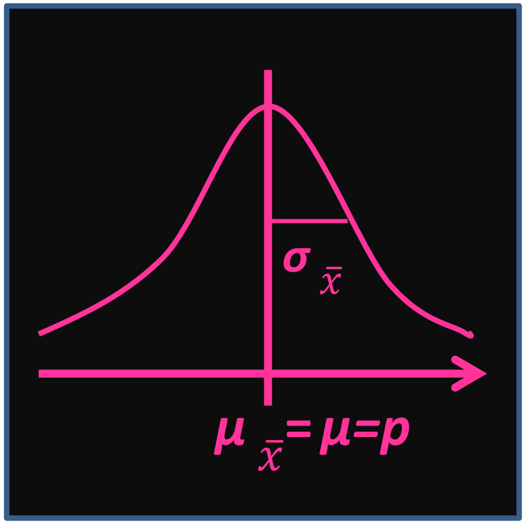
Computing performance interval. Example

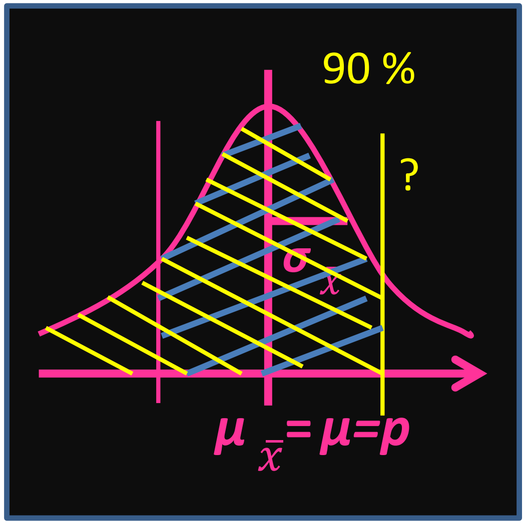
$\sigma_{\bar{x}}$ = 0.0435

How many such standard deviations away from the samplings mean we need to be to have 80% confidence that any random sample mean is within this interval?

Because the mean of the distribution of the sampling means is equal to the real mean μ, answering the previous question will answer: how big an interval should we allocate around μ, such that any random sampling of size N will have its mean within this interval

The probability of the variable to be less than the upper mark is 40+50=90%





Computing performance interval. Example

$\sigma_{\bar{x}} = 0.0435$

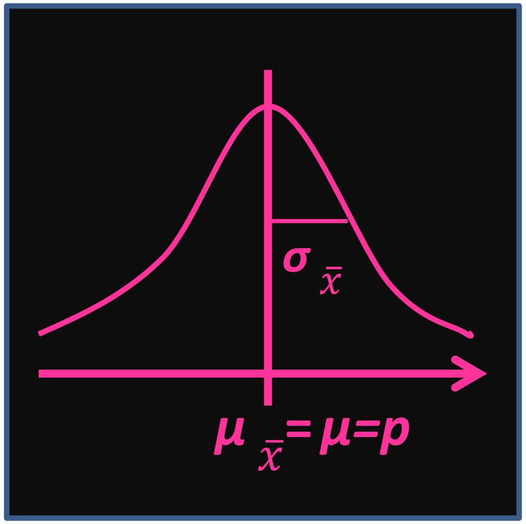
The probability of the variable to be less than the upper mark is 40+50=90%

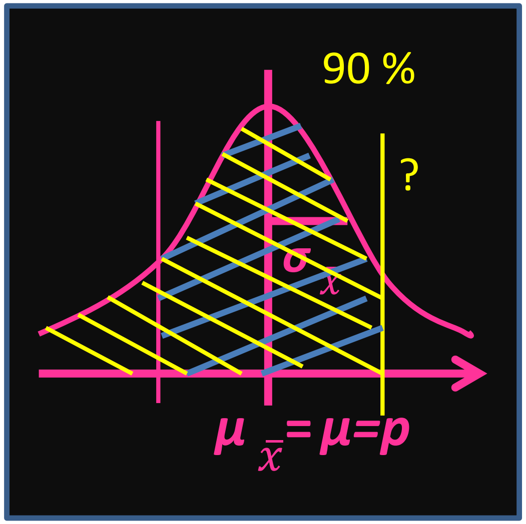
Cumulative probability up to this point

Z-table

z	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	.500	.504	.508	.512	.516	.520	.524	.528	.532	.536
0.1	.540	.544	.548	.552	.556	.560	.564	.568	.571	.575
0.2	.580	.583	.587	.591	.595	.599	.603	.606	.610	.614
0.3	.618	.622	.626	.630	.633	.637	.641	.644	.648	.652
0.4	.655	.659	.663	.666	.670	.674	.677	.681	.684	.688
0.5	.692	.695	.699	.702	.705	.709	.712	.716	.719	.722
0.6	.726	.729	.732	.736	.740	.742	.745	.749	.752	.755
0.7	.758	.761	.764	.767	.770	.773	.776	.779	.782	.785
0.8	.788	.791	.794	.797	.800	.802	.805	.808	.811	.813
0.9	.816	.819	.821	.824	.826	.829	.832	.834	.837	.839
1.0	.841	.844	.846	.849	.851	.853	.855	.858	.850	.862
1.1	$\boxed{.864}$	.867	.869	.871	.873	.875	.877	.879	.881	.883
1.2	.885	.887	.889	.891	.893	.894	.896	.898	.900	.902

How many standard deviations above the mean

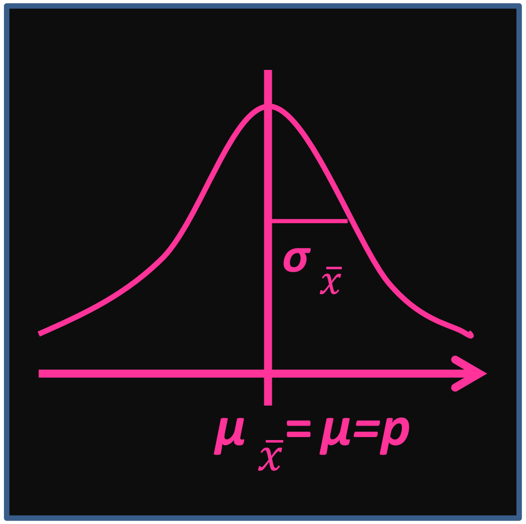




Computing performance interval. Example

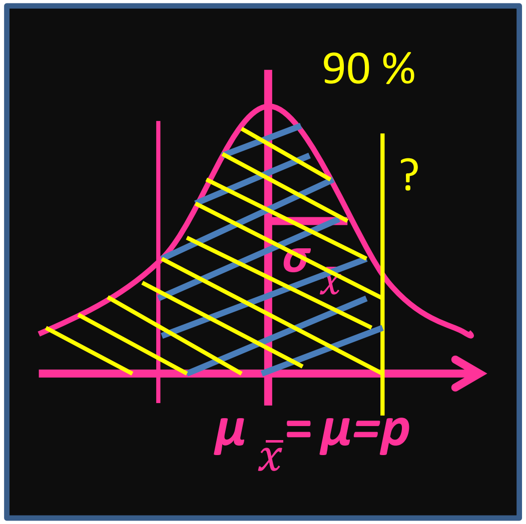
$\sigma_{\bar{x}}$ = 0.0435

Our sample mean is less than real mean plus 1.28 standard deviations with probability 90%



Z-table

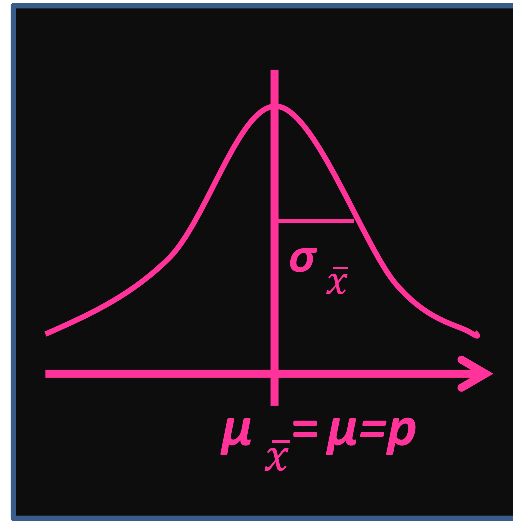
z	0.0	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	.500	.504	.508	.512	.516	.520	.524	.528	.532	.536
0.1	.540	.544	.548	.552	.556	.560	.564	.568	.571	.575
0.2	.580	.583	.587	.591	.595	.599	.603	.606	.610	.614
0.3	.618	.622	.626	.630	.633	.637	.641	.644	.648	.652
0.4	.655	.659	.663	.666	.670	.674	.677	.681	.684	.688
0.5	.692	.695	.699	.702	.705	.709	.712	.716	.719	.722
0.6	.726	.729	.732	.736	.740	.742	.745	.749	.752	.755
0.7	.758	.761	.764	.767	.770	.773	.776	.779	.782	.785
0.8	.788	.791	.794	.797	.800	.802	.805	.808	.811	.813
0.9	.816	.819	.821	.824	.826	.829	.832	.834	.837	.839
1.0	.841	.844	.846	.849	.851	.853	.855	.858	.850	.862
1.1	.864	.867	.869	.871	.873	.875	.877	.879	.881	.883
1.2	.885	.887	.889	.891	.893	.894	.896	.898	$\boxed{.900}$	.902



Computing performance interval. Example

$\sigma_{\bar{x}}$ = 0.0435

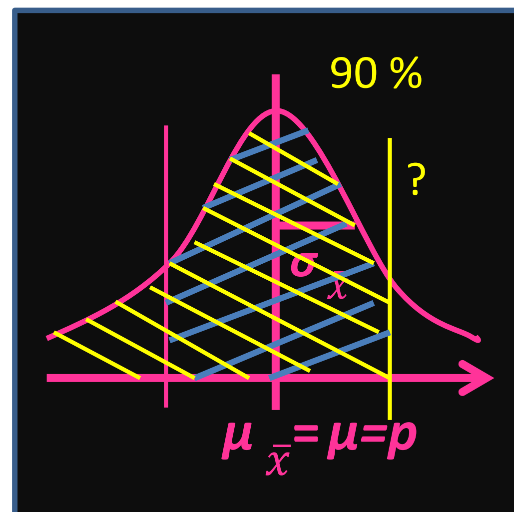
Our sample mean is less than real mean plus 1.28 standard deviations with probability 90%



Our sample mean $\bar{x}$=0.75 falls within 1.28 $\sigma_{\bar{x}}$ from the real mean $\mu=p$

or

the real mean $\mu=p$ is within 1.28 $\sigma_{\bar{x}}$ from the sample mean $\bar{x}$=0.75.



The real mean $\mu=p$ is between:

[$\bar{x}$ - 1.28 $\sigma_{\bar{x}}$, $\bar{x}$ - 1.28 $\sigma_{\bar{x}}$]

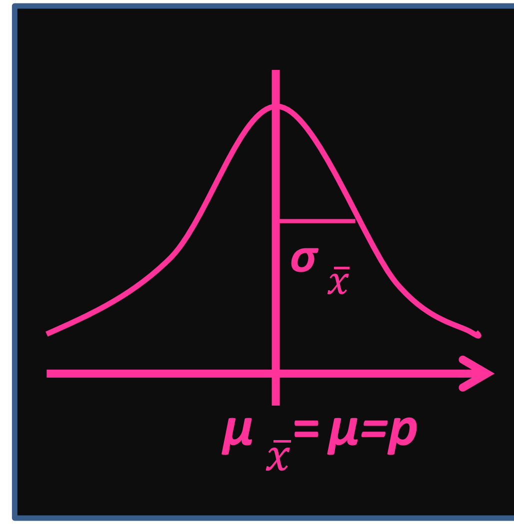
[0.75-1.28*0.0435, 0.75+1.28*0.0435]

[0.69, 0.805]

Computing performance interval. Result

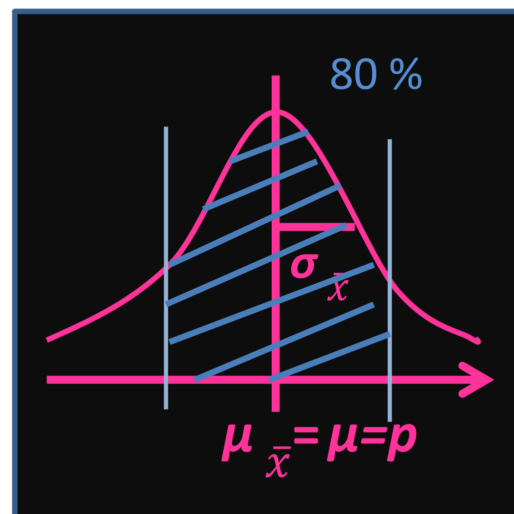
The real mean $\mu = p$ is between:

[0.69, 0.805] with the probability 80%



We can say that with confidence 80% the correctness of our classifier on real datasets is between 69% and 80.5%

Confidence – is a level of reliability of estimating the population parameter (in this case, the mean of the real population, $\mu = p$) from the sample data.



We may also say that the result [0.69, 0.805] is statistically significant with significance level 10%: significance=100%-confidence

Computing confidence interval of classifier's success rate <u>in practice</u>

- Estimate real standard deviation by computing sample standard deviation:

$$\sigma^2 \approx \Sigma_i^N (mean_X - x_i)^2/(N-1)$$

- For confidence interval C, find z-value for C/2+0.5 (from the z-table)
- Real μ=p is within:

$$p = \bar{x} \pm z \frac{\sigma}{\sqrt{N}}$$