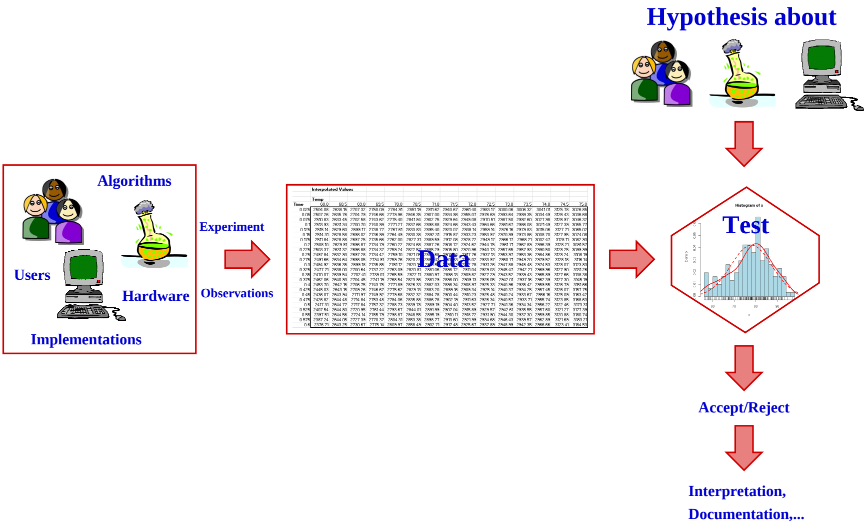
Experiments and Evaluation

Part 3: Performing and Evaluating Experiments

Manfred Jaeger

Aalborg University

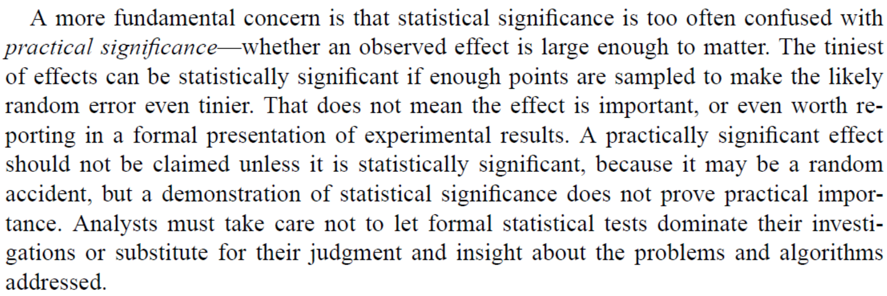
Review



Interpolated Values

Time	Temp 68.0	68.5	69.0	69.5	70.0	70.5	71.0	71.5	72.0	72.5	73.0	73.5	74.0	74.5	75.0
0.025	2504.08	2638.15	2707.32	2750.09	2784.91	2851.19	2911.62	2940.67	2961.40	2983.17	3000.06	3006.32	3041.01	3125.78	3026.85
0.05	2507.26	2635.76	2704.79	2746.66	2779.96	2846.35	2907.00	2934.98	2955.07	2976.69	2993.64	2999.35	3034.49	3126.43	3036.68
0.075	2510.83	2633.45	2702.58	2743.62	2775.40	2841.84	2902.75	2929.64	2949.08	2970.51	2987.50	2992.60	3027.98	3126.97	3046.32
0.1	2513.93	2631.34	2700.70	2740.99	2771.27	2837.66	2898.88	2924.66	2943.43	2964.66	2981.67	2986.08	3021.49	3127.39	3055.77
0.125	2515.14	2629.60	2699.17	2738.77	2767.61	2833.83	2895.40	2920.07	2938.14	2959.14	2976.16	2979.83	3015.06	3127.71	3065.02
0.15	2514.31	2628.58	2698.02	2736.99	2764.49	2830.38	2892.31	2915.87	2933.23	2953.97	2970.99	2973.86	3008.70	3127.95	3074.08
0.175	2511.84	2628.88	2697.25	2735.66	2762.00	2827.31	2889.59	2912.08	2928.72	2949.17	2966.17	2968.21	3002.47	3128.11	3082.93
0.2	2508.10	2629.91	2696.87	2734.79	2760.22	2824.68	2887.26	2908.72	2924.62	2944.75	2961.71	2962.89	2996.39	3128.21	3091.57
0.225	2503.37	2631.32	2696.88	2734.37	2759.24	2822.57	2885.29	2905.80	2920.96	2940.73	2957.65	2957.93	2990.50	3128.25	3099.99
0.25	2497.84	2632.93	2697.28	2734.42	2759.10	2821.09	[illegible]	[illegible]	2917.76	2937.13	2953.97	2953.36	2984.86	3128.24	3108.19
0.275	2491.66	2634.64	2698.05	2734.91	2759.76	2820.23	[illegible]	[illegible]	[illegible]	2933.97	2950.71	2949.20	2979.52	3128.18	3116.14
0.3	2484.92	2636.35	2699.18	2735.85	2761.12	[illegible]	[illegible]	[illegible]	[illegible]	2931.26	2947.88	2945.48	2974.53	3128.07	3123.83
0.325	2477.71	2638.00	2700.64	2737.22	2763.09	2820.81	2881.06	2898.72	2911.04	2929.03	2945.47	2942.21	2969.96	3127.90	3131.26
0.35	2470.07	2639.54	2702.41	2739.01	2765.59	2822.11	2880.97	2898.13	2909.82	2927.29	2943.52	2939.43	2965.89	3127.66	3138.38
0.375	2462.06	2640.93	2704.45	2741.19	2768.54	2823.98	2881.29	2898.00	2909.13	2926.05	2942.01	2937.16	2962.39	3127.30	3145.19
0.4	2453.70	2642.15	2706.75	2743.75	2771.89	2826.33	2882.03	2898.34	2908.97	2925.33	2940.96	2935.42	2959.55	3126.79	3151.66
0.425	2445.03	2643.15	2709.26	2746.67	2775.62	2829.13	2883.20	2899.16	2909.34	2925.14	2940.37	2934.25	2957.45	3126.07	3157.75
0.45	2436.07	2643.94	2711.97	2749.92	2779.68	2832.32	2884.78	2900.44	2910.23	2925.48	2940.24	2933.67	2956.16	3125.09	3163.42
0.475	2426.82	2644.48	2714.84	2753.48	2784.06	2835.88	2886.78	2902.19	2911.63	2926.34	2940.57	2933.71	2955.74	3123.85	3168.63
0.5	2417.31	2644.77	2717.84	2757.32	2788.73	2839.78	2889.19	2904.40	2913.52	2927.71	2941.36	2934.34	2956.22	3122.46	3173.31
0.525	2407.54	2644.80	2720.95	2761.44	2793.67	2844.01	2891.99	2907.04	2915.89	2929.57	2942.61	2935.55	2957.60	3121.27	3177.39
0.55	2397.51	2644.56	2724.14	2765.79	2798.87	2848.55	2895.19	2910.11	2918.72	2931.90	2944.30	2937.30	2959.85	3120.88	3180.74
0.575	2387.24	2644.05	2727.39	2770.37	2804.31	2853.38	2898.77	2913.60	2921.99	2934.68	2946.43	2939.57	2962.89	3121.69	3183.21
0.6	2376.71	2643.25	2730.67	2775.14	2809.97	2858.49	2902.71	2917.48	2925.67	2937.89	2948.99	2942.35	2966.66	3123.41	3184.53

Warnings and Disclaimers



A more fundamental concern is that statistical significance is too often confused with *practical significance*—whether an observed effect is large enough to matter. The tiniest of effects can be statistically significant if enough points are sampled to make the likely random error even tinier. That does not mean the effect is important, or even worth reporting in a formal presentation of experimental results. A practically significant effect should not be claimed unless it is statistically significant, because it may be a random accident, but a demonstration of statistical significance does not prove practical importance. Analysts must take care not to let formal statistical tests dominate their investigations or substitute for their judgment and insight about the problems and algorithms addressed.

[R. L. Rardin and R. Uzsoy: Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial]

- Go to a bank and get 1000 5Kr. coins

- Go to a bank and get 1000 5Kr. coins

- Toss each coin 100 times, and record the number of times it comes up *heads* or *tails*

- Go to a bank and get 1000 5Kr. coins

- Toss each coin 100 times, and record the number of times it comes up *heads* or *tails*
- Use the Binomial test of significance at level 5% to test for each coin the hypothesis that

$$P(heads) = P(tails) = 1/2$$

- Go to a bank and get 1000 5Kr. coins

- Toss each coin 100 times, and record the number of times it comes up *heads* or *tails*
- Use the Binomial test of significance at level 5% to test for each coin the hypothesis that

$$P(heads) = P(tails) = 1/2$$

- If all coins are fair, then the hypothesis will be rejected in approximately 5% = 50 of the experiments

- Go to a bank and get 1000 5Kr. coins

- Toss each coin 100 times, and record the number of times it comes up *heads* or *tails*
- Use the Binomial test of significance at level 5% to test for each coin the hypothesis that

$$P(heads) = P(tails) = 1/2$$

- If all coins are fair, then the hypothesis will be rejected in approximately 5% = 50 of the experiments
- Bring these 50 coins back to the bank ...

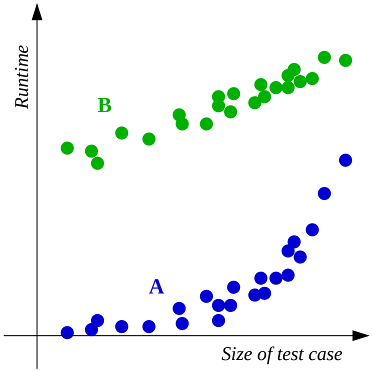
- Go to a bank and get 1000 5Kr. coins

- Toss each coin 100 times, and record the number of times it comes up *heads* or *tails*
- Use the Binomial test of significance at level 5% to test for each coin the hypothesis that

$$P(heads) = P(tails) = 1/2$$

- If all coins are fair, then the hypothesis will be rejected in approximately 5% = 50 of the experiments
- Bring these 50 coins back to the bank ...

Multiple Testing: when a large number of tests are performed, then the outcomes will contain some *false rejections* of the null Hypothesis. Any given (true) hypothesis can be (eventually) rejected by testing it often enough (if tests are independent).

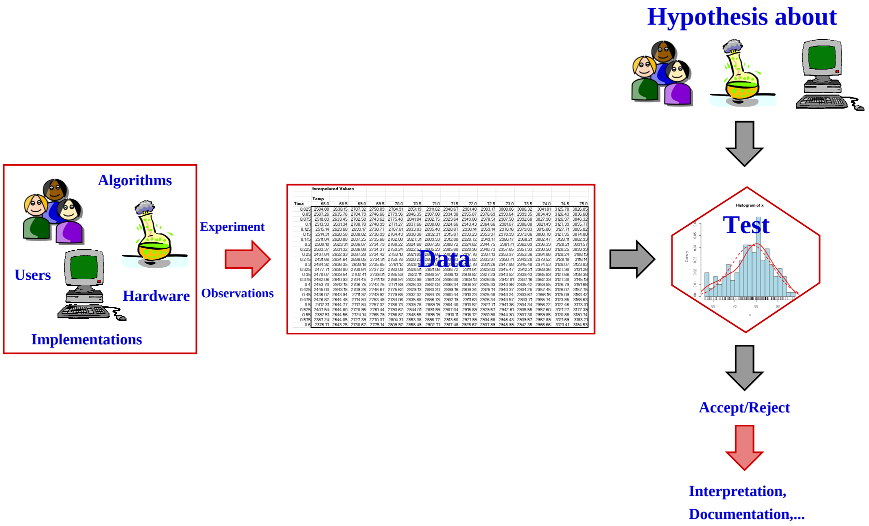


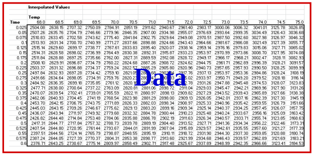
Complexity analysis: B (linear) is better than **A** (quadratic?).

Testing: Average runtime of **A** is less than of **B** (statistically significant).

- Which statement is more relevant depends on whether test cases are representative for size of future cases

“Experimental Design”




Interpolated Values

Time	Temp 68.0	68.5	69.0	69.5	70.0	70.5	71.0	71.5	72.0	72.5	73.0	73.5	74.0	74.5	75.0
0.025	2504.08	2638.15	2707.32	2750.09	2784.91	2851.19	2911.62	2940.67	2961.40	2983.17	3000.06	3006.32	3041.01	3125.78	3026.85
0.05	2507.26	2635.76	2704.79	2746.66	2779.96	2846.35	2907.00	2934.98	2955.07	2976.69	2993.64	2999.35	3034.49	3126.43	3036.68
0.075	2510.83	2633.45	2702.58	2743.62	2775.40	2841.84	2902.75	2929.64	2949.08	2970.51	2987.50	2992.60	3027.98	3126.97	3046.32
0.1	2513.93	2631.34	2700.70	2740.99	2771.27	2837.66	2898.88	2924.66	2943.43	2964.66	2981.67	2986.08	3021.49	3127.39	3055.77
0.125	2515.14	2629.60	2699.17	2738.77	2767.61	2833.83	2895.40	2920.07	2938.14	2959.14	2976.16	2979.83	3015.06	3127.71	3065.02
0.15	2514.31	2628.58	2698.02	2736.99	2764.49	2830.38	2892.31	2915.87	2933.23	2953.97	2970.99	2973.86	3008.70	3127.95	3074.08
0.175	2511.84	2628.88	2697.25	2735.66	2762.00	2827.31	2889.59	2912.08	2928.72	2949.17	2966.17	2968.21	3002.47	3128.11	3082.93
0.2	2508.10	2629.91	2696.87	2734.79	2760.22	2824.68	2887.26	2908.72	2924.62	2944.75	2961.71	2962.89	2996.39	3128.21	3091.57
0.225	2503.37	2631.32	2696.88	2734.37	2759.24	2822.57	2885.29	2905.80	2920.96	2940.73	2957.65	2957.93	2990.50	3128.25	3099.99
0.25	2497.84	2632.93	2697.28	2734.42	2759.10	2821.05	[illegible]	2903.84	2917.76	2937.13	2953.97	2953.36	2984.86	3128.24	3108.19
0.275	2491.66	2634.64	2698.05	2734.91	2759.76	2820.23	[illegible]	[illegible]	[illegible]	2933.97	2950.71	2949.20	2979.52	3128.18	3116.14
0.3	2484.92	2636.35	2699.18	2735.85	2761.12	2820.16	[illegible]	[illegible]	[illegible]	2931.26	2947.88	2945.48	2974.53	3128.07	3123.83
0.325	2477.71	2638.00	2700.64	2737.22	2763.09	2820.81	2881.06	2898.72	2911.04	2929.03	2945.47	2942.21	2969.96	3127.90	3131.26
0.35	2470.07	2639.54	2702.41	2739.01	2765.59	2822.11	2880.97	2898.13	2909.82	2927.29	2943.52	2939.43	2965.89	3127.66	3138.38
0.375	2462.06	2640.93	2704.45	2741.19	2768.54	2823.98	2881.29	2898.00	2909.13	2926.05	2942.01	2937.16	2962.39	3127.30	3145.19
0.4	2453.70	2642.15	2706.75	2743.75	2771.89	2826.33	2882.03	2898.34	2908.97	2925.33	2940.96	2935.42	2959.55	3126.79	3151.66
0.425	2445.03	2643.15	2709.26	2746.67	2775.62	2829.13	2883.20	2899.16	2909.34	2925.14	2940.37	2934.25	2957.45	3126.07	3157.75
0.45	2436.07	2643.94	2711.97	2749.92	2779.68	2832.32	2884.78	2900.44	2910.23	2925.48	2940.24	2933.67	2956.16	3125.09	3163.42
0.475	2426.82	2644.48	2714.84	2753.48	2784.06	2835.88	2886.78	2902.19	2911.63	2926.34	2940.57	2933.71	2955.74	3123.85	3168.63
0.5	2417.31	2644.77	2717.84	2757.32	2788.73	2839.78	2889.19	2904.40	2913.52	2927.71	2941.36	2934.34	2956.22	3122.46	3173.31
0.525	2407.54	2644.80	2720.95	2761.44	2793.67	2844.01	2891.99	2907.04	2915.89	2929.57	2942.61	2935.55	2957.60	3121.27	3177.39
0.55	2397.51	2644.56	2724.14	2765.79	2798.87	2848.55	2895.19	2910.11	2918.72	2931.90	2944.30	2937.30	2959.85	3120.88	3180.74
0.575	2387.24	2644.05	2727.39	2770.37	2804.31	2853.38	2898.77	2913.60	2921.99	2934.68	2946.43	2939.57	2962.89	3121.69	3183.21
0.6	2376.71	2643.25	2730.67	2775.14	2809.97	2858.49	2902.71	2917.48	2925.67	2937.89	2948.99	2942.35	2966.66	3123.41	3184.53

Admission statistics from Berkeley University, 1973:

Applicant Gender	Admitted
Male	44%
Female	35%

Is there gender discrimination against female applicants?

Admission statistics from Berkeley University, 1973:

Applicant Gender	Admitted
Male	44%
Female	35%

Is there gender discrimination against female applicants?

By departments:

Department	Male admitted	Female admitted
A	62%	82%
B	63%	68%
C	37%	34%
D	33%	35%
E	28%	24%
F	6%	7%

- There seems to be gender discrimination at the university as a whole, but not at any single department ...

Simpson's paradox - resolved

Adding distribution of applicants to the picture (hypothetical numbers):

			% of applicants applying to department	
Department	Male admitted	Female admitted	male	female
A	62%	82%	23%	10%
B	63%	68%	20%	10%
C	37%	34%	22%	15%
D	33%	35%	20%	25%
E	28%	24%	10%	20%
F	6%	7%	5%	20%

☞ Male applicants have a higher admission rate overall, but a lower one in (almost) all departments, because female students apply for the programs that are harder to get into.

When collecting data for evaluating the differences between setups A and B, make sure that test cases do not differ systematically with regard to factors other than defined by A and B (**confounding factors**).

- Run algorithms A and B on the same hardware
- Randomly assign test cases or test users to setups A and B
- Paired samples (within-subject design): apply both setups to each test case/user
  - Easy when test cases are computational problems
  - For test users: randomize the order in which setups A,B are shown to the users
- **Stratified Sampling:** Include in the sample used for statistical analysis equal numbers of test cases for every configuration of the confounding factor(s).

Analyzing data that is not collected in **controlled experiments**: try to identify all factors that could potentially have influenced the observations.

Test Cases: Data

For some types of computational problems there exist repositories of benchmark problems:

Satisfiability problems – SATLIB:

`http://www.satlib.org/`

Machine Learning data sets – UCI ML Repository:

`http://archive.ics.uci.edu/ml/`

Optimization:

`http://plato.la.asu.edu/bench.html`

Plus:

- Availability
- Reproducibility
- Historical results available

Minus:

- Older and less interesting datasets
- Development can be driven by optimization on benchmarks
- “Best published” results may be difficult to improve upon (and may be too good to be true)
- Invites multiple testing

Create a generator for (random) data cases.

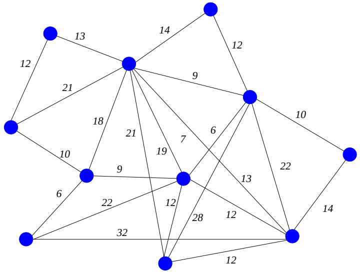
Plus:

- Easy to generate in large quantities
- Some data characteristics can be precisely tuned (size/complexity of problem instances)
- Can be shared with others (no proprietary or privacy concerns)

Minus:

- Difficult to get the structure of “real” data in random data (TSP example)
- Easy to manipulate to obtain desired outcomes

Input case for Traveling Salesman Problem:



Creating random instances for the Traveling Salesman Problem (cf. Rardin,Uzsoy):

- 1st approach: create distance matrix with random entries. Problems: distances will not satisfy triangle inequality, and most tours will have about equal length
- 2nd approach: randomly distribute points in the plane, and use the Euclidean distance between points. Problems: how to distribute points, so that resulting graph resembles typical TSP problem?

The data used as experimental test cases should be representative for the inputs of the intended application domain.

- For general algorithmic problems (e.g. TSP): test cases should cover wide spectrum of possible applications
- For concrete application scenarios (e.g. design of web shop homepage): test cases must be closely representative for use cases (best: live data collection in deployed system)

Temptation: use test cases that make your setup look good!

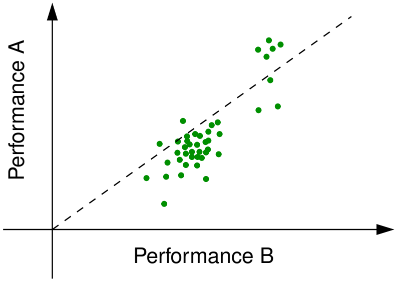
The data used as experimental test cases should be representative for the inputs of the intended application domain.

- For general algorithmic problems (e.g. TSP): test cases should cover wide spectrum of possible applications
- For concrete application scenarios (e.g. design of web shop homepage): test cases must be closely representative for use cases (best: live data collection in deployed system)

Temptation: use test cases that make your setup look good!

Example:

- A first experiments yields these results



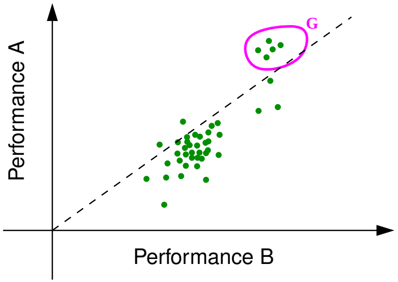
The data used as experimental test cases should be representative for the inputs of the intended application domain.

- For general algorithmic problems (e.g. TSP): test cases should cover wide spectrum of possible applications
- For concrete application scenarios (e.g. design of web shop homepage): test cases must be closely representative for use cases (best: live data collection in deployed system)

Temptation: use test cases that make your setup look good!

Example:

- A first experiments yields these results
- Temptation: conduct another experiment with test cases resembling those in subset G.



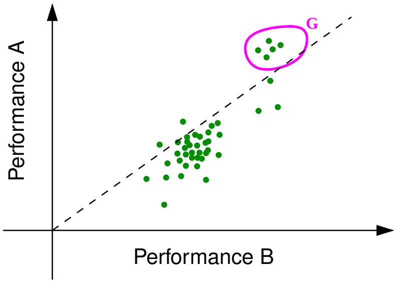
The data used as experimental test cases should be representative for the inputs of the intended application domain.

- For general algorithmic problems (e.g. TSP): test cases should cover wide spectrum of possible applications
- For concrete application scenarios (e.g. design of web shop homepage): test cases must be closely representative for use cases (best: live data collection in deployed system)

Temptation: use test cases that make your setup look good!

Example:

- A first experiments yields these results
- Temptation: conduct another experiment with test cases resembling those in subset G.
- This need not be unethical, if all results are reported (and/or special nature of cases in G is acknowledged)



Experimental Procedures

Experiments: Evaluation vs. Development

A setup can be controlled by parameters:

- Parameters controlling time/space/accuracy tradeoff of optimization algorithm
- Parameters controlling layout of web shop homepage
- . . .
- Parameters representing more high-level design choices

Experiments often interleave:

- Performance evaluation of setup
- Tuning of parameters to optimize performance

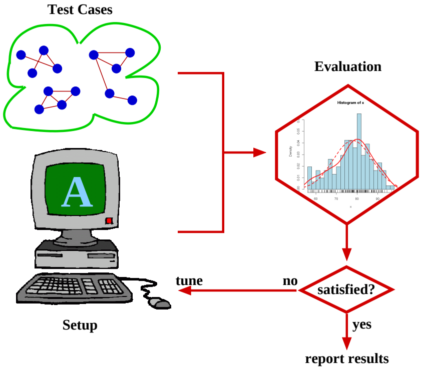
A setup may contain a large number of tunable parameters:

$$A \in \mathbb{R},\ B \in \mathbb{Z},\ \gamma \in [0,1],\ t \in \{0,1\},\ \ldots$$

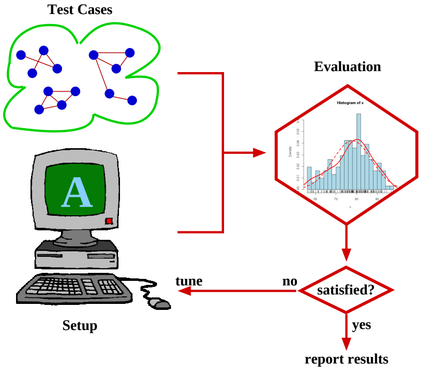
Usually impossible to try all combinations of parameter values! Simple procedure:

- Using some “best guess” default setting for other parameters, find the best value of the first parameter
- Using best value of first parameter, default settings for others, find best value of second parameter
- . . .

Not guaranteed to find best setting, but good heuristic.

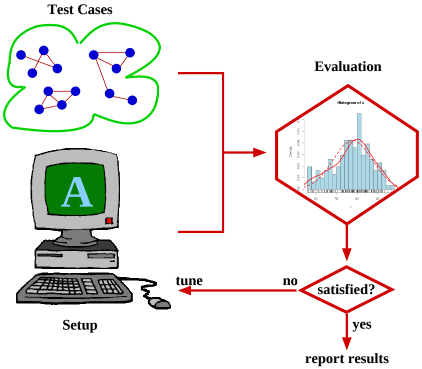


Any problems with this process?



Any problems with this process?

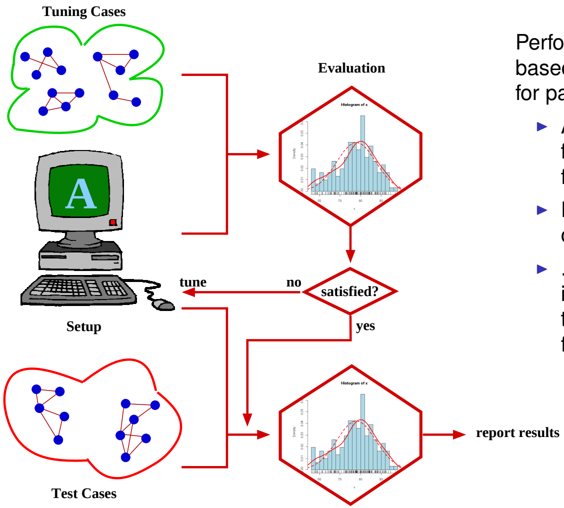
Problem 1: Overfitting. The setup is optimized to perform best on the test cases. Performance on future cases may be worse than expected.



Any problems with this process?

Problem 1: Overfitting. The setup is optimized to perform best on the test cases. Performance on future cases may be worse than expected.

Problem 2: Multiple Testing. If evaluation contains statistical tests, then spurious rejections of null hypothesis may occur (mitigated by the fact that tests for different parameter settings are not independent – not necessarily 5% false rejections!).



Perform final performance evaluation based on test cases that were not used for parameter tuning

- Avoids overfitting: test cases in final evaluation representative of future cases
- Final significance analysis based on a single test
- . . . but, in reality: almost impossible to completely eliminate the “tuning on the test cases” fallacy

Make sure:

- ‘Our” system **A** and competitor **B** tested on the same hardware
- No other processes running on machine during testing

Even then, one often finds:

- “Our” **A** finely tuned
- Competitor **B** run in default configuration
- Expertise lacking to get best performance out of **B**
- $\rightsquigarrow$ comparison biased in favor of our system

Documentation

One often reads (here: in Machine Learning papers):

. . . we conducted experiments on 10 datasets from the UCI repository . . .

Suspicion: experiments were conducted on more datasets, but these 10 gave the best results.

Ideally:

- All experiments and their results are reported, including negative or inconclusive ones

Second best option:

- Admit that the selected test cases are well suited for proposed setup A. Explain why.

Other researchers should be able to independently reproduce experimental results:

- In publication: precisely describe experimental procedure
  - Test cases
  - Preprocessing (data cleaning, outlier removal, . . .)
  - Tune/Test approach
  - Hardware platform
- Make all relevant information publicly available (supplementary material published on the web)
  - Code for your implementation/algorithm
  - Test data (if not already publicly available)
  - Questionnaires used in user studies
  - . . .