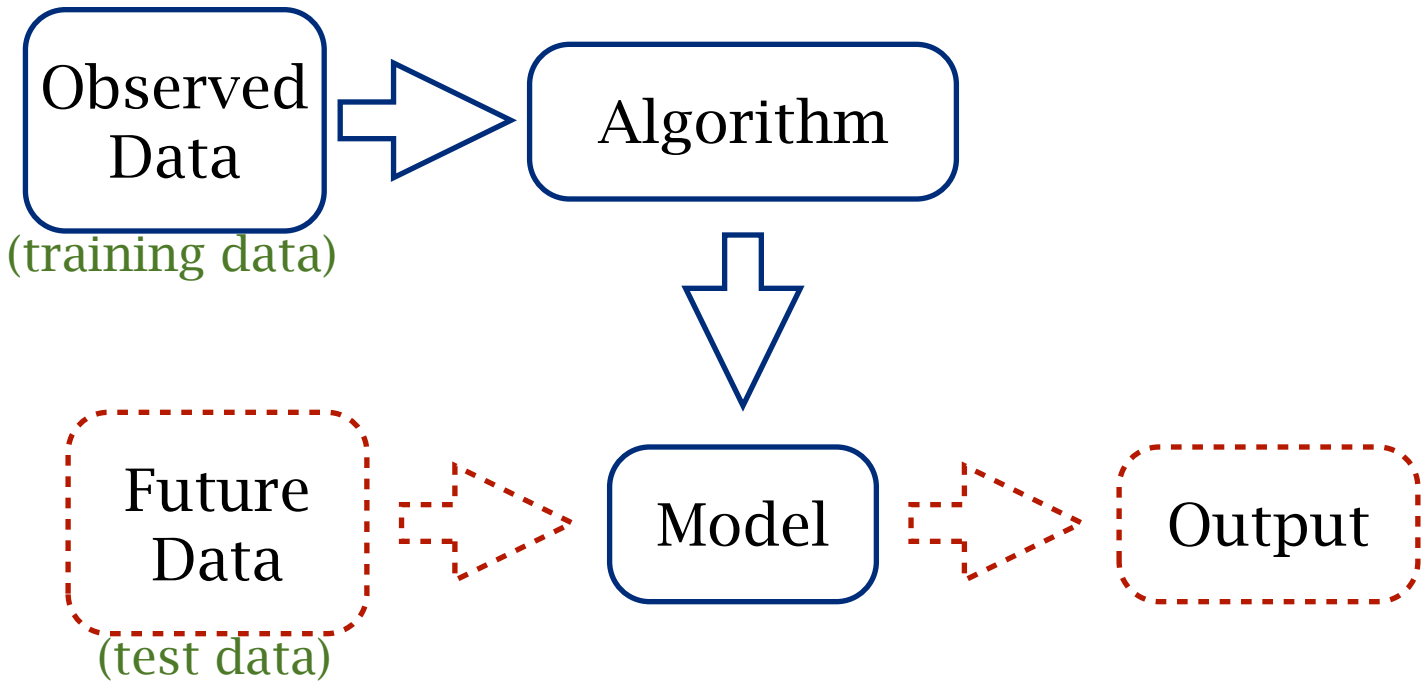
Mini Lecture: Experiment Design and Analysis

http://cs.nju.edu.cn/yuy/course_dm12.ashx

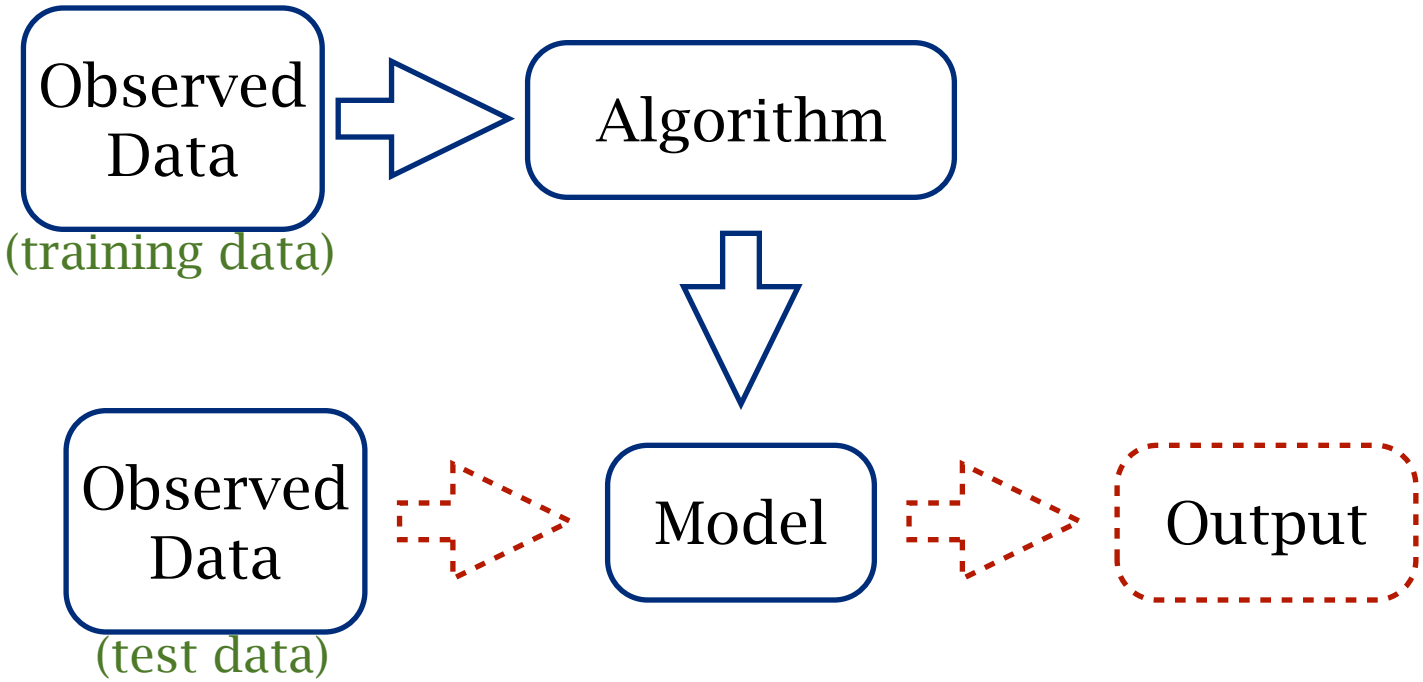
A common mining system structure



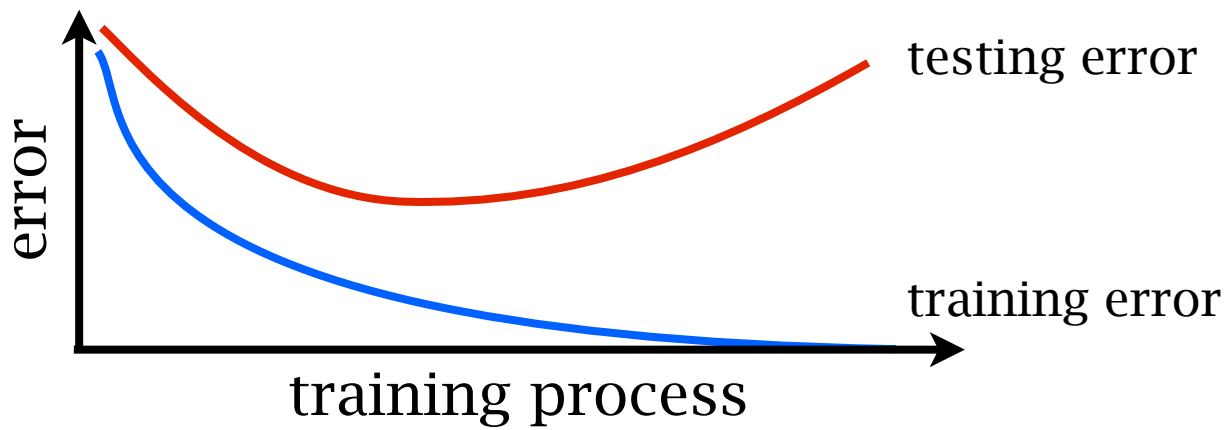
In experiment stage, we don't have the future data

How do we evaluate our algorithms and models?

A wrong way!

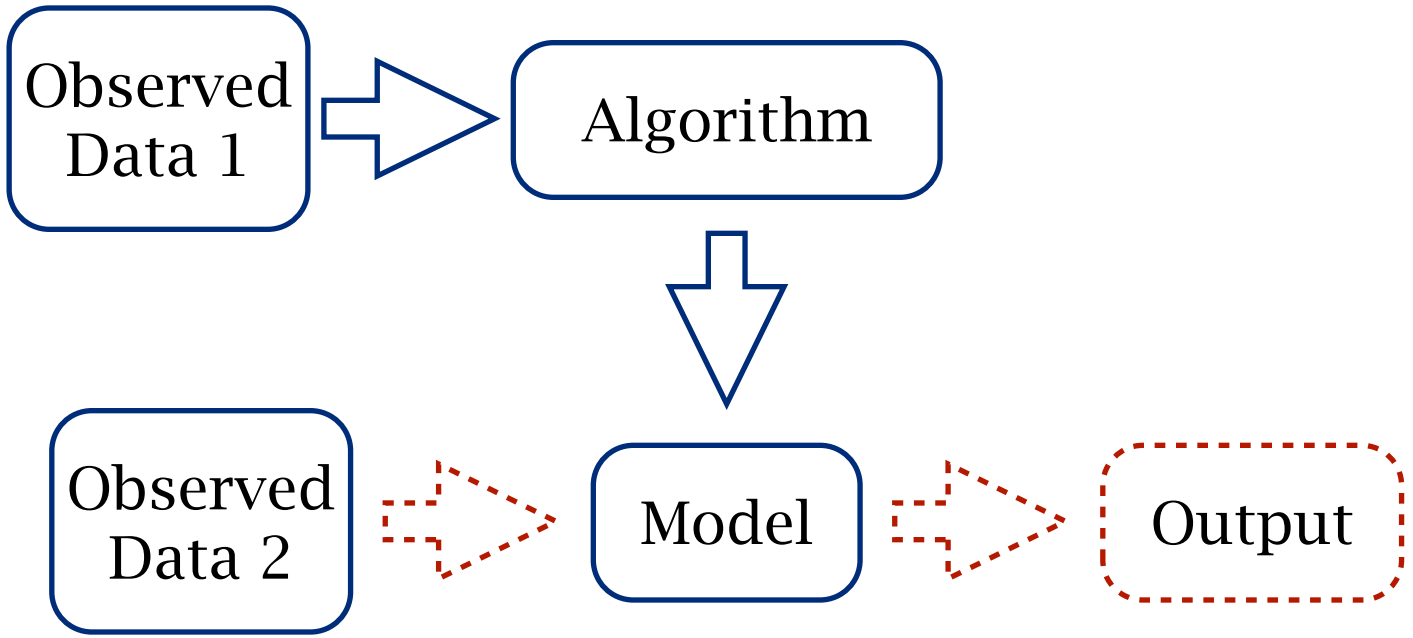


Never use the training data to evaluate your algorithm



A right way

split the training data into non-overlap parts



In other words, you should simulate the real situation

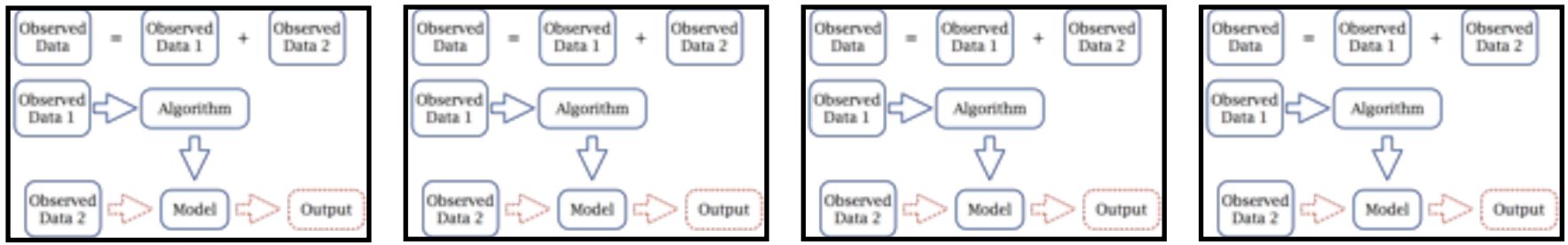
A right way

split the training data into non-overlap parts

Observed Data = Observed Data 1 + Observed Data 2

Randomness!

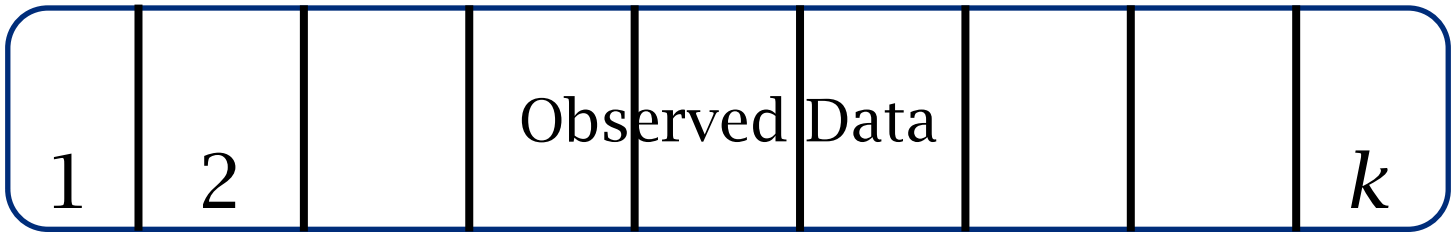
Use many different splits of the training data



and report the average performance

A right way

k-fold cross-validation



1st fold: leave the 1st block as the test data
2nd fold: leave the 2nd block as the test data
...
k-st fold: leave the k-st block as the test data

n-times *k*-fold cross-validation

hold-one-out

=*n*-fold cross-validation
(n is the number of training instances)

m-times hold-k%-out

sampling k% data as the test data

Performance comparison

"my algorithm has error 11%. It is perfect!"

Performance comparison

"my algorithm has error 11%. It is perfect!"

Comparison with baselines is necessary in order to show your superior.

Performance comparison

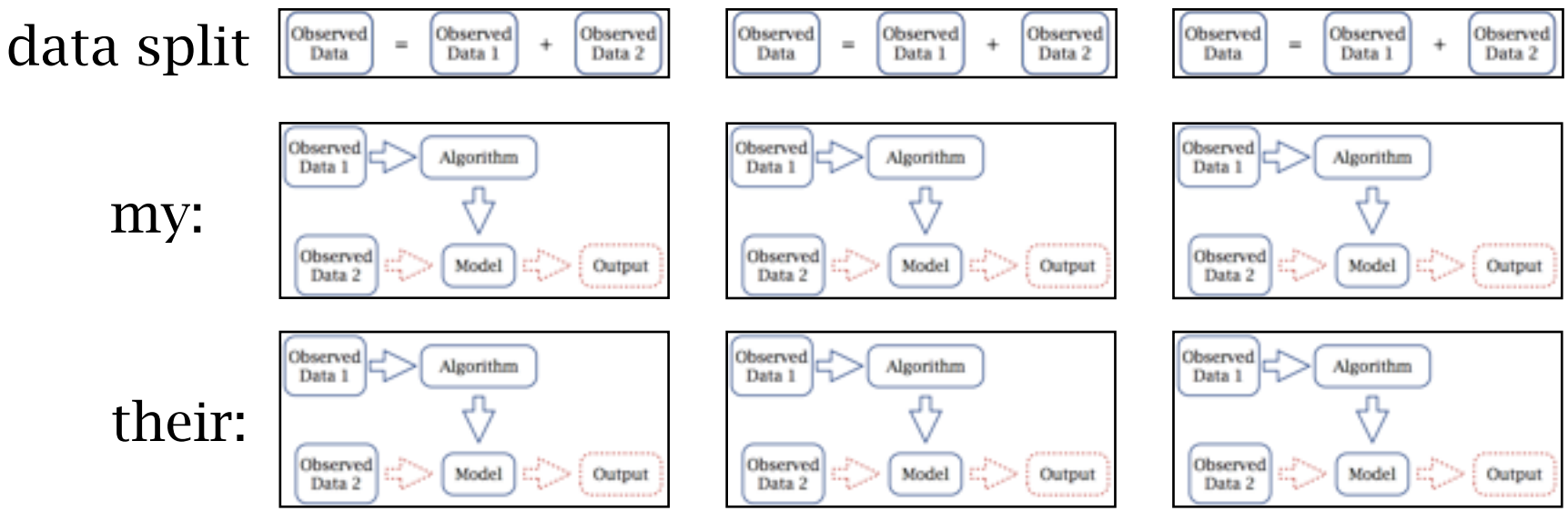
"their stupid algorithm has error 10%, my cleaver algorithm has error 9%. Mine is better!"

Performance comparison

"their stupid algorithm has error 10%, my cleaver algorithm has error 9%. Mine is better!"

statistical hypothesis test:

how large is the probability my algorithm is better?



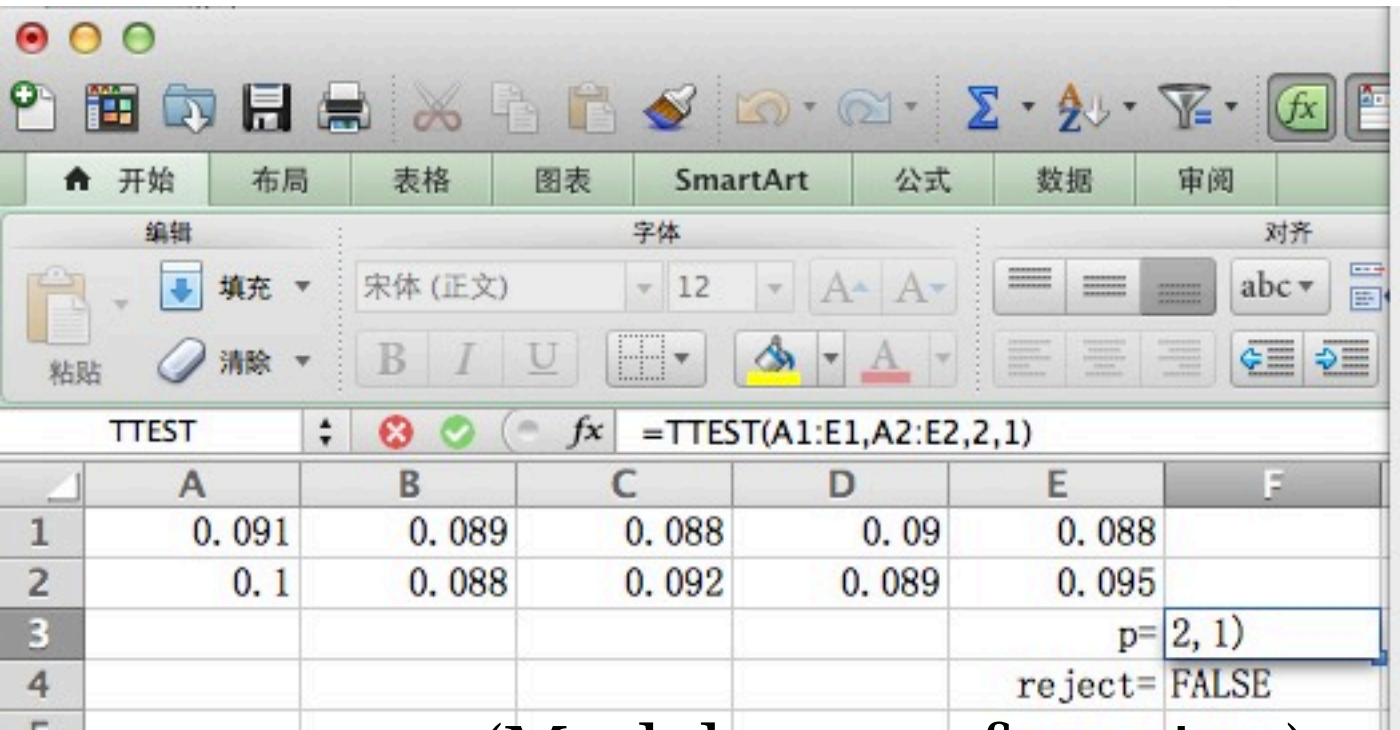
Performance comparison

pair-wise *t*-test: Gaussian distribution

my:	0.091	0.089	0.088	0.090	0.088
their:	0.100	0.088	0.092	0.089	0.095

probability threshold: 0.05

(Excel: ttest function)



(Matlab: ttest function)