CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Today

- Comparing Learning Algorithms

Sampling Distribution of Number of Errors

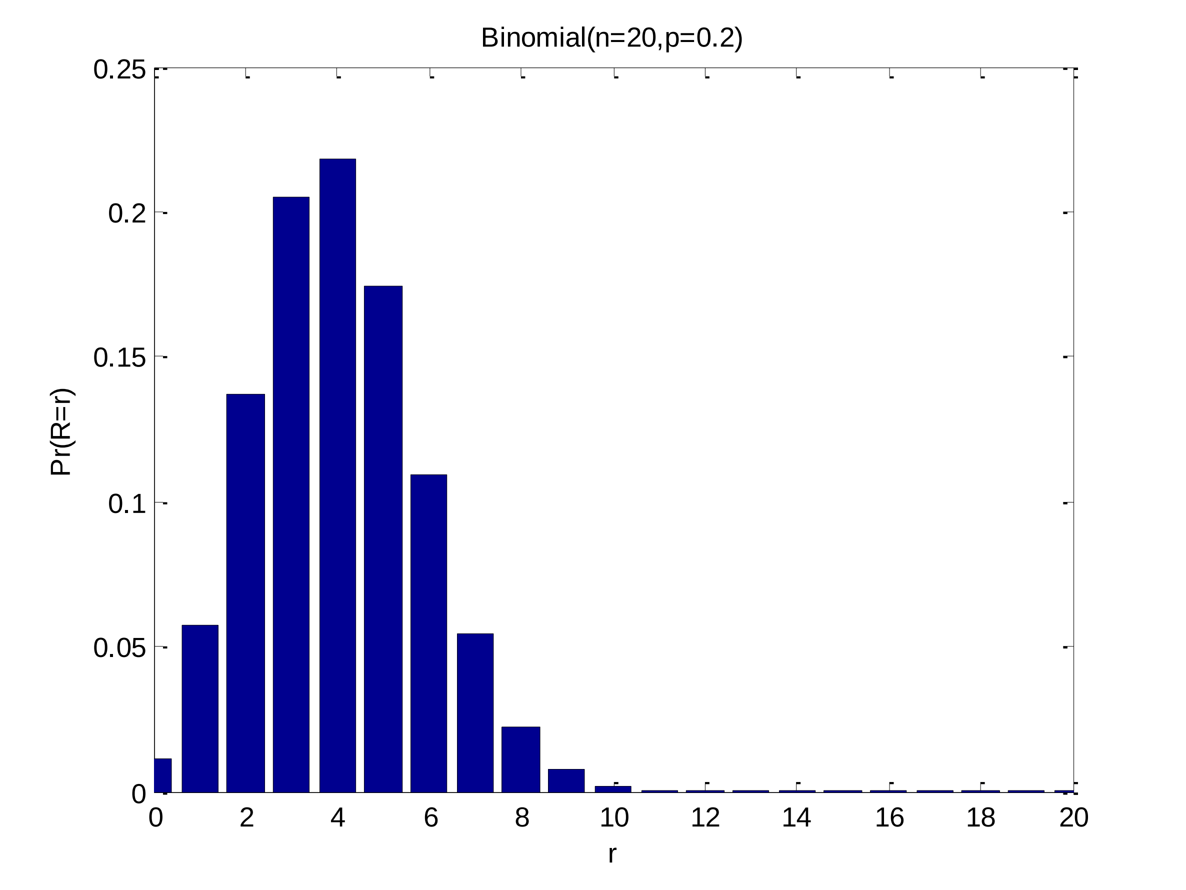
- Let R be a r.v. denoting the *number* of errors in an evaluation experiment

$$r = \sum_{x \in S} \delta(y_x, \hat{y}_x)$$

- What is the sampling distribution of R?

Sampling Distribution of R

- It is a Binomial distribution



$$B(R=r;n,p)=\binom{n}{r}p^r(1-p)^{n-r}$$

Useful Binomial Facts

- Expectation of a Binomial random variable R with distribution $B(n,e_D)$

$$E(R) = ne_D$$

- Variance of a Binomial random variable with distribution $B(n,e_D)$

$$V(R) = ne_D(1 - e_D)$$

Parameter Estimation

- Notice that in this case, we are working with a distribution whose parameters are unknown
  – We are trying to *estimate* e_D , given r and n

- Suppose we only did a single experiment with n examples and observed r errors
  – What is a good estimate of e_D?

Parameter Estimation

- It is $e_S = r/n$. Why?
- This is the estimate that, under the Binomial distribution, *maximizes the likelihood of the observed number of errors*:

$$\hat{e}_D = \arg\max_p B(R = r; n, p) = e_S = \frac{r}{n}$$

- Called the Maximum Likelihood Estimate, or MLE

Variance

- Given the sampling distribution, we can now talk about *the variance in our estimate*

$$\hat{e}_D = e_S = \frac{r}{n}$$

- Notice that the error rate r.v. $E_D = R/n$
- So $$V(E_D) = V\left(\frac{R}{n}\right) = \frac{1}{n^2} V(R)$$

$$V(R) = n e_D (1 - e_D)$$

$$V(E_D) = \frac{e_D (1 - e_D)}{n}, \text{using } \hat{e}_D = \frac{r}{n}$$

Example

- We use ID3 to learn a decision tree. On a test set with 100 examples the resulting tree misclassifies 20 examples.
  - What is the expected error rate of this tree?
  - What is the variance in our estimate?

$$\hat{e}_D = E(E_D) = r / n = 20 / 100 = 0.2$$

$$V(E_D) = \frac{\hat{e}_D(1-\hat{e}_D)}{n} = (r / n)(1 - r / n) / n$$

$$= 0.2(1-0.2) / 100 = 0.0016$$

Example

- We use ID3 to learn a decision tree. On a test set with 10000 examples the resulting tree misclassifies 2000 examples.
  - What is the expected error rate of this tree?
  - What is the variance in our estimate?

$$\hat{e}_D = E(E_D) = r / n = 2000 / 10000 = 0.2$$

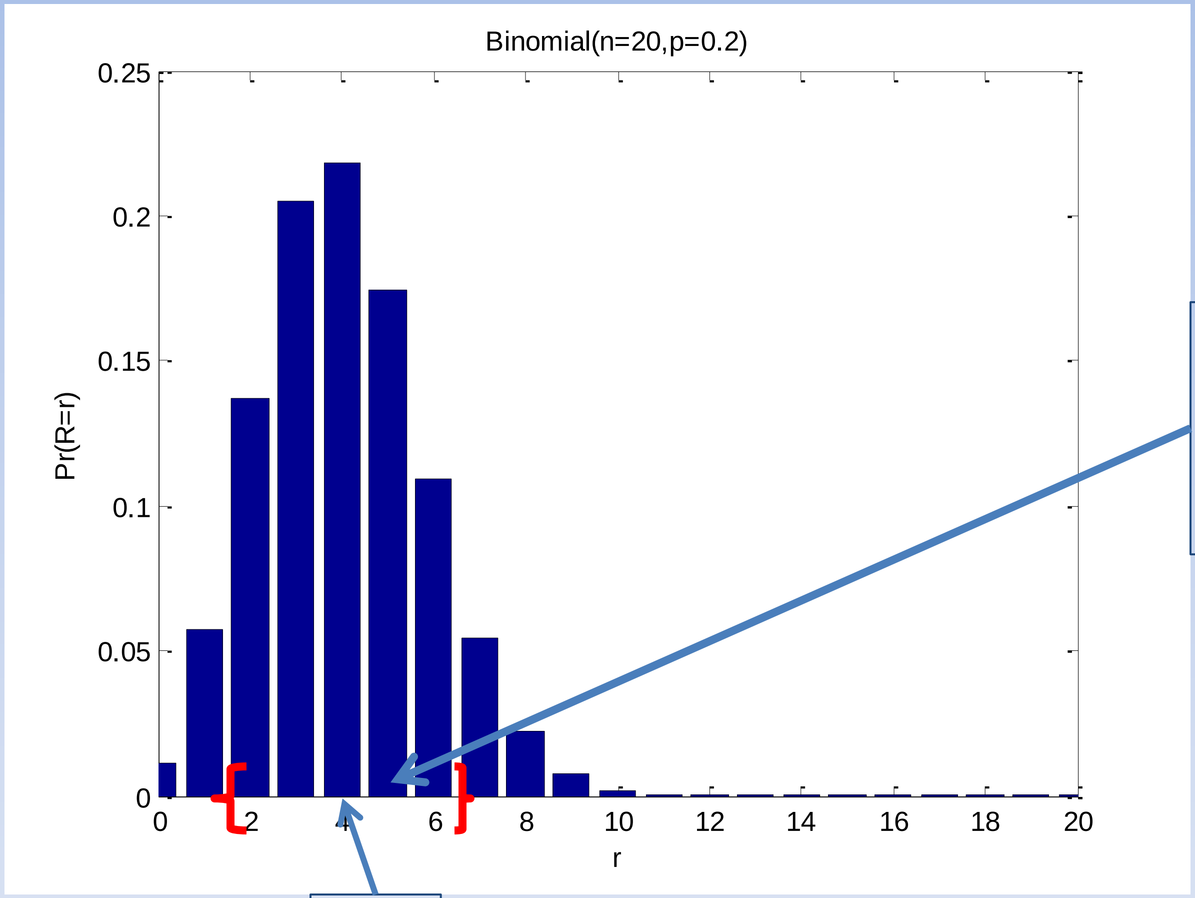
$$V(E_D) = \frac{\hat{e}_D(1-\hat{e}_D)}{n} = (r / n)(1 - r / n) / n$$

$$= 0.2(1-0.2) / 10000 = 0.16e-4$$

Confidence Intervals

- How do we use the variance estimate?
  - We can use it to describe the uncertainty in our estimate of E_D
  - We produce an interval around $\hat{e}_D$ in which a new estimate of E_D will fall with probability C
  - Called the C% confidence interval for E_D

Confidence Interval for R



85% CI: With prob 0.85, the true r will be in the range (2,6).

$E(R)$

Finding C% CI

- If n is large enough, the Binomial is well-approximated by a Gaussian distribution

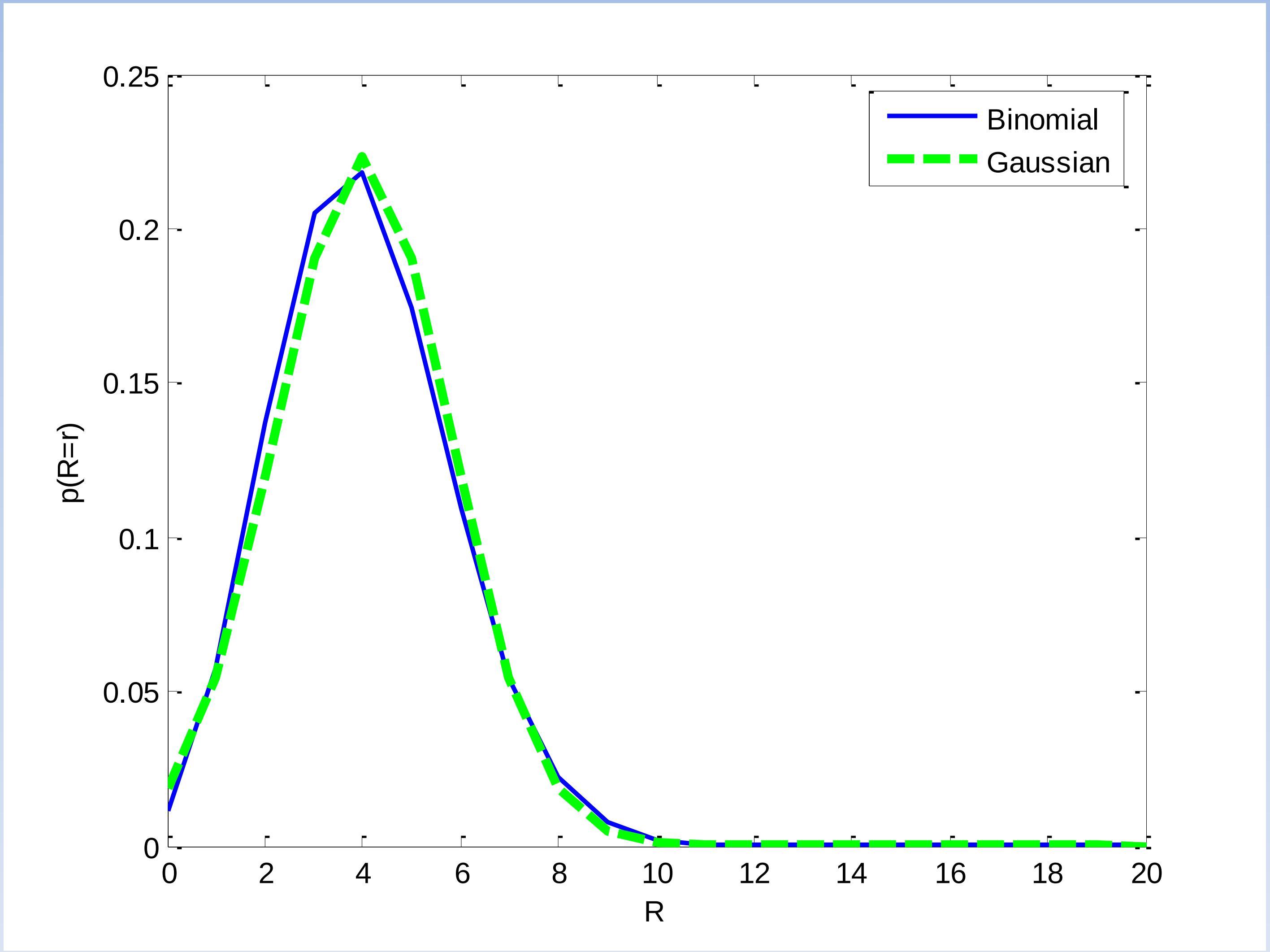
$$p(r;\mu,\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}\left(\frac{r-\mu}{\sigma}\right)^2};E(r)=\mu;V(r)=\sigma^2$$

- With parameters

$$\mu=np$$

$$\sigma=\sqrt{np(1-p)}$$

Normal Approximation



How does this help?

- There are tables available that give the size of the interval around μ as a function of σ that contains C% of the probability, for various C

- For example, see table 5.1 in Mitchell
  - Thus an interval of width $\pm 1.96\sigma$ around μ contains the 95% confidence interval

Example

- We use ID3 to learn a decision tree. On a test set with 100 examples the resulting tree misclassifies 20 examples.
  - What is the 95% CI?

$$\hat{e}_D = r/n = 20/100 = 0.2$$

$$V(E_D) = \frac{\hat{e}_D(1-\hat{e}_D)}{n} = 0.2(1-0.2)/100 = 0.0016$$

So σ=0.04 and with prob 0.95, a different estimate would lie in the range $(0.2 \pm 1.96 \times 0.04) = (0.1216, 0.2784)$

Example

- We use ID3 to learn a decision tree. On a test set with 10000 examples the resulting tree misclassifies 2000 examples.
  - What is the 95% CI?

$\hat{e}_D = r/n = 2000/10000 = 0.2$

$V(E_D) = \dfrac{\hat{e}_D(1-\hat{e}_D)}{n} = 0.2(1-0.2)/10000 = 0.16e-4$

So σ=$0.4e-2$ and with prob 0.95, a different estimate would
lie in the range $(0.2 \pm 1.96 \times 0.004) = (0.19216, 0.20784)$

Recap: Issue #1

- Suppose we collect some test data from a binary classification problem and evaluate a classifier. The accuracy is x.

- Then we (or someone else) repeats the experiment with another set of test data from the same problem, collected independently of the first set.
  - What can we say about the accuracy in this case?

Summary: Issue #1

- Determine sampling distribution of measure
- Estimate sampling distribution parameters using MLE on test set
  - (If necessary, approximate using standard distribution such as Gaussian)
- Use tables to figure C% CI
  - Usually use C=95
  - The true measure will lie in that interval with C % probability

Issue #2

- We have a conjecture, “Classifier/Algorithm A is *better than* B for this learning problem”
- How do we verify or reject this conjecture?
  - Fundamental question in all of science
    - “Theory A explains these observations better than B”
- One answer: Use *statistical hypothesis testing*

2.1 Comparing Classifiers

- Suppose we have two classifiers and we want to estimate the difference between their accuracies
  - We observe their errors e_{S,C_1} and e_{S,C_2} in separate experiments
  - They look different, but this could just be random variation in the sample
- We want to know, "What is the probability that $e_{D,C_1} \neq e_{D,C_2}$?"

Sampling Distribution

- Here the appropriate measure is the difference of the error rates

$$F = E_{D,C_1} - E_{D,C_2}$$

- What is the sampling distribution of F?

$$E(F) = e_{S,C_1} - e_{S,C_2} = \left(\frac{r_1}{n_1} - \frac{r_2}{n_2}\right)$$

$$V(F) = V(E_{D,C_1}) + V(E_{D,C_2}) = \frac{e_{S,C_1}(1 - e_{S,C_1})}{n_1} + \frac{e_{S,C_2}(1 - e_{S,C_2})}{n_2}$$

Comparing Classifiers

- Establish a "Null hypothesis" that we will try to reject with high (say 95%) probability
  - E.g. $E_{D,C_1} - E_{D,C_2} = 0$
  - Presumed true until hypothesis test shows otherwise
  - Negation is called "alternative hypothesis"
- Find sampling distribution of LHS and determine if RHS lies within 95% CI of mean
  - If it does, null hypothesis CANNOT be rejected
  - If it does not, null hypothesis CAN be rejected

Example

- On a test set with 100 examples a decision tree misclassifies 20 examples. On the same test set, a neural network misclassifies 25 examples. Are these two classifiers actually different on this problem?

$$F = r_1 / n_1 - r_2 / n_2 = 0.05$$

$$V(F) = 0.2(1-0.2)/100 + 0.25(1-0.25)/100$$

$$= 0.0016 + 0.001875 = 0.003475$$

So σ=0.059 and the 95% CI is

$(0.05 \pm 1.96 \times 0.059)$=$(-0.1245, 0.2245)$

Since zero lies in the 95% CI, the null hypothesis CANNOT be rejected (with 95% confidence).

Example

- On a test set with 1000 examples a decision tree misclassifies 200 examples. On the same test set, a neural network misclassifies 250 examples. Are these two classifiers actually different on this problem?

$$F = r_1 / n_1 - r_2 / n_2 = 0.05$$

$$V(F) = 0.2(1-0.2)/1000 + 0.25(1-0.25)/1000$$

$$= 0.00016 + 0.0001875 = 0.0003475$$

So σ=0.019 and the 95% CI is

$$(0.05 \pm 1.96 \times 0.019) = (0.014, 0.086)$$

Since zero does not lie in the 95% CI, the null hypothesis CAN be rejected (with 95% confidence).

#2.2: Comparing Learning Algorithms

- This is different from the classifier comparison because the training set will vary as well
- Let $A(Tr)$ and $B(Tr)$ denote the classifiers learned by algorithms A and B on train set Tr
- Let $$\begin{aligned} E_A &= E_{Tr\sim D^n}(\Pr_{x\sim D}(\, y_x \neq \hat{y}_x | A(Tr))) \\ &= E_{Tr\sim D^n}(E_{D,A(Tr)}) \end{aligned}$$
- We are looking for an estimate of $E_A - E_B$

Paired Testing

- When comparing algorithms, we'll usually train and test them on the *same data*
- This will usually give us better (narrower) CI's than if we use separate train/test sets
- This is called paired testing

$$E_A - E_B = E_{Tr\sim D^n}(E_{D,A(Tr)} - E_{D,B(Tr)})$$

$vs.$

$$E_A - E_B = E_{Tr\sim D^n}(E_{D,A(Tr)}) - E_{Tr\sim D^n}(E_{D,B(Tr)})$$

Comparing Algorithms

- Our null hypothesis is: “the error rates of the two algorithms are equal”, i.e. neither is any better than the other

- To evaluate an algorithm we’ll usually use n-fold CV
  - This gives an estimate of E_A in the previous slide

Comparing Algorithms

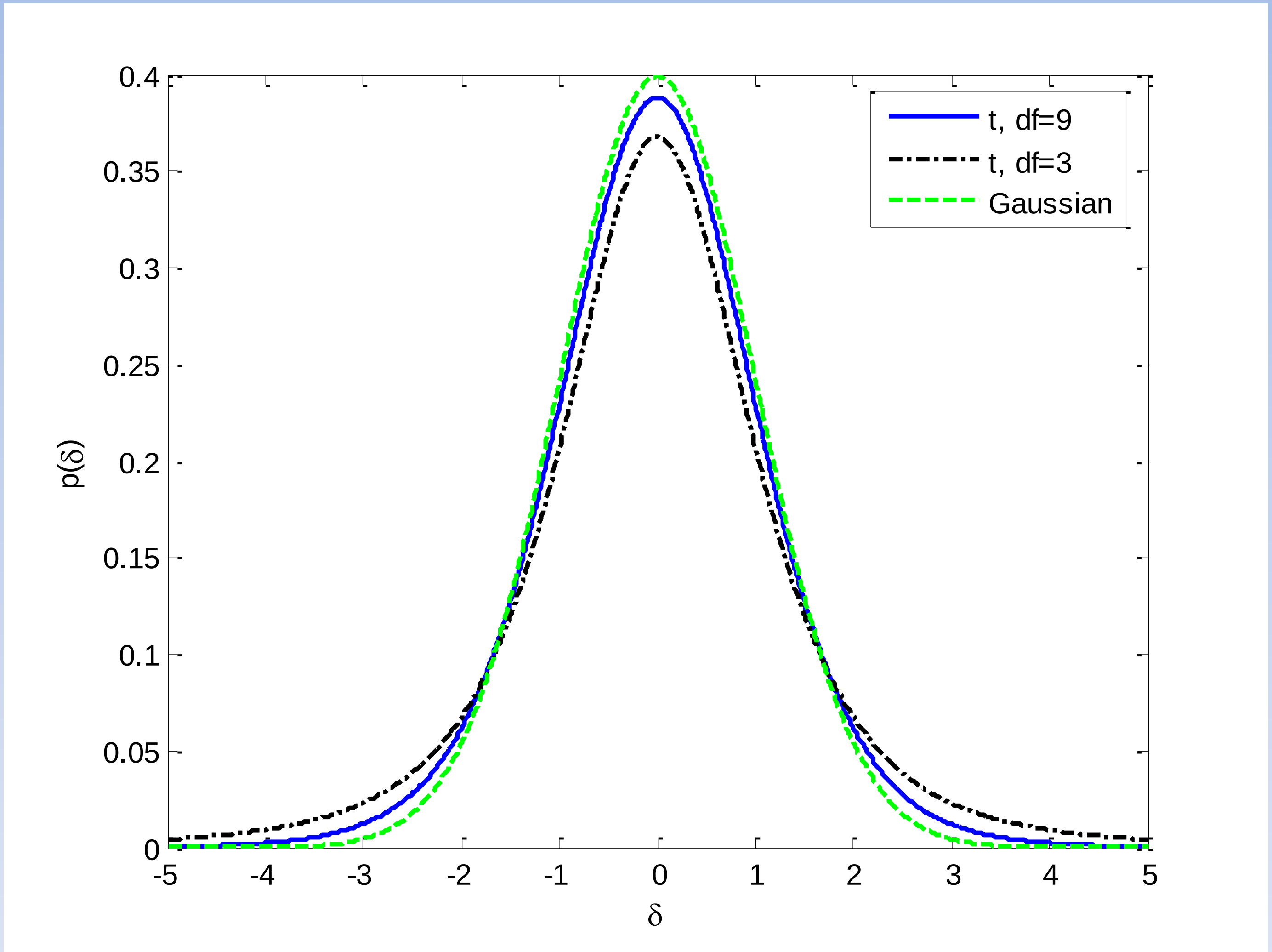
- Perform cross validation to measure the quantities of interest, E_A and E_B
- Get a number of measurements
- Each measurement will vary because of variation in the training/testing sample

Example

Fold	Error rate of Algorithm A	Error rate of Algorithm B	$E_A - E_B$
1	5%	3%	2%
2	1%	3%	-2%
3	8%	4%	4%
4	5%	1%	4%
5	1%	4%	-3%
Average	**4%**	**3%**	**1%**

Our initial estimate of the difference between A and B is 1%.
But maybe this is just due to randomness in the data?
Well, suppose we could do 5-fold cv many many times and plot average $E_A - E_B$. What would that look like?

t-distribution



The t-test

- If n was large enough, can use Gaussian here with sample means and variances to get a CI
  – Note here n is the *number of folds*, NOT the number of test examples

- For small n, use a t-test
  – Key difference: Sample variance is adjusted to produce a distribution with more mass in the tails
  – As n increases, approximation with Gaussian improves

t-distribution parameters

- $E_A - E_B$ has a t-distribution with parameters δ, s and "degrees of freedom" n-1
- Mean δ is the average of $E_A - E_B$ across n folds
- Standard Deviation s is given by:

$$s = \sqrt{\frac{\sum_{i=1}^{n} \left(\delta_i - \delta\right)^2}{n(n-1)}}$$

- Degrees of freedom n is related to the number of experiments we did (in 5-fold cv $n=5$)

Using the *t*-test

$$\text{Let } \delta_i = e_{S,A(Tr_i)} - e_{S,B(Tr_i)}$$

$$\text{Let } \delta = \frac{1}{n}\sum_{i=1}^{n}\delta_i; \text{ Let } s = \sqrt{\frac{\sum_{i=1}^{n}\left(\delta_i - \delta\right)^2}{n(n-1)}}$$

- Then use the *t*-distribution table to check if zero is contained in the 95% CI centered around δ

$$0 \in \delta \pm t_{C,n-1} s \,?$$

From table

Example

Fold	Error rate of Algorithm A	Error rate of Algorithm B	$E_A - E_B$
1	5%	3%	2%
2	1%	3%	-2%
3	8%	4%	4%
4	5%	1%	4%
5	1%	4%	-3%
Average	**4%**	**3%**	**1%**

Example

- For our table, $\delta=0.01$ and $s=0.015$ and $t_{0.95,4}=2.776$ ($t_{0.95,9}=2.262$)

- The 95% CI is $[-0.031, 0.051]$

- Clearly zero lies in the 95% CI, so the null hypothesis cannot be rejected
  - So maybe A and B are not different after all

One-way ANOVA

- If we need to compare more than two algorithms, can use this
- Null hypothesis: All the algorithms have equal errors
- Compares "between-means" variances to average variances within each sample with F-test
- If "between" variances are much more than "within" variances then means are unlikely to be the same

Mann-Whitney-Wilcoxon signed-rank test

- What if the classifier produces confidence estimates?
- If we can rank the predictions, we can calculate a statistic called "U" based on the ranks

$$U_1 = \sum_i R_{1,i} - \frac{n_1(n_1+1)}{2}$$

- For large enough samples, U can be approximated with a normal distribution as well
- We can show that the area under ROC is a "normalized" version of U

Bootstrap

- All previous methods relied on knowing the sampling distribution of the statistic we are interested in

- The bootstrap is a procedure where we get the properties of the statistic using *empirical resampling* from the observations

Example

- Suppose we have a set of iid examples and we want to get a CI for F1
- Repeatedly draw an equal sized sample (with replacement) from our test examples and measure F1
  – A “bootstrap replicate”
- This creates an empirical sampling distribution
- Then for the original data, measure F1 and ask how unusual that is in the empirical distribution

Pros and Cons

- Very easy to do, makes few assumptions, can estimate very complex things

- Assumes sample is representative
  - If not, can produce biased estimates