Introduction to Prognostics
Annual Conference of the PHM Society 2019

Matteo Corbetta
SGT Inc., NASA Ames Research Center

September 24, 2019 Scottsdale, AZ
Acknowledgments:

- Dr. Kai Goebel and the PHM Society
- Previous tutorial presenters
- SGT Inc., Diagnostics & Prognostics Group, NASA Ames
- Prof. Yongming Liu and his team for the crack growth dataset
Topics of the tutorial

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)
Your feelings during this tutorial if:

you know (some) PHM

Uh? PHM? prediction?

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)
Today’s material

Download this presentation and tutorial code at:
phmsociety.org/events/conference/phm/19/tutorials

Scripts and dataset:
particleFilterPrediction.py    gpRegression.py    CO2data.txt

Libraries we’ll use:
numpy    scipy    matplotlib

Instructions to install Python and libraries:
README.txt
What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)
Definition

**Prognostics** is an engineering discipline focused on predicting the time at which a system or a component will no longer perform its intended function*

Approaches to prognostics
Thanks to Prof. J. W. Hines, PHM Tutorial 2009

Type I: Reliability-based

\[ \lambda = \lambda(t), \text{MTTF, MTBF, ...} \]
Approaches to prognostics

Thanks to Prof. J. W. Hines, PHM Tutorial 2009

Type I: Reliability-based
\[ \lambda = \lambda(t), \text{MTTF, MTBF, ...} \]

Type II: Stress-based
E.g., proportional hazard models
\[ \lambda = \lambda(t, z), \text{where } z \text{ are "stressors"} \]
Approaches to prognostics

Type I: Reliability-based

\[ \lambda = \lambda(t), \text{ MTTF, MTBF, ...} \]

Type II: Stress-based

E.g., proportional hazard models
\[ \lambda = \lambda(t, z), \text{ where } z \text{ are } “\text{stressors}” \]

Type III: Condition-based ← what we’ll see today

Modeling individual failure mechanisms, cumulative damage models, state extrapolation, ...
Why prognostics?

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)
Why prognostics?

Thanks to: Dr. Abhinav Saxena
Why prognostics?

Safety

prevent unexpected failures
minimize impact on other systems
be prepared to initiate contingency plans
Why prognostics?

**Logistics**
reduce spare parts stock
logistics footprint
Why prognostics?

Maintenance
reduce unnecessary interventions
”Just-in-time” approach
optimize fleet maintenance
Why prognostics?

Reliability & Performance

product reputation
reduced safety factors
Why prognostics?

**Safety**
- prevent unexpected failures
- minimize impact onto other systems
- implement contingency plans

**Logistics**
- reduce spare parts stock
- logistics footprint

**Maintenance**
- reduce unnecessary interventions
- "Just-in-time" approach
- optimize fleet maintenance

**Reliability & Performance**
- product reputation
- reduced safety factors

Thanks to: Dr. N. Scott Clements. Please refer to his tutorial [PHM Tutorial 2011](#) for more information on industrial applications
Prognostic process

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)
What we are trying to predict

Future behavior

Calculate the future values of the quantities of interest to infer future behavior of the system

End-of-life

Calculate the time-to-failure or the remaining useful life (RUL) of a component/system, which current condition is known with certain confidence.
Steps of the prognostic process

1. Anomaly / fault detection

2. Identification and isolation

3. Quantification

4. Prognosis
Source of information (and uncertainty)
Measures

- Ground truth measures are hard to come by, and sometimes there’s no term for comparison (e.g., Golden Gate bridge)
- Many times measures are noisy, corrupted by systematic errors or faulty measurement systems
- Health-related quantities are typically hard to measure (i.e., measures are intrusive or destructive)
- Some times it’s simply not possible (physically or economically) to measure some variables
Models

- Models are mere representations of reality
- Models do not (typically) accommodate all physical phenomena affecting the system. If they do, they may not be suitable for real-time applications
- They always require calibration and validation.
- They may require correction terms to be updated in real-time (every time) or to be tuned on a case-basis
Environmental & operational conditions (input)

- Varying environmental conditions can drastically change algorithm performance (or even make algorithms useless)
- Many damage-sensitive features are also affected by operational profiles (e.g., vibrations in a wind turbine generator change with produced power)
- Environmental variables may be unknown, hard to measure or their future values hard to predict (i.e., wind speed and direction in urban environments)
- Finding causal relationships: dependencies from external factors are hard to quantify
Computing methods

- Rounding errors or machine precision may not be negligible for the problem we’re looking at.
- If the algorithm goal is minimization or filtering, they may get stuck into a local minima (e.g., the results change at different runs).
- They often need tuning of parameters, or in case of data-driven methods, their performance depends on the amount of training data.
- Convergence not always guaranteed.
Prognosis in a cartoon

Tracking health vs. tracking degradation

- Normal performance zone
- Degraded performance zone
- Hazard zone
- Acceptable risk threshold
- Critical failure threshold
- Expected failure
- End of operation point

Time

System's health indicator

System's degradation indicator
Prognosis in a cartoon

Thanks to: Dr. Abhinav Saxena, GE
See his prognostics tutorial from Annual PHM Conference 2010 here.
Examples (with codes)

What is prognostics?

Why prognostics?

Prognostic process

Examples (with codes)
Example 1
Data-driven $\text{CO}_2$ concentration prediction
CO$_2$ concentration prediction

Monthly average atmospheric CO$_2$ concentrations (in parts per million by volume, ppmv) collected at the Mauna Loa Observatory in Hawaii between 1958-2001$^1$.

What will the CO$_2$ concentration be after 2001?

$^1$ credits for the idea to Rasmussen and Williams, Gaussian Processes for Machine Learning, and Sci-kit learn, and NOAA for the dataset.
We use Gaussian Processes (GP) to predict the concentration over the years after 2001. The process $f(x)$ is a GP if it can be specified by a mean and covariance function:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

**The covariance function** $k(x, x')$ **is the key** containing info about time-correlations and dispersion.

Once we learn the covariance function, we can perform predictions far from training points.
GP - prior vs posterior

Building the covariance function

The covariance function $k$ is the key: To fit the CO$_2$ time series, we build $k$ as a sum of elementary covariance functions:

$$k_1(x, x') = \theta_1^2 \exp \left( -\frac{1}{2} \frac{(x-x')^2}{\theta_2^2} \right)$$  \text{long-term rising trend}

$$k_2(x, x') = \theta_3^2 \exp \left( -\frac{(x-x')^2}{2\theta_4^2} - \frac{2\sin^2(\pi(x-x'))}{\theta_5^2} \right)$$  \text{periodicity}

$$k_3(x, x') = \theta_6^2 \left( 1 + \frac{(x-x')^2}{2\theta_8^2\theta_7^2} \right)^{-\theta_8}$$  \text{medium term irregularities}

$$k_4(x, x') = \theta_9^2 \exp \left( -\frac{(x_p-x_q)^2}{2\theta_{10}^2} \right) + \theta_{11}^2 \delta_{p,q}$$  \text{noise}

$$k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + k_4(x, x')$$

$$\theta = [\theta_1, \theta_2, \ldots, \theta_{11}]$$
Find hyper-parameter vector $\theta$

Find the hyper parameters $\theta$ that best fit the training data. We do so by maximizing the marginal likelihood $p(y|X)$ in log-form:

$$
\log p(y|X) = -\frac{1}{2} y^T (k + \sigma_n^2 I)^{-1} y - \frac{1}{2} \log |k + \sigma_n^2 I| - \frac{n}{2} \log 2\pi
$$

covariance function
model error (noise)

See Rasmussen & Williams, GP for ML, 2006.
Code

open gpRegression.py. make sure CO2data.txt is in the same folder.
Prediction

Using (sub-)optimal parameters found via differential evolution algorithm\(^2\).

\(^{2}\)DE should converge towards the optimal parameters. For this problem, different runs produced different results, suggesting either that the population size or number of iterations was too small.
Prediction

- Observations
- Prediction
- Prediction from Scikit-learn

±σ from Scikit-learn
Prediction

What’s the concentration today?

[Graph showing CO₂ concentration at Mauna Loa, Hawaii from 1960 to 2030, with predictions for 2018 and 2019.]
A few things to remember

What about model validation???

Here's some options you should try:

▶ Split dataset into training and validation
▶ Cross-validation with batches, leave-one-out, etc.
▶ Gather more data
▶ Try adding/removing different covariance functions
Example 2
Fatigue crack growth prognosis using particle filter
Fatigue crack growth prognosis

Data from 2019 PHM data challenge:

- fatigue crack growth at rivet holes
- tensile, constant amplitude fatigue loading

Thanks to: Prof. Yongming Liu and his team, ASU

Visit the 2019 PHM Data challenge website for more information.
Given the set of sequential measures of crack length, can we predict the number of cycles to reach final length $a = 7.22$ mm (i.e., 55,000 load cycle)?
Bayesian filtering equations

Chapman-Kolmogorov and Bayesian updating

\[
p(X_k | Y_{k-1}) = \int_{-\infty}^{\infty} p(X_k | X_{k-1}) p(X_{k-1} | Y_{k-1}) \, dX_{k-1}
\]

\[
p(X_k | Y_k) = \frac{p(X_k | Y_{k-1}) p(Y_k | X_k)}{p(Y_k | Y_{k-1})}
\]

Far-ahead prediction stage:

\[
p(X_{k+l} | Y_k) = \int_{\mathcal{X}} p(X_k | Y_k) \left[ \prod_{j=k+1}^{k+l} p(X_j | X_{j-1}) \right] \, dX_{k:k+l-1}
\]
Particle filtering pseudo-code

Input: \( x_{k-1}^{(i)} \), \( i = 1, \ldots, N_s \), and \( y_k \)
Output: \( p(X_k | Y_k) \), \( p(RUL_k | Y_k) \)

1. Approximate posterior pdf
   \( x_k^{(i)} \sim p(X_k | x_{k-1}^{(i)}) \) ← propagate samples with model function
   \( \ell (y_k | x_k^{(i)}) \) ← compute likelihood for all samples
   \( w_k^{(i)} \propto w_{k-1}^{(i)} \ell (z_k | x_k^{(i)}) \) ← assign weights
   \( p(X_k | Y_k) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta_{x_k, x_k^{(i)}} \) ← approx. posterior pdf

2. Systematic re-sampling
   \( x_k^{(j)} \sim p(X_k | Y_k) \): \( \Pr\{x_k^{(j)} = x_k^{(i)}\} = w_k^{(i)} \)
   \( w_k^{(j)} = 1 / N_s \) \( \forall \ j = 1, \ldots, N_s \)

3. Prognosis
   for \( i = 1, 2, \ldots, N_s \) do
     \( l = 0 \)
     while \( x_k^{(i)} \in \text{safe domain} \) do
       \( x_k^{(i)} \sim p(X_{k+l} | x_k^{(i)}) \)
       \( l += 1 \)
     end
     \( t_f = t_{k+l} \) ← extract time at which sample \( i \) reached threshold \( x_{th} \)
     \( \text{RUL}_{k}^{(i)} = t_f - t_k \) ← extract remaining useful life for sample \( i \)
   end
Assign variables

\[
x = [a, \log C, m]^T \quad \text{augmented state vector}
\]
\[
z \rightarrow z = a + \epsilon_g \quad \text{unbiased, noisy measures}
\]
\[
u \rightarrow u = \Delta S = 95 \text{ MPa} \quad \text{applied stress range} (R \approx 0.05)
\]
\[
\theta = [\log C, m]^T \quad \text{state model parameter vector}
\]
\[
\epsilon_f = [e^\omega, \epsilon_{\log C}, \epsilon_m]^T \quad \text{state model error}
\]

where:
\[
\omega \sim \mathcal{N}\left(-\frac{\sigma^2_\omega}{2}, \sigma^2_\omega\right), \quad [\epsilon_{\log C}, \epsilon_m] \sim \mathcal{MVN}(\mathbf{0}, \Sigma_\theta), \quad \epsilon_g \sim \mathcal{N}(0, \sigma^2_g)
\]
open particleFilterPrediction.py
Prediction

![Graph showing predicted semi-crack length vs. load cycle with observed, measured, and estimated data points. The graph includes a shaded area representing the 95% σ-band.](image)

- **Observed (true)** data points indicate the actual measurements.
- **Measured (synthetic)** data points reflect synthetic measurements.
- **Estimated** data points are based on predictions.
- The shaded area represents the 95% confidence band for the estimated values.
Prediction

specimen life from first detection, %

normalized remaining useful life, %

true RUL
95% σ-band
predicted RUL
RUL ±10% EOL band
RUL ±10% RUL cone
A few things to remember

- The model error (or process noise) $e^\omega$ has that form for a reason. Please see Corbetta et al. MSSP 2018, 104; 305:322

- Try to implement Kernel smoothing instead of artificial dynamics for better performance (see Liu J, West M. In Sequential Monte Carlo methods in practice 2001; 197:223 Springer, NY.)

- Using unbounded processes to estimate bounded parameters usually results in poor performance
Useful links

Prognostics Center of Excellence (PCoE)

Web page:
http://prognostics.nasa.gov

Data repository:
https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/
THANK YOU
SPASIBO
DANKE
PAHMET!