

Impact of censored sampling on the performance of restart strategies

Matteo Gagliolo^{1,2} Jürgen Schmidhuber^{1,3}

¹[IDSIA](#), Lugano, Switzerland

²University of Lugano, Faculty of Informatics, Switzerland

³TU, Munich, Germany

12th International Conference on Principles and
Practice of Constraint Programming

Roadmap

Restart strategies

Survival analysis

Experiments

Conclusion

Restart strategies - what?

A restart strategy consists in executing a sequence of runs of a randomized algorithm, in order to solve a same problem instance, stopping each run k after a time $T(k)$ if no solution is found, and restarting the algorithm with a different random seed.

Restart strategies - when?

Whether a restart strategy can be effective or not depends on the shape of the runtime distribution (*RTD*).

Restart strategies - when?

Whether a restart strategy can be effective or not depends on the shape of the runtime distribution (*RTD*).
Examples:

Restart strategies - when?

Whether a restart strategy can be effective or not depends on the shape of the runtime distribution (*RTD*).

Examples:

- ▶ **Heavy-tailed** distributions

$$F(t) \rightarrow_{t \rightarrow \infty} 1 - Ct^{-\alpha}$$

Restart strategies - when?

Whether a restart strategy can be effective or not depends on the shape of the runtime distribution (*RTD*).

Examples:

- ▶ **Heavy-tailed** distributions

$$F(t) \rightarrow_{t \rightarrow \infty} 1 - Ct^{-\alpha}$$

This means: most runs are short, some take a very long time...

E.g., backtracking SAT/CP solvers on structured underconstrained problems (Gomes et al. 2000).

Restart strategies - when?

Whether a restart strategy can be effective or not depends on the shape of the runtime distribution (*RTD*).

Examples:

- ▶ **Heavy-tailed** distributions

$$F(t) \rightarrow_{t \rightarrow \infty} 1 - Ct^{-\alpha}$$

This means: most runs are short, some take a very long time...

E.g., backtracking SAT/CP solvers on structured underconstrained problems (Gomes et al. 2000).

- ▶ **Exponential** distributions

$$F(t) = 1 - e^{-\lambda t}$$

Restart strategies - when?

Whether a restart strategy can be effective or not depends on the shape of the runtime distribution (*RTD*).

Examples:

- ▶ **Heavy-tailed** distributions

$$F(t) \rightarrow_{t \rightarrow \infty} 1 - Ct^{-\alpha}$$

This means: most runs are short, some take a very long time...

E.g., backtracking SAT/CP solvers on structured underconstrained problems (Gomes et al. 2000).

- ▶ **Exponential** distributions

$$F(t) = 1 - e^{-\lambda t}$$

E.g., local search (Hoos et al. 1999).

Restart strategies - how?

Luby et. al (1993) proved that the optimal restart strategy is *uniform*, i. e., one in which a constant $T(k) = T$ is used to bound each run.

Restart strategies - how?

Luby et. al (1993) proved that the optimal restart strategy is *uniform*, i. e., one in which a constant $T(k) = T$ is used to bound each run.

They show that, in this case, the expected value of the total run-time t_T can be evaluated as

$$E(t_T) = \frac{T - \int_0^T F(\tau) d\tau}{F(T)}$$

Restart strategies - how?

Luby et. al (1993) proved that the optimal restart strategy is *uniform*, i. e., one in which a constant $T(k) = T$ is used to bound each run.

They show that, in this case, the expected value of the total run-time t_T can be evaluated as

$$E(t_T) = \frac{T - \int_0^T F(\tau) d\tau}{F(T)}$$

If F is known I can evaluate an *optimal* strategy $T(k) = T^*$.

$$T^* = \arg \min_T E(t_T)$$

Restart strategies - how?

If F is completely unknown I can use Luby's **universal** strategy:

Restart strategies - how?

If F is completely unknown I can use Luby's **universal** strategy:

$$T(k) = \{1, 1, 2, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 4, 8, 1, \dots\}$$

whose performance t_U is, with high probability, within a logarithmic factor worse than the *expected* total run-time $E(t_{T^*})$ of the optimal strategy.

Restart strategies - how?

Couldn't we just **estimate** F ?

Restart strategies - how?

Couldn't we just **estimate** F ?

We could then use the estimated \hat{F} in place of F , to evaluate a **sub**-optimal strategy \hat{T} .

$$\hat{T} = \arg \min_T \frac{T - \int_0^T \hat{F}(\tau) d\tau}{\hat{F}(T)}$$

Restart strategies - how?

Couldn't we just **estimate** F ?

We could then use the estimated \hat{F} in place of F , to evaluate a **sub**-optimal strategy \hat{T} .

$$\hat{T} = \arg \min_T \frac{T - \int_0^T \hat{F}(\tau) d\tau}{\hat{F}(T)}$$

Issues:

Restart strategies - how?

Couldn't we just **estimate** F ?

We could then use the estimated \hat{F} in place of F , to evaluate a **sub**-optimal strategy \hat{T} .

$$\hat{T} = \arg \min_T \frac{T - \int_0^T \hat{F}(\tau) d\tau}{\hat{F}(T)}$$

Issues:

- ▶ F might **differ** on each problem instance.

Restart strategies - how?

Couldn't we just **estimate** F ?

We could then use the estimated \hat{F} in place of F , to evaluate a **sub**-optimal strategy \hat{T} .

$$\hat{T} = \arg \min_T \frac{T - \int_0^T \hat{F}(\tau) d\tau}{\hat{F}(T)}$$

Issues:

- ▶ F might **differ** on each problem instance.
- ▶ Sampling F might take lot of time...

Restart strategies - how?

Couldn't we just **estimate** F ?

We could then use the estimated \hat{F} in place of F , to evaluate a **sub-optimal** strategy \hat{T} .

$$\hat{T} = \arg \min_T \frac{T - \int_0^T \hat{F}(\tau) d\tau}{\hat{F}(T)}$$

Issues:

- ▶ F might **differ** on each problem instance.
- ▶ Sampling F might take lot of time... **especially if the algorithm RTD has heavy tails!**

Learning restart strategies

Intuitively, there should be:

- ▶ a *trade-off* between the **precision** of \hat{F} and the **time** spent for collecting training data.

Learning restart strategies

Intuitively, there should be:

- ▶ a *trade-off* between the **precision** of \hat{F} and the **time** spent for collecting training data.
- ▶ a correlation between the **precision** of \hat{F} and the **performance** of \hat{T}

Learning restart strategies

Intuitively, there should be:

- ▶ a *trade-off* between the **precision** of \hat{F} and the **time** spent for collecting training data.
- ▶ a correlation between the **precision** of \hat{F} and the **performance** of \hat{T}

So there should be a trade-off between training time and performance of \hat{T} .

Survival analysis

What are the events whose distribution is studied most?

Survival analysis

What are the events whose distribution is studied most?

- ▶ **DEATH** (*Survival analysis* in medicine, sociology, biology,...)

Survival analysis

What are the events whose distribution is studied most?

- ▶ **DEATH** (*Survival analysis* in medicine, sociology, biology,...)
- ▶ **FAILURE** (*Lifetime distribution estimation* in engineering)

Survival analysis

Examples:

Survival analysis

Examples:

- ▶ “Out of 10 individuals under treatment, 2 died after 23 and 150 days respectively, 1 dropped the study at day 122, the rest is still alive (day 180)”

Survival analysis

Examples:

- ▶ “Out of 10 individuals under treatment, 2 died after 23 and 150 days respectively, 1 dropped the study at day 122, the rest is still alive (day 180)”
- ▶ “Out of 100 lightbulbs under test, 12 went off at times (2h, 14h, ...), the rest is still working after 500h...”

Survival analysis

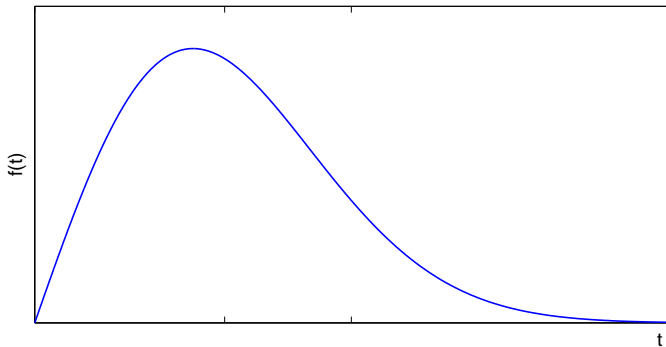
Examples:

- ▶ “Out of 10 individuals under treatment, 2 died after 23 and 150 days respectively, 1 dropped the study at day 122, the rest is still alive (day 180)”
- ▶ “Out of 100 lightbulbs under test, 12 went off at times (2h, 14h, ...), the rest is still working after 500h...”

A common situation: *incomplete* data.

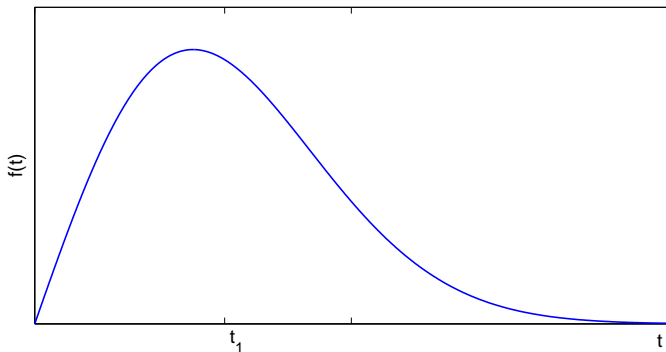
Survival analysis - Parametric models

Graphical example:



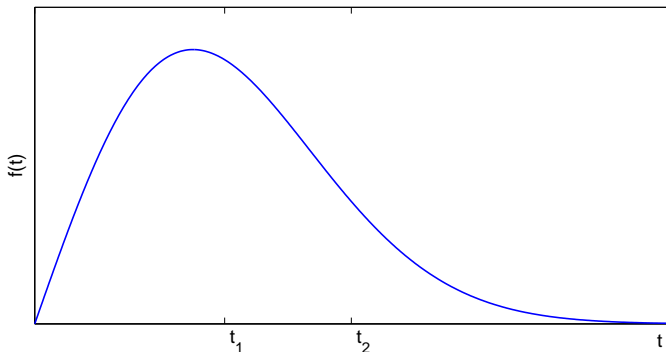
Survival analysis - Parametric models

Graphical example: one component **fails** at time t_1 ...



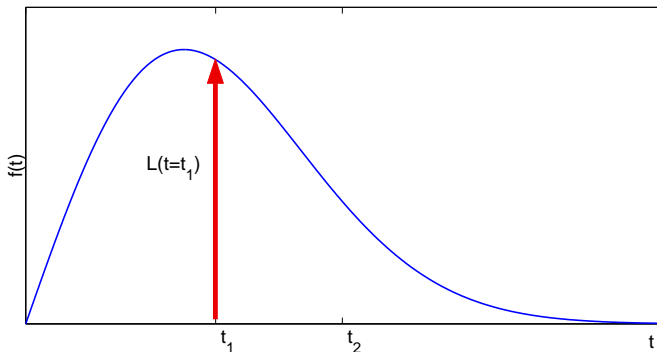
Survival analysis - Parametric models

Graphical example: one component **fails** at time t_1 ... one is reported to last until time t_2



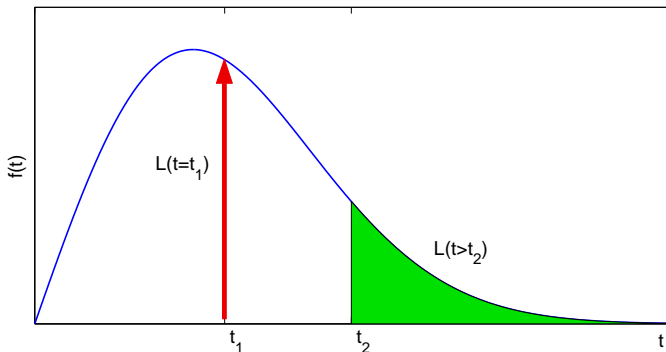
Survival analysis - Parametric models

Graphical example: one component **fails** at time t_1 ... one is reported to last until time t_2



Survival analysis - Parametric models

Graphical example: one component **fails** at time t_1 ... one is reported to last until time t_2



Censored sampling

- ▶ Using survival analysis techniques to deal with incomplete data is **better** than discarding it, but **worse** than having the real data..

Censored sampling

- ▶ Using survival analysis techniques to deal with incomplete data is **better** than discarding it, but **worse** than having the real data..
- ▶ In our case, collecting runtime samples with **more** censoring means spending **less** time training

Censored sampling

- ▶ Using survival analysis techniques to deal with incomplete data is **better** than discarding it, but **worse** than having the real data..
- ▶ In our case, collecting runtime samples with **more** censoring means spending **less** time training
- ▶ Censored sampling introduces a *trade-off* between the **precision** of an estimate \hat{F} , and the **time** spent gathering the training samples.

Censored sampling

- ▶ Using survival analysis techniques to deal with incomplete data is **better** than discarding it, but **worse** than having the real data..
- ▶ In our case, collecting runtime samples with **more** censoring means spending **less** time training
- ▶ Censored sampling introduces a *trade-off* between the **precision** of an estimate \hat{F} , and the **time** spent gathering the training samples.
- ▶ Intuitively we would expect the performance of a restart strategy to depend on how well \hat{F} models the unknown F .

Censored sampling

- ▶ Using survival analysis techniques to deal with incomplete data is **better** than discarding it, but **worse** than having the real data..
- ▶ In our case, collecting runtime samples with **more** censoring means spending **less** time training
- ▶ Censored sampling introduces a *trade-off* between the **precision** of an estimate \hat{F} , and the **time** spent gathering the training samples.
- ▶ Intuitively we would expect the performance of a restart strategy to depend on how well \hat{F} models the unknown F .

So here comes the question:

“How big is the impact of censored sampling on the performance of a restart strategy?”

Experiments - Benchmark

- ▶ Algorithm: Satz-Rand (Gomes et al. 2000)

Experiments - Benchmark

- ▶ Algorithm: Satz-Rand (Gomes et al. 2000)
- ▶ SATLIB Benchmark: Morphed GCP (Gent et. al. 1999): 9 groups of 100 instances (group i has structure parameter $p = 2^{-i}$)

Experiments - Benchmark

- ▶ Algorithm: Satz-Rand (Gomes et al. 2000)
- ▶ SATLIB Benchmark: Morphed GCP (Gent et. al. 1999): 9 groups of 100 instances (group i has structure parameter $p = 2^{-i}$)
- ▶ Various parametric (mixture) models, and the non-parametric Kaplan-Meier estimator (1958).

Experiments - A simple learning algorithm

For each group of instances:

Experiments - A simple learning algorithm

For each group of instances:

- ▶ randomly pick 50 instances

Experiments - A simple learning algorithm

For each group of instances:

- ▶ randomly pick 50 instances
- ▶ start 20 parallel runs of the solver for each instance

Experiments - A simple learning algorithm

For each group of instances:

- ▶ randomly pick 50 instances
- ▶ start 20 parallel runs of the solver for each instance
- ▶ stop **all** runs when a fraction $c \in [0, 1)$ ends

Experiments - A simple learning algorithm

For each group of instances:

- ▶ randomly pick 50 instances
- ▶ start 20 parallel runs of the solver for each instance
- ▶ stop **all** runs when a fraction $c \in [0, 1)$ ends
- ▶ train a model \hat{F} of the RTD **for the group**

Experiments - A simple learning algorithm

For each group of instances:

- ▶ randomly pick 50 instances
- ▶ start 20 parallel runs of the solver for each instance
- ▶ stop **all** runs when a fraction $c \in [0, 1)$ ends
- ▶ train a model \hat{F} of the RTD **for the group**
- ▶ obtain the corresponding \hat{T} minimizing $E(t_T)$

Experiments - A simple learning algorithm

For each group of instances:

- ▶ randomly pick 50 instances
- ▶ start 20 parallel runs of the solver for each instance
- ▶ stop **all** runs when a fraction $c \in [0, 1)$ ends
- ▶ train a model \hat{F} of the RTD **for the group**
- ▶ obtain the corresponding \hat{T} minimizing $E(t_T)$
- ▶ test \hat{T} on the remaining 50 instances

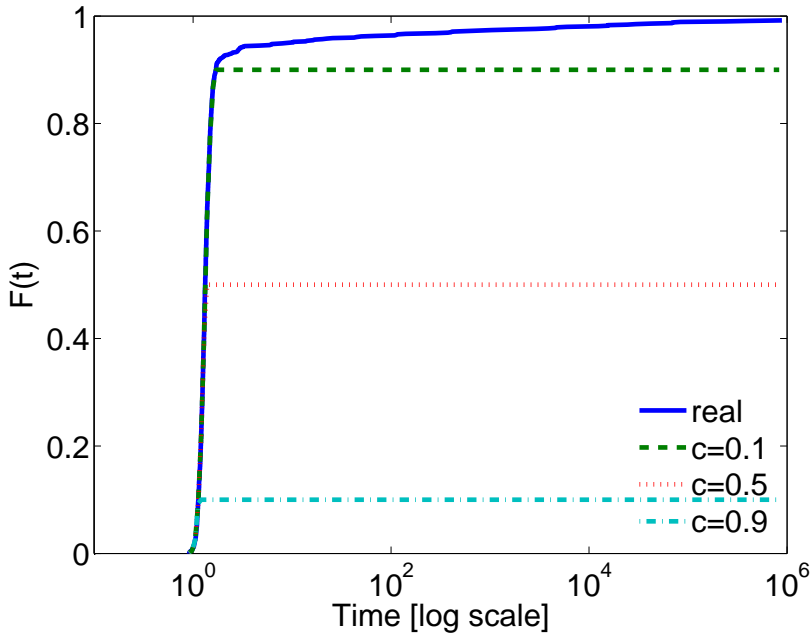
Experiments - A simple learning algorithm

For each group of instances:

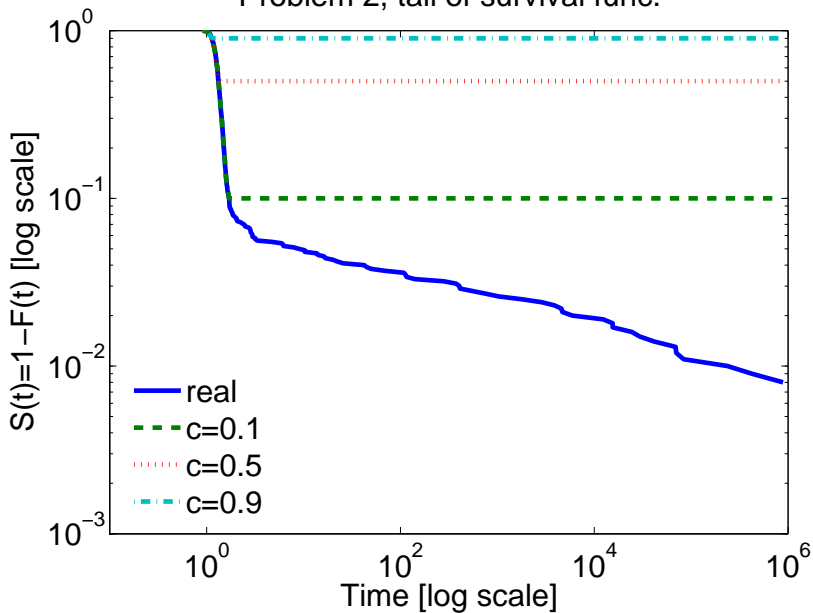
- ▶ randomly pick 50 instances
- ▶ start 20 parallel runs of the solver for each instance
- ▶ stop **all** runs when a fraction $c \in [0, 1)$ ends
- ▶ train a model \hat{F} of the RTD **for the group**
- ▶ obtain the corresponding \hat{T} minimizing $E(t_T)$
- ▶ test \hat{T} on the remaining 50 instances

We can repeat this for different levels of c , evaluating the performance of the corresponding \hat{T}

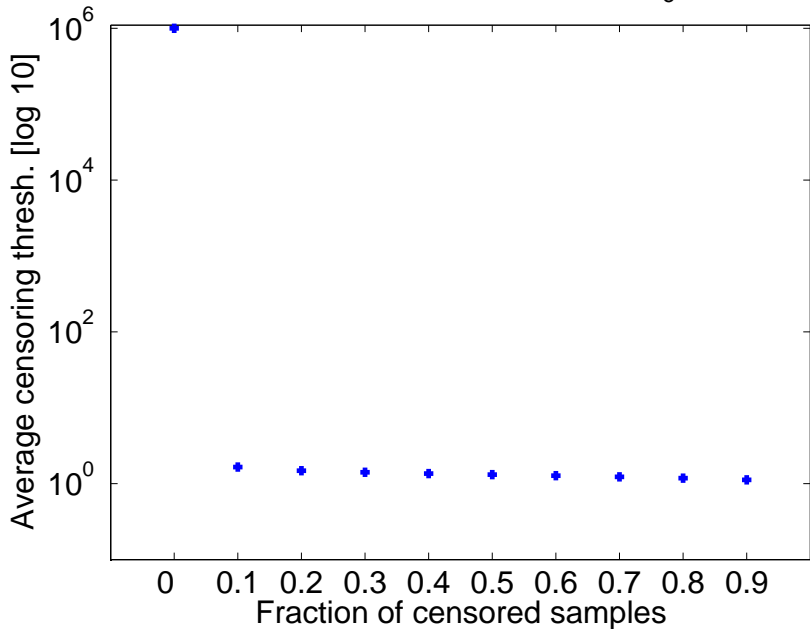
Problem 2, CDF



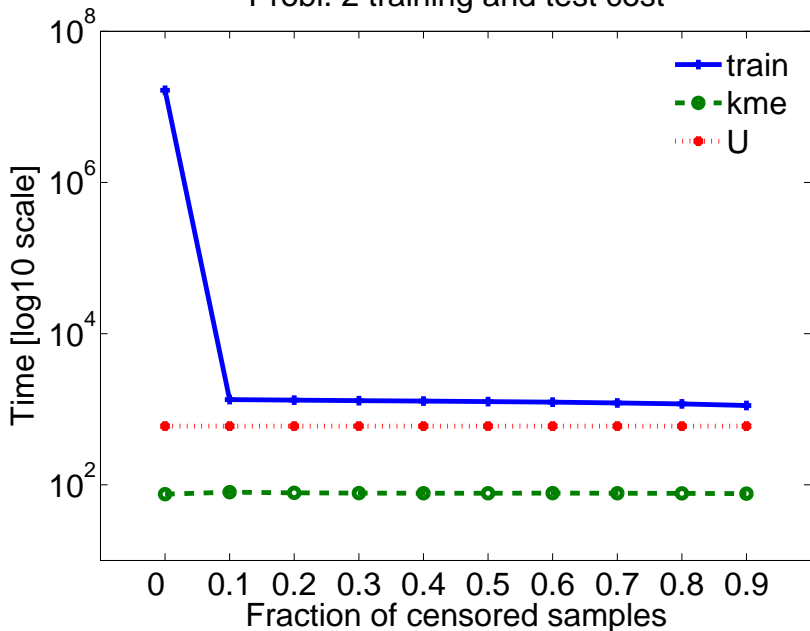
Problem 2, tail of survival func.



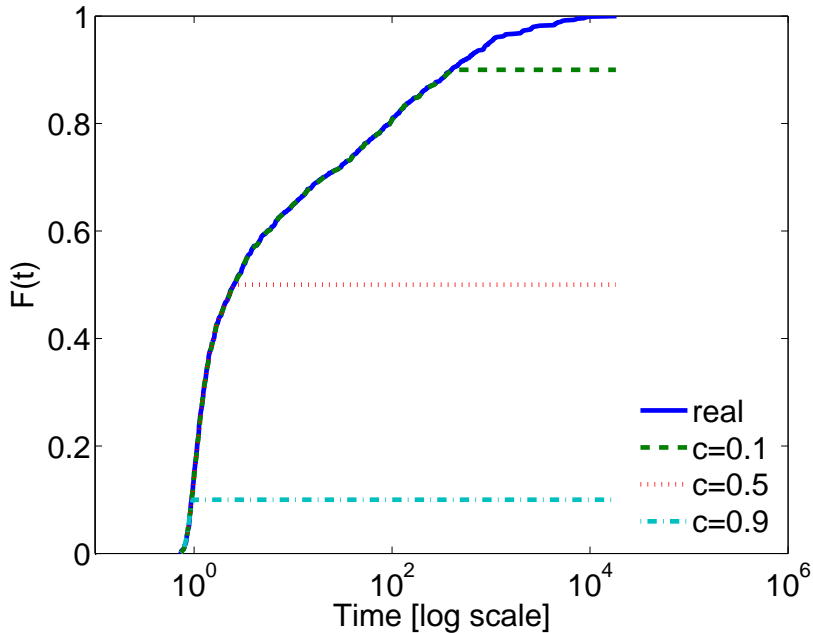
Problem 2, censoring thresh. t_c



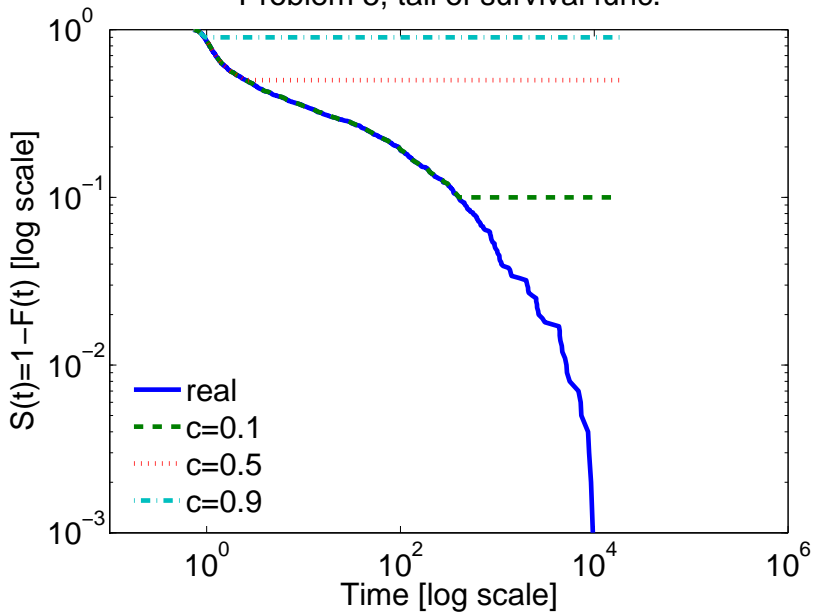
Probl. 2 training and test cost



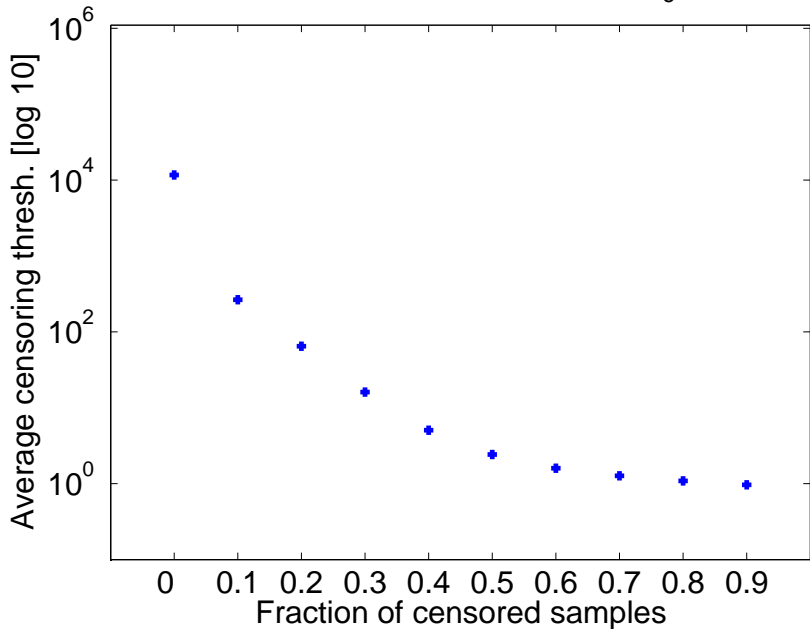
Problem 5, CDF



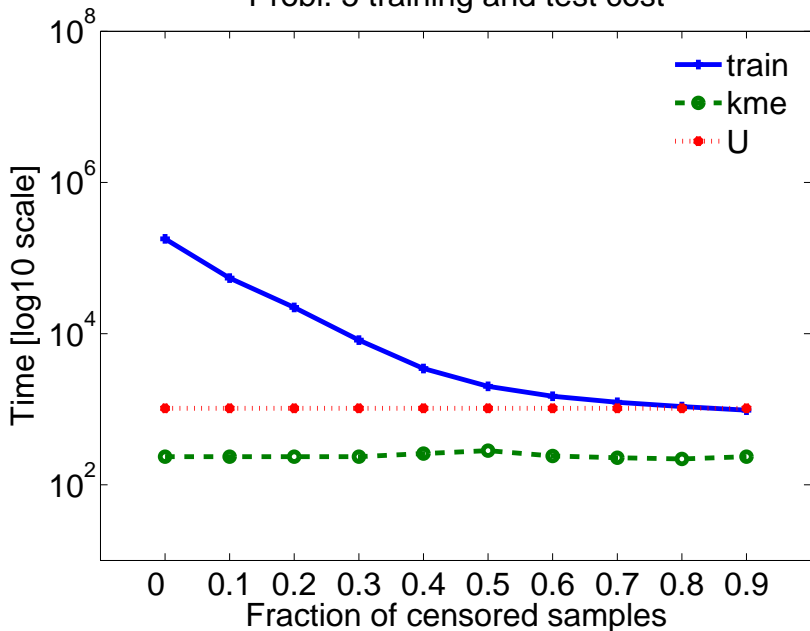
Problem 5, tail of survival func.



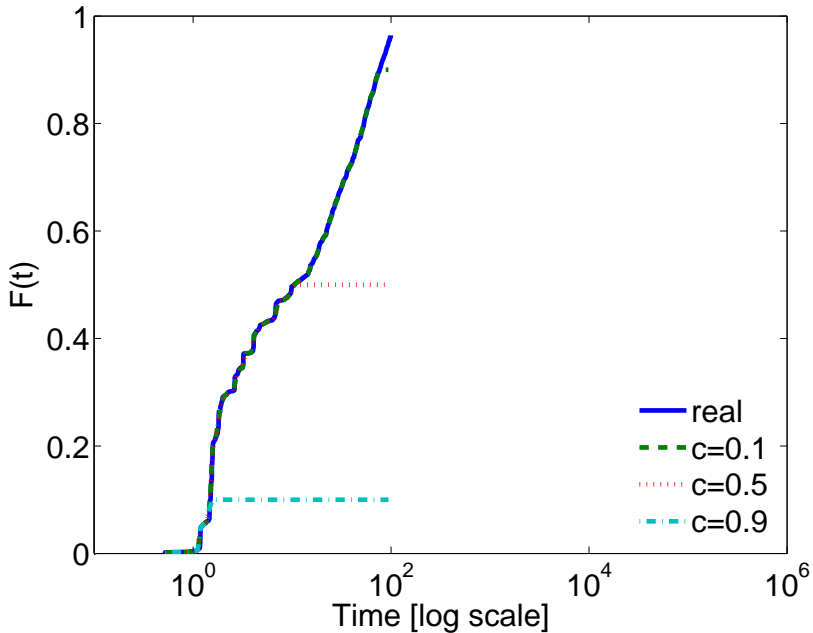
Problem 5, censoring thresh. t_c



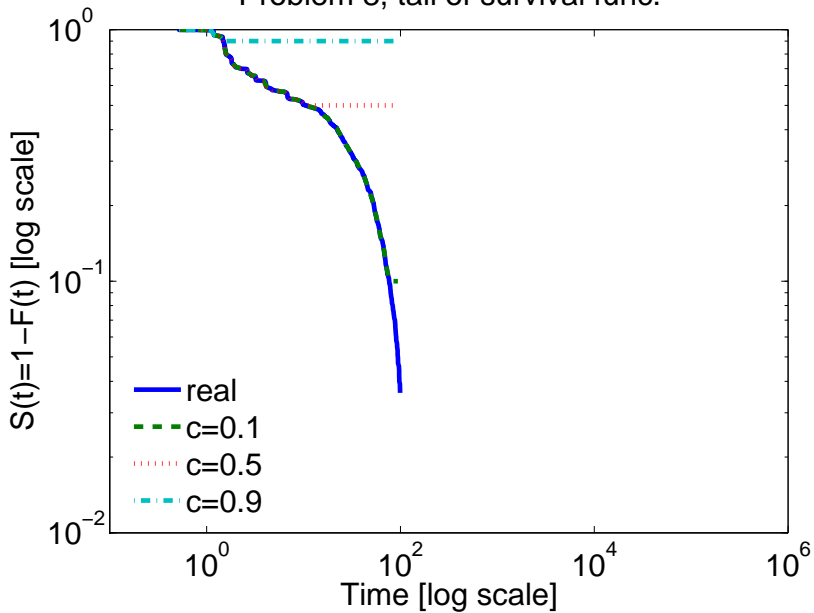
Probl. 5 training and test cost



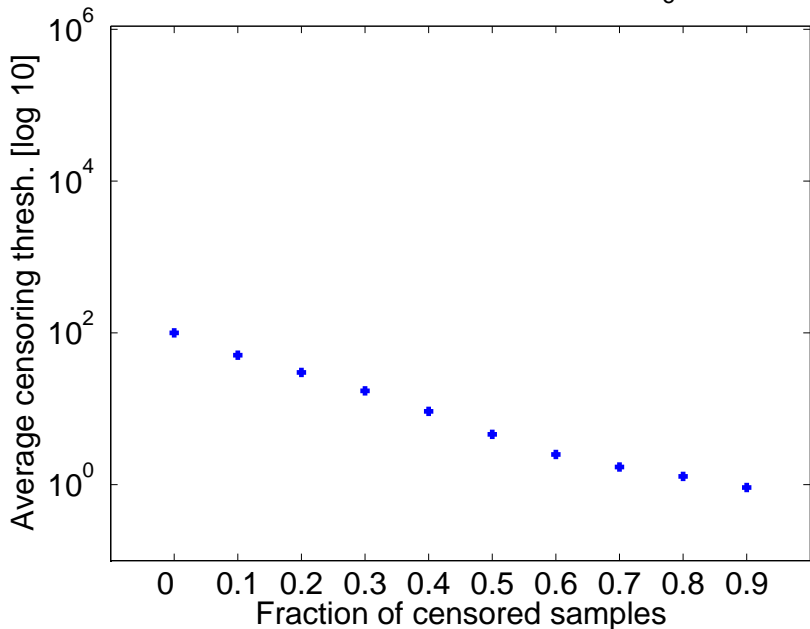
Problem 8, CDF



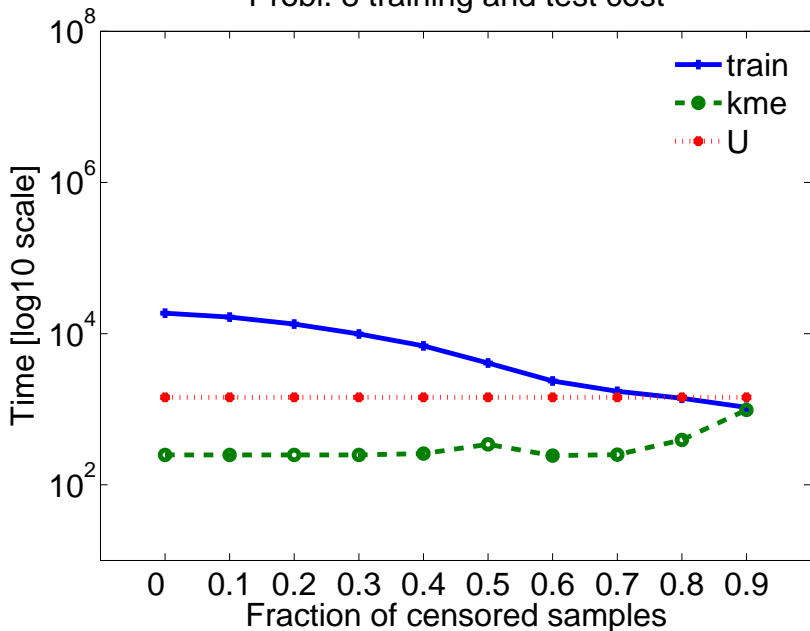
Problem 8, tail of survival func.



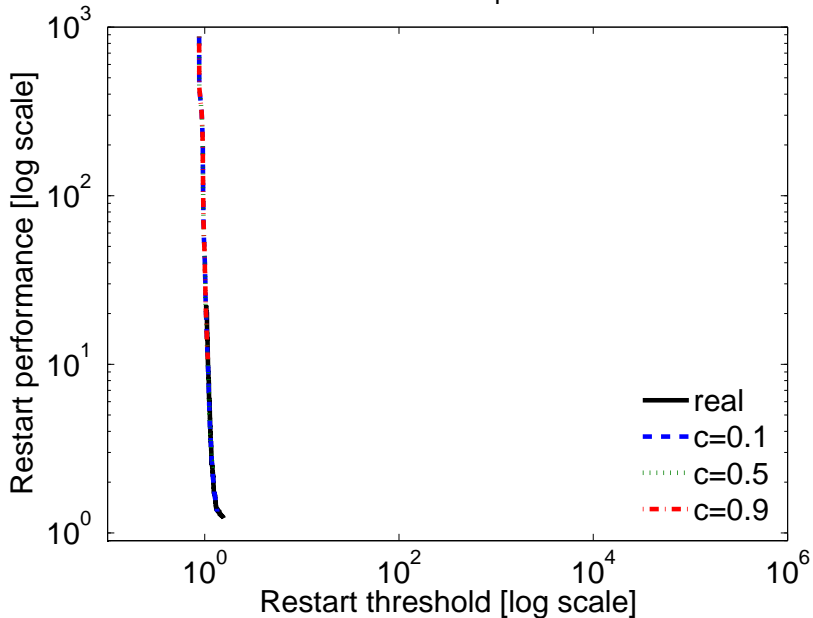
Problem 8, censoring thresh. t_c



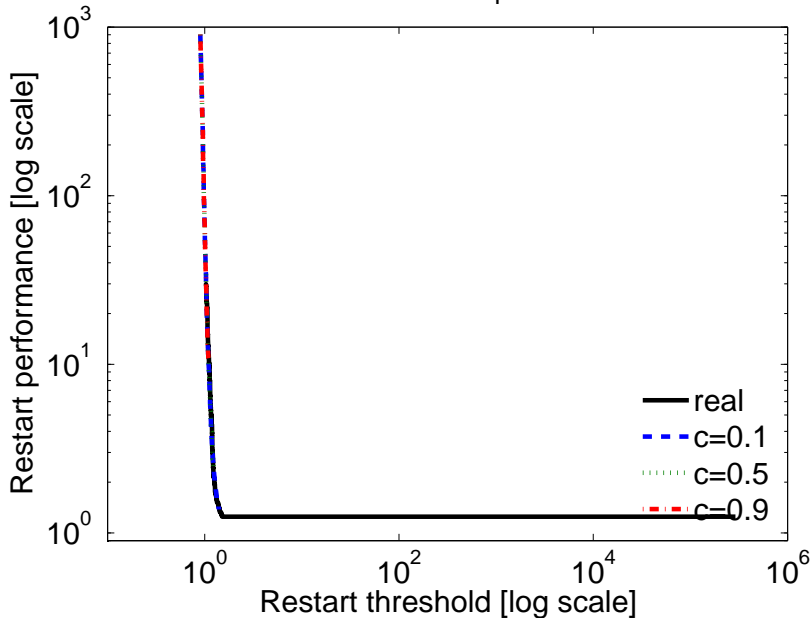
Probl. 8 training and test cost



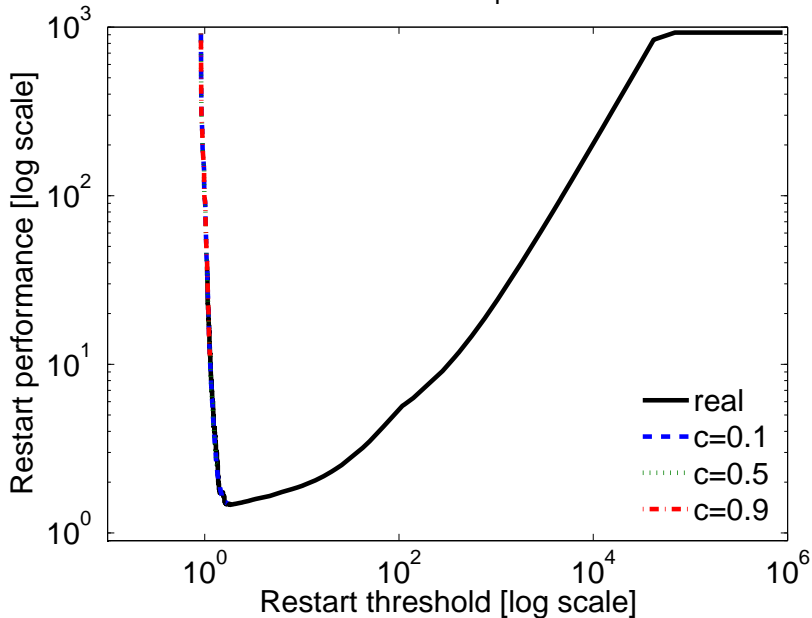
Problem 0, cost t_T of restart



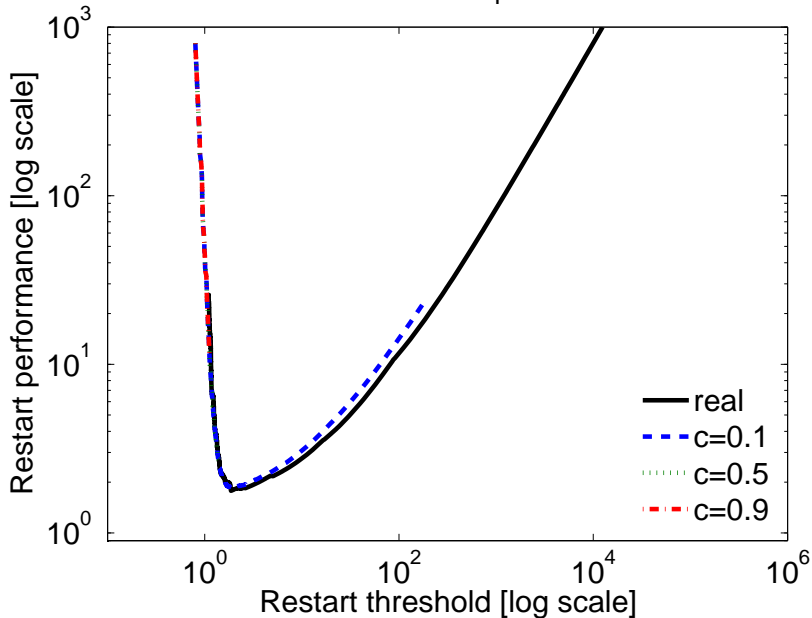
Problem 1, cost t_T of restart



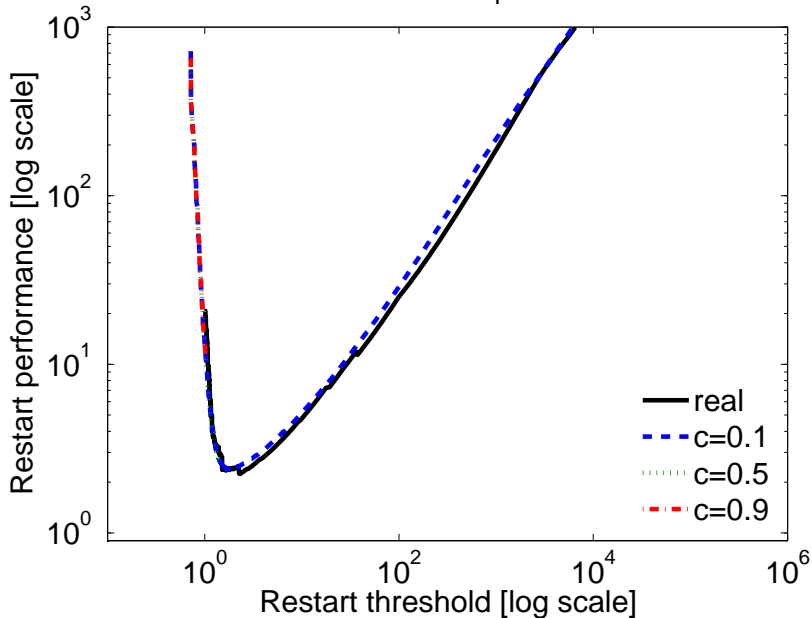
Problem 2, cost t_T of restart



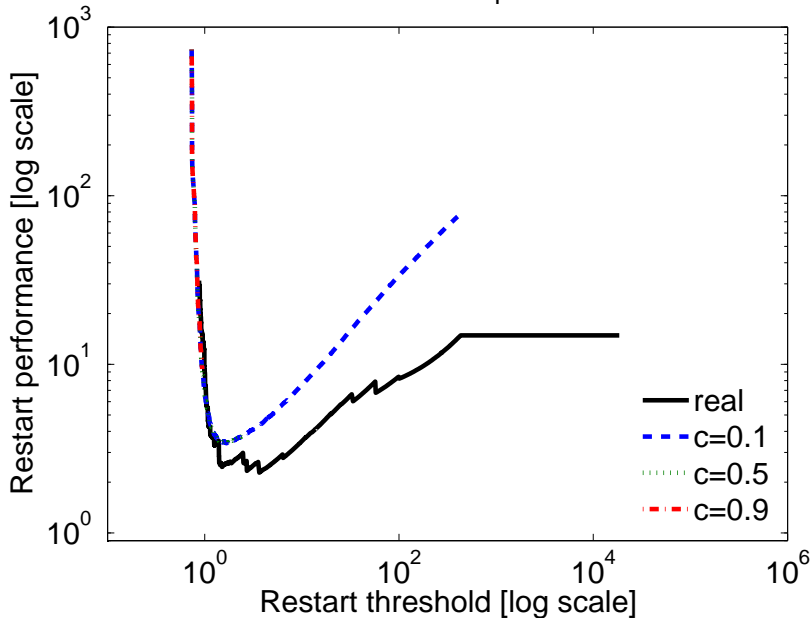
Problem 3, cost t_T of restart



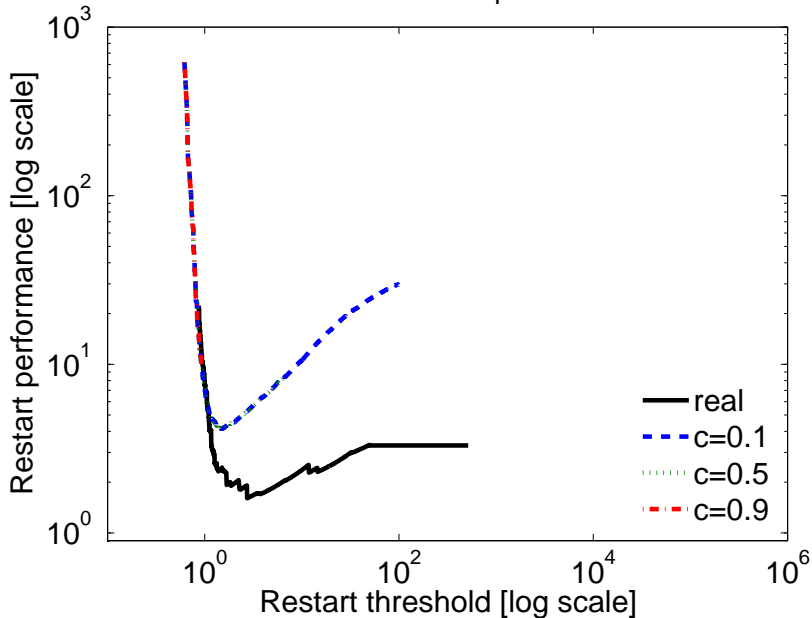
Problem 4, cost t_T of restart



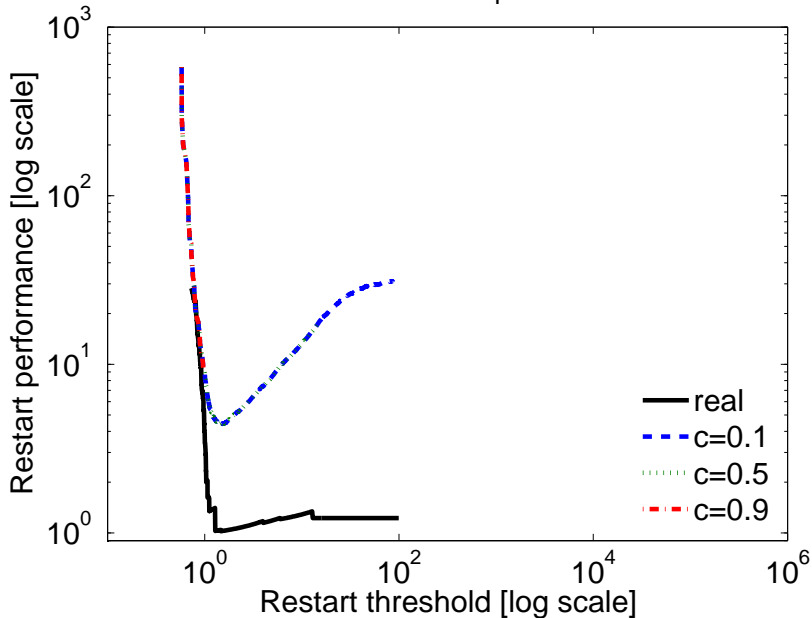
Problem 5, cost t_T of restart



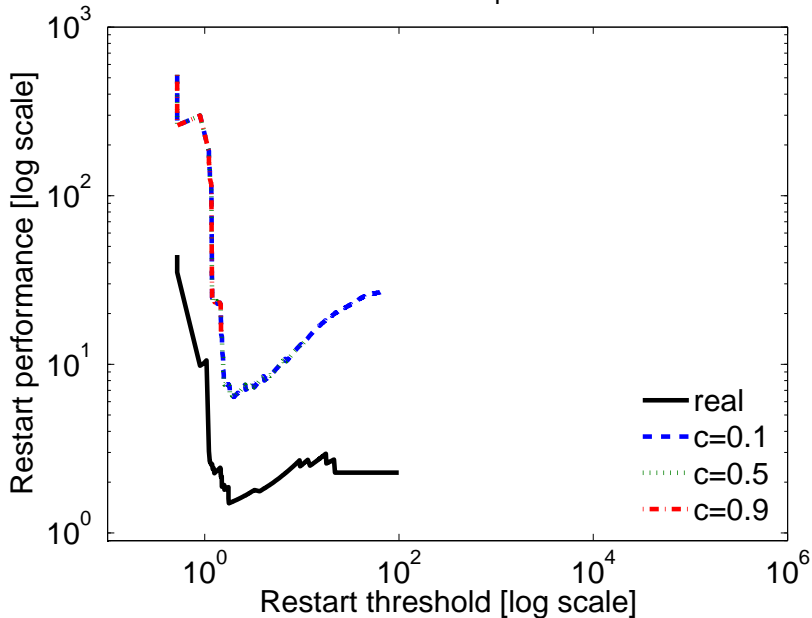
Problem 6, cost t_T of restart



Problem 7, cost t_T of restart



Problem 8, cost t_T of restart



Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.

more on: <http://www.idsia.ch/~matteo/>

Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.
- ▶ The impact on performance is pretty **low**!

more on: <http://www.idsia.ch/~matteo/>

Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.
- ▶ The impact on performance is pretty **low**!
- ▶ If a model of the RTD is used to improve performance, its quality should be measured *in terms of performance*.

more on: <http://www.idsia.ch/~matteo/>

Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.
- ▶ The impact on performance is pretty **low**!
- ▶ If a model of the RTD is used to improve performance, its quality should be measured *in terms of performance*.

What next?

more on: <http://www.idsia.ch/~matteo/>

Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.
- ▶ The impact on performance is pretty **low**!
- ▶ If a model of the RTD is used to improve performance, its quality should be measured *in terms of performance*.

What next?

- ▶ To further reduce training cost: *life-long* learning of a restart strategy (IJCAI 2007)

more on: <http://www.idsia.ch/~matteo/>

Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.
- ▶ The impact on performance is pretty **low**!
- ▶ If a model of the RTD is used to improve performance, its quality should be measured *in terms of performance*.

What next?

- ▶ To further reduce training cost: *life-long* learning of a restart strategy (IJCAI 2007)
- ▶ To discriminate among instances: *conditional* models (work in progress..)

more on: <http://www.idsia.ch/~matteo/>

Conclusions

- ▶ The impact of censored sampling on model precision can be quite **high**.
- ▶ The impact on performance is pretty **low**!
- ▶ If a model of the RTD is used to improve performance, its quality should be measured *in terms of performance*.

What next?

- ▶ To further reduce training cost: *life-long* learning of a restart strategy (IJCAI 2007)
- ▶ To discriminate among instances: *conditional* models (work in progress..)
- ▶ Answer some **QUESTIONS?**

more on: <http://www.idsia.ch/~matteo/>

Survival analysis - Parametric models

Both for Maximum Likelihood and Bayesian method, one needs to evaluate the likelihood of the parameter θ given the data.

Assuming i.i.d. samples, this amounts to evaluating the likelihood given each sample separately

$$L(\theta|\{t_i\}) = \prod_i L(\theta|t_i)$$

Survival analysis - Notation

- ▶ $n(t)$ number of individual/components *at risk* at time t
- ▶ T_i time of event (failure/death)
- ▶ C_i *censoring* time
- ▶ $t_i = \min(T_i, C_i)$ **observed** time sample
- ▶ ν_i censoring indicator
- ▶ $F(t) = \Pr\{T \leq t\}$ lifetime CDF $F(t) = \int_0^\infty f(t)dt$
- ▶ $S(t) = 1 - F(t)$ *survival* function
- ▶ $h(t) = \frac{f(t)}{S(t)}$ *hazard* function

Survival Analysis - Non-parametric models

Empirical CDF, without censoring ($n(t) = n$)

$$\hat{F}(t) = \sum_{t_j < t} \frac{1}{n}$$

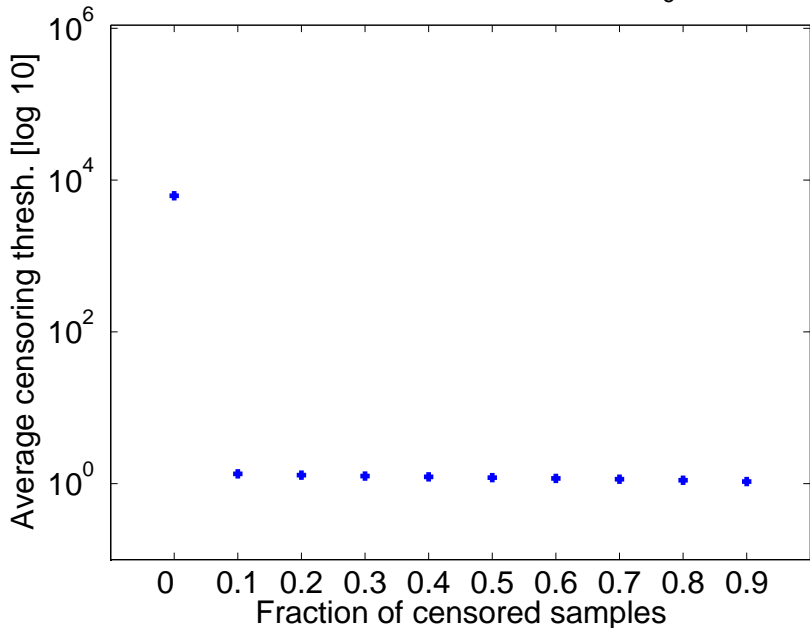
With censoring: Product-Limit Estimator (Kaplan, Meier, 1958)

$$\hat{h}(t) = \frac{\sum_{t_i = t} 1}{\sum_{t_i \geq t} 1}$$

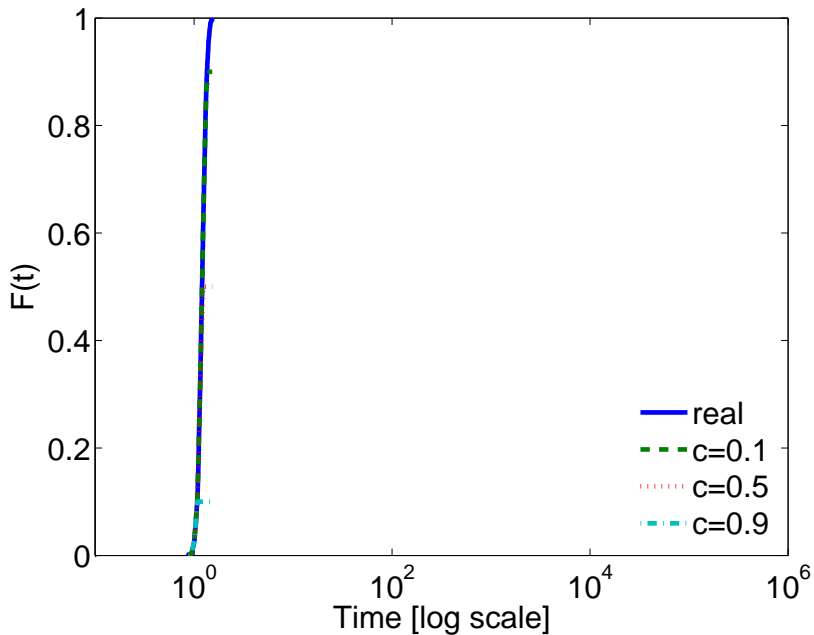
In both cases:

- ▶ $F(t)$ is a step function
- ▶ $f(t)$ is a train of pulses $f(t) = \sum_i w_i \delta(t - t_i)$

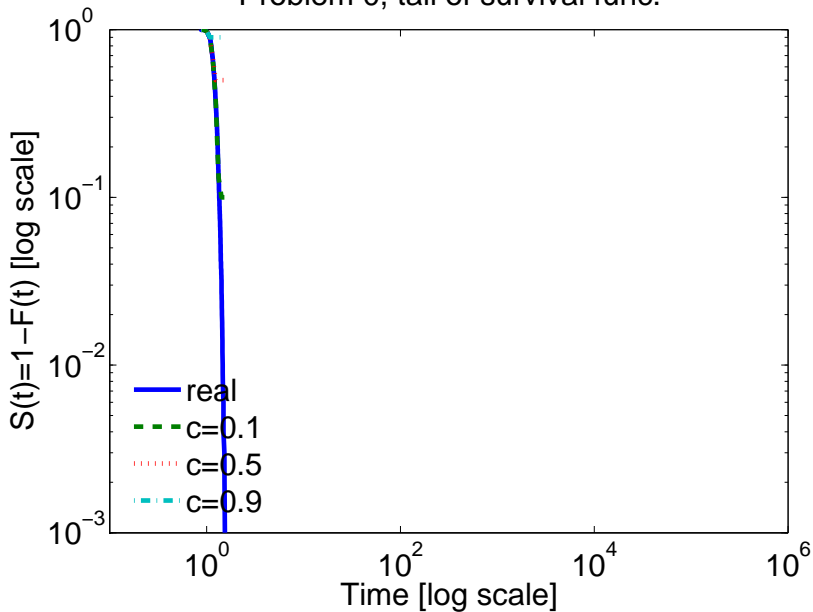
Problem 0, censoring thresh. t_c



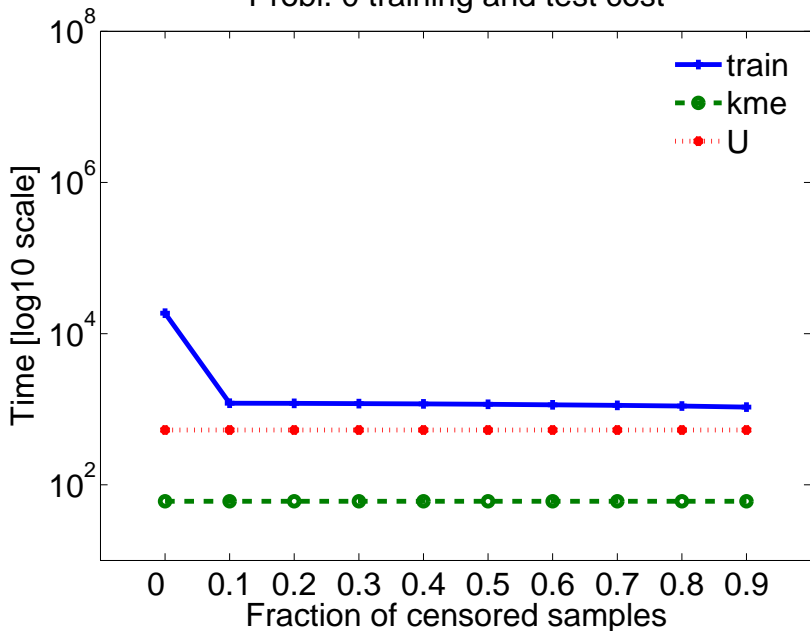
Problem 0, CDF



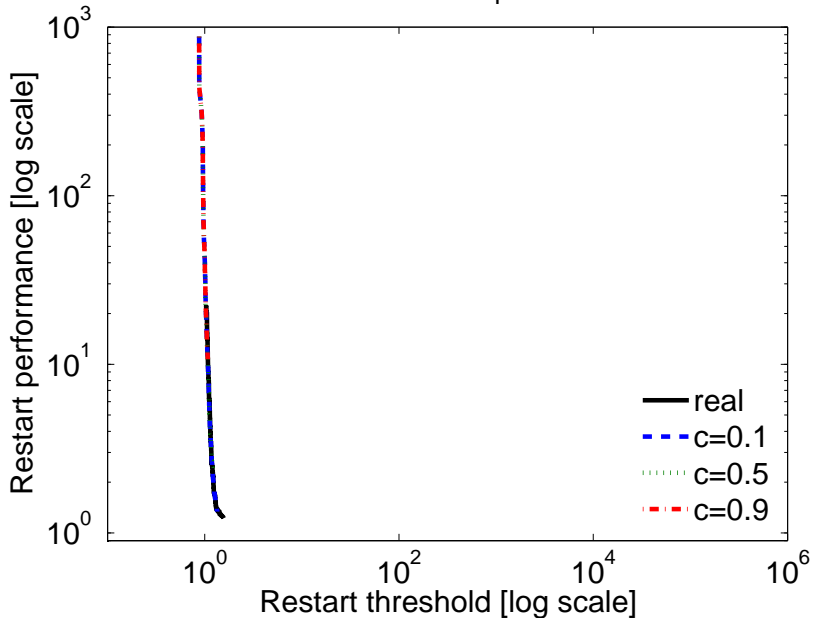
Problem 0, tail of survival func.



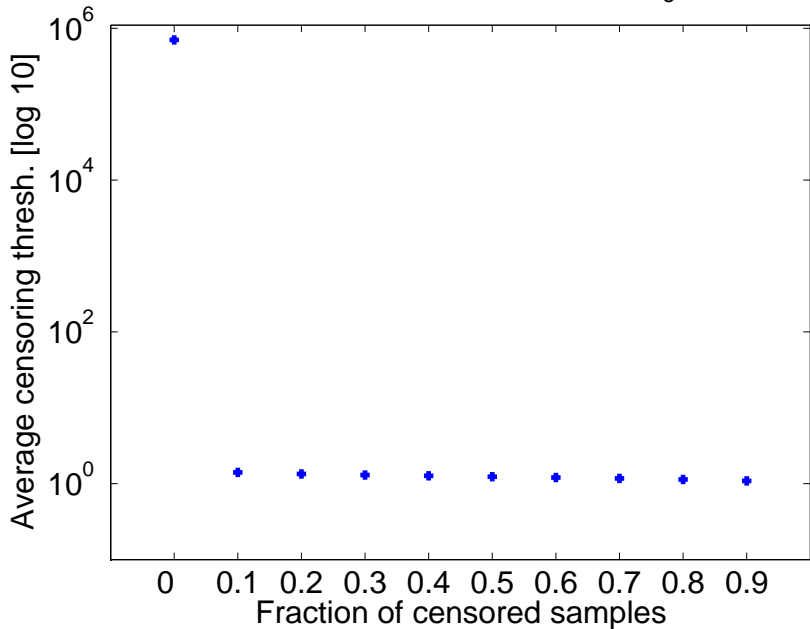
Probl. 0 training and test cost



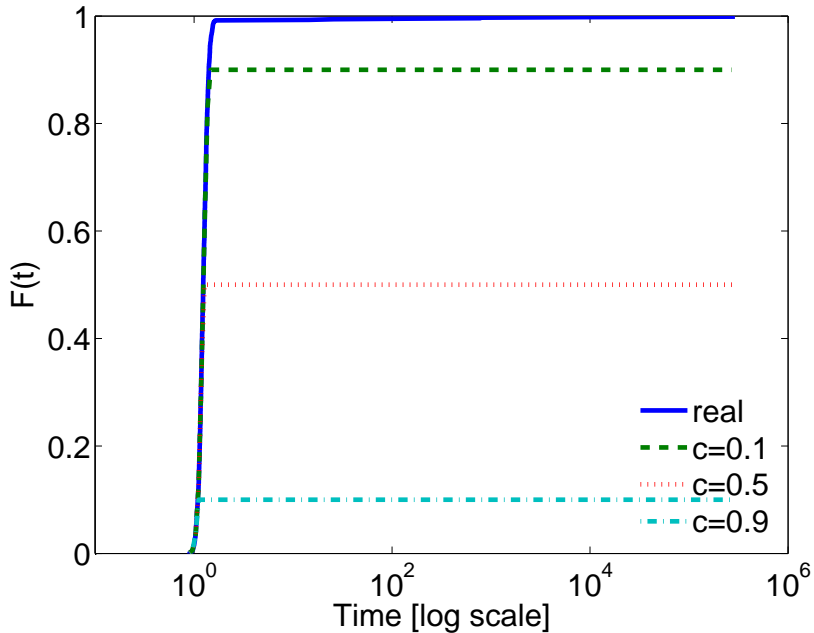
Problem 0, cost t_T of restart



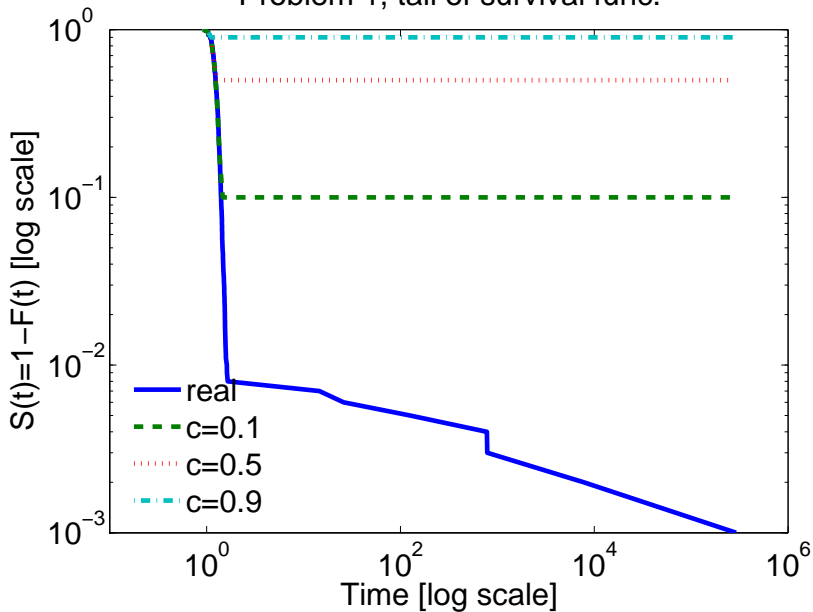
Problem 1, censoring thresh. t_c



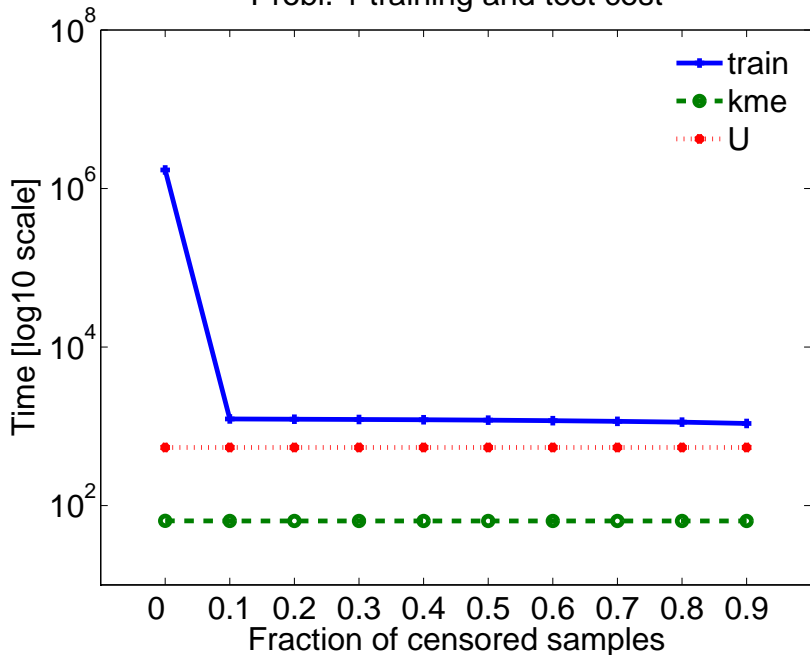
Problem 1, CDF



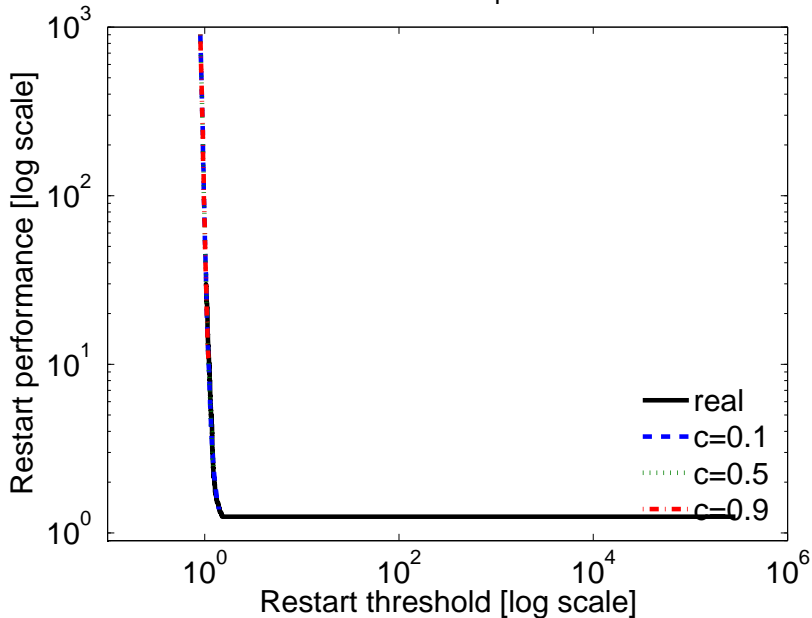
Problem 1, tail of survival func.



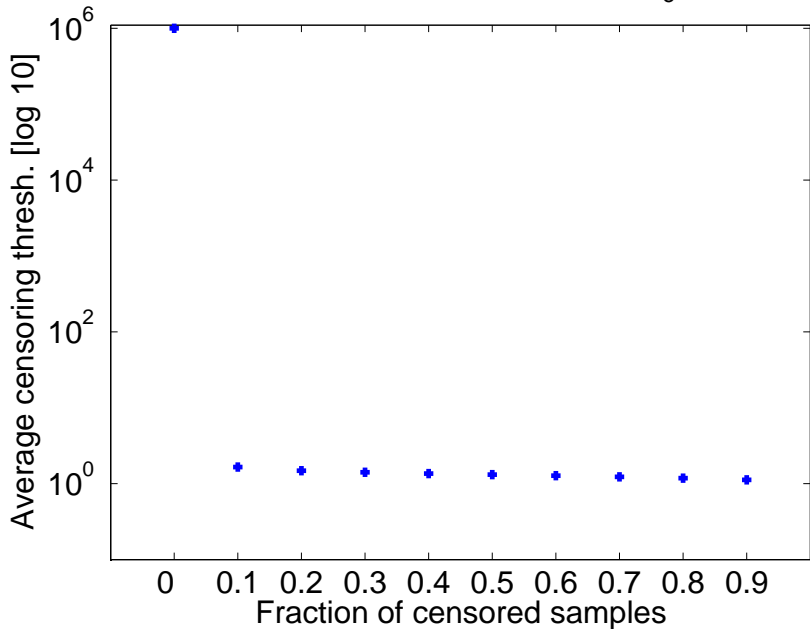
Probl. 1 training and test cost



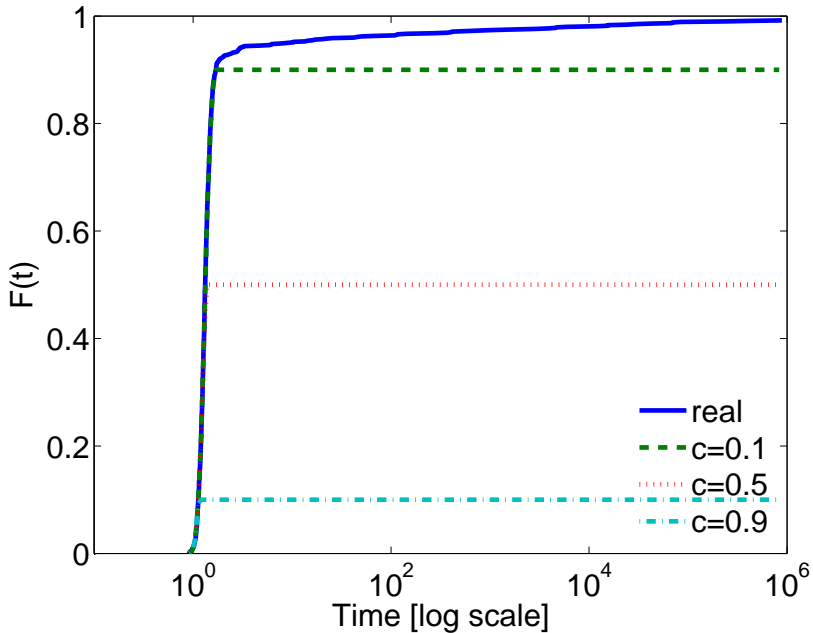
Problem 1, cost t_T of restart



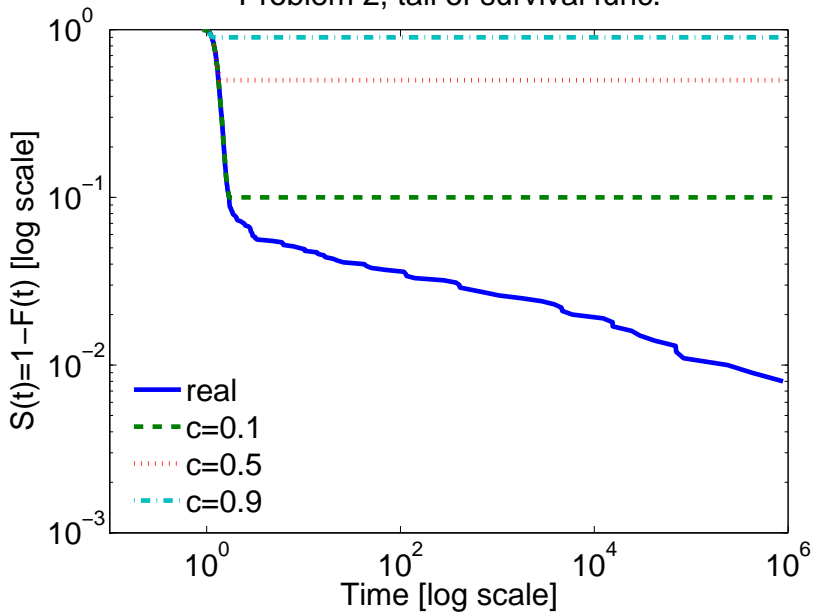
Problem 2, censoring thresh. t_c



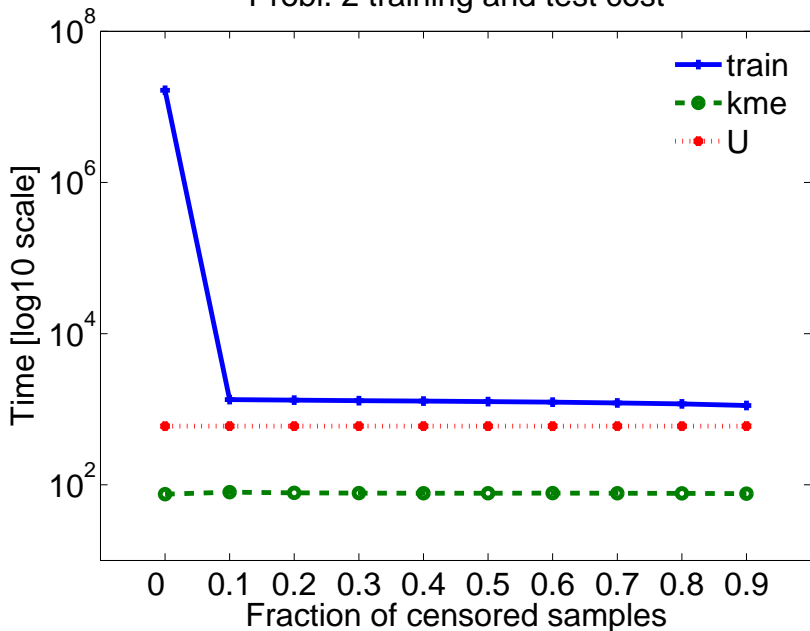
Problem 2, CDF



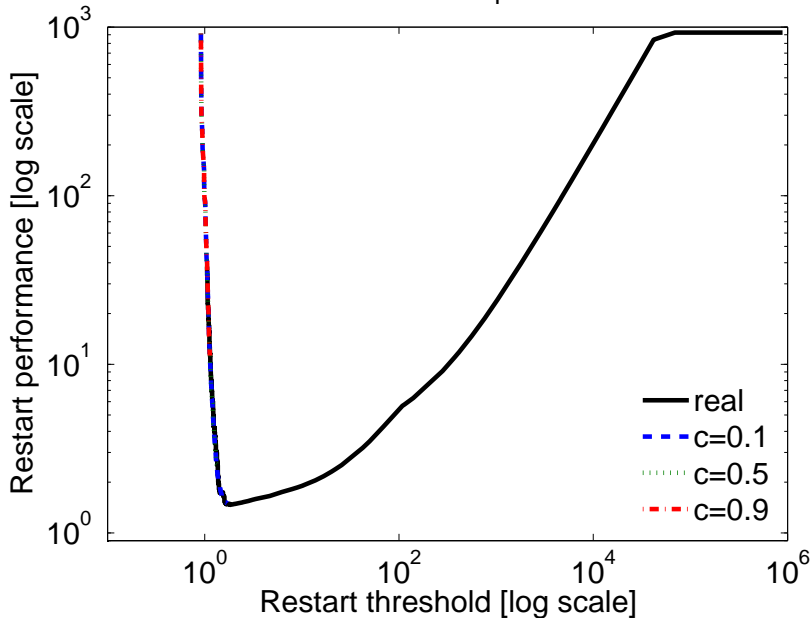
Problem 2, tail of survival func.



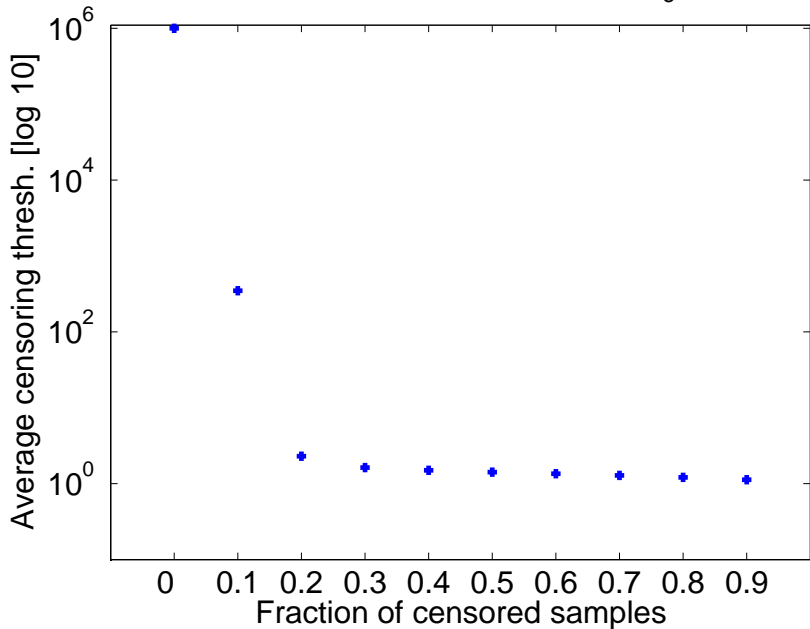
Probl. 2 training and test cost



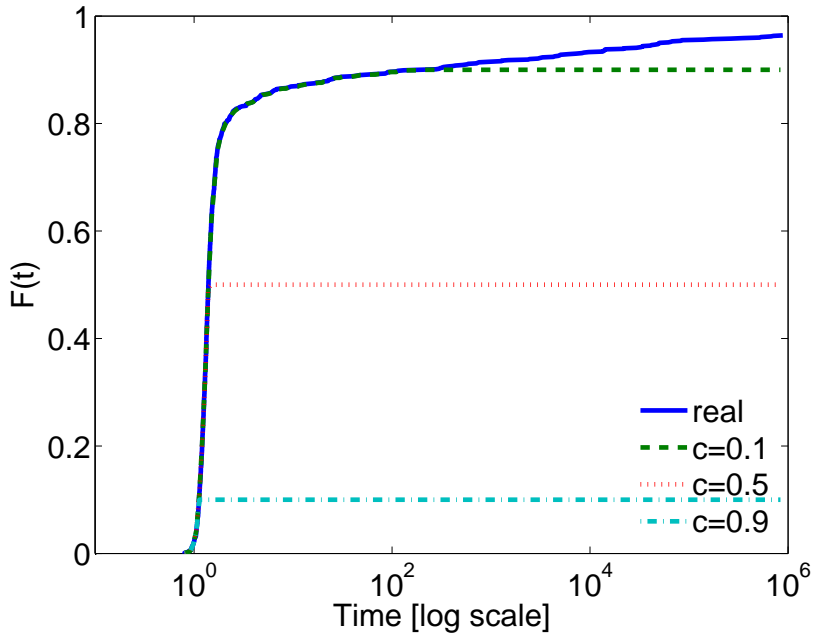
Problem 2, cost t_T of restart



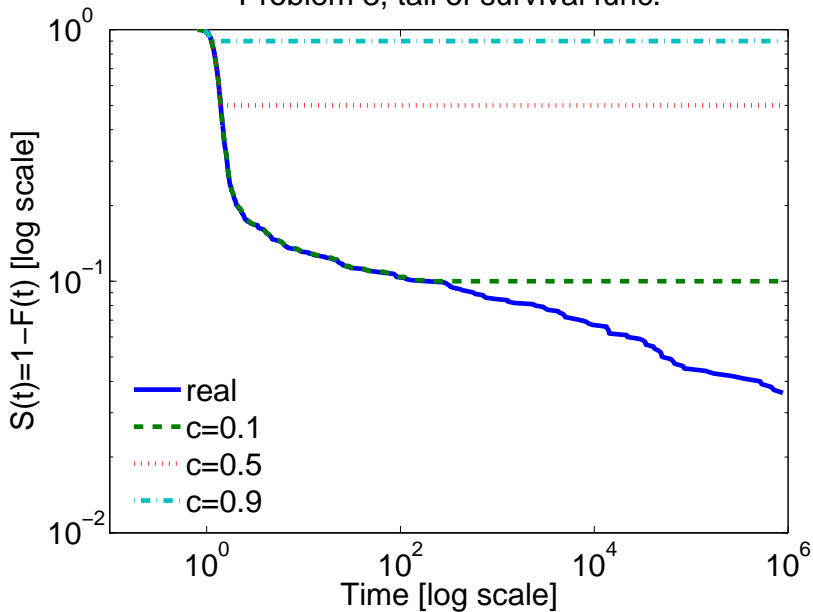
Problem 3, censoring thresh. t_c



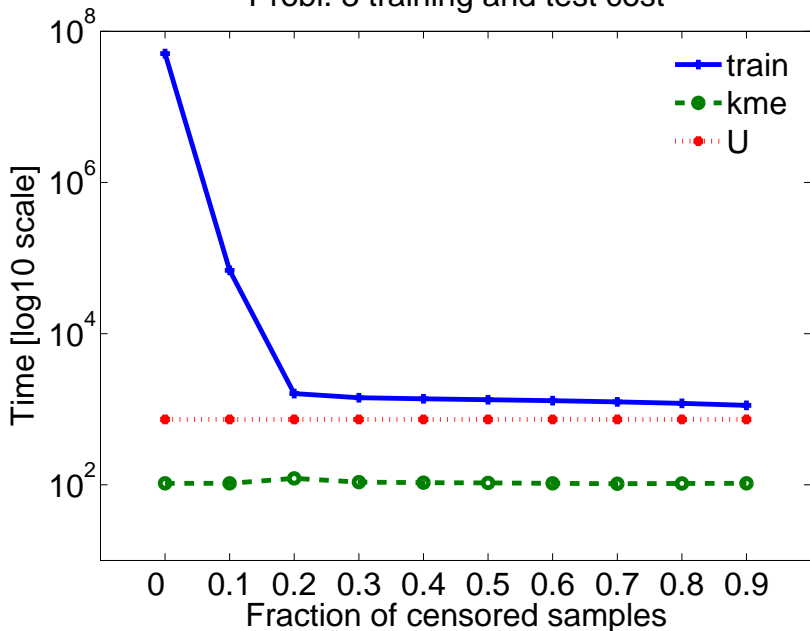
Problem 3, CDF



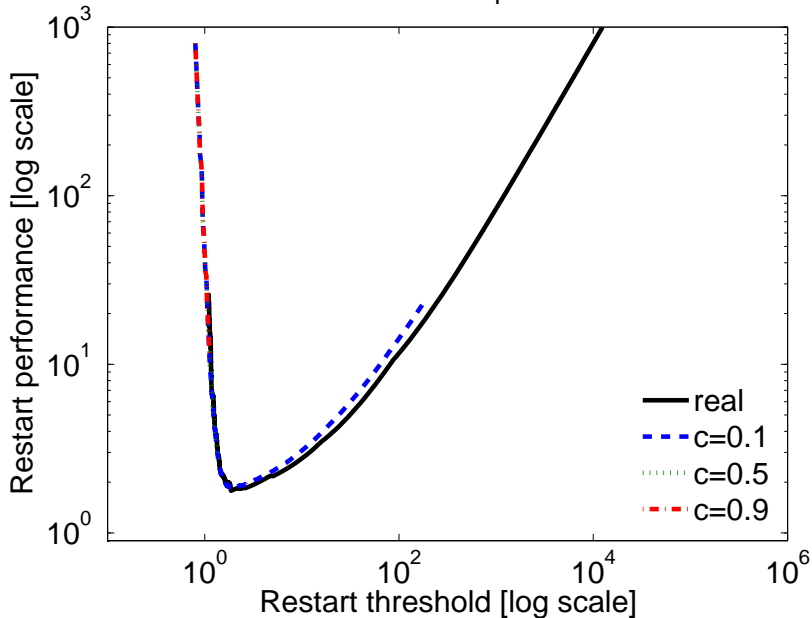
Problem 3, tail of survival func.



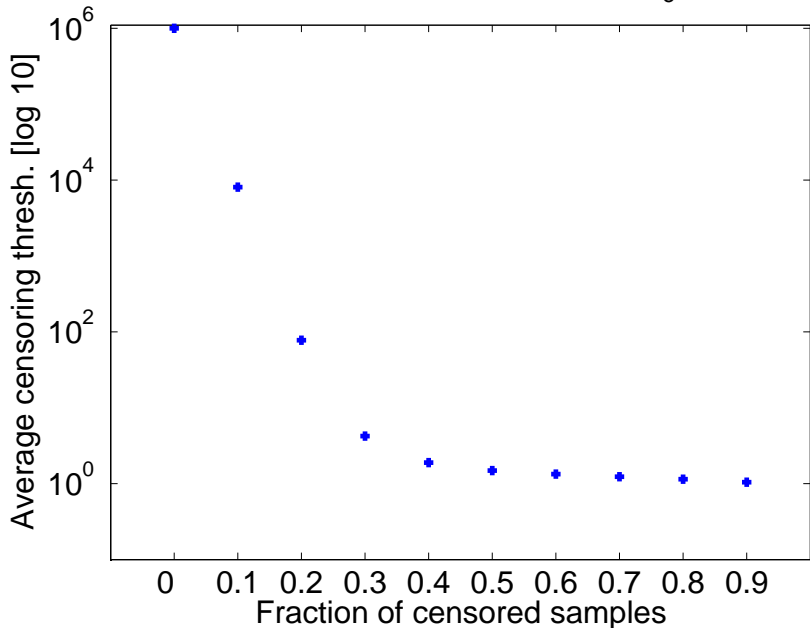
Probl. 3 training and test cost



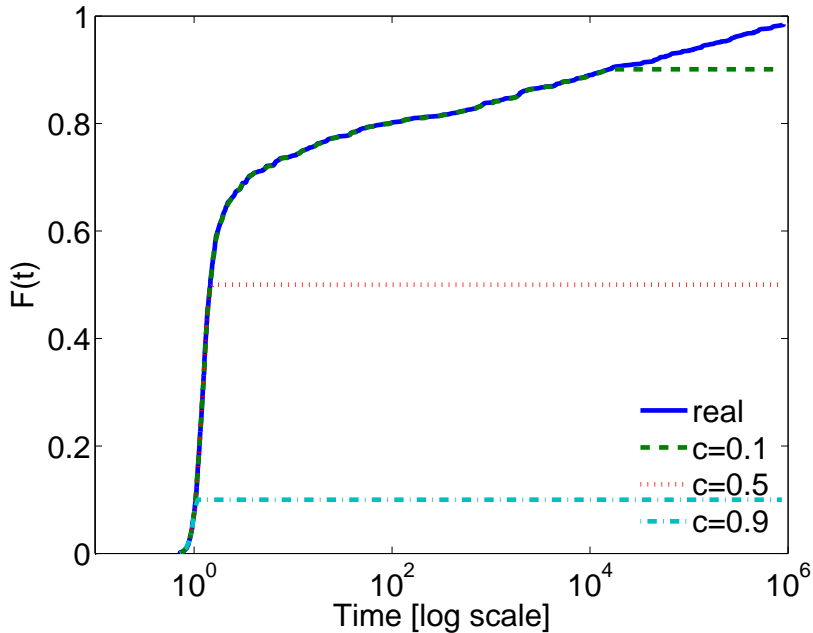
Problem 3, cost t_T of restart



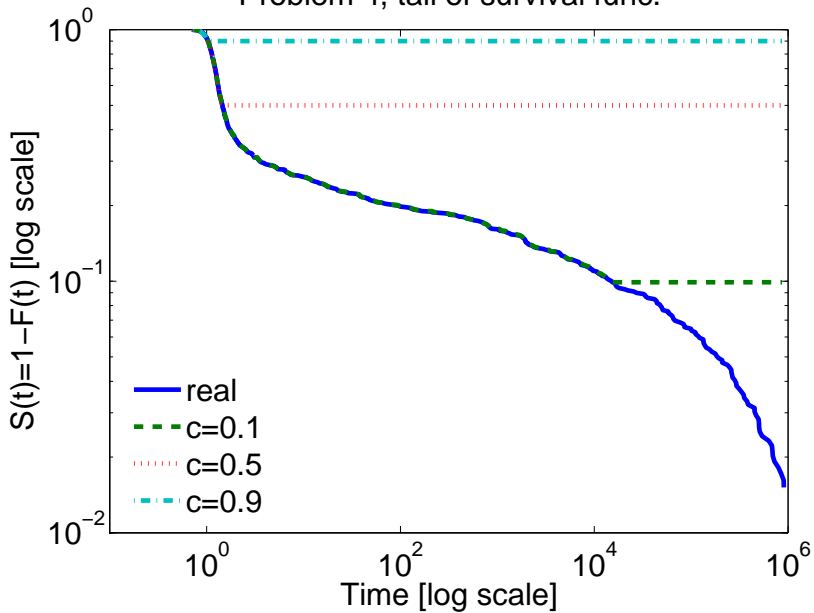
Problem 4, censoring thresh. t_c



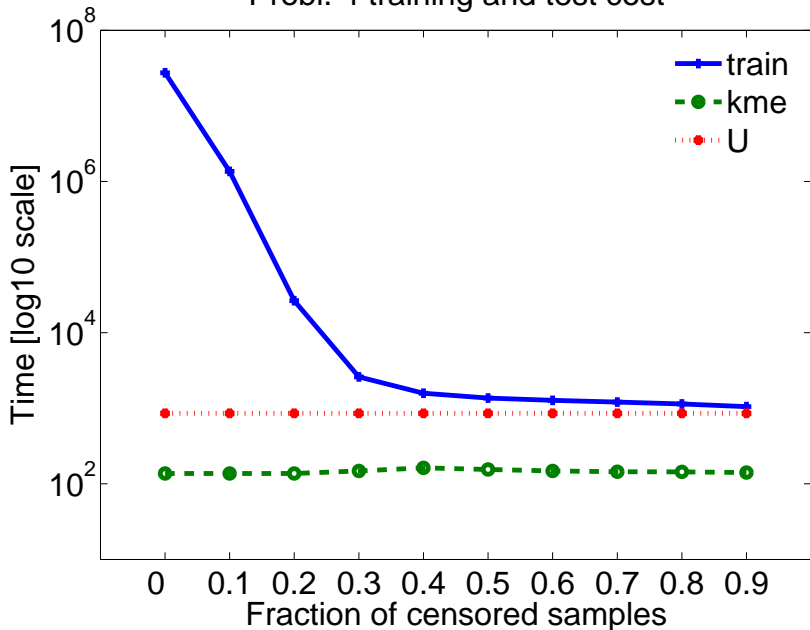
Problem 4, CDF



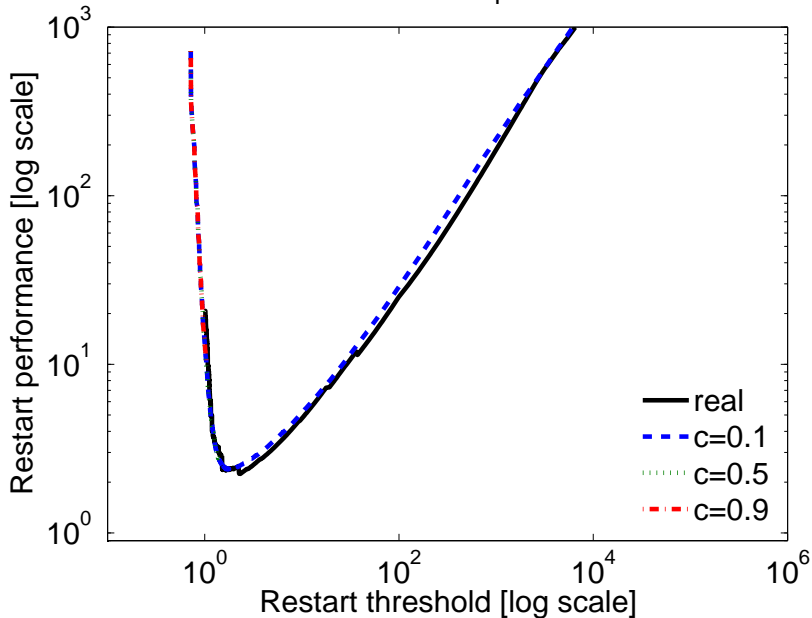
Problem 4, tail of survival func.



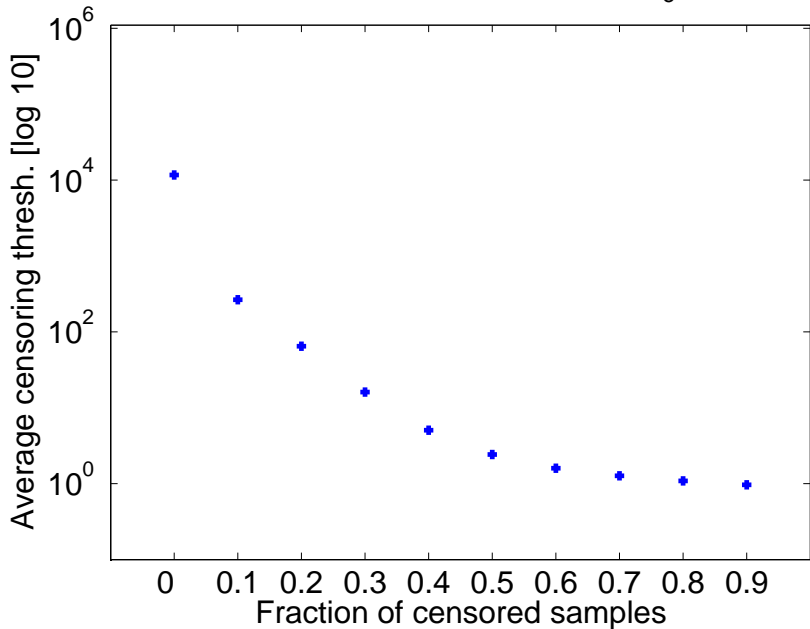
Probl. 4 training and test cost



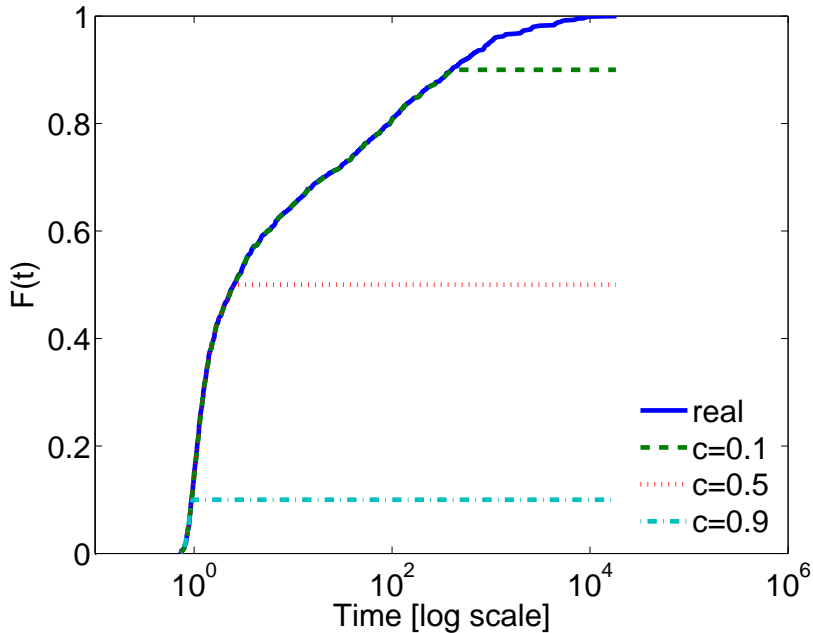
Problem 4, cost t_T of restart



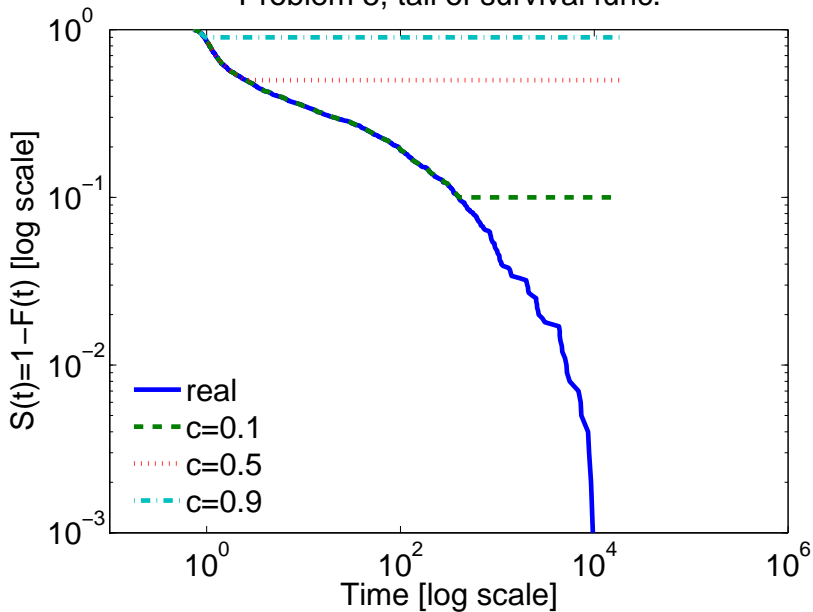
Problem 5, censoring thresh. t_c



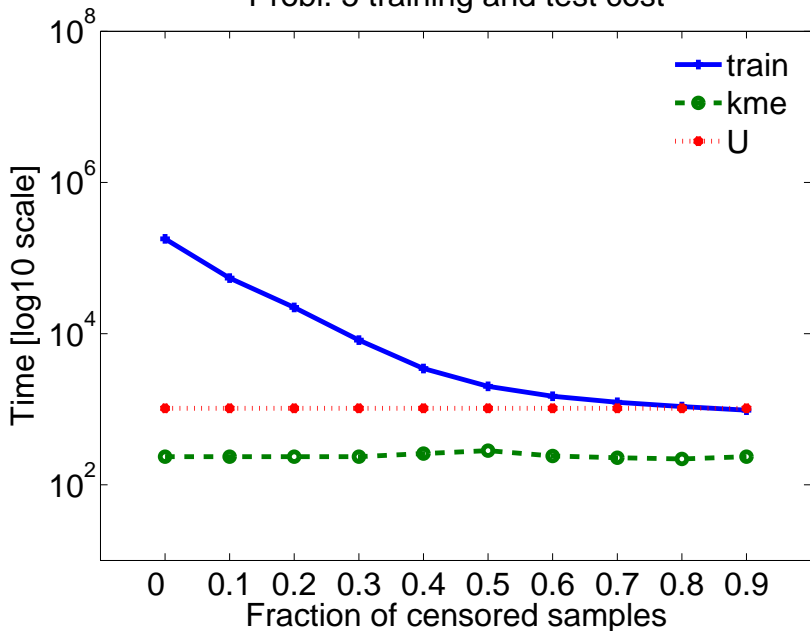
Problem 5, CDF



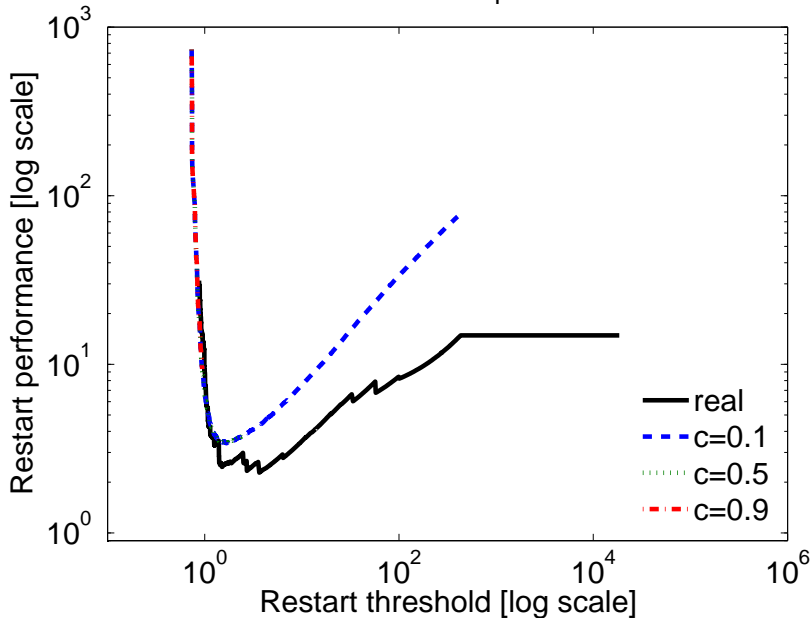
Problem 5, tail of survival func.



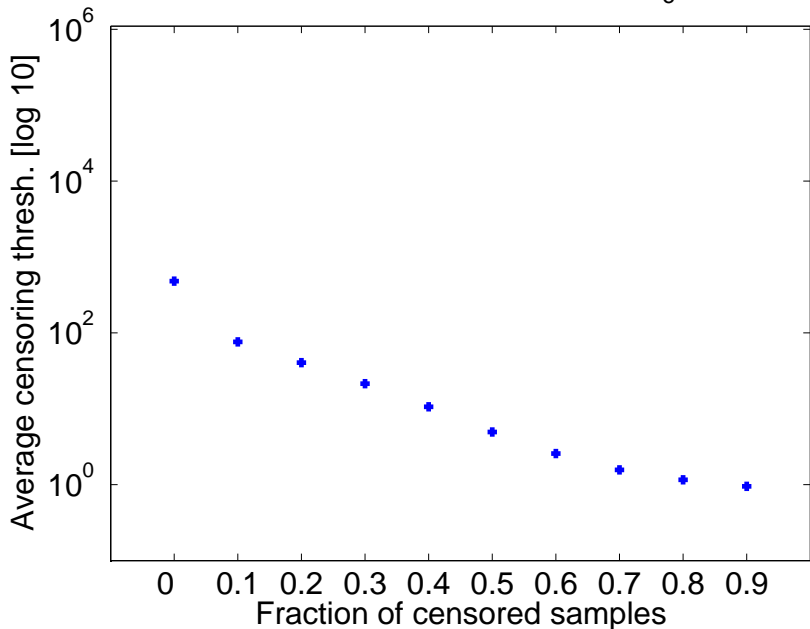
Probl. 5 training and test cost



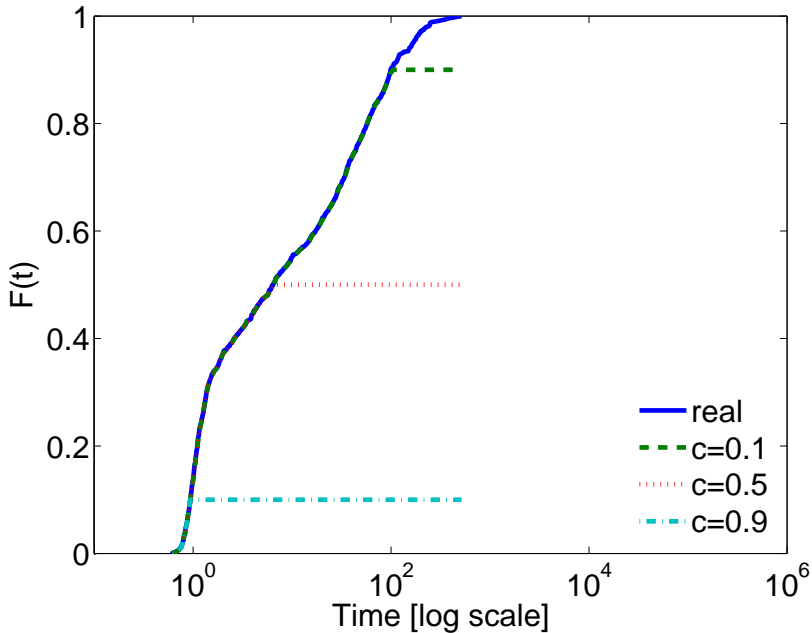
Problem 5, cost t_T of restart



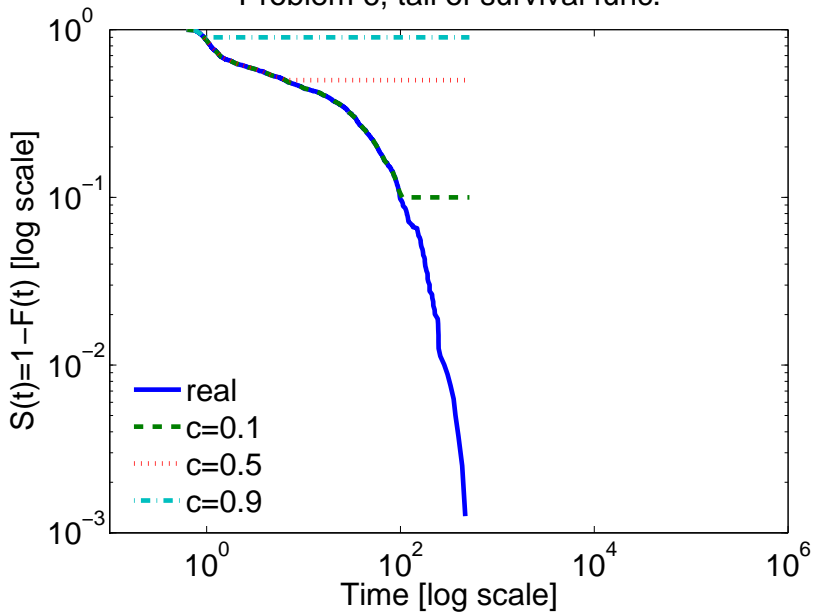
Problem 6, censoring thresh. t_c



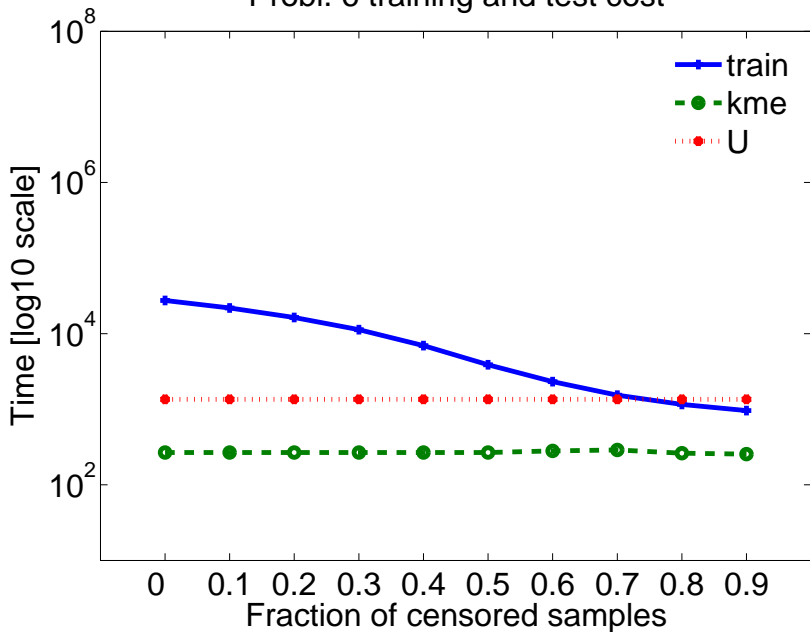
Problem 6, CDF



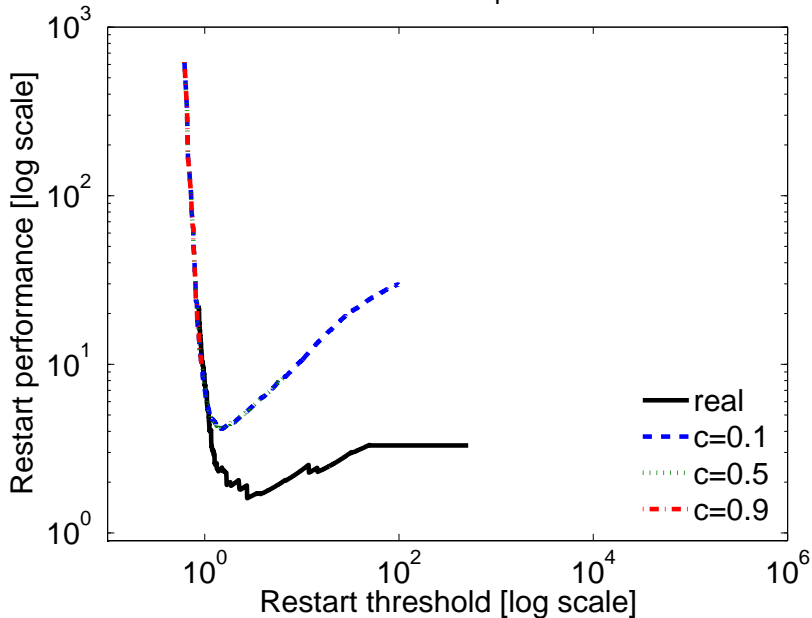
Problem 6, tail of survival func.



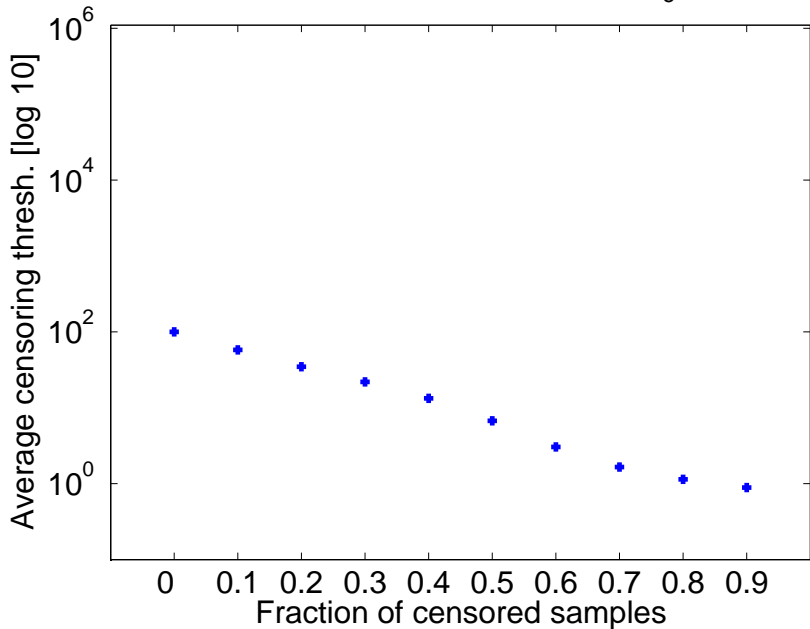
Probl. 6 training and test cost



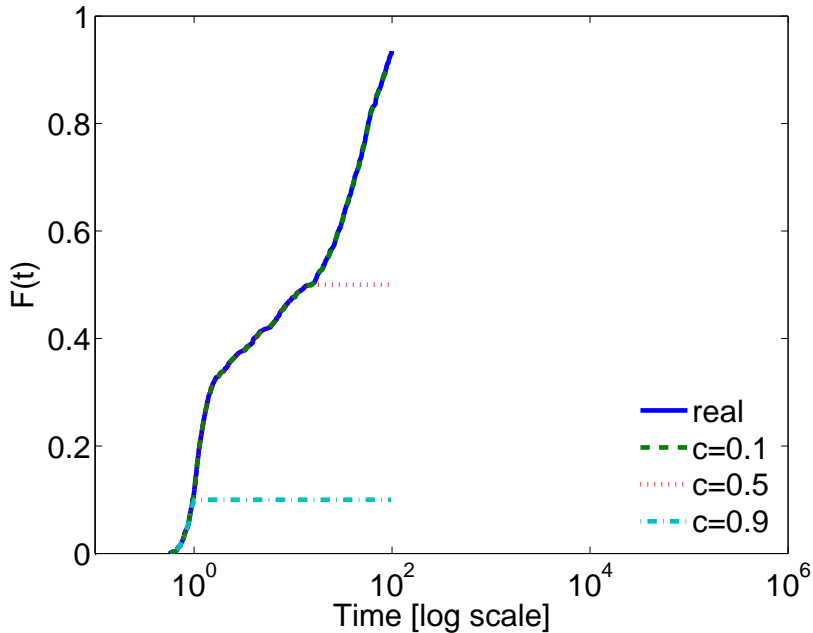
Problem 6, cost t_T of restart



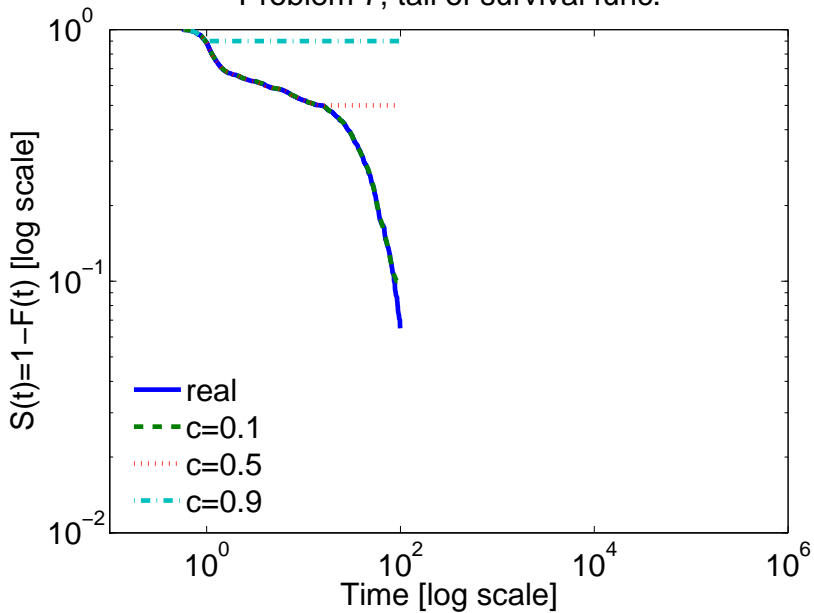
Problem 7, censoring thresh. t_c



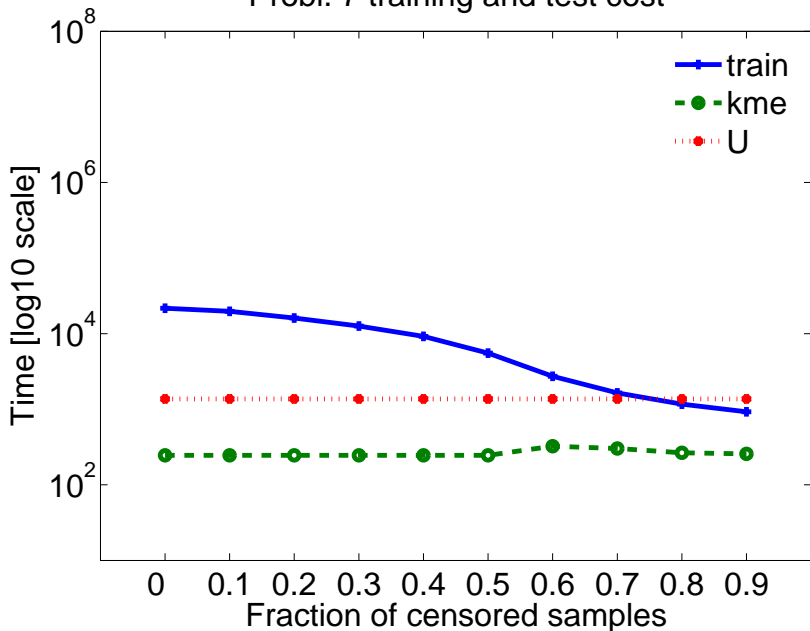
Problem 7, CDF



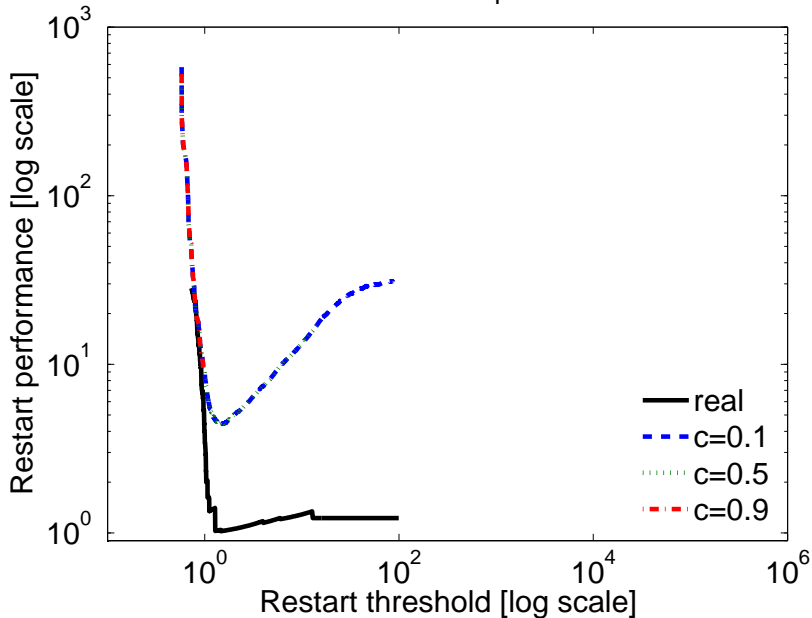
Problem 7, tail of survival func.



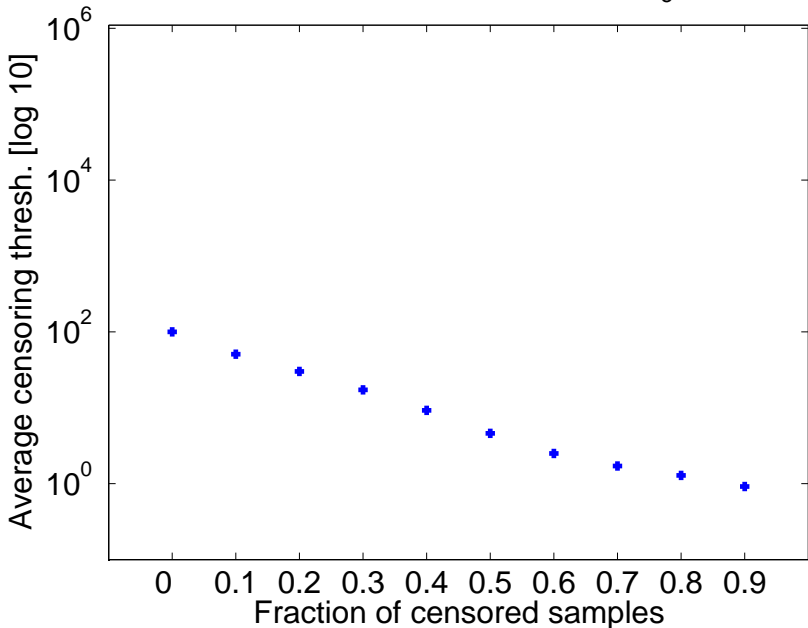
Probl. 7 training and test cost



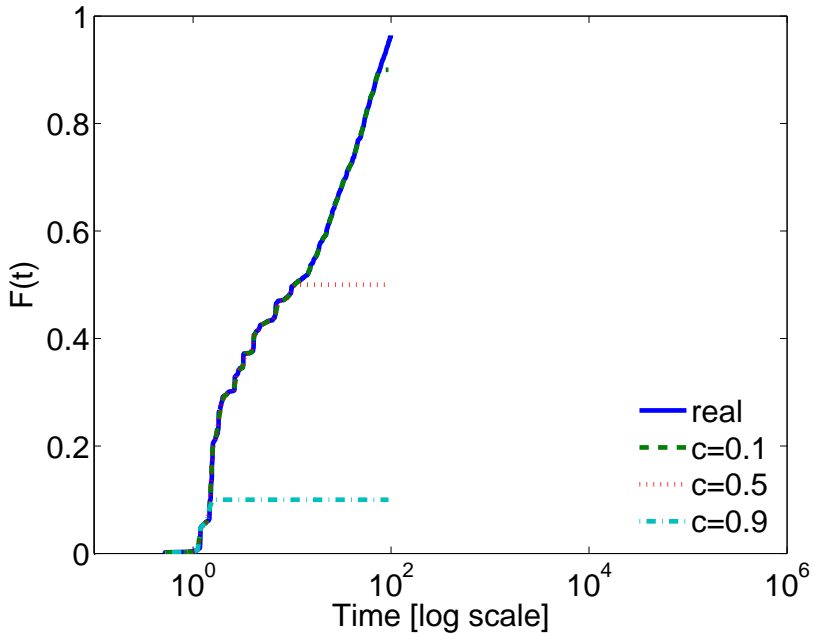
Problem 7, cost t_T of restart



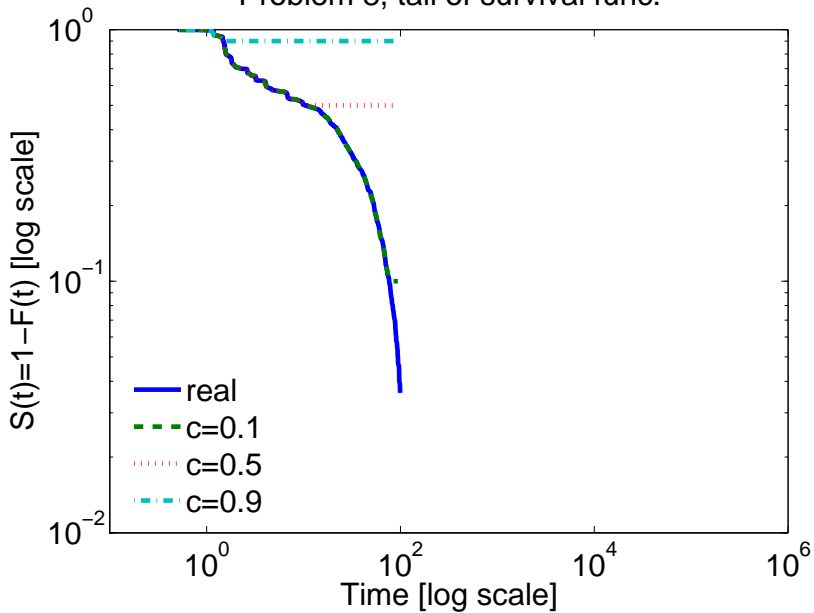
Problem 8, censoring thresh. t_c



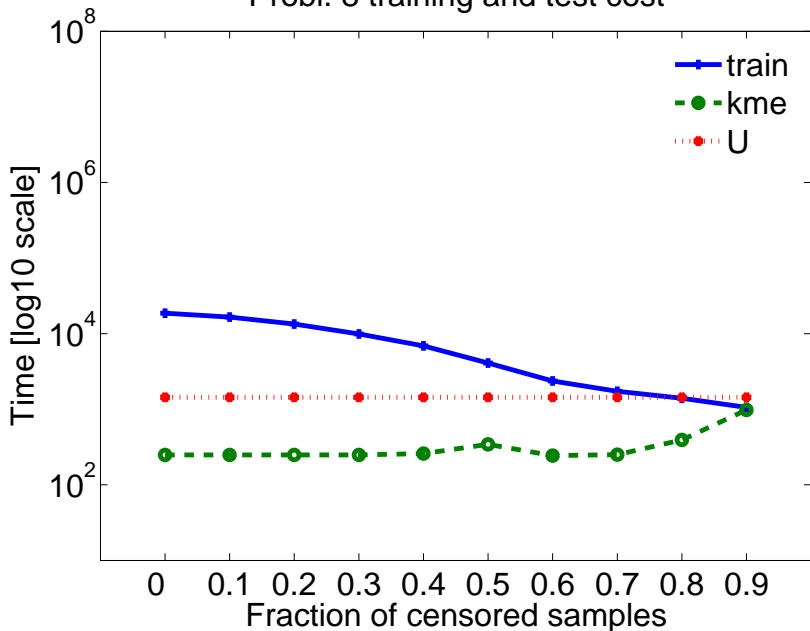
Problem 8, CDF



Problem 8, tail of survival func.



Probl. 8 training and test cost



Problem 8, cost t_T of restart

