Learning to Predict User Replies in Interactive Job Scheduling[1]

Johannes Varga[a], Günther R. Raidl[a], Tobias Rodemann[b]

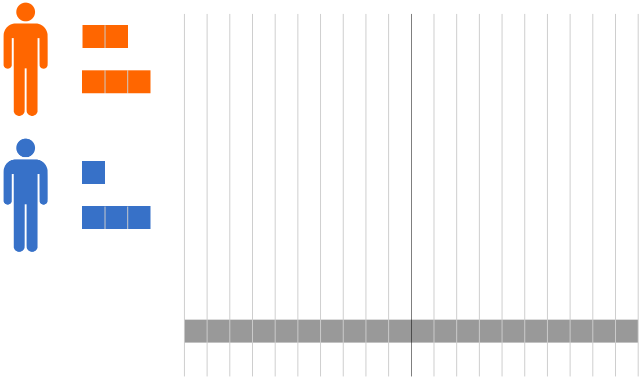
[a]Institute of Logic and Computation, TU Wien, Vienna, Austria

[b]Honda Research Institute Europe, Offenbach, Germany

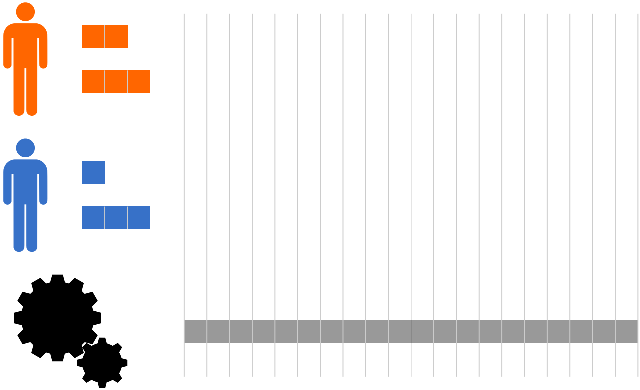
May 6, 2024

[1] J. Varga acknowledges the financial support from Honda Research Institute Europe.

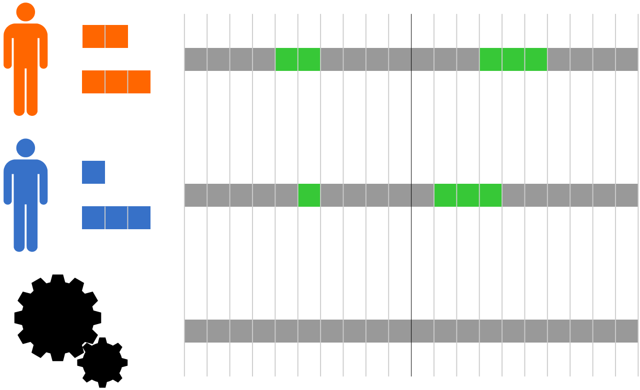
Scheduling Setting



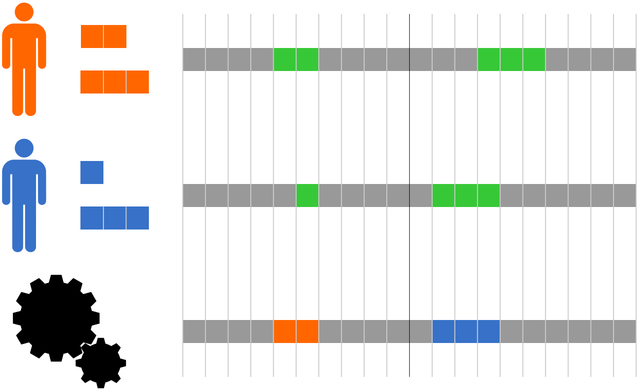
Scheduling Setting



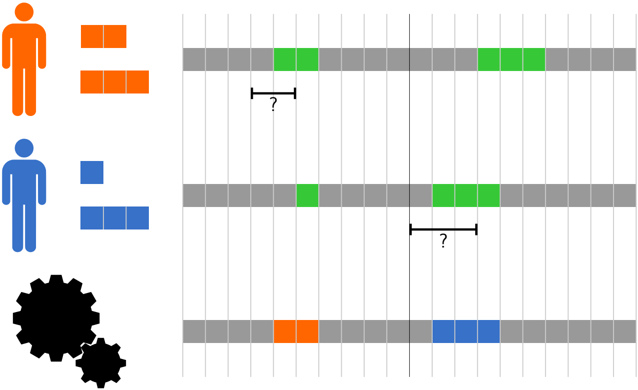
Scheduling Setting



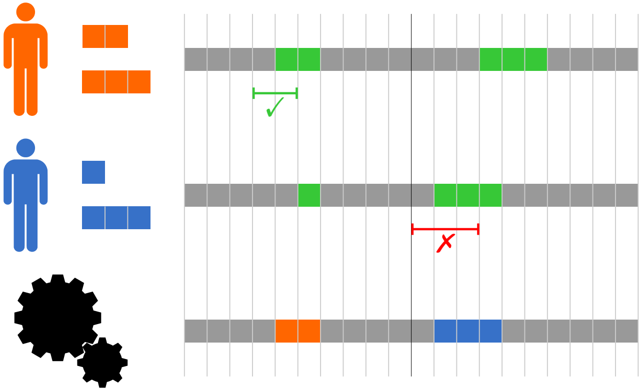
Scheduling Setting



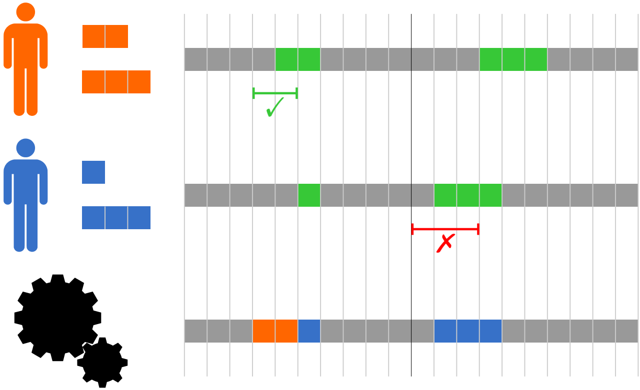
Scheduling Setting



Scheduling Setting



Scheduling Setting



Scheduling Setting

Core problem:

- Discrete time planning horizon of multiple days
- Multiple users
- Multiple jobs per user
- Multiple machines
- Schedule jobs non-preemptively on machine

Objective:

- time dependent costs c_{it} for using the machine i at timestep t
- penalty q_j for not scheduling a job j
- not scheduling a job is more expensive than scheduling it

Scheduling setting

User availabilities:

- Limit job running times
- Only partially known
- Complement knowledge with interaction

Interaction:

- B rounds of interaction
- each with up to b queries
- Query: Time interval
- Reply: Yes/no

Prediction task

Criteria for queries:

1. Good response likely $\longrightarrow$ Model users in probabilistic way
2. Improve the schedule $\longrightarrow$ Optimize

Train Sample:

- Proposed time intervals I^{prop}
- Accepted time intervals I^{acc}
- Rejected time intervals I^{rej}

Test Sample: additionally

- Potential queries I^{pred}
- Labels $\hat{l}^{\text{pred}} : I^{\text{pred}} \rightarrow \{\textbf{false}, \textbf{true}\}$

One model for all users and days

Bayesian Learning[2] and Probabilistic Programming[3]

Bayes theorem:

$$\underbrace{\mathbb{P}(\theta \mid D)}_{\text{Posterior}} \propto \overbrace{\underbrace{\mathbb{P}(D \mid \theta)}_{\text{Likelihood}} \underbrace{\mathbb{P}(\theta)}_{\text{Prior}}}^{\text{Joint Probability}} \tag{1}$$

Probabilistic program: Calculate joint (log) probability

Advantages:

- Sample efficient
- Very flexible

→ Model ...

  - ... bias from proposed intervals
  - ... uncertainty of training samples

- Uncertainty measure

[2] Schoot et al. 2021.

[3] Meent et al. 2018; Wingate, Stuhlmüller, and Goodman 2011.

Markov Model[4]

Model availabilites with Markov process

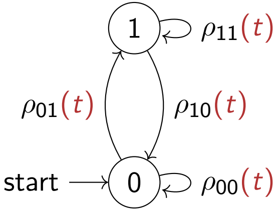


Figure: Two-state Markov Chain

Use time-independent and time-dependent transition probabilities

[4] Varga et al. 2023.

Time Interval Model

n availability intervals $[t_i^{\text{start}}, t_i^{\text{end}}]$, $i = 1, \ldots, n$ throughout the day

Rounded normally distributed endpoints:

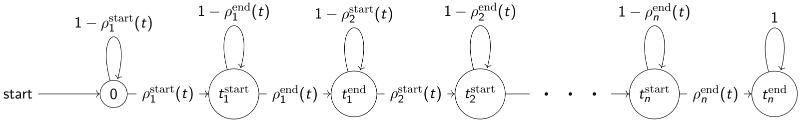
$$t_i^{\text{start}} \sim \texttt{Round(Normal(}\mu_i^{\text{start}}, \sigma_i^{\text{start}}\texttt{)}, [1, t^{\max}]\texttt{)} \qquad (2)$$

$$t_i^{\text{end}} \sim \texttt{Round(Normal(}\mu_i^{\text{end}}, \sigma_i^{\text{end}}\texttt{)}, [1, t^{\max}]\texttt{)} \qquad (3)$$

Condition: Ordered

$$t_1^{\text{start}} < t_1^{\text{end}} + 1 < t_2^{\text{start}} < \ldots < t_n^{\text{end}} + 1 \qquad (4)$$

As Markov chain:



Probabilistic Program

Algorithm 1: Probabilistic program to condition on training samples S^{train}.

Input: Training samples S^{train}

sample $\theta \sim \texttt{Prior(Model)}$;
for $s^{\text{train}} = (I^{\text{prop}}, I^{\text{acc}}, I^{\text{rej}})$ **in** S^{train} **do**
  sample $T^{\text{avail}*} \sim \texttt{Model}(\theta)$;
  for $[t_1, t_2]$ **in** I^{prop} **do**
    $I \leftarrow \texttt{Subintervals}(T^{\text{avail}*}, t_2 - t_1 + 1)$;
    sample $[t_1', t_2'] \sim \texttt{Uniform}(I)$;
    observe $t_1' = t_1$ *and* $t_2' = t_2$;
  end
  for $[t_1, t_2]$ **in** I^{acc} **do**
    observe $[t_1, t_2] \subseteq T^{\text{avail}*}$;
  end
  for $[t_1, t_2]$ **in** I^{rej} **do**
    observe $[t_1, t_2] \nsubseteq T^{\text{avail}*}$;
  end
end

Inference Procedure

Algorithm 2: Model-independent part of the sampling procedure.

Input: Training samples $S^{\text{train}} = \{s_1^{\text{train}}, \ldots, s_{|S^{\text{train}}|}^{\text{train}}\}$, $n^{\text{samples}} \in \mathbb{N}$

Output: Sets of parameters $\{\theta_1, \ldots, \theta_{n^{\text{samples}}}\}$ distributed according to the posterior distribution.

$T_k^{\text{avail}*} \leftarrow T \qquad \forall k \in \{1, \ldots, |S^{\text{train}}|\}$;
$\theta \leftarrow$ `InitParameters(Model)` ;
for j **in** $\{1, \ldots, n^{\text{samples}}\}$ **do**
    $T^{\text{avail}*} \leftarrow$ `SampleTavail(`θ, $T^{\text{avail}*}$, S^{train}`)` ;
    $\theta \leftarrow$ `SampleParameters(`θ, $T^{\text{avail}*}$`)` ;
    $\theta_k \leftarrow \theta$;
end
return $\{\theta_1, \ldots, \theta_{n^{\text{samples}}}\}$;

Sample $T^{\text{avail}*}$

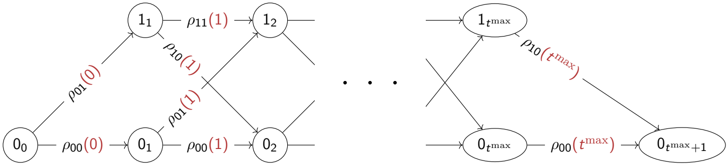
θ fixed, account for I^{prop}, I^{acc}, I^{rej}

For each training sample

1. Generate probability graph ($I^{\text{prop}} \triangleq I^{\text{acc}}$)
2. Sample random path $\rightarrow$ set of availabilities
3. Account for I^{prop} $\rightarrow$ Metropolis Hastings

Probability graph

- Unroll Markov chain of model $\Rightarrow$ Paths $\leftrightarrow$ availability sets
- Account for I^{prop}, I^{acc} and I^{rej} by manipulating graph
- Calculate (conditional) probability of next state for each state



Sample Parameters

Markov models:

- Each transition: Bernoulli distribution
- Transitions known ($T^{\text{avail}*}$ is fixed)

→ Sample transition probabilities from beta distribution

Time interval model:

- t_i^{start}, t_i^{end}: Rounded normally distributed
- t_i^{start}, t_i^{end} known (for each training sample)

→ Sample mean and variance from Normal inverse χ^2 distribution[5]

- Correct with Metropolis Hastings

[5] Gelman et al. 1995, Chapter 3.

Datasets

Collect datasets:

- Four instances of scheduling problem with
  - five machines
  - 30 users
  - four jobs per user $\rightarrow$ 120 jobs

$\rightarrow$ Four weeks training data

- Simulate and record interaction (5 interaction rounds)
- Test data: Additionally compute queries (all possible short intervals) and labels

Two datasets:

- Generated user availabilities: two intervals with normally distributed start time and duration
- User availabilities based on Dutch Time-use-Survey[6]

[6] Sociaal en Cultureel Planbureau 2005.

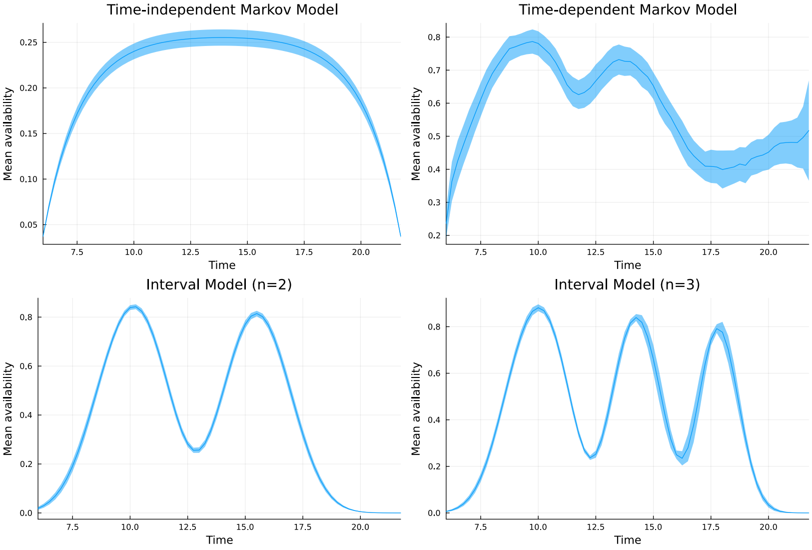
Computing Environment

- Julia 1.10.0[7]
- Probabilistic programming framework: Gen.jl[8]
- AMD Ryzen 9 5900X
- Markov models: 1000 iterations, use samples 500:10:1000
- Time interval model: 100 iterations, use samples 50:100
- Train times $< 70s$, Test times $< 40s$

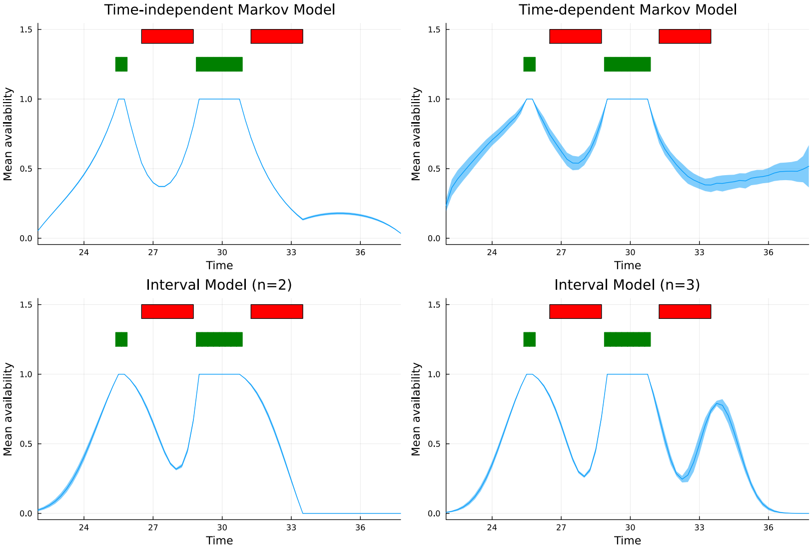
[7] Bezanson et al. 2017.

[8] Cusumano-Towner et al. 2019.

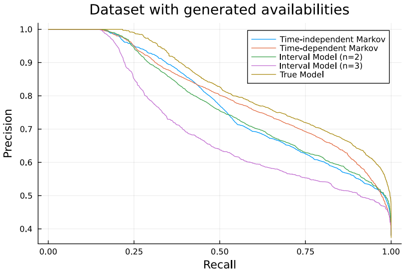
Average User Availabilities

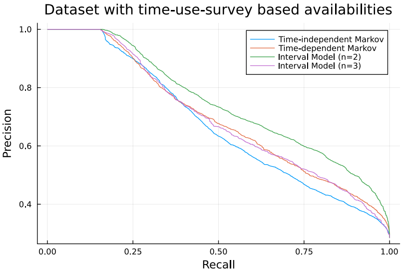


Average User Availabilities After Interaction



Precision-Recall Curves





Varying the Number of Rounds for Test Samples

Table: Area-under-curve of precision recall plots.

Number of Rounds	**Time-independent Markov**	**Time-dependent Markov**	**Interval Model (n=2)**	**Interval Model (n=3)**
0	0.608 ± 0.001	0.655 ± 0.009	$\mathbf{0.718} \pm \mathbf{0.003}$	0.661 ± 0.012
1	0.622 ± 0.001	0.664 ± 0.009	$\mathbf{0.724} \pm \mathbf{0.003}$	0.667 ± 0.012
2	0.638 ± 0.001	0.672 ± 0.009	$\mathbf{0.732} \pm \mathbf{0.002}$	0.675 ± 0.011
3	0.649 ± 0.001	0.682 ± 0.008	$\mathbf{0.739} \pm \mathbf{0.002}$	0.684 ± 0.011
4	0.662 ± 0.001	0.690 ± 0.008	$\mathbf{0.747} \pm \mathbf{0.002}$	0.691 ± 0.011
5	0.672 ± 0.001	0.697 ± 0.008	$\mathbf{0.753} \pm \mathbf{0.002}$	0.697 ± 0.011

Conclusion and Future Work

Bayesian learning of three user models for interactive scheduling problem

Time-dependent Markov and time interval model ($n = 2$) work best

Also good performance before any interaction

Future work

- Vary number of intervals in interval model
- Learn differences between days and users
- Active learning
- Recognize drift in user behavior

References I

- Bezanson, Jeff et al. (2017). "Julia: A fresh approach to numerical computing". In: *SIAM Review* 59.1, pp. 65–98.
- Cusumano-Towner, Marco F. et al. (2019). "Gen: A General-purpose Probabilistic Programming System with Programmable Inference". In: *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*. PLDI 2019. New York, NY, USA: ACM, pp. 221–236.
- Gelman, Andrew et al. (1995). *Bayesian data analysis*. New York: Chapman and Hall/CRC.
- Meent, Jan-Willem van de et al. (2018). *An introduction to probabilistic programming*. Preprint arXiv:1809.10756.
- Schoot, Rens van de et al. (2021). "Bayesian statistics and modelling". In: *Nature Reviews Methods Primers* 1.1, p. 1.

References II

Sociaal en Cultureel Planbureau (2005). *Tijdsbestedingsonderzoek 2005 - TBO 2005*. Version V2. DOI: 10.17026/dans-znn-5xvz. URL: https://doi.org/10.17026/dans-znn-5xvz.

Varga, Johannes et al. (2023). "Interactive Job Scheduling with Partially Known Personnel Availabilities". In: *OLA 2023: Optimization and Learning*. Ed. by B. Dorronsoro et al. Vol. 1824. Communications in Computer and Information Science. Springer, pp. 236–247.

References III

Wingate, David, Andreas Stuhlmüller, and Noah Goodman (2011). “Lightweight implementations of probabilistic programming languages via transformational compilation”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. PMLR, pp. 770–778.