Writeup

Yannik Pitcan
Panos Lambrianides
Ram Akella
Anil Aswani
Phil Kaminsky

August 28, 2019

Current Direction

Problem 1

The first problem can be split into two parts:

- System identification – estimate an underlying state and then use model-based approaches to solve for optimal repair strategy to reduce total maintenance cost.
- Determining optimal actions without knowledge of underlying states (model-free reinforcement-learning approach)

For the first step (estimation), one can use a variant of EM (expectation maximization) algorithm to determine underlying parameters. This method however comes with some normality assumptions we may not necessarily want for our model.

Problem 2

The second problem, we ask how we combine prior knowledge about the data generating process when determining the optimal repair strategy that minimizes costs. For example, if we have a controlled system generating data, this alters the observed machine measurements as opposed to an uncontrolled system. This is because if the machine is deteriorating, we repair and restore the system back to full health so our measurements seen will be different than if we let the machine fail.

The alternative route would be to avoid parameter estimation altogether and work with objective functions to assess which distribution our data is coming from. The objective function encapsulates enough information about our underlying distributions for it to act as a means of identifying our data generation process. Reinforcement learning can be of use because we can avoid dealing with too complex data by only considering data points corresponding to high value functions or high probabilities.

Model Specifications

To summarize, the two problems we want to solve are

- Can we use RL approaches to determine information about the time to failure (RUL) without resorting to parametric estimation?
- How can we understand optimal repair methods when we blend prior knowledge of our machine data? For example, do observed system measurements indicate a control policy is in place?

Experiment 1

As before, we create a simulation for a machine running until it reaches failure. The underlying states $(1, 2, 3, 4)$ represent the health level of the machine, with 1 indicating very-low/failing and 4 very-high/perfect condition. A level of 2 denotes low but not failing and 3 moderately high health.

Our state transitions are given by

$$\begin{bmatrix} p_{11} & 0 & 0 & 0 \\ p_{21} & p_{22} & 0 & 0 \\ 0 & p_{32} & p_{33} & 0 \\ 0 & 0 & p_{43} & p_{44} \end{bmatrix}$$

where p_{ij} denotes the probability of the system health moving from level i to j. The above is for the case where no action is taken.

The actions here are either to repair $a = 1$ or do nothing $a = 0$.

When a repair is done, then

$$\begin{bmatrix} 0 & p'_{12} & p'_{13} & p'_{14} \\ 0 & 0 & p'_{23} & p'_{24} \\ 0 & 0 & 0 & p'_{34} \\ 0 & 0 & 0 & p'_{44} \end{bmatrix}$$

is our transition matrix, where we use p' to denote these are probabilities when we do a repair.

Our observations depend on whether we repair the system or not. Furthermore, the observation space is continuous with the observations distributed by pdfs f_0 and f_1 for actions 'no-repair' and 'repair' respectively.

If we don't repair the system, then define $O(o|s, a = 0) \sim f_0(s)$. Else, if we do repair, then $O(o|s, a = 1) \sim f_1(s)$.

If we use discrete observation buckets, then we have a matrix form

$$O(0) = \begin{bmatrix} NoRepair & 1 & 2 & 3 & 4 \\ 1 & o_{11} & o_{12} & o_{13} & o_{14} \\ 2 & o_{21} & o_{22} & o_{23} & o_{24} \\ 3 & o_{31} & o_{32} & o_{33} & o_{34} \\ 4 & o_{41} & o_{42} & o_{43} & o_{44} \end{bmatrix}$$

and

$$O(1) = \begin{bmatrix} Repair & 1 & 2 & 3 & 4 \\ 1 & o'_{11} & o'_{12} & o'_{13} & o'_{14} \\ 2 & o'_{21} & o'_{22} & o'_{23} & o'_{24} \\ 3 & o'_{31} & o'_{32} & o'_{33} & o'_{34} \\ 4 & o'_{41} & o'_{42} & o'_{43} & o'_{44} \end{bmatrix}$$

Lastly, our reward function takes the following form:

$$c(4 - s) + \sum_{s=1}^{4} I_M(a = 1|s)$$

where I is an indicator function. c is a negative constant so this represents increasing (negative) rewards with respect to the state, if there is no action taken and vice versa (decreasing, positive w.r.t the state)if there is an action.

Another way of writing this is as follows

$$R(0) = \begin{bmatrix} NoRepair & Reward \\ 1 & r_1 \\ 2 & r_2 \\ 3 & r_3 \\ 4 & r_4 \end{bmatrix}$$

$$R(1) = \begin{bmatrix} Repair & Reward \\ 1 & r_1' \\ 2 & r_2' \\ 3 & r_3' \\ 4 & r_4' \end{bmatrix}$$

We can define this reward function as $R(a) = c(4 - s) + \sum_{s=1}^{4} I(a = 1|s)$ where $R_s(0) = r_s$ and $R_s(1) = r_s'$.

The overarching goal is to use reinforcement learning methods with POMDPs to give an optimal strategy for repairing the system that minimizes the repair costs. And we believe that, in solving the POMDP, we can determine the time of failure implicity with a good choice of an objective function.

The two routes are as follows:

- Solving this machine failure model as a POMDP using model-based methods such as dynamic programming (value-iteration)
  - In this approach, we do estimation of the underlying states first by using an EM based approach for example.
  - Other methods here involve linear interpolation of states based on system measurements. Since we know the machine begins at full health and ends at failure, we can use the measurements at start and end times and linearly interpolate for health levels.
- Model-free approaches that circumvent guesses of the underlying state transition matrices.

In the first route, we could estimate underlying states via solving a POMDP with a cost function dependent on the observed measurement instead of the state. So $R(a) = f(o, a)$ where f may be a linear function for simplicity, but not necessarily.

Experiment 2: Combining RL with Sequential Analysis

Here, we would like to learn optimal policies for repairing a system when we have assumptions on the data generating process.

- No prior distribution – our data is generated via bootstrapping. For example, with the C-MAPPS data, we could resample from the trajectories.
- One prior
  - Do we trust this prior distribution? Or should we ignore that and just bootstrap?
- We have two prior distributions and we would like to understand which one generates our data, or how they interact with each other. For example, we can combine two priors via a "weighted average" (partial update) or fully via a Bayesian update.

What we would to solve is, if our data is generated from $\epsilon_t P_0 + (1 - \epsilon_t) P_1$, where P_0 and P_1 are the two priors, does $\epsilon_t \to 0$ or to a constant over time?

In this setting, we can think of our $z_t = \{x_t, y_t\}$ pairs coming from data sources P_0 and P_1.

Define $p(z_t|P_0)$ and $p(z_t|P_1)$ to be the probabilities of observing z_t given P_0 and P_1 respectively at time t. Then

$$p(P_j|z_t) = \frac{\epsilon_t p(z_t|P_0)I(j=0) + (1-\epsilon_t)p(z_t|P_1)I(j=1)}{\epsilon_t p(z_t|P_0) + (1-\epsilon_t)p(z_t|P_1)}.$$

But now, we want to understand

$$p(\{P_{j_1}, \ldots, P_{j_T}\}|\{z_1, \ldots, z_T\})$$

where each P_{jt} is either P_0 or P_1.

The simplistic approach to estimate ϵ would be to do a maximization of the above quantity over all assignments of $\{P_{j1}, \ldots, P_{jT}\}$ to the set $\{P_0, P_1\}^T$. This method, however, is computationally intensive and furthermore does not account for time-dependence in our data.

We propose a method which uses RL to optimize an objective function which implicitly tells us the proportion of time data comes from P_0 vs P_1. At time t, after we observe $z_1, \ldots, z_t$, we can do one of three actions. Either we choose P_0 or P_1 as our hypothesis of the prior, or we continue exploration, which incurs a cost C. If we choose incorrectly between P_0 and P_1, we incur a cost L and if we were correct, we don't incur any cost.

This is now a three action POMDP where the observations are the z_i's and the underlying states are P_0 and P_1.

Our observation distributions are given by the pdfs $O_s(z)$, which can be one of $O_0(z)$ or $O_1(z)$, depending on whether the underlying data comes from P_0 or P_1 at that time.

State transitions, however, require a little more thought. For now, the transitions between our two states is just a 2x2 matrix

$$\begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix}$$

For our cost function, define this again to be $R_s(a) = C * I(a=2) + L * I(a<2, s \neq a)$ where $s \in \{0,1\}$ represents the underlying distribution and $a \in \{0,1,2\}$ represents choosing P_0, P_1, or neither. So if $a=1$, we chose P_1. One way of dealing with the time-dependent structure here is by incorporating a discount factor γ that weighs the most recent more. Thus we give more weight to the most recent observations and rewards.

In this setting, we cannot use a model-based approach since we don't have knowledge about our state transitions. But with a model-free approach such as deep variational reinforcement learning `https://arxiv.org/pdf/1806.02426.pdf`, we can learn the state transition matrix and thus infer the proportion of time data comes from each generating process.

Appendix: End Of Life Analysis Background

The goal of the preliminary analysis is to create a Bayesian probabilistic survival model, using incomplete data sets in a general way. The goal of this section is to define the model and methodology used and to outline options and identify risk areas.

Let T be the continuous nonnegative random variable representation the end of life time of a machine in some population. Let f(t) be the pdf of T and let the distribution function be

$$F(t) = P(t \leq t) = \int_0^t f(u)du \tag{0.1}$$

The probability of a machine functioning until time t is given by the function

$$S(t) = 1 - F(t) = P(T > t) \tag{0.2}$$

where S(0)=1 and $S(\infty) = \lim_{t\to\infty} S(t) = 0$. The hazard function h(t) is defined as the instantaneous rate of failure at time t

$$h(t) = \frac{f(t)}{S(t)} \tag{0.3}$$

The interpretation is that $h(t)\Delta t$ is the approximate probability of failure in $(t, t+\Delta t)$. It is easy to show that

$$h(t) = -\frac{d}{dt} log(S(t)) \tag{0.4}$$

Integrating both sides and exponentiating we get

$$S(t) = exp(-\int_0^t h(u)du) \tag{0.5}$$

The hazard function H(t) and the survivor function S(t) are related by

$$S(t) = exp(-H(t)) \tag{0.6}$$

and the hazard function h(t) has the properties

$$h(t) \geq 0 \quad \text{and} \quad \int_0^\infty h(t)dt = \infty \tag{0.7}$$

Finally the survival pdf is given by

$$f(t) = h(t)exp(-\int_0^t h(u)du) \tag{0.8}$$

The Weibull distribution

The Weibull distribution is one of the most commonly used distributions to model survival pdf, and is the distribution we will use to model survival pdf

$$f(t) = \begin{cases} \alpha\gamma t^{\alpha-1} exp(-\gamma t^\alpha) & t > 0, \alpha > 0, \gamma > 0 \\ 0 \quad \text{otherwise} \end{cases} \tag{0.9}$$

The Weibull distribution is denoted by $\mathcal{W}(\alpha, \gamma)$ where α and γ are the parameters of the distribution. For this distribution the hazard distribution h(t) is monotonically increasing when $\alpha > 1$ and monotonically decreasing when $0\alpha < 1$. It follows that the survivor function is

$$S(t) = exp(-\gamma t^\alpha) \tag{0.10}$$

and the hazard function is

$$h(t) = \gamma\alpha t^{\alpha-1} \tag{0.11}$$

and the cumulative hazard function H(t) is given by

$$H(t) = \gamma t^\alpha \tag{0.12}$$

Proportional Hazard Model

The hazard function depends on both time and a set of covariates. The proportional hazards model separates these components by by specifying that the hazard at time t, for a machine whose covariate vector is **x** is given by the linear relatioship

$$h(t|\mathbf{x}) = h_0(t)exp(\mathbf{x}'\beta) \tag{0.13}$$

where β is a vector of regression coefficients.

Censoring

We expect the end of life sensor data is right censored, that is, the survival times are only known for a portion of machines under study. The likelihood function for righ censored data will be constructed as follows. Suppose there are n machines and associated with the i^{th} individual is a survival time t_i and a fixed censoring time c_i. The t_o are assumed to be independent and identically distributed with density f(t) and survival function S(t). The exact survival time for machine i, t_i, will be observed only if $t_i \leq c_i$, The data can be represented as the n pairs of random variables (y_i, ν_i) where

$$y_i = \min(t_i, c_i) \tag{0.14}$$

and

$$\nu_i = \begin{cases} 1 & \text{if} \quad t_i \leq c_i, \\ 0 & \text{if} \quad t_i > c_i \end{cases} \tag{0.15}$$

Then the likelihood function for $(\beta, h_0(.))$ for a set of right censored data is given by

$$\begin{aligned} L(\beta, h_0(t)|D) \propto \prod_{i=1}^{n} [h_o(y_i) exp(\eta_i)]^{\nu_i} (S_0(y_i)^{exp(\eta_i)}) \\ \prod_{i=1}^{n} [h_o(y_i) exp(\eta_i)]^{\nu_i} exp \left\{ -\sum_{i=1}^{n} exp(\eta_i) H_0(y_i) \right\} \end{aligned} \tag{0.16}$$

where $D = (n, \mathbf{y}, X, \nu)$, $\mathbf{y} = (y_1, y_2, ..., y_n)'$ and $\nu = (\nu_1, \nu_2, ..., \nu_n)$, $\eta_i = \mathbf{x}_i' \beta$ is the linear predictor of machine i, and

$$S_0(t) = exp(-\int_0^t h_0(u) du) = exp(-H_0(t)) \tag{0.17}$$

is the baseline survivor function.