

Predicting Battery Life from Usage Trajectory Patterns

Toshihiro Takahashi and Tsuyoshi Idé
IBM Research - Tokyo
{e30137, goodidea}@jp.ibm.com

Abstract

This paper addresses the task of predicting the battery capacity degradation ratio for a given usage pattern. This is an interesting pattern recognition task, where each usage pattern is represented as a trajectory in a feature space, and the prediction model captures the previous usage trajectory patterns. The main technical challenge here is how to build a good model from a limited number of training samples. To tackle this, we introduce a new smoothing technique in the trajectory space. The trajectory smoothing technique is shown to be equivalent of a novel regularization scheme for linear regression. Using real Li-ion battery data, we show that our approach outperforms existing methods.

1. Introduction

In recent years, growing attention has been paid to batteries in many industrial applications such as electric vehicles (EVs), electric grids, and factory energy management systems. In spite of the ubiquitous use of batteries, users often must worry about the degradation of their batteries. For example, in EVs, since battery degradation directly affects the navigation distance, the issue of degradation is very critical. In most practical applications, the task of battery life prediction is one of the key technologies.

This paper addresses the task of battery life prediction. One of the most important properties of batteries is the strong dependence of the battery life on the previous usage pattern, which is typically represented as a time series of physical-chemical conditions. This is an interesting pattern recognition task, where we need to capture previous usage patterns, to make predictions for arbitrary usage histories. In battery community, some latest researches focus this task [4, 5]. But they are using only single degradation factor (cycle count, electric current, or duration) for predicting the battery degradation. We need to deal with combination of these factors

for practical predictions. We formulate this task as “trajectory regression,” which can be viewed as a variant of trajectory analysis, one of the recent hot topics in the pattern recognition community [6, 3, 2, 1]. We characterize a battery usage pattern as a trajectory in a feature space of physical-chemical variables, and encode the previous usage pattern in the regression function.

In real world battery modeling, one of the critical issues is a lack of a sufficient amount of data. In battery life modeling, a prediction model is typically built based on the data from battery endurance tests. However, since each endurance test takes a long time, the number of samples is always quite small. In the present context, this means that we do not have enough trajectories to cover all of the possible usage patterns. This is our main technical challenge.

To handle this difficult problem, we attempt to take advantage of the natural notion that similar physical-chemical states should produce similar degradations. Very interestingly, such an analysis leads us to a new formulation for trajectory regression with a novel regularization term. To the best of our knowledge, this is the first work that formulates the task of battery life prediction as trajectory regression, and we show the utility of the trajectory regression using real-world Li-ion battery data.

2. Problem Setting

We assume that we are given a data set of N pairs of a trajectory and a degradation ratio:

$$\mathcal{D} \equiv \{(P^{(n)}, y^{(n)}) | n = 1, 2, \dots, N\}, \quad (1)$$

where $P^{(n)}$ is the n -th trajectory pattern observed, and $y^{(n)}$ is the amount of degradation corresponding to $P^{(n)}$. Formally, our goal is to predict the amount of degradation y for an arbitrary trajectory pattern P by using a regression function ϕ as

$$y = \phi(P|\theta), \quad (2)$$

where θ is a set of model parameters.

Each trajectory represents a usage history of a battery. Although how to represent battery usage patterns is not at all trivial, based on careful physical chemistry analysis, we found that three variables, temperature, state-of-charge (SoC), and a quantity named depth-of-discharge (DoD), are sufficient. To make the problem tractable, we discretize the three-dimensional space into $N_T \times N_S \times N_D$ voxels, where N_T , N_S , and N_D are the total number of voxels along the temperature, SoC, and DoD dimensions, respectively. In addition, we also found that the amount of battery degradation is proportional to how long the battery is used, and how much electric current is flowing at each voxel.

To conclude the problem setting, each usage trajectory P is represented as (\mathbf{f}, \mathbf{g}) , where \mathbf{f} and \mathbf{g} are $(N_T \times N_S \times N_D)$ -dimension feature vectors of the trajectory, representing the duration and electric current at each voxel, respectively (see Figure 1). For example, if the battery continuously remains at a single voxel, then only a single entry is nonzero in both \mathbf{f} and \mathbf{g} . Finally, the regression function ϕ is represented as

$$\phi(\mathbf{f}, \mathbf{g} | \boldsymbol{\theta}) \equiv \sum_c \alpha_c f_c + \sum_c \beta_c g_c, \quad (3)$$

where c runs over all of the voxels, and $\boldsymbol{\theta} \equiv (\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the coefficients representing the dependency of the nature of the battery degradation on voxel locations. Since each of the trajectories, represented as (\mathbf{f}, \mathbf{g}) , is directly observed, our goal is to determine the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, based on the training data.

3. Formulation

To determine the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, we consider an optimization problem for the objective function as

$$\Psi(\boldsymbol{\theta}) \equiv L(\boldsymbol{\theta}) + R(\boldsymbol{\theta}), \quad (4)$$

where $L(\boldsymbol{\theta})$ is a loss function term that represents the error of the model as measured by the training data, and $R(\boldsymbol{\theta})$ is the smoothing regularization function term that is introduced to augment the data, as explained in detail later.

For the loss function, the most natural assumption is the quadratic loss:

$$L(\boldsymbol{\theta}) \equiv \sum_{n=1}^N \left(y^{(n)} - \phi(P^{(n)} | \boldsymbol{\theta}) \right)^2, \quad (5)$$

which is justified by a Gaussian noise model. Note that $P^{(n)} = (\mathbf{f}^{(n)}, \mathbf{g}^{(n)})$ is the feature vector of the n -th usage trajectory in the training set.

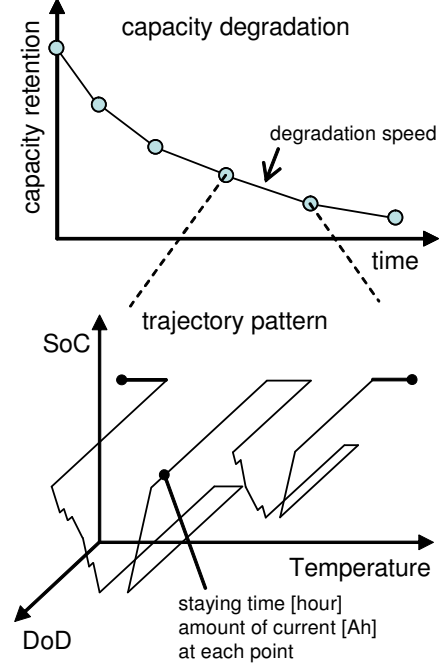


Figure 1. Outline of data set

For the smoothing regularization function, we note that the training data cannot cover the entire trajectory space. Since the length and the shape of a trajectory is arbitrary (see Figure 1), it is quite rare to find a trajectory in the training data that is exactly the same as an input trajectory P . This fact motivates us to introduce a smoothing term in the objective function. The smoothing term is also useful to reduce the risk of overfitting. In the linear model in Eq. (3), we have many coefficients that would be much larger than the number of observed trajectories.

To address this problem, we use a well-known property of batteries that the degradation rate at each state is well-approximated by the interpolation over its neighbors. Based on this, we introduce a regularization term

$$R(\boldsymbol{\alpha}, \boldsymbol{\beta}) \equiv \lambda_\alpha \sum_c \left(\alpha_c - \frac{1}{3} Q_{\text{all}}(\boldsymbol{\alpha}, c) \right)^2 \quad (6)$$

$$+ \lambda_\beta \sum_c \left(\beta_c - \frac{1}{3} Q_{\text{all}}(\boldsymbol{\beta}, c) \right)^2. \quad (7)$$

Intuitively, this term represents the natural assumption that the value of a coefficient should not deviate very much from its neighbors. Here λ_α and λ_β are parameters which control the degree of smoothing, and $Q_{\text{all}}(\boldsymbol{\alpha}, c)$ is the sum of three interpolations in the di-

mensions of the temperature, SoC, and DoD as

$$Q_{\text{all}}(\boldsymbol{\alpha}, c) \equiv Q_T(\boldsymbol{\alpha}, c) + Q_S(\boldsymbol{\alpha}, c) + Q_D(\boldsymbol{\alpha}, c), \quad (8)$$

and $Q_T(\boldsymbol{\alpha}, c)$ is a simple interpolator as

$$Q_T(\boldsymbol{\alpha}, c) \equiv \begin{cases} -\alpha_{T-2,S,D} + 2\alpha_{T-1,S,D} & (T = N_T) \\ -\alpha_{T+2,S,D} + 2\alpha_{T+1,S,D} & (T = 1) \\ \frac{1}{2}(\alpha_{T-1,S,D} + \alpha_{T+1,S,D}) & (\text{other}) \end{cases}.$$

$Q_S(\boldsymbol{\alpha}, c)$ and $Q_D(\boldsymbol{\alpha}, c)$ are also defined in the same way.

Now our problem is to minimize the objective function $\Psi(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. The next section explains how the optimization problem is solved.

4. Solving the Optimization Problem

In this section, we derive a matrix representation of the objective function in order to reduce the seemingly complicated optimization problem to a simple quadratic program.

First let us define a data matrix as

$$W \equiv \begin{pmatrix} \mathbf{f}^{(1)} & \dots & \mathbf{f}^{(N)} \\ \mathbf{g}^{(1)} & \dots & \mathbf{g}^{(N)} \end{pmatrix}^\top,$$

and the coefficient vector is

$$\boldsymbol{\gamma} \equiv \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix}.$$

Then we easily rewrite the loss function as

$$L(\boldsymbol{\gamma}) = \|\mathbf{y} - W\boldsymbol{\gamma}\|^2. \quad (9)$$

Now let us consider the regularization term. For a simplified expression, we introduce the indicator vectors $\mathbf{k}_{c,\text{all}}$ such that $\boldsymbol{\alpha}^\top \mathbf{k}_{c,\text{all}}$ reproduces the operation of $Q_{\text{all}}(\boldsymbol{\alpha}, c)$. We also introduce another indicator vector \mathbf{p}_c , whose c' -th element is just $\delta_{c,c'}$ (Kronecker's delta). Using this notation, we can rewrite Eq. (7) as

$$\begin{aligned} R(\boldsymbol{\gamma}) &= \lambda_\alpha \sum_c \left\| \boldsymbol{\alpha}^\top \left(\mathbf{p}_c - \frac{1}{3} \mathbf{k}_{c,\text{all}} \right) \right\|^2 \\ &+ \lambda_\beta \sum_c \left\| \boldsymbol{\beta}^\top \left(\mathbf{p}_c - \frac{1}{3} \mathbf{k}_{c,\text{all}} \right) \right\|^2 \\ &= \boldsymbol{\gamma}^\top \begin{pmatrix} \lambda_\alpha \mathbf{R}_s & 0 \\ 0 & \lambda_\beta \mathbf{R}_s \end{pmatrix} \boldsymbol{\gamma} \\ &= \boldsymbol{\gamma}^\top \mathbf{R} \boldsymbol{\gamma}, \end{aligned}$$

where

$$\begin{aligned} \mathbf{k}_{c,\text{all}} &\equiv \mathbf{k}_{c,T} + \mathbf{k}_{c,S} + \mathbf{k}_{c,D}, \\ \mathbf{R}_s &\equiv \sum_c \left(\mathbf{p}_c - \frac{1}{3} \mathbf{k}_{c,\text{all}} \right) \left(\mathbf{p}_c - \frac{1}{3} \mathbf{k}_{c,\text{all}} \right)^\top, \\ \mathbf{R} &\equiv \begin{pmatrix} \lambda_\alpha \mathbf{R}_s & 0 \\ 0 & \lambda_\beta \mathbf{R}_s \end{pmatrix}. \end{aligned}$$

Here is the matrix representation of the objective function with smoothing regularizer

$$\Psi(\boldsymbol{\gamma}|\boldsymbol{\lambda}) = \|\mathbf{y} - W\boldsymbol{\gamma}\|^2 + \boldsymbol{\gamma}^\top \mathbf{R} \boldsymbol{\gamma}, \quad (10)$$

and the global optimum solution $\boldsymbol{\gamma}^*$ is given as the solution of a quadratic problem as

$$\text{Minimize } \frac{1}{2} \boldsymbol{\gamma}^\top \mathbf{Q} \boldsymbol{\gamma} + \mathbf{c}^\top \boldsymbol{\gamma}, \quad (11)$$

$$\text{subject to } \boldsymbol{\gamma} \geq 0, \quad (12)$$

where $\mathbf{Q} = 2W^\top W + \mathbf{R}$, and $\mathbf{c} = -2W^\top \mathbf{y}$. To solve this quadratic problem, we can use well known algorithms such as the interior point method or the subgradient method.

5. Experiments

In this section, we validated our model using real-world battery data set.

5.1. Data sets

We used two data sets, SIMULATOR and ENDURANCE, for our performance evaluations.

The SIMULATOR data was generated by a battery simulator carefully designed to reproduce the behavior of Li-ion batteries based on physical-chemical laws. Table 1 shows the summary of the data. This data includes six cycle patterns for training, but 20%-80% cycle patterns are lacking. For accurate prediction, the prediction model must also handle usage patterns that fall into the unobserved region.

Table 1. Summary of SIMULATOR data set

training data	SoC cycle pattern	DoD
bat-tr-1	0% - 100%	100%
bat-tr-2	0% - 20%	20%
bat-tr-3	20% - 40%	20%
bat-tr-4	40% - 60%	20%
bat-tr-5	60% - 80%	20%
bat-tr-6	80% - 100%	20%
test data	SoC cycle pattern	DoD
bat-tts-1	0% - random	random
bat-tts-2	20% - random	random \leq 80%
bat-tts-3	40% - random	random \leq 60%
bat-tts-4	60% - random	random \leq 40%
bat-tts-5	80% - random	random \leq 20%

The ENDURANCE data is a real-world Li-ion battery data set. In this data, we tested hundreds of batteries under various temperature and charge-discharge patterns. We also performed additional tests under some realistic situations to validate the performance of our approach. The details are omitted due to space limitations.

5.2. Methods compared

We compared the proposed approach (denoted by PROPOSED) with k -nearest neighbor regression (KNN). In KNN, given an input trajectory, the regression function selects its neighbors N in the training data. The neighbors N are selected based on the Euclidian distance in the three-dimensional state space spanned by temperature, SoC, and DoD. A predicted value is computed as the average of the neighbors' y .

5.3. Results

Table 2 shows the relative squared errors versus PROPOSED. We see that our approach is about 1.9 - 2.2 times better than KNN for both data sets. Figure 2-a compares the actual and predicted values of y as measured by capacity retention in the SIMULATOR data set. We clearly see that PROPOSED gives a much closer curve to ACTUAL than KNN.

The poor performance of KNN can be attributed to its inability to extrapolate. For example, as one of the characteristic properties of the SIMULATOR data set, the degradation is especially significant around 100%. Thus the 0%-100% cycle is much faster than the 0%-80% cycle. However, KNN selects the 0%-100% cycle (`bat-tr-1`) as a neighbor of the 0%-80% cycle. This explains why KNN always predicts a faster degradation rate than ACTUAL.

Figure 2-b shows the coefficients in the dimension of SoC for SIMULATOR. We see that PROPOSED produces a smoother distribution of the coefficients, in spite of the fact that we used only the 6 cycle patterns for training.

Table 2. Relative squared error

	PROPOSED	KNN
SIMULATOR	1	2.20
ENDURANCE	1	1.89

6. Conclusion

We formulated the task of battery life prediction as a trajectory regression, where a battery usage pattern is represented as a trajectory in a feature space. We showed that our trajectory regression framework corresponds to a novel regularized linear regression. We demonstrated the high predictability of our method with real-world Li-ion battery data. For future work, we will look further into mathematical properties of the objective Eq. (10), particularly in the context of Laplacian regularization.

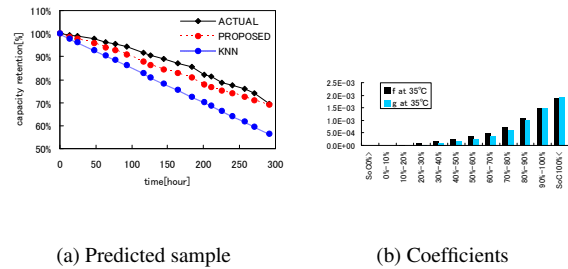


Figure 2. Result of SIMULATOR

References

- [1] T. Idé and M. Sugiyama. Trajectory regression on road networks. In *Proc. of the 25th AAAI Conf. on Artificial intelligence (AAAI-11)*, pages 203–208, 2011.
- [2] Y. Liu, X. Li, and W. Hu. Semi-supervised trajectory learning using a multi-scale key point based trajectory representation. In *Proc. the 20th Intl. Conf. Pattern Recognition, ICPR '10*, pages 3525–3528, 2010.
- [3] E. Ricci, F. Tobia, and G. Zen. Learning pedestrian trajectories with kernels. In *Proc. the 20th Intl. Conf. Pattern Recognition, ICPR '10*, pages 149–152, 2010.
- [4] P. Rong and M. Pedram. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, 14(5):441–451, 2006.
- [5] J. Wang, P. Liu, J. Hicks-Garner, E. Sherman, S. Soukiazian, M. Verbrugge, H. Tataria, J. Musser, and P. Finamore. Cycle-life model for graphite-lifepo4 cells. *Journal of Power Sources*, 196(8):3942–3948, 2010.
- [6] Z. Zhang, K. Huang, and T. Tan. Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes. In *Proc. the 18th Intl. Conf. Pattern Recognition - Volume 03, ICPR '06*, pages 1135–1138, 2006.